

Controllo assi Trajexia

TJ1-MC04

TJ1-MC16

MANUALE DI PROGRAMMAZIONE



Avviso

I prodotti OMRON sono destinati all'uso in accordo con le procedure appropriate da parte di un operatore qualificato e solo per gli scopi descritti in questo manuale.

In questo manuale vengono utilizzate le seguenti convenzioni per indicare e classificare le precauzioni. Attenersi sempre alle istruzioni fornite.

La mancata osservanza di tali precauzioni potrebbe causare lesioni o danni a cose e persone.

Definizione di informazioni di carattere precauzionale



PERICOLO

Indica una situazione di immediato pericolo che, se non evitata, sarà causa di lesioni gravi o mortali.



AVVERTENZA

Indica una situazione di potenziale pericolo che, se non evitata, può essere causa di lesioni gravi o mortali.



Attenzione

Indica una situazione di potenziale pericolo che, se non evitata, può essere causa di danni o lesioni non gravi a persone o cose.

Marchi e copyright

PROFIBUS è un marchio registrato di PROFIBUS International.

MECHATROLINK è un marchio registrato di Yaskawa Corporation.

DeviceNet è un marchio registrato di Open DeviceNet Vendor Assoc INC.

CIP è un marchio registrato di Open DeviceNet Vendor Assoc INC.

Trajexia è un marchio registrato di OMRON.

Motion Perfect è un marchio registrato di Trio Motion Technology Ltd.

© OMRON, 2007

Tutti i diritti riservati. Nessuna parte della presente pubblicazione può essere riprodotta, memorizzata in un sistema, trasmessa in qualsivoglia formato o mezzo, meccanico, elettronico, tramite fotocopia, registrazione o altro, senza previo consenso scritto di OMRON. Non viene assunta alcuna responsabilità brevettuale in relazione all'uso delle informazioni contenute nel presente manuale. Inoltre, in considerazione del fatto che OMRON punta costantemente a migliorare la qualità dei propri prodotti, le informazioni contenute nel presente manuale sono soggette a modifiche senza preavviso. Sebbene il presente manuale sia stato redatto con la massima attenzione, OMRON non si assume alcuna responsabilità in relazione ad eventuali errori od omissioni, né alcuna responsabilità in relazione ad eventuali danni derivanti dalle informazioni in esso contenute.

Informazioni sul manuale

Il presente manuale descrive le procedure di installazione e funzionamento del Controllo assi Trajexia.

Leggere attentamente il presente manuale e i manuali correlati elencati nella tabella riportata di seguito e assicurarsi di avere compreso le informazioni fornite prima di procedere all'installazione o all'azionamento delle unità di controllo assi Trajexia. Accertarsi di leggere attentamente le precauzioni indicate nella sezione successiva.

Nome	N. cat.	Contenuto
Controllo assi Trajexia MANUALE DI AVVIO RAPIDO	I50E	Descrive come apprendere rapidamente il funzionamento di Trajexia, effettuando lo spostamento di un singolo asse mediante MECHATROLINK-II, in una configurazione di prova.
Controllo assi Trajexia MANUALE DI RIFERIMENTO HARDWARE	I51E	Descrive la procedura di installazione e le specifiche hardware delle unità Trajexia, nonché la filosofia del sistema Trajexia.
Controllo assi Trajexia MANUALE DI PROGRAMMAZIONE	I52E	Descrive i comandi BASIC da utilizzare per la programmazione di Trajexia, spiega i protocolli di comunicazione e il software Trajexia Tools, fornisce esempi pratici e informazioni per la risoluzione dei problemi.
Manuale del servoazionamento Sigma-II	SIEP S800000 15	Descrive le procedure di installazione e funzionamento dei servoazionamenti Sigma-II.
Manuale dei servoazionamenti della serie JUNMA	TOEP-C71080603 01-OY	Descrive le procedure di installazione e funzionamento dei servoazionamenti JUNMA.
Manuale JUSP-NS115	SIEP C71080001	Descrive le procedure di installazione e funzionamento del modulo applicativo MECHATROLINK-II.

Nome	N. cat.	Contenuto
Manuale di Sigma-III con interfaccia MECHATROLINK	SIEP S800000 11	Descrive le procedure di installazione e funzionamento dei servoazionamenti Sigma-III con l'interfaccia MECHATROLINK.
Inverter V7	TOEP C71060605 02-OY	Descrive le procedure di installazione e funzionamento degli inverter V7.
Inverter F7Z	TOE S616-55 1-OY	Descrive le procedure di installazione e funzionamento degli inverter F7Z.
Inverter G7	TOE S616-60	Descrive le procedure di installazione e funzionamento degli inverter G7.
Interfaccia SI-T MECHATROLINK per gli inverter G7 e F7	SIBP-C730600-08	Descrive le procedure di installazione e funzionamento delle interfacce MECHATROLINK per gli inverter G7 ed F7.
Interfaccia ST-T/V7 MECHATROLINK per gli inverter V7	SIBP-C730600-03	Descrive le procedure di installazione e funzionamento delle interfacce MECHATROLINK per gli inverter V7.
Moduli IO MECHATROLINK	SIE C887-5	Descrive le procedure di installazione e funzionamento dei moduli di ingresso e uscita MECHATROLINK e del ripetitore MECHATROLINK-II.
Comandi di comunicazione delle serie SYSMAC CS/CJ	W342	Descrive il protocollo di comunicazione e i comandi FINS.



AVVERTENZA

La mancata lettura o comprensione delle informazioni fornite in questo manuale può procurare lesioni personali, anche mortali, danneggiare il prodotto o causarne un funzionamento non corretto. Leggere ogni capitolo per intero e accertarsi di avere compreso le informazioni in esso contenute e quelle correlate prima di eseguire una delle procedure od operazioni descritte.

Funzioni supportate dalle versioni delle unità

Durante lo sviluppo di Trajexia, nuove funzionalità sono state aggiunte all'unità di controllo dopo il rilascio sul mercato.

Tali funzionalità sono implementate nel firmware e/o nell'FPGA dell'unità di controllo.

Nella tabella riportata di seguito, viene descritta la funzionalità applicabile in relazione alla versione del firmware e dell'FPGA di TJ1-MC__.

Funzionalità	Versione firmware TJ1-MC__	Versione FPGA TJ1-MC__
Supporto totale TJ1-FL02	V1.6509	21 e superiore.
Supporto comandi BASIC FINS_COMMS	V1.6509	Tutte le versioni
Supporto TJ1-DRT	V1.6509	Tutte le versioni
Supporto TJ1-MC04 e TJ1-ML04	V1.6607	21 e superiore

Verificare le versioni del firmware e dell'FPGA di TJ1-MC__

Collegare TJ1-MC__ al software Trajexia Tools. Vedere il Manuale di programmazione.

Aprire la finestra del terminale e digitare i seguenti comandi:

Nella finestra del terminale, digitare `PRINT VERSION`. Il parametro della versione restituisce il numero di versione corrente del firmware dell'unità di controllo assi.

Nella finestra del terminale, digitare `PRINT FPGA_VERSION SLOT(-1)`.

Il parametro restituisce il numero di versione corrente dell'FPGA di TJ1-MC__.

1	Avvertenze e precauzioni di sicurezza	16
1.1	Destinatari del manuale	16
1.2	Precauzioni generali	16
1.3	Precauzioni di sicurezza	16
1.4	Precauzioni relative all'ambiente operativo.....	17
1.5	Precauzioni relative all'applicazione	18
1.6	Precauzioni per l'assemblaggio dell'unità	21
2	Sistema Trajexia.....	22
2.1	Introduzione	22
2.1.1	Hardware Trajexia	23
2.1.2	Questo manuale	23
2.2	Programmazione BASIC multitasking.....	23
2.3	Programmazione in BASIC.....	24
2.3.1	Istruzioni per assi, sistema e task.....	24
2.3.2	Aree di memoria	24
2.3.3	Strutture dei dati e variabili	25
2.3.4	Specifiche matematiche	27
2.4	Esecuzione del movimento.....	28
2.4.1	Generatore di movimento	28
2.4.2	Messa in sequenza.....	29
2.4.3	Caricamento dei movimenti	29
2.5	Interfaccia della riga di comando	30
2.6	Programmi in Basic.....	30
2.6.1	Gestione dei programmi	30
2.6.2	Compilazione di programmi.....	31
2.6.3	Esecuzione del programma.....	31
3	Comandi BASIC	33
3.1	Categorie	33
3.1.1	Comandi degli assi	33
3.1.2	Parametri degli assi	34
3.1.3	Comandi e parametri di comunicazione	36
3.1.4	Costanti	36
3.1.5	Comandi, funzioni e parametri di I/O	36
3.1.6	Funzioni e operandi matematici	37
3.1.7	Comandi di programma	37
3.1.8	Comandi di controllo del programma	38
3.1.9	Parametri e modificatori di slot	38

3.1.10	Comandi e funzioni di sistema	38
3.1.11	Parametri di sistema	39
3.1.12	Comandi e parametri dei task	40
3.2	Tutti i comandi BASIC.....	41
3.2.1	+ (Addizione)	41
3.2.2	- (Sottrazione).....	41
3.2.3	* (Moltiplicazione)	41
3.2.4	/ (Divisione).....	41
3.2.5	^ (Potenza)	42
3.2.6	= (Uguale a).....	42
3.2.7	= (Assegnazione)	42
3.2.8	<> (Diverso da).....	42
3.2.9	>(Maggiore di)	43
3.2.10	>= (Maggiore di o uguale a)	43
3.2.11	< (Minore di)	43
3.2.12	<= (Minore o uguale a)	43
3.2.13	\$ (Dati esadecimali).....	44
3.2.14	' (Campo di commento)	44
3.2.15	: (Separatore di istruzioni)	44
3.2.16	#.....	44
3.2.17	ABS	45
3.2.18	ACC	45
3.2.19	ACCEL.....	45
3.2.20	ACOS	45
3.2.21	ADD_DAC	46
3.2.22	ADDAX	46
3.2.23	ADDAX_AXIS	47
3.2.24	AIN.....	47
3.2.25	AND	47
3.2.26	AOUT	48
3.2.27	ASIN	48
3.2.28	ATAN	48
3.2.29	ATAN2.....	48
3.2.30	ATYPE.....	49
3.2.31	AUTORUN.....	49
3.2.32	AXIS	49
3.2.33	AXIS_DISPLAY	50
3.2.34	AXIS_ENABLE	50
3.2.35	AXISSTATUS	50
3.2.36	B_SPLINE	51

3.2.37	BASE	51
3.2.38	BASICERROR	52
3.2.39	BATTERY_LOW	53
3.2.40	BREAK_RESET	53
3.2.41	CAM	53
3.2.42	CAMBOX	55
3.2.43	CANCEL	56
3.2.44	CHECKSUM	56
3.2.45	CHR	56
3.2.46	CLEAR	57
3.2.47	CLEAR_BIT	57
3.2.48	CLEAR_PARAMS	57
3.2.49	CLOSE_WIN	57
3.2.50	CLUTCH_RATE	57
3.2.51	COMMSERROR	58
3.2.52	COMMSTYPE	58
3.2.53	COMPILE	58
3.2.54	CONNECT	59
3.2.55	CONSTANT	59
3.2.56	CONTROL	59
3.2.57	COPY	60
3.2.58	COS	60
3.2.59	CREEP	60
3.2.60	D_GAIN	60
3.2.61	D_ZONE_MAX	61
3.2.62	D_ZONE_MIN	61
3.2.63	DAC	61
3.2.64	DAC_OUT	61
3.2.65	DAC_SCALE	61
3.2.66	DATE	62
3.2.67	DATE\$	62
3.2.68	DATUM	62
3.2.69	DATUM_IN	63
3.2.70	DAY	64
3.2.71	DAY\$	64
3.2.72	DECEL	64
3.2.73	DEFPOS	64
3.2.74	DEL	65
3.2.75	DEMAND_EDGES	65
3.2.76	DEVICENET	66

3.2.77	DIR	67
3.2.78	DISABLE_GROUP	67
3.2.79	DISPLAY	67
3.2.80	DPOS	68
3.2.81	DRIVE_ALARM	68
3.2.82	DRIVE_CLEAR	69
3.2.83	DRIVE_CONTROL	69
3.2.84	DRIVE_INPUTS	70
3.2.85	DRIVE_MONITOR	70
3.2.86	DRIVE_READ	71
3.2.87	DRIVE_RESET	71
3.2.88	DRIVE_STATUS	72
3.2.89	DRIVE_WRITE	73
3.2.90	EDIT	73
3.2.91	ELSE	74
3.2.92	ELSEIF	74
3.2.93	ENCODER	74
3.2.94	ENCODER_BITS	74
3.2.95	ENCODER_CONTROL	74
3.2.96	ENCODER_ID	75
3.2.97	ENCODER_RATIO	75
3.2.98	ENCODER_READ	75
3.2.99	ENCODER_STATUS	76
3.2.100	ENCODER_TURNS	76
3.2.101	ENCODER_WRITE	76
3.2.102	ENDIF	76
3.2.103	ENDMOVE	77
3.2.104	EPROM	77
3.2.105	ERROR_AXIS	77
3.2.106	ERROR_LINE	77
3.2.107	ERRORMASK	78
3.2.108	ETHERNET	78
3.2.109	EX	79
3.2.110	EXP	79
3.2.111	FALSE	79
3.2.112	FAST_JOG	79
3.2.113	FASTDEC	80
3.2.114	FE	80
3.2.115	FE_LATCH	80
3.2.116	FE_LIMIT	80

3.2.117	FE_LIMIT_MODE	81
3.2.118	FE_RANGE	81
3.2.119	FHOLD_IN	81
3.2.120	FHSPEED	82
3.2.121	FINS_COMMS	82
3.2.122	FLAG	84
3.2.123	FLAGS	84
3.2.124	FOR..TO..STEP..NEXT	85
3.2.125	FORWARD	86
3.2.126	FPGA_VERSION	86
3.2.127	FRAC	86
3.2.128	FRAME	86
3.2.129	FREE	87
3.2.130	FS_LIMIT	87
3.2.131	FWD_IN	87
3.2.132	FWD_JOG	88
3.2.133	GET	88
3.2.134	GLOBAL	89
3.2.135	GOSUB..RETURN	89
3.2.136	GOTO	89
3.2.137	HALT	90
3.2.138	HEX	90
3.2.139	HLM_COMMAND	90
3.2.140	HLM_READ	91
3.2.141	HLM_STATUS	92
3.2.142	HLM_TIMEOUT	93
3.2.143	HLM_WRITE	94
3.2.144	HLS_NODE	95
3.2.145	(HW_PSWITCH)	95
3.2.146	I_GAIN	96
3.2.147	IDLE	96
3.2.148	IEEE_IN	96
3.2.149	IEEE_OUT	96
3.2.150	IF..THEN..ELSE..ENDIF	97
3.2.151	IN	97
3.2.152	INDEVICE	98
3.2.153	INITIALISE	98
3.2.154	INPUT	99
3.2.155	INT	99
3.2.156	INVERT_IN	99

3.2.157	INVERT_STEP	100
3.2.158	INVERTER_COMMAND	100
3.2.159	INVERTER_READ	101
3.2.160	INVERTER_WRITE	102
3.2.161	JOGSPEED	103
3.2.162	KEY	103
3.2.163	LAST_AXIS	104
3.2.164	LINKAX	104
3.2.165	LINPUT	104
3.2.166	LIST	105
3.2.167	LIST_GLOBAL	105
3.2.168	LN	106
3.2.169	LOCK	106
3.2.170	MARK	106
3.2.171	MARKB	107
3.2.172	MECHATROLINK	107
3.2.173	MERGE	108
3.2.174	MHELICAL	109
3.2.175	MOD	109
3.2.176	MOTION_ERROR	109
3.2.177	MOVE	110
3.2.178	MOVEABS	111
3.2.179	MOVECIRC	112
3.2.180	MOVELINK	114
3.2.181	MOVEMODIFY	115
3.2.182	MPOS	116
3.2.183	MSPEED	116
3.2.184	MTYPE	116
3.2.185	NAIO	117
3.2.186	NEG_OFFSET	117
3.2.187	NEW	117
3.2.188	NEXT	117
3.2.189	NIO	117
3.2.190	NOT	118
3.2.191	NTYPE	118
3.2.192	OFF	118
3.2.193	OFFPOS	118
3.2.194	ON	119
3.2.195	ON.. GOSUB	119
3.2.196	ON.. GOTO	119

3.2.197	OP	119
3.2.198	OPEN_WIN	120
3.2.199	OR	120
3.2.200	OUTDEVICE	121
3.2.201	OUTLIMIT	121
3.2.202	OV_GAIN	121
3.2.203	P_GAIN	122
3.2.204	PI	122
3.2.205	PMOVE	122
3.2.206	POS_OFFSET	123
3.2.207	POWER_UP	123
3.2.208	PRINT	123
3.2.209	PROC	124
3.2.210	PROC_STATUS	124
3.2.211	PROCESS	125
3.2.212	PROCNUMBER	125
3.2.213	PROFIBUS	125
3.2.214	PSWITCH	126
3.2.215	RAPIDSTOP	127
3.2.216	READ_BIT	127
3.2.217	REG_POS	128
3.2.218	REG_POSB	128
3.2.219	REGIST	128
3.2.220	REMAIN	130
3.2.221	REMOTE_ERROR	130
3.2.222	RENAME	130
3.2.223	REP_DIST	131
3.2.224	REP_OPTION	131
3.2.225	REPEAT..UNTIL	131
3.2.226	RESET	132
3.2.227	RETURN	132
3.2.228	REV_IN	132
3.2.229	REV_JOG	133
3.2.230	REVERSE	133
3.2.231	RS_LIMIT	133
3.2.232	RUN	133
3.2.233	RUN_ERROR	134
3.2.234	RUNTYPE	134
3.2.235	S_REF	135
3.2.236	S_REF_OUT	135

3.2.237	SCOPE	136
3.2.238	SCOPE_POS	137
3.2.239	SELECT	137
3.2.240	SERVO	137
3.2.241	SERVO_PERIOD	137
3.2.242	SET_BIT	138
3.2.243	SETCOM	138
3.2.244	SGN	139
3.2.245	SIN	139
3.2.246	SLOT	139
3.2.247	SPEED	139
3.2.248	SQR	140
3.2.249	SRAMP	140
3.2.250	STEP	140
3.2.251	STEP_RATIO	140
3.2.252	STEPLINE	141
3.2.253	STOP	141
3.2.254	SYSTEM_ERROR	142
3.2.255	T_REF	142
3.2.256	TABLE	143
3.2.257	TABLEVALUES	143
3.2.258	TAN	144
3.2.259	THEN	144
3.2.260	TICKS	144
3.2.261	TIME	144
3.2.262	TIME\$	145
3.2.263	TO	145
3.2.264	TRANS_DPOS	145
3.2.265	TRIGGER	145
3.2.266	TROFF	145
3.2.267	TRON	146
3.2.268	TRUE	146
3.2.269	TSIZE	146
3.2.270	UNITS	147
3.2.271	UNLOCK	147
3.2.272	UNTIL	147
3.2.273	VERIFY	147
3.2.274	VERSION	147
3.2.275	VFF_GAIN	148
3.2.276	VP_SPEED	148

3.2.277	VR.....	148
3.2.278	VRSTRING	149
3.2.279	WA	149
3.2.280	WAIT IDLE.....	150
3.2.281	WAIT LOADED	150
3.2.282	WAIT UNTIL	150
3.2.283	WDOG	151
3.2.284	WHILE..WEND	151
3.2.285	XOR.....	152
4	Protocolli di comunicazione	153
4.1	Interfacce disponibili	153
4.2	Ethernet	153
4.2.1	Comunicare direttamente con Trajexia dal proprio computer.....	154
4.2.2	Comunicare a distanza con Trajexia	155
4.2.3	Protocollo Trajexia Tools	156
4.2.4	Protocollo del server FINS	156
4.2.5	Protocollo client FINS	158
4.3	Protocollo seriale	158
4.3.1	Master Host Link	159
4.3.2	Slave Host Link	163
4.3.3	Protocollo definito dall'utente	165
4.4	PROFIBUS	167
4.4.1	Introduzione.....	167
4.4.2	Configurazione della comunicazione.....	167
4.4.3	Stato della comunicazione.....	172
4.5	DeviceNet	173
4.5.1	Introduzione.....	173
4.5.2	Configurazione della comunicazione.....	173
4.5.3	Stato della comunicazione.....	178
4.6	MECHATROLINK-II	179
5	Interfaccia Trajexia Tools.....	180
5.1	Introduzione	180
5.2	Specifiche e connessioni	180
5.2.1	Specifiche del PC	180
5.2.2	Installazione del software Trajexia Tools.....	181
5.2.3	Collegamento a TJ1-MC__.....	186
5.3	Progetti	191
5.3.1	Progetti di Trajexia Tools.....	191
5.3.2	Finestra Check Project	192

5.4	Finestra delle applicazioni di Trajexia Tools	194
5.4.1	Pannello di controllo	194
5.4.2	Barra dei menu	194
5.4.3	Barra degli strumenti	195
5.5	Descrizioni dei menu	196
5.5.1	Menu Project	196
5.5.2	Menu Controller	198
5.5.3	Menu Program	203
5.5.4	Menu Tools	205
5.5.5	Menu Options	221
5.5.6	Menu Window	224
5.5.7	Menu Help	224
6	Esempi e suggerimenti.....	225
6.1	Esempi procedurali	225
6.1.1	Programma Startup	225
6.1.2	Impostazioni del guadagno	229
6.1.3	Impostazione del parametro degli assi UNITS e del rapporto di riduzione	239
6.1.4	Mappatura degli ingressi e delle uscite del servozionamento	251
6.1.5	Ricerca dell'origine	253
6.1.6	Registrazione.....	259
6.1.7	Registrazione e monitoraggio	269
6.2	Esempi pratici	279
6.2.1	Programma shell	279
6.2.2	Programma di inizializzazione	283
6.2.3	Programma per asse singolo	286
6.2.4	Posizione tramite individuazione del prodotto	287
6.2.5	Posizione su una griglia	289
6.2.6	Programma per un alimentatore di sacchetti	291
6.2.7	Tabella CAM all'interno di un programma	294
6.2.8	Programma per lama semovente	295
6.2.9	Programma di correzione	298
7	Soluzione dei problemi.....	300
7.1	Tensione e strumenti di analisi	300
7.2	TJ1-MC	300
7.2.1	Errori di sistema.....	300
7.2.2	Errori degli assi.....	300
7.2.3	Errori delle unità	301
7.2.4	Errori di configurazione.....	302
7.2.5	Sostituire la batteria.....	302

7.3	TJ1-PRT	302
7.3.1	Errori di sistema.....	302
7.3.2	Errore di comunicazione nei dati di I/O.....	303
7.4	TJ1-DRT	304
7.4.1	Errori di sistema.....	304
7.4.2	Errore di comunicazione nei dati di I/O.....	304
7.5	TJ1-ML_.....	304
7.5.1	Errori di sistema.....	304
7.5.2	Errori del bus	304
7.6	TJ1-FL02	305
7.6.1	Errori di sistema.....	305

1 Avvertenze e precauzioni di sicurezza

1.1 Destinatari del manuale

Il presente manuale è destinato a personale qualificato nella gestione di impianti elettrici (elettrotecnici o equivalenti), responsabile della progettazione, installazione e gestione di sistemi e strutture di automazione industriale.

1.2 Precauzioni generali

L'utente deve utilizzare il prodotto in base alle specifiche riportate nel presente manuale.

Prima di utilizzare il prodotto in condizioni non descritte nel manuale o di applicarlo a sistemi di controllo nucleare, sistemi ferroviari, sistemi per l'aviazione, veicoli, apparecchiature di sicurezza, stabilimenti petrolchimici e qualunque altro sistema, macchina o apparecchiatura, il cui utilizzo errato può avere un serio impatto sull'incolumità di persone o l'integrità di cose, rivolgersi al proprio rappresentante OMRON.

1.3 Precauzioni di sicurezza



AVVERTENZA

Quando l'unità è alimentata, non tentare di aprirla e non toccarne le parti interne.

Tali azioni comportano il rischio di scosse elettriche.



AVVERTENZA

Quando il sistema è alimentato, non toccare i terminali o le morsettiere.

Tali azioni comportano il rischio di scosse elettriche.



AVVERTENZA

Non cortocircuitare mai i terminali positivo e negativo delle batterie. Non caricare le batterie, né smontarle, deformarle sottoponendole a pressione o gettarle nel fuoco. Le batterie potrebbero esplodere, incendiarsi o perdere liquido.



AVVERTENZA

Il cliente è tenuto a implementare meccanismi di sicurezza per guasti ed errori allo scopo di garantire la sicurezza in caso di segnali errati, mancanti o anomali provocati da guasti a carico delle linee di segnale, cadute di tensione temporanee o altre cause. Disattendere queste precauzioni potrebbe essere causa di gravi incidenti.



AVVERTENZA

È responsabilità del cliente provvedere all'installazione sotto forma di circuiti esterni, ovvero esterni all'unità di controllo assi Trajexia, di circuiti di arresto di emergenza, circuiti di interblocco, circuiti di finecorsa e altre misure di sicurezza analoghe. Disattendere queste precauzioni potrebbe essere causa di gravi incidenti.



AVVERTENZA

Quando si verifica il sovraccarico o il cortocircuito dell'uscita a 24 Vc.c. (alimentazione I/O dell'unità TJ1), potrebbe verificarsi un abbassamento di tensione e una conseguente disattivazione delle uscite. Come soluzione a tali problemi, il sistema deve essere dotato di misure di sicurezza esterne.



AVVERTENZA

In caso di sovraccarico dei transistor di uscita (protezione), le uscite TJ1 vengono disattivate. Come soluzione a tali problemi, il sistema deve essere dotato di misure di sicurezza esterne.

**AVVERTENZA**

L'unità TJ1 disattiverà il WDOG nel caso in cui la relativa funzione di autodiagnostica rilevi un errore. Come soluzione a tali problemi, il sistema deve essere dotato di misure di sicurezza esterne.

**AVVERTENZA**

Applicare adeguate misure di sicurezza ai circuiti esterni, ovvero non interni all'unità di controllo assi Trajexia (definita "TJ1"), in modo da garantire la massima sicurezza in caso di anomalie dovute al malfunzionamento dell'unità TJ1 o ad altri fattori esterni che influiscono sul funzionamento di TJ1. Disattendere queste precauzioni potrebbe essere causa di gravi incidenti.

**AVVERTENZA**

Non tentare di smontare, riparare o modificare alcuna unità. Qualsiasi intervento in tal senso potrebbe provocare errori di funzionamento, incendi o scosse elettriche.

**Attenzione**

Verificare lo stato di sicurezza sull'unità di destinazione prima di trasferire un programma o modificare la memoria. La mancata osservanza di questa precauzione prima di procedere a tali operazioni comporta il rischio di lesioni.

**Attenzione**

Per i programmi utente scritti nell'unità di controllo assi non viene eseguito il backup automatico nella memoria flash di TJ1 (funzione di memoria flash).

**Attenzione**

Durante il cablaggio dell'alimentazione cc, fare attenzione a non invertire la polarità (+/-). Il collegamento errato potrebbe provocare errori di funzionamento nel sistema.

**Attenzione**

Serrare le viti sulla morsettiera del modulo di alimentazione, applicando la coppia specificata in questo manuale. La presenza di viti allentate può provocare bruciature o errori di funzionamento.

1.4**Precauzioni relative all'ambiente operativo****Attenzione**

Non utilizzare l'unità nei luoghi elencati qui di seguito. Disattendere questa precauzione può causare errori di funzionamento, scosse elettriche o bruciature.

- Luoghi esposti alla luce solare diretta.
- Luoghi con temperature o tassi di umidità fuori dell'intervallo di valori riportato nelle specifiche.
- Luoghi soggetti a formazione di condensa a causa di considerevoli escursioni termiche.
- Luoghi esposti a gas corrosivi o infiammabili.
- Luoghi esposti a polvere (in particolare polvere metallica) o agenti salini.
- Luoghi esposti ad acqua, oli o agenti chimici.
- Luoghi soggetti a urti o vibrazioni.

**Attenzione**

Adottare misure adeguate e sufficienti quando si installano sistemi nei luoghi elencati qui di seguito.

In caso contrario, potrebbero verificarsi errori di funzionamento.

- Luoghi soggetti a elettricità statica o interferenze di altro tipo.
- Luoghi in cui sono presenti forti campi elettromagnetici.
- Luoghi potenzialmente esposti a radioattività.
- Luoghi in prossimità di fonti di alimentazione.

**Attenzione**

L'ambiente in cui viene utilizzato il sistema TJ1 può avere un grande impatto sulla vita utile e sull'affidabilità del sistema. L'utilizzo in ambienti operativi non appropriati può provocare errori di funzionamento, guasti e altri problemi non prevedibili nel sistema TJ1.

Accertarsi che l'ambiente operativo rispetti le condizioni richieste per l'installazione e che tali condizioni siano mantenute per l'intera vita utile del sistema.

1.5 Precauzioni relative all'applicazione

**AVVERTENZA**

Avviare il sistema solo dopo aver verificato che gli assi sono presenti e sono del tipo corretto.

Il numero degli assi flessibili cambia se durante l'avvio si verificano errori di rete MECHATROLINK-II oppure se la configurazione di rete MECHATROLINK-II viene modificata.

**AVVERTENZA**

Verificare la corretta esecuzione del programma utente prima di eseguirlo sull'unità.

La mancata verifica del programma può provocare un funzionamento imprevisto.

**Attenzione**

Utilizzare sempre la tensione di alimentazione specificata nel presente manuale.

Una tensione errata potrebbe provocare errori di funzionamento o bruciature.

**Attenzione**

Adottare misure adeguate per garantire che la tensione e la frequenza nominali della corrente di alimentazione siano sempre corrette. In particolare, fare molta attenzione in luoghi dove l'alimentazione è instabile.

Un'alimentazione di corrente impropria può provocare errori di funzionamento.

**Attenzione**

Installare interruttori esterni e adottare altre misure di sicurezza per evitare cortocircuiti nel cablaggio esterno.

Misure insufficienti di protezione da cortocircuiti potrebbero causare bruciature.

**Attenzione**

Non applicare ai moduli di ingresso una tensione superiore alla tensione di ingresso nominale.

Una tensione eccessiva potrebbe causare bruciature.

**Attenzione**

Non applicare ai moduli di uscita tensioni o carichi superiori alla corrente di carico massima.

Tensioni o carichi eccessivi potrebbero causare bruciature.

**Attenzione**

Scollegare il terminale di messa a terra funzionale quando si eseguono i test per verificare la tensione di resistenza.

In caso contrario, potrebbero verificarsi bruciature.

**Attenzione**

Durante l'installazione delle unità, effettuare sempre un collegamento a terra di classe 3 (a 100Ω o inferiore). Un collegamento a terra non di classe 3 potrebbe provocare scosse elettriche.

**Attenzione**

Spegnere sempre il sistema e scollegare l'alimentazione prima di eseguire una delle operazioni elencate qui di seguito. La mancata interruzione dell'alimentazione potrebbe provocare errori di funzionamento o scosse elettriche.

- Montaggio o smontaggio di moduli di espansione, CPU o altre unità.
- Assemblaggio di moduli.
- Impostazione di DIP switch o di selettori rotanti.
- Collegamento o cablaggio dei cavi.
- Collegamento o scollegamento di connettori.

**Attenzione**

Accertarsi che tutte le viti di montaggio, le viti dei morsetti e le viti dei connettori dei cavi siano serrate in base alla coppia specificata in questo manuale. Il serraggio a una coppia non corretta potrebbe provocare errori di funzionamento.

**Attenzione**

Durante il cablaggio, lasciare l'etichetta di protezione dalla polvere attaccata all'unità. La rimozione di tale etichetta potrebbe provocare errori di funzionamento.

**Attenzione**

Una volta completato il cablaggio, rimuovere l'etichetta di protezione contro la polvere per garantire un'appropriata dispersione del calore. Lasciare attaccata l'etichetta protettiva, potrebbe provocare errori di funzionamento.

**Attenzione**

Per il cablaggio utilizzare terminali a crimpare. Non collegare direttamente ai terminali fili scoperti. Il collegamento di fili scoperti potrebbe provocare bruciature.

**Attenzione**

Controllare attentamente il cablaggio di tutti i componenti prima di attivare l'alimentazione. Un cablaggio errato può essere causa di bruciature.

**Attenzione**

Eseguire il cablaggio correttamente. Un cablaggio errato può essere causa di bruciature.

**Attenzione**

Montare l'unità solo dopo aver verificato attentamente il cablaggio della morsettiera.

**Attenzione**

Accertarsi che le morsettiere, le prolunghe e altri componenti dotati di dispositivi di bloccaggio siano correttamente bloccati in posizione. L'errato bloccaggio di questi componenti può causare errori di funzionamento.

**Attenzione**

Prima di cambiare la modalità operativa del sistema, accertarsi l'azione che non produca effetti negativi sul sistema. Disattendere questa precauzione potrebbe provocare un funzionamento imprevisto.

**Attenzione**

Riavviare il funzionamento solo dopo aver trasferito nella nuova CPU il contenuto della memoria VR e di tabella necessario per il funzionamento. Disattendere questa precauzione potrebbe provocare un funzionamento imprevisto.

**Attenzione**

Quando si sostituiscono componenti, accertarsi che le specifiche tecniche del pezzo di ricambio siano appropriate. Disattendere questa precauzione può causare errori di funzionamento o bruciature.

**Attenzione**

Non tirare né piegare i cavi oltre il limite di resistenza naturale. Ciò potrebbe provocarne la rottura.

**Attenzione**

Prima di toccare il sistema, toccare un oggetto metallico con messa a terra per scaricare l'elettricità statica accumulata. In caso contrario, potrebbero verificarsi errori di funzionamento o danni.

**Attenzione**

I cavi UTP non sono schermati. In ambienti soggetti a interferenze, utilizzare un sistema con cavo schermato a doppi intrecciati (STP) e hub appropriati per un ambiente FA. Non installare cavi a doppi intrecciati su linee ad alta tensione. Non installare cavi a doppi intrecciati nei pressi di dispositivi che generano interferenze. Non installare cavi a doppi intrecciati in luoghi soggetti a tassi di umidità elevati. Non installare cavi a doppi intrecciati in luoghi soggetti a sporco o polvere eccessivi, nebbia d'olio o altri fattori contaminanti.

**Attenzione**

Per collegare le unità, utilizzare i cavi di connessione dedicati specificati nei manuali dell'operatore. L'utilizzo di cavi per computer RS-232C normalmente reperibili in commercio può provocare guasti ai dispositivi esterni o all'unità di controllo assi.

**Attenzione**

È possibile che le uscite restino attivate a causa di un errore di funzionamento delle uscite del transistor incorporato o di altri circuiti interni. Come soluzione a tali problemi, il sistema deve essere dotato di misure di sicurezza esterne.

**Attenzione**

Quando viene attivata l'alimentazione e se un programma BASIC è impostato sulla modalità di esecuzione automatica, l'unità TJ1 inizierà a funzionare in modalità RUN.

1.6 Precauzioni per l'assemblaggio dell'unità



Attenzione

Installare l'unità in modo appropriato.
Un'installazione inappropriata dell'unità può comportare errori di funzionamento.



Attenzione

Accertarsi di montare l'unità di terminazione fornita insieme a TJ1-MC__ nell'unità all'estrema destra.
Se l'unità di terminazione non viene montata in modo appropriato, l'unità TJ1 non funzionerà correttamente.

2 Trajexia

2.1 Introduzione

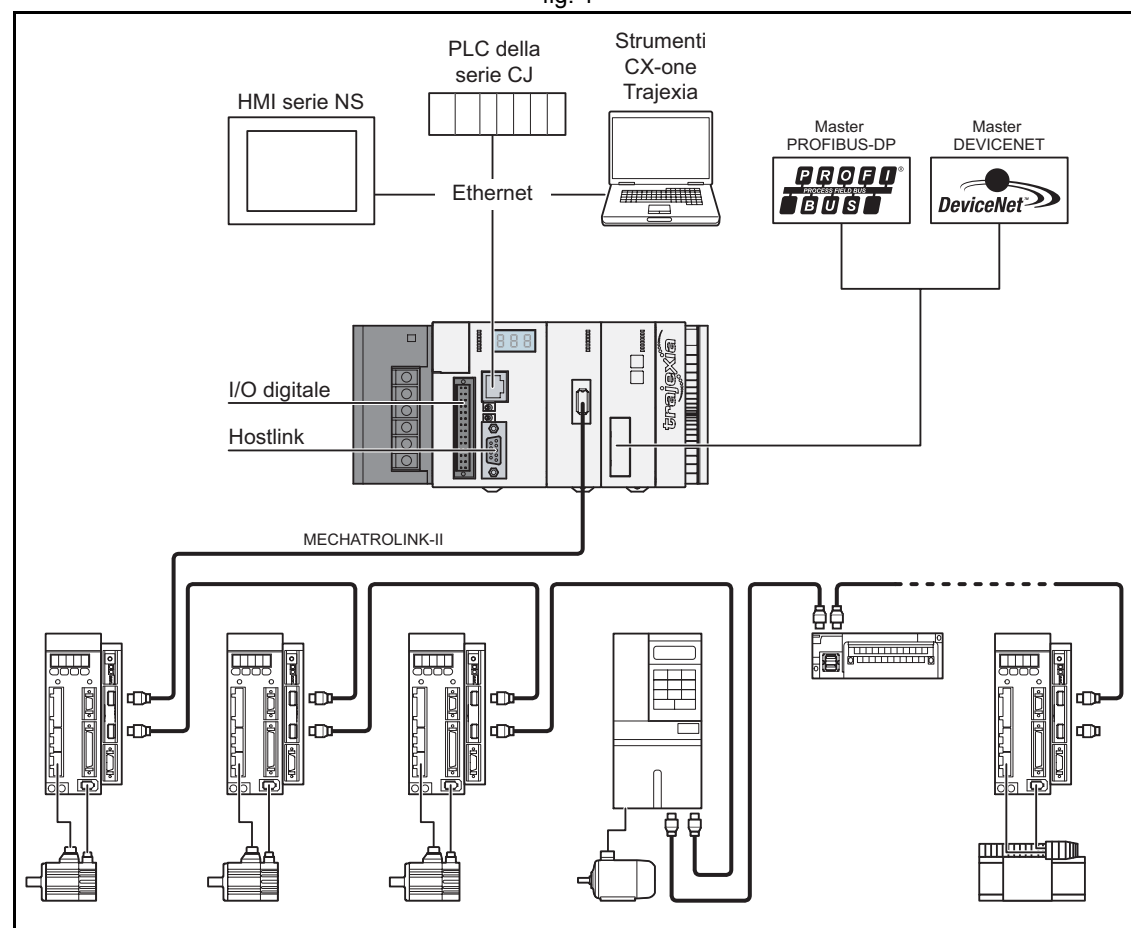
Trajexia è la piattaforma di controllo assi di OMRON che offre le prestazioni e la facilità d'uso di un sistema dedicato.

Si tratta di un sistema modulare stand alone dotato della massima flessibilità e scalabilità. L'elemento propulsivo di Trajexia è il coordinatore di movimento multitasking TJ1. Grazie a un DSP da 32 bit, è in grado di eseguire task come camma elettronica, sincronizzazione, controllo di registrazione e interpolazione usando semplici comandi di movimento.

Trajexia offre un controllo per un massimo di 16 assi su un bus MECHATROLINK-II o un controllo tradizionale analogico o a impulsi, con un monitoraggio della posizione, della velocità e della coppia indipendente per ogni asse. Inoltre, l'esauriente serie di istruzioni sul movimento, ne rende la programmazione semplice e intuitiva.

È possibile scegliere tra una vasta gamma di inverter e servo ad azionamento diretto, lineari e rotativi tra i migliori del mondo. Il sistema è scalabile fino a un massimo di 16 assi e 8 inverter e moduli di I/O.

fig. 1



2.1.1 Hardware Trajexia

L'hardware Trajexia è descritto nel Manuale di riferimento hardware di Trajexia, pertanto si consiglia di leggere innanzitutto questo manuale. Trajexia offre i vantaggi indicati di seguito

Connettività diretta via Ethernet

La porta Ethernet incorporata di Trajexia fornisce una connettività diretta e rapida per PC, PLC, HMI e altri dispositivi, fornendo al tempo stesso un accesso completo ai servomotori tramite bus MECHATROLINK-II. Consente lo scambio di messaggi espliciti tramite Ethernet e attraverso MECHATROLINK-II, offrendo totale trasparenza fino al livello di attuatore e rendendo possibile l'accesso remoto.

Tutela del know-how dell'utente

Il metodo di cifratura di Trajexia garantisce la completa protezione e riservatezza del know-how dell'utente.

Porta seriale e I/O locali

Una porta seriale fornisce una connettività diretta con qualsiasi OMRON PLC, HMI o altro dispositivo sul campo. I 16 ingressi e le 8 uscite sono I/O dell'unità controllo assi, integrati e liberamente configurabili per consentire l'impostazione di Trajexia in base al design della macchina utilizzata.

Master MECHATROLINK-II

Il master MECHATROLINK-II controlla fino a 16 servo, inverter o I/O, sempre consentendo totale trasparenza in tutto il sistema. MECHATROLINK-II offre una precisione nei tempi e una velocità di comunicazione essenziali per garantire un perfetto controllo del movimento dei servo. È possibile selezionare il tempo di ciclo del movimento tra 0,5 m/s, 1 m/s o 2 m/s.

TJ1-FL02 (modulo assi flessibile)

Il TJ1-FL02 consente di ottenere il pieno controllo di due attuatori tramite un'uscita analogica o treno di impulsi. Inoltre supporta i protocolli principali di encoder assoluti consentendo il collegamento di un encoder esterno al sistema.

Servomotori e inverter

È disponibile un'ampia scelta di servo rotanti, lineari e ad azionamento diretto, nonché di inverter, per soddisfare ogni requisito di compattezza, prestazioni e affidabilità. Gli inverter collegati a MECHATROLINK-II vengono azionati con lo stesso tempo di ciclo di aggiornamento dei servomotori.

I/O remoti:

Gli I/O del bus MECHATROLINK-II consentono l'espansione del sistema, mantenendo i dispositivi sotto un unico motion bus.

PROFIBUS-DP

Lo slave PROFIBUS-DP fornisce alla macchina la connettività con la rete PROFIBUS.

DeviceNet

Lo slave DeviceNet fornisce alla macchina la connettività con la rete DeviceNet.

2.1.2 Questo manuale

Questo Manuale di programmazione fornisce le informazioni dedicate per:

- La descrizione e l'uso dei comandi BASIC
- I protocolli di comunicazione necessari per Trajexia
- L'uso e la descrizione dei diversi elementi dell'interfaccia Trajexia Tools
- Esempi di programmi e le procedure ottimali di programmazione
- Risoluzione dei problemi e individuazione dei guasti.

2.2 Programmazione BASIC multitasking

Le unità TJ1-MC__ (unità controllo assi) dispongono di una versione multitasking del linguaggio di programmazione BASIC. Il linguaggio per il controllo assi è in larga misura basato su una versione di BASIC in formato token e i programmi sono compilati sotto forma di token prima di essere eseguiti.

Il multitasking è semplice da impostare e da usare e consente la programmazione di macchine molto complesse. Conferisce al TJ1-MC__ un considerevole vantaggio sui sistemi equivalenti a un solo task, in quanto consente applicazioni modulari laddove processi logicamente collegati possono essere raggruppati nello stesso programma del task, semplificando in tal modo l'architettura e la progettazione del codice.

Se le dimensioni della memoria lo consentono, il TJ1-MC__ può contenere fino a 14 programmi. L'esecuzione dei programmi è controllata dall'utente in BASIC. I comandi, le funzioni e i parametri BASIC qui presentati sono disponibili nel capitolo 3.

2.3 Programmazione in BASIC

Il linguaggio di programmazione BASIC consiste, tra le altre cose, in comandi, funzioni e parametri. Queste istruzioni in BASIC sono gli elementi costitutivi necessari per controllare il funzionamento di TJ1-MC__.

I comandi sono parole riconosciute dal processore che producono l'esecuzione di una certa azione, ma non restituiscono un valore. Ad esempio, **PRINT** è una parola riconosciuta che genererà la stampa, su un dispositivo di uscita specificato, del valore delle funzioni o variabili che seguono il comando.

Le funzioni sono parole riconosciute dal processore che producono l'esecuzione di una certa azione e restituiscono un valore correlato a quell'azione. Ad esempio, la funzione **ABS** prenderà il valore del relativo parametro e ne restituirà il valore assoluto, che sarà poi usato da qualche altra funzione o comando. Ad esempio **ABS(-1)** restituirà il valore 1, che potrà essere utilizzato dal comando **PRINT** per generare una stringa da inviare a un determinato dispositivo.

I parametri sono parole riconosciute dal processore, contenenti un valore specifico. Tale valore può essere letto e, sempre che non sia di sola lettura, scritto. I parametri sono utilizzati per determinare e monitorare il comportamento del sistema. Ad esempio, **ACCEL** determina la velocità di accelerazione di un determinato asse.

2.3.1 Istruzioni per assi, sistema e task

I comandi, le funzioni e i parametri si applicano a uno o più assi, ai task oppure al sistema in generale.

Istruzioni per gli assi

I comandi di controllo del movimento e i parametri degli assi si applicano a uno o più assi. I parametri degli assi determinano e controllano la reazione degli assi ai comandi dati e all'ambiente esterno. Ogni asse è dotato di un insieme di parametri che gli consente di funzionare in maniera indipendente da tutti gli altri. I comandi di controllo del movimento sono in grado di controllare contemporaneamente uno o più assi, sebbene ciascun asse abbia un comportamento proprio. A ogni avvio, i parametri degli assi vengono riportati ai valori predefiniti.

I comandi e i parametri agiscono su un asse di base o su un gruppo di assi, secondo quanto specificato nel comando **BASE**. Il comando **BASE** viene impiegato per cambiare questo gruppo di assi di base e ciascun task dispone di un gruppo proprio che può essere cambiato in qualsiasi momento. L'asse di base predefinito è 0.

Comandi o parametri che dipendono da assi individuali possono essere programmati per agire su un asse di base temporaneo includendo nel comando che dipende dall'asse la funzione **AXIS** come modificatore. Un asse di base temporaneo risulta valido solo per il comando o parametro dopo il quale compare **AXIS**.

Istruzioni per i task

I parametri dei task si applicano a task singoli e controllano, ad esempio, i task per quanto riguarda la gestione degli errori. Il modificatore **PROC** consente all'utente di accedere al parametro di un certo task. Senza **PROC** viene preso in considerazione il task corrente. Il comando **BASE** (vedere sopra) si riferisce a task specifici e può essere utilizzato con il modificatore **PROC**.

Istruzioni per il sistema

Si tratta di istruzioni che regolano le funzioni generali del sistema. In pratica, si tratta di tutte quelle istruzioni che non appartengono ai primi due gruppi.

2.3.2 Aree di memoria

Nell'unità di controllo assi di Trajexia è possibile individuare tre aree principali di memoria:

- Memoria di I/O.
- Memoria VR.
- Memoria TABLE.

Memoria di I/O

La memoria di I/O viene utilizzata per tenere i dispositivi di ingresso e di uscita collegati Trajexia. È suddivisa in due sottoaree: una per la memoria di I/O digitale e l'altra per la memoria di I/O analogica. La memoria di I/O digitale contiene gli stati di ingresso e di uscita dei dispositivi di I/O digitali. Ha una capacità di 256 bit in ingresso (punti di ingresso) e 256 bit in uscita (punti di uscita). È possibile accedere agli ingressi di questa memoria utilizzando il comando **IN**. Per accedere alle uscite, è possibile utilizzare il comando **OUT**. La memoria di I/O analogica contiene i valori di ingresso e di uscita dei dispositivi di I/O analogici. Dispone di una capacità di 36 canali di ingresso e 36 canali di uscita. I canali di ingresso analogici sono accessibili attraverso il comando **AIN**. I canali di uscita analogici sono accessibili attraverso il comando **AOUT**.

Memoria VR

La memoria VR solitamente viene usata se vi sono dati o valori che devono essere globali, vale a dire accessibili da tutti i programmi del progetto contemporaneamente. La dimensione di questa memoria è di 1024 slot con indici da 0 a 1023. Uno slot di memoria è dotato di un indirizzo che usa la macro **VR(x)** dove **x** è l'indice dello slot di memoria. È possibile accedere alla memoria VR per la lettura e la scrittura. La scrittura viene eseguita tramite assegnazione matematica utilizzando il comando **=** del programma. Il contenuto di questa memoria viene conservato nella RAM alimentata a batteria e pertanto non va perduto in assenza di alimentazione. La memoria VR viene mantenuta anche quando la batteria viene sostituita, sempre che tale sostituzione avvenga con la dovuta rapidità.

Memoria TABLE

La memoria TABLE solitamente viene usata se vi sono dati o valori che devono essere globali, vale a dire accessibili da tutti i programmi del progetto contemporaneamente. Se la memoria VR viene impiegata anch'essa per definire diversi dati e valori globali, la memoria TABLE viene utilizzata per quantitativi molto maggiori di dati globali che necessitano anche di essere organizzati secondo un determinato ordine. Per questo motivo, la memoria TABLE è solitamente utilizzata per la memorizzazione di dati, profili di movimento e log di dati di TABLE. Alcuni comandi BASIC che forniscono dati di questo tipo e dimensione, ad esempio **SCOPE**, **CAM** e **CAMBOX**, richiedono l'utilizzo della memoria TABLE per scrivere i propri risultati. La dimensione di questa memoria è di 64000 slot con indici che vanno da 0 a 63999. Anche

la memoria TABLE è accessibile per la lettura e la scrittura, ma le modalità di accesso per ciascuna di queste due funzioni è diversa. Prima di essere letto, un determinato slot di memoria TABLE deve essere in primo luogo definito e scritto utilizzando il comando **TABLE(x, valore1, valore2,...)** dove **x** è l'indice dello slot d'inizio della memoria TABLE che occorre definire e **valore1, valore2**, ecc., i valori scritti nella memoria TABLE agli indici **x, x+1**, ecc. Una volta definito e scritto, lo slot di memoria TABLE può essere letto utilizzando il comando **TABLE(x)**, in cui **x** rappresenta l'indice dello slot di memoria TABLE. Se si cerca di leggere uno slot di memoria TABLE non definito, si verifica un errore che viene riportato da TJ1-MC___. Il contenuto della memoria TABLE viene conservato nella RAM alimentata a batteria e pertanto non va perduto in caso di assenza di alimentazione. La memoria TABLE viene mantenuta anche quando la batteria viene sostituita, sempre che tale sostituzione avvenga con la dovuta rapidità.

2.3.3 Strutture dei dati e variabili

I programmi BASIC sono in grado di memorizzare dati numerici in diversi tipi di variabili. Alcune variabili hanno funzioni predefinite, ad esempio i parametri degli assi e i parametri di sistema; altre variabili possono essere definite dal programmatore se richiesto in fase di programmazione. In questa sezione vengono spiegate le variabili TABLE, globali e locali di TJ1-MC___. Viene inoltre spiegato l'uso delle etichette.

Variabili TABLE

La memoria TABLE è una struttura a matrice che contiene una serie di numeri. Questi numeri vengono impiegati, ad esempio, per specificare nel profilo le posizioni di un comando **CAM** o **CAMBOX**. Possono essere inoltre utilizzati per memorizzare dati da impiegare in futuro, ad esempio i parametri usati per definire un oggetto da elaborare.

La memoria TABLE è condivisa da tutti i task di TJ1-MC___. Questo vuol dire che i valori scritti da un task nella memoria TABLE possono essere letti da altri task.

I valori TABLE possono essere scritti e letti utilizzando il comando **TABLE**. La lunghezza massima della matrice è di 64000 elementi, da **TABLE(0)** a **TABLE(63999)**. La matrice TABLE è inizializzata fino all'elemento definito più elevato.

Variabili globali

Le variabili globali, definite nella memoria VR, sono condivise da tutti i task di TJ1-MC__. Questo vuol dire che se un programma che sta eseguendo il task 2 imposta VR(25) su un certo valore, altri programmi che stanno eseguendo task diversi possono leggere quello stesso valore da VR(25). Questo risulta particolarmente utile nella sincronizzazione di due o più task, anche se occorre assicurarsi di evitare la scrittura simultanea sulla stessa variabile da parte di più programmi. L'unità di controllo dispone di 1024 variabili globali, da VR(0) a VR(1023). Le variabili vengono lette e scritte utilizzando il comando **VR**.



I vari task in esecuzione possono accedere ai dati TABLE e VR. Quando si usano le variabili VR o TABLE, assicurarsi di usare solo un task per scrivere su una particolare variabile. Ciò consente di evitare i problemi derivanti dall'imprevista scrittura su una stessa variabile da parte di due programmi.

Variabili locali

Nei programmi è possibile dichiarare variabili nominate o variabili locali per un task particolare. Questo significa che due o più programmi eseguiti su task differenti possono utilizzare una variabile con lo stesso nome, il cui valore sarà però differente per ciascun programma. Le variabili locali possono essere lette esclusivamente dal task in cui sono dichiarate. Le variabili locali vengono sempre azzerate all'avvio di un programma. È possibile azzerare le variabili locali usando il comando **CLEAR** oppure il comando **RESET**. Il massimo numero di variabili locali che possono essere dichiarate è 255. Solo i primi 16 caratteri del nome sono significativi. Le variabili locali non definite restituiscono zero. Non è possibile dichiarare variabili locali sulla riga di comando.

Etichette

I programmi BASIC vengono eseguiti in sequenza, una riga per volta. Per alterare la sequenza di esecuzione si possono usare i comandi BASIC **GOTO** e **GOSUB**. Per definire un'etichetta, occorre collocarla come primo elemento di una riga e farla terminare con il segno dei due punti (:). Le etichette possono essere stringhe di caratteri di qualsiasi lunghezza, anche se solo i primi 15 caratteri sono significativi.

Uso delle variabili e delle etichette

Ciascun task ha le proprie etichette e le proprie variabili locali. Si prendano in considerazione, a titolo di esempio, i due programmi mostrati qui di seguito.

<pre>start: FOR a = 1 to 100 MOVE (a) WAIT IDLE NEXT a GOTO start</pre>	<pre>start: a=0 REPEAT a = a + 1 PRINT a UNTIL a = 300 GOTO start</pre>

Questi due programmi, se vengono eseguiti simultaneamente in task diversi, hanno una versione propria della variabile **a** e dell'etichetta **start**. Se è necessario conservare dati in comune tra due o più programmi, occorre usare le variabili VR. In alternativa, se il quantitativo di dati da conservare è notevole, è possibile usare la memoria TABLE. Per rendere più leggibile un programma quando si usa una variabile VR globale, è possibile ricorrere a due metodi diversi. Il primo consiste nell'usare come costante, all'interno della variabile VR, una variabile locale nominata. La variabile locale costante, tuttavia, deve essere dichiarata in tutti i programmi che usano la variabile globale VR. L'esempio di seguito mostra come usare VR(3) utilizzando questo approccio per mantenere un parametro di lunghezza condiviso da diversi programmi:

<pre>start: GOSUB Initial VR(length) = x </pre> <pre>Initial: length = 3 RETURN</pre>	<pre>start: GOSUB Initial MOVE (VR(length)) PRINT (VR(length)) </pre> <pre>Initial: length = 3 RETURN</pre>

L'altro approccio garantisce persino una maggiore leggibilità e utilizza il comando **GLOBAL** per dichiarare il nome come riferimento a una delle variabili VR globali. Il nome può quindi essere utilizzato dall'interno del

programma contenente la definizione **GLOBAL** e di tutti gli altri programmi. Prima che il nome venga usato in altri programmi è necessario assicurarsi che sia eseguito il programma contenente la definizione **GLOBAL**. La procedura ottimale consiste nel definire i nomi globali nel programma d'avvio. In questo modo, l'esempio sopra diventa:

```
'La dichiarazione nel programma d'avvio
GLOBAL length, 3

'In altri programmi eseguiti dopo il programma d'avvio
start:                                start:
    length = x                        MOVE(length)
    ...                               PRINT(length)
    ...                               ...
```

2.3.4 Specifiche matematiche

Formato numerico

Per i valori numerici, TJ1-MC__ utilizza due formati principali: numeri a virgola mobile in singola precisione e numeri interi in singola precisione. Internamente, il formato dei numeri a virgola mobile in singola precisione è un valore a 32 bit. Ha un campo esponente a 8 bit, un bit per il segno e un campo frazione a 23 bit, con un 1 implicito quale 24esimo bit. I numeri a virgola mobile hanno un intervallo valido che va da $\pm 5,9 \times 10^{-39}$ a $\pm 3,4 \times 10^{38}$. I numeri interi sono, essenzialmente, numeri a virgola mobile con esponente zero. Ciò implica che i numeri interi hanno un'ampiezza di 24 bit. L'intervallo dei numeri interi è quindi compreso tra -16.777.216 e 16.777.215. I valori numerici che ricadono fuori di questo intervallo saranno a virgola mobile.



Tutti i calcoli matematici sono eseguiti in formato a virgola mobile. Questo significa che i calcoli di/con valori più grandi possono dare risultati di precisione limitata. L'utente dovrebbe tenerlo presente nello sviluppo dell'applicazione per il controllo del movimento.

Formato esadecimale

TJ1-MC__ supporta l'assegnazione e la stampa di valori esadecimali. Un numero esadecimale viene immesso antepoendo il carattere \$ al numero. L'intervallo valido è compreso tra 0x0 e 0xFFFFF. Esempio:

```
>> VR(0)=$FF
>> PRINT VR(0)
255.0000
```

È possibile stampare un valore in formato esadecimale utilizzando la funzione **HEX**. I valori negativi producono valori esadecimali con complemento a 2 (24 bit). L'intervallo valido è compreso tra -8.388.608 e 16.777.215. Esempio:

```
>> TABLE(0, -10, 65536)
>> PRINT HEX(TABLE(0)), HEX(TABLE(1))
FFFFFF6 10000
```

Posizionamento

Per il posizionamento, TJ1-MC__ effettuerà un arrotondamento per eccesso se il valore frazionario calcolato per la distanza dal margine dell'encoder supera lo 0,9. In caso contrario, il valore frazionario sarà arrotondato per difetto. La posizione misurata interna e la posizione richiesta degli assi, rappresentata dai parametri degli assi **MPOS** e **DPOS**, hanno contatori a 32 bit.

Confronto a virgola mobile

Per evitare risultati di confronto inaspettati, la funzione di confronto considera nulla una piccola differenza tra valori diversi. Pertanto due valori con una differenza inferiore a $1,19 \times 10^{-6}$ sono considerati uguali.

Precedenza

Di seguito viene fornita la precedenza degli operatori:

1. Meno unario, **NOT**
2. ^
3. / *
4. **MOD**
5. + -
6. = <> > >= <= <
7. **AND OR XOR**
8. Da sinistra a destra

Per assicurare la precedenza dei vari operatori, il modo migliore è usare le parentesi.

2.4 Esecuzione del movimento

Ciascun task di TJ1-MC__ dispone di un insieme di buffer che mantiene le informazioni provenienti dai comandi di movimento impartiti.

2.4.1 Generatore di movimento

Il generatore di movimento dispone di due buffer di movimento per ciascun asse. Un buffer, denominato **MTYPE**, contiene il movimento effettivo, vale a dire il movimento in esecuzione sull'asse in quel momento. L'altro buffer, chiamato **NTYPE**, contiene il movimento successivo, che viene eseguito una volta che il movimento effettivo è stato completato. Per una spiegazione dettagliata, vedere capitolo 2.8 "Buffer di movimento" nel Manuale di riferimento hardware di Trajexia.

I programmi BASIC sono separati dal programma del generatore di movimento, che controlla il movimento degli assi. Il generatore di movimento dispone di funzioni separate per ciascun asse. Ogni asse può essere programmato con parametri propri (ad esempio velocità, accelerazione) e può muoversi in maniera indipendente e simultanea oppure essere collegato ad altri assi utilizzando comandi speciali.

Quando un comando di movimento viene elaborato, il generatore di movimento attende che il movimento sia terminato e il buffer relativo all'asse interessato si sia vuotato, prima di caricarvi l'informazione di movimento successiva.



Se i buffer dei task sono pieni, l'esecuzione del programma viene sospesa fino a quando i buffer non sono di nuovo disponibili. Ciò vale anche per i task della riga di comando: nessun comando può essere dato in tale circostanza. In un caso simile, Trajexia Tools si scollegherà. Il parametro dei task **PMOVE** sarà impostato su TRUE quando i buffer dei task sono pieni e reimpostato su FALSE quando sono di nuovo disponibili.

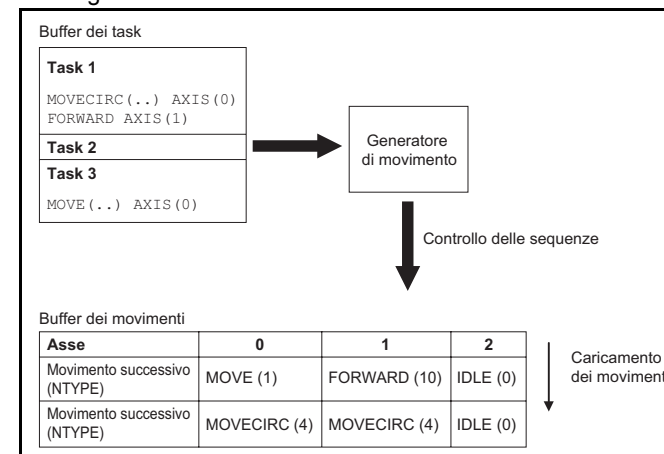
2.4.2 Messa in sequenza

Ogni volta che si interrompe un ciclo di servoazionamento (vedere la sezione 2.6.3), il generatore di movimento esamina i buffer **NTYPE** per vedere se ve ne sono di disponibili. In caso affermativo, verifica i buffer dei task per vedere se vi è un movimento in attesa di essere caricato. Se un movimento può essere caricato, i dati relativi a tutti gli assi specificati vengono caricati dai buffer dei task ai buffer **NTYPE** e i buffer dei task corrispondenti vengono contrassegnati come inattivi. Questo processo viene denominato controllo delle sequenze.

2.4.3 Caricamento dei movimenti

Quando il controllo delle sequenze è terminato, i buffer **MTYPE** vengono sottoposti a verifica per vedere se è possibile caricare dei movimenti. Se i buffer **MTYPE** richiesti sono disponibili, il movimento viene caricato dai buffer **NTYPE** nei buffer **MTYPE** e i buffer **NTYPE** vengono contrassegnati come inattivi. Questo processo viene denominato caricamento del movimento. Se i buffer **MTYPE** contengono un movimento valido, questo viene elaborato. Quando il movimento è stato completato, i buffer **MTYPE** vengono contrassegnati come inattivi.

fig. 2



2.5 Interfaccia della riga di comando

L'interfaccia della riga di comando offre all'utente un'interfaccia diretta per eseguire i comandi e accedere ai parametri del sistema. Quando TJ1-MC__ è collegato, utilizzare la finestra del terminale di Trajexia Tools. Per ulteriori dettagli, vedi sezione 5.5.4. TJ1-MC__ colloca in un buffer gli ultimi 10 comandi impartiti sulla riga di comando. Premendo i tasti cursore su e giù, è possibile passare in modo ciclico da un comando nel buffer all'altro.

2.6 Programmi in Basic

TJ1-MC__ può memorizzare fino a 14 programmi, a condizione che la capacità della memoria non venga superata. Per la gestione di questi file di programma, TJ1-MC__ supporta semplici istruzioni di gestione molto simili al sistema di archiviazione DOS di un computer. Il pacchetto software Trajexia Tools viene impiegato per memorizzare e caricare i programmi in un computer per l'archiviazione, la stampa e la modifica. Dispone inoltre di diverse funzionalità per il monitoraggio e il debug dell'unità di controllo. Fare riferimento al capitolo 5.

2.6.1 Gestione dei programmi

Trajexia Tools crea automaticamente un progetto che contiene i programmi da usare per un'applicazione. I programmi del progetto sono tenuti sia nell'unità di controllo che nel computer. Ogniqualvolta un programma viene creato o modificato, Trajexia Tools modifica entrambe le copie consentendo così di avere sempre un backup accurato fuori dell'unità di controllo. Trajexia Tools verifica che le due versioni del progetto siano identiche, attuando un controllo di ridondanza ciclica. Se le due versioni sono diverse, Trajexia Tools consente di copiare la versione di TJ1-MC__ sul disco e viceversa. I programmi del computer sono memorizzati in un file di testo ASCII. Possono pertanto essere stampati, modificati e copiati usando un semplice editor di testo. TJ1-MC__ contiene i programmi originali sotto forma di token; per questo motivo gli stessi programmi avranno dimensioni inferiori su TJ1-MC__ che non sul computer.

Memorizzazione dei programmi

I programmi di TJ1-MC__ sono conservati nella RAM alimentata a batteria e pertanto non vanno perduti in caso di assenza di alimentazione. Si tratta di un sistema analogo a quello della memoria VR e della memoria TABLE. Il contenuto della memoria RAM per i programmi viene mantenuto quando la batteria si trova in TJ1-MC__. I programmi non vanno perduti neppure quando la batteria viene sostituita, sempre che tale sostituzione avvenga con la dovuta rapidità. Per conservare i programmi in assenza di batteria per periodi più lunghi è necessario copiare i programmi nella memoria Flash dell'unità di controllo utilizzando il comando **EPROM** e quindi leggerli quando si riattiva l'alimentazione; quest'ultima operazione è determinata dal parametro di sistema **POWER_UP**.

Comandi di programma

TJ1-MC__ dispone di diversi comandi BASIC che consentono la creazione, la modifica e la cancellazione dei programmi. Trajexia Tools è dotato di pulsanti che eseguono anche queste operazioni; pertanto solitamente non è richiesto l'uso di questi comandi nei programmi.

Comando	Funzione
SELECT	Seleziona un programma per la modifica, la cancellazione, ecc.
NEW	Cancella il programma attualmente selezionato, un programma specificato o tutti i programmi.
DIR	Mostra la directory con tutti i programmi.
COPY	Crea il duplicato di un programma specificato.
RENAME	Rinomina un programma specificato.
DEL	Cancella il programma attualmente selezionato o un programma specificato.
LIST	Mostra il programma attualmente selezionato o un programma specificato.

2.6.2 Compilazione di programmi

Il sistema TJ1-MC__ compila i programmi automaticamente se necessario. Normalmente non è necessario forzare la compilazione dei programmi da parte di TJ1-MC__. È tuttavia possibile compilare programmi mediante il menu **Program** di Trajexia Tools.

Il sistema TJ1-MC__ compila i programmi automaticamente nelle seguenti circostanze.

- Se è stato modificato, il programma selezionato viene compilato prima di essere eseguito.
- Se il programma selezionato è stato modificato prima di passare a un altro programma, esso viene compilato.
- Il programma selezionato viene compilato usando il comando **COMPILE**.

Durante la compilazione, la sintassi e la struttura del programma vengono verificate. Se la compilazione non riesce, verrà visualizzato un messaggio e non sarà generato alcun codice di programma. Nella casella della directory di Trajexia Tools comparirà una croce rossa.

Non è possibile eseguire i programmi quando si verificano errori di compilazione. Gli errori devono essere corretti e il programma ricompilato. Il processo di compilazione include anche quando segue:

- Rimozione dei commenti.
- Compilazione dei numeri nel formato del processore interno.
- Conversione delle espressioni in notazione polacca inversa per l'esecuzione.
- Precalcolo delle posizioni variabili.
- Calcolo e integrazione delle destinazioni con struttura ad anello.



Dal momento che il processo di compilazione richiede della memoria libera, è possibile che si verifichino errori di compilazione quando la memoria disponibile non è sufficiente.

2.6.3 Esecuzione del programma

La temporizzazione dell'esecuzione dei vari task e l'aggiornamento degli I/O di TJ1-MC__ sono incentrati sul periodo del ciclo di servoazionamento del sistema. Tale periodo è determinato dal parametro di sistema **SERVO_PERIOD**. Il periodo del ciclo di servoazionamento di TJ1-MC__ è di 0,5, 1,0 o 2,0 m/s.

Aggiornamento degli I/O

Lo stato degli I/O di TJ1-MC__ viene aggiornato all'inizio di ciascun ciclo di servoazionamento.

- Lo stato acquisito degli ingressi digitali viene trasferito alla variabile di ingresso del sistema **IN**. Notare che questo è lo stato acquisito nel ciclo di servoazionamento precedente.
- Le uscite analogiche delle velocità di riferimento vengono aggiornate.
- Le uscite digitali sono aggiornate in base allo stato della variabile di uscita del sistema **OP**.
- Viene acquisito lo stato degli ingressi digitali.

Notare che non si verifica alcuna elaborazione automatica dei segnali di I/O, salvo che per la registrazione. Ciò significa che tutte le azioni devono essere programmate all'interno dei programmi BASIC.

Comandi idonei

Trajexia Tools consente di eseguire, mettere in pausa e arrestare i programmi in diversi modi, tramite i pulsanti del pannello di controllo e le finestre di modifica. Per controllare l'esecuzione è possibile inserire i seguenti comandi sulla riga di comando.

Comando	Funzione
RUN	Esegue il programma attualmente selezionato o un programma specificato; a scelta, su un numero di task specificato.
STOP	Arresta il programma attualmente selezionato o un programma specificato.
HALT	Arresta tutti i programmi del sistema.
PROCESS	Visualizza tutti i task in esecuzione.

L'utente può allocare esplicitamente la priorità dei task in base alla quale si desidera che venga eseguito il programma BASIC. Quando un programma utente viene eseguito senza una specifica allocazione di task, gli viene assegnata la priorità di task disponibile più elevata.

Impostazione dei programmi da eseguire all'avvio

È possibile stabilire l'esecuzione automatica dei programmi al momento dell'accensione a livelli di priorità diversi. Se necessario, il computer può restare collegato come interfaccia operatore oppure può essere rimosso, lasciando i programmi in esecuzione autonoma.

L'esecuzione automatica dei programmi all'avvio viene impostata in Trajexia Tools attraverso la selezione **Set Power Up Mode...** del menu **Program**.

Questa operazione stabilisce quali programmi devono essere eseguiti automaticamente e con quale priorità. Lo stesso risultato si ottiene con il comando BASIC **RUNTYPE**, mentre con il comando **DIR** è possibile visualizzare lo stato corrente.

Per maggiori informazioni su controllo dei programmi, multitasking e tempi di ciclo, vedere le sezioni 2.2 e 2.3 del Manuale di riferimento hardware di Trajexia.

3 Comandi BASIC

3.1 Categorie

In questa sezione sono elencati tutti i comandi BASIC divisi per categorie.

Le categorie sono:

- Comandi degli assi.
- Parametri degli assi.
- Comandi e parametri di comunicazione.
- Costanti.
- Comandi, funzioni e parametri di I/O.
- Funzioni e operazioni matematiche.
- Comandi di programma.
- Comandi di controllo del programma.
- Parametri e modificatori di slot.
- Comandi e funzioni di sistema.
- Parametri di sistema.
- Comandi e parametri dei task.

Gli elenchi forniscono solo un riferimento di rapida consultazione. Nella prossima sezione, i comandi sono elencati in ordine alfabetico e corredati di una descrizione completa.

3.1.1 Comandi degli assi

Nome	Descrizione
ACC	Cambia ACCEL e DECEL contemporaneamente.
ADD_DAC	Somma il valore DAC di un asse all'uscita analogica dell'asse di base.
ADDAX	Stabilisce un collegamento a un asse sovrapposto. Tutti i movimenti della posizione richiesta per l'asse sovrapposto saranno aggiunti a qualsiasi movimento attualmente in corso di esecuzione.
B_SPLINE	Espande il profilo archiviato nella memoria TABLE utilizzando la funzione matematica B-Spline.
BASE	Utilizzato per impostare l'asse di base a cui vengono applicati i comandi e i parametri.

Nome	Descrizione
CAM	Sposta un asse in base ai valori di un profilo di movimento archiviato nella matrice di variabili della memoria TABLE .
CAMBOX	Sposta un asse in base ai valori di un profilo di movimento archiviato nella matrice di variabili della memoria TABLE . Il movimento è collegato al movimento misurato di un altro asse, formando così una trasmissione software a variazione continua.
CANCEL	Cancella il movimento di un asse.
CONNECT	Collega la posizione richiesta di un asse ai movimenti misurati dell'asse che è stato specificato per asse_motore al fine di fornire una trasmissione elettronica.
DATUM	Esegue una delle 7 sequenze di ricerca dell'origine per posizionare un asse su una posizione assoluta o reimpostare un errore di movimento.
DEFPOS	Definisce la posizione corrente come nuova posizione assoluta.
DISABLE_GROUP	Raggruppa gli assi per la disabilitazione degli errori.
DRIVE_ALARM	Controlla l'allarme corrente.
DRIVE_CLEAR	Cancella lo stato di allarme del servozionamento.
DRIVE_READ	Legge il parametro specificato del servozionamento.
DRIVE_RESET	Reimposta il servozionamento.
DRIVE_WRITE	Scriva un valore specifico nel parametro specificato del servozionamento.
ENCODER_READ	Legge un parametro dell'encoder assoluto EnDat.
ENCODER_WRITE	Scriva un parametro dell'encoder assoluto EnDat.
FORWARD	Sposta in avanti un asse costantemente alla velocità impostata nel parametro SPEED .
(HW_PSWITCH)	Attiva e disattiva l'interruttore hardware all'uscita 0 di TJ1-FL02, quando le posizioni predefinite vengono raggiunte.
MECHATROLINK	Inizializza il bus MECHATROLINK-II ed esegue varie operazioni sulle stazioni MECHATROLINK-II collegate al bus.
MHELICAL	Interpola 3 assi ortogonali producendo un movimento elicoidale.
MOVE	Sposta alla velocità, accelerazione e decelerazione richieste uno o più assi nella posizione specificata, come incremento rispetto alla posizione corrente.

Nome	Descrizione
MOVEABS	Sposta alla velocità, accelerazione e decelerazione richieste uno o più assi nella posizione specificata, come posizione assoluta.
MOVECIRC	Interpola 2 assi ortogonali producendo un arco di circonferenza.
MOVELINK	Crea un movimento lineare dell'asse di base collegato tramite trasmissione software alla posizione misurata di un asse di collegamento.
MOVEMODIFY	Cambia la posizione finale assoluta del movimento lineare corrente ad asse singolo (MOVE o MOVEABS).
RAPIDSTOP	Cancella il movimento corrente da tutti gli assi.
REGIST	Acquisisce la posizione di un asse quando sull'encoder viene individuato un ingresso di registrazione o il fase Z.
REVERSE	Sposta indietro un asse costantemente alla velocità impostata nel parametro SPEED .
STEP_RATIO	Imposta il rapporto per l'uscita del motore passo-passo dell'asse.

3.1.2 Parametri degli assi

Nome	Descrizione
ACCEL	Contiene la velocità di accelerazione dell'asse.
ADDAX_AXIS	Contiene il numero dell'asse al quale l'asse di base è attualmente collegato tramite ADDAX .
ATYPE	Contiene il tipo di asse.
AXIS_DISPLAY	Seleziona le informazioni che sono rappresentate dai LED sul coperchio frontale di TJ1-FL02.
AXIS_ENABLE	Abilita e disabilita un asse particolare, indipendentemente da un altro asse.
AXISSTATUS	Contiene lo stato dell'asse.
CLOSE_WIN	Definisce la fine della finestra in cui è previsto la tacca di registrazione.
CLUTCH_RATE	Definisce il cambiamento del rapporto di connessione quando viene utilizzato il comando CONNECT .
CREEP	Contiene la velocità di slittamento.

Nome	Descrizione
D_GAIN	Contiene il guadagno di controllo derivativo.
DAC_SCALE	Imposta la scala e la polarità applicate ai valori DAC .
DATUM_IN	Contiene il numero d'ingresso da usare come ingresso dell'origine.
DECEL	Contiene la velocità di decelerazione dell'asse.
DEMAND_EDGES	Contiene il valore corrente del parametro degli assi DPOS contenuto nei fronti dell'encoder.
DPOS	Contiene la posizione richiesta generata dai comandi di movimento.
DRIVE_CONTROL	Seleziona i dati da monitorare attraverso DRIVE_MONITOR per gli assi collegati tramite il bus MECHATROLINK-II. Per gli assi collegati tramite TJ1-FL02, DRIVE_CONTROL imposta le uscite di TJ1-FL02.
DRIVE_INPUTS	Contiene i dati di I/O del driver collegato al bus MECHATROLINK-II. I dati vengono aggiornati a ogni ciclo di servoazionamento.
DRIVE_MONITOR	Controlla i dati del servoazionamento collegato al bus MECHATROLINK-II. I dati vengono aggiornati a ogni ciclo di servoazionamento.
DRIVE_STATUS	Contiene lo stato corrente del servoazionamento.
ENCODER	Contiene una copia non elaborata del registro hardware dell'encoder.
ENCODER_BITS	Imposta il numero di bit per l'encoder assoluto collegato a TJ1-FL02.
ENCODER_CONTROL	Controlla la modalità operativa dell'encoder assoluto EnDat.
ENCODER_ID	Restituisce il valore ID dell'encoder assoluto collegato a TJ1-FL02.
ENCODER_RATIO	Imposta il valore di scala per i conteggi encoder in entrata.
ENCODER_STATUS	Restituisce lo stato dell'encoder assoluto Tamagawa.
ENCODER_TURNS	Restituisce il conteggio multigiro dell'encoder assoluto.
ENDMOVE	Mantiene la posizione finale del movimento corrente.
ERRORMASK	Contiene il valore di maschera che determina se MOTION_ERROR si verifica in base allo stato dell'asse.
FAST_JOG	Contiene il numero d'ingresso da usare come ingresso di jog rapido.
FASTDEC	Definisce il rapporto di decelerazione da rampa a zero quando un asse raggiunge il fincorsa o la posizione.

Nome	Descrizione
FE	Contiene l'errore di inseguimento (Following Error, FE).
FE_LATCH	Contiene il valore FE che ha fatto sì che l'asse mettesse l'unità di controllo nello stato MOTION_ERROR .
FE_LIMIT	Contiene il valore massimo ammesso per l'errore di inseguimento.
FE_LIMIT_MODE	Definisce in che modo FE influenzi lo stato MOTION_ERROR .
FE_RANGE	Contiene i limiti per l'avvertenza relativa all'errore di inseguimento.
FHOLD_IN	Contiene il numero d'ingresso da usare come ingresso per la sospensione dell'alimentazione.
FHSPEED	Contiene la velocità di sospensione dell'alimentazione.
FS_LIMIT	Contiene la posizione assoluta del limite software di avanzamento.
FWD_IN	Contiene il numero d'ingresso da usare come ingresso per il limite di avanzamento.
FWD_JOG	Contiene il numero d'ingresso da usare come ingresso di un jog in avanti.
I_GAIN	Contiene il guadagno di controllo integrale.
INVERT_STEP	Indirizza un inverter hardware al circuito di uscita del motore passo-passo.
JOGSPEED	Imposta la velocità di jog.
LINKAX	Contiene il numero dell'asse di collegamento durante qualsiasi movimento collegato.
MARK	Individua l'evento di registrazione primario in un ingresso di registrazione.
MARKB	Individua l'evento di registrazione secondario in un ingresso di registrazione.
MERGE	È un interruttore software che può essere utilizzato per abilitare o disabilitare l'unione di movimenti consecutivi.
MPOS	È la posizione dell'asse misurata dall'encoder.
MSPEED	Rappresenta il cambiamento della posizione rilevata nell'ultimo ciclo di servozionamento.
MTYPE	Contiene il tipo di movimento attualmente in esecuzione.

Nome	Descrizione
NTYPE	Contiene il tipo di movimento presente nel buffer dei movimenti successivi.
OFFPOS	Contiene un offset che sarà applicato alla posizione richiesta senza influenzare il movimento in nessun altro modo.
OPEN_WIN	Definisce l'inizio della finestra in cui è previsto la tacca di registrazione.
OUTLIMIT	Contiene il limite che circonda l'uscita della velocità di riferimento da TJ1-MC__.
OV_GAIN	Contiene il guadagno per il controllo della velocità di uscita.
P_GAIN	Contiene il guadagno per il controllo proporzionale.
REG_POS	Contiene la posizione in cui si è verificato un evento di registrazione.
REG_POSB	Contiene la posizione in cui si è verificato l'evento di registrazione secondario.
REMAIN	È la distanza rimanente fino alla fine del movimento corrente.
REMOTE_ERROR	Restituisce il numero di errori sulla connessione MECHATROLINK-II del servozionamento.
REP_DIST	Contiene o imposta la distanza di ripetizione.
REP_OPTION	Controlla l'applicazione del parametro REP_DIST per gli assi.
REV_IN	Contiene il numero d'ingresso da usare come ingresso per il limite di marcia indietro.
REV_JOG	Contiene il numero d'ingresso da usare come ingresso di un jog indietro.
RS_LIMIT	Contiene la posizione assoluta del limite software di marcia indietro.
S_REF	Contiene il valore della velocità di riferimento che viene applicata quando l'asse si trova in modalità ad anello aperto.
S_REF_OUT	Contiene il valore della velocità di riferimento che viene applicata al servozionamento sia in modalità ad anello chiuso che in modalità ad anello aperto.
SERVO	Determina se l'asse funziona in modalità di controllo servo oppure ad anello aperto.
SPEED	Contiene la velocità richiesta in unità/s.
SRAMP	Contiene il fattore della curva a S.

Nome	Descrizione
T_REF	Contiene il valore della coppia di riferimento che viene applicata al servomotore.
TRANS_DPOS	Contiene la posizione richiesta per l'asse all'uscita della trasformazione del contorno.
UNITS	Contiene il fattore di conversione delle unità.
VERIFY	Seleziona modalità operative differenti sull'asse di un'uscita motore passo-passo.
VFF_GAIN	Contiene il guadagno di velocità per il feed-forward.
VP_SPEED	Contiene la velocità del profilo di velocità.

3.1.3 Comandi e parametri di comunicazione

Nome	Descrizione
FINS_COMMS	Invia la memoria di lettura e la memoria di scrittura FINS a un server FINS designato.
HLM_COMMAND	Esegue un comando Host Link specifico sullo slave.
HLM_READ	Legge i dati dello slave Host Link trasferendoli alla matrice di variabili VR o TABLE.
HLM_STATUS	Rappresenta lo stato dell'ultimo comando del master Host Link.
HLM_TIMEOUT	Definisce il tempo di timeout del master Host Link.
HLM_WRITE	Scrive i dati sullo slave Host Link trasferendoli dalla matrice di variabili VR o TABLE.
HLS_NODE	Definisce il numero di modulo dello slave per il protocollo slave Host Link.
SETCOM	Imposta le comunicazioni seriali.

3.1.4 Costanti

Nome	Descrizione
FALSE	Pari al valore numerico 0.
OFF	Pari al valore numerico 0.
ON	Pari al valore numerico 1.
PI	Pari al valore numerico 3,1416.
TRUE	Pari al valore numerico -1.

3.1.5 Comandi, funzioni e parametri di I/O

Nome	Descrizione
GET	Attende l'arrivo di un unico carattere e assegna il codice ASCII del carattere alla variabile.
IN	Restituisce il valore degli ingressi digitali.
INDEVICE	Definisce i parametri del dispositivo di ingresso predefinito.
INPUT	Attende la ricezione di una stringa e assegna il valore numerico alla variabile.
KEY	Restituisce TRUE o FALSE a seconda che il carattere sia stato ricevuto oppure no.
LINPUT	Attende una stringa e l'inserisce nelle variabili VR.
OP	Imposta una o più uscite oppure restituisce lo stato delle prime 24 uscite.
OUTDEVICE	Definisce il dispositivo di uscita predefinito.
PRINT	Invia una serie di caratteri a una porta seriale.
PSWITCH	Attiva un'uscita quando viene raggiunta una posizione predefinita e disattiva l'uscita quando viene raggiunta una seconda posizione.

3.1.6 Funzioni e operandi matematici

Nome	Descrizione
+ (ADDIZIONE)	Somma due espressioni.
- (SOTTRAZIONE)	Calcola la differenza tra due espressioni.
* (MOLTIPLICAZIONE)	Moltiplica due espressioni.
/ (DIVISIONE)	Calcola la divisione due espressioni.
^ (POTENZA)	Trasferisce l'esponente di un'espressione ad un'altra.
= (UGUALE A)	Verifica due espressioni per vedere se sono uguali.
= (ASSEGNAZIONE)	Assegna un'espressione a una variabile.
<> (DIVERSO DA)	Controlla due espressioni per vedere se sono diverse.
> (MAGGIORE DI)	Verifica due espressioni per vedere se l'espressione di sinistra è maggiore di quella di destra.
>= (MAGGIORE DI O UGUALE A)	Verifica due espressioni per vedere se l'espressione di sinistra è maggiore o uguale a quella di destra.
< (MINORE DI)	Verifica due espressioni per vedere se l'espressione di sinistra è minore di quella di destra.
<= (MINORE O UGUALE A)	Verifica due espressioni per vedere se l'espressione di sinistra è minore o uguale a quella di destra.
ABS	Restituisce il valore assoluto di un'espressione.
ACOS	Restituisce l'arcocoseno di un'espressione.
AND	Esegue un'operazione AND sui bit corrispondenti dei numeri interi di due espressioni.
ASIN	Restituisce l'arcoseno di un'espressione.
ATAN	Restituisce l'arcotangente di un'espressione.
ATAN2	Restituisce l'arcotangente di un numero complesso diverso da zero dato da due espressioni.
COS	Restituisce il coseno di un'espressione.
EXP	Restituisce il valore esponenziale di un'espressione.
FRAC	Restituisce la parte frazionaria di un'espressione.

Nome	Descrizione
IEEE_IN	Restituisce il numero a virgola mobile in formato IEEE, rappresentato da 4 byte.
IEEE_OUT	Restituisce il singolo byte estratto dal numero a virgola mobile in formato IEEE.
INT	Restituisce il numero intero di un'espressione.
LN	Restituisce il logaritmo naturale di un'espressione.
MOD	Restituisce il modulo di due espressioni.
NOT	Esegue un'operazione NOT sui bit corrispondenti dei numeri interi di due espressioni.
OR	Esegue un'operazione OR tra i bit corrispondenti dei numeri interi di due espressioni.
SGN	Restituisce il segno di un'espressione.
SIN	Restituisce il seno di un'espressione.
SQR	Restituisce la radice quadrata di un'espressione.
TAN	Restituisce la tangente di un'espressione.
XOR	Esegue una funzione XOR tra i bit corrispondenti dei numeri interi di due espressioni.

3.1.7 Comandi di programma

Nome	Descrizione
' (CAMPO DI COMMENTO)	Consente a una riga di non essere eseguita.
: (SEPARATORE DI ISTRUZIONI)	Consente la presenza di più istruzioni su una riga.
AUTORUN	Avvia tutti i programmi che sono stati impostati per essere eseguiti all'avvio.
COMPILE	Compila il programma corrente.
COPY	Copia in un nuovo programma un programma esistente nell'unità di controllo assi.

Nome	Descrizione
DEL	Elimina un programma dall'unità di controllo assi.
DIR	Visualizza una lista dei programmi presenti nell'unità di controllo assi, la loro dimensione e il loro RUNTYPE sull'uscita standard.
EDIT	Consente la modifica di un programma tramite un terminale VT100.
EPROM	Memorizza un programma nella memoria flash.
LIST	Stampa il programma sull'uscita standard.
NEW	Elimina tutte le righe del programma dell'unità di controllo assi.
PROCESS	Restituisce lo stato di funzionamento e il numero di ogni task in esecuzione.
RENAME	Cambia il nome di un programma dell'unità di controllo assi.
RUN	Esegue un programma.
RUNTYPE	Determina se un programma viene azionato all'avvio e su quale task deve essere eseguito.
SELECT	Specifica il programma corrente.
STEPLINE	Esegue una singola riga di un programma.
STOP	Interrompe l'esecuzione del programma.
TROFF	Sospende una registrazione alla riga corrente e riprende la normale esecuzione del programma.
TRON	Crea un'interruzione in un programma.

3.1.8 Comandi di controllo del programma

Nome	Descrizione
FOR..TO..STEP..NEXT	L'esecuzione ad anello consente al segmento di un programma di essere ripetuto con un aumento o una diminuzione della variabile.
GOSUB..RETURN	Passa a una subroutine della riga immediatamente successiva all'etichetta. L'esecuzione del programma torna all'istruzione successiva se viene fornito un "RETURN" a pagina 132.
GOTO	Passa alla riga contenente l'etichetta.

Nome	Descrizione
IF..THEN..ELSE..ENDIF	Controlla il flusso del programma in base ai risultati della condizione.
ON.. GOSUB o ON.. GOTO	Attiva un conditional jump verso una delle diverse etichette.
REPEAT..UNTIL	L'esecuzione ad anello consente la ripetizione del segmento di programma fino a quando la condizione diventa "TRUE" a pagina 146.
WHILE..WEND	L'esecuzione ad anello consente la ripetizione del segmento di programma fino a quando la condizione non diventa FALSE .

3.1.9 Parametri e modificatori di slot

Nome	Descrizione
COMMSTYPE	Contiene il tipo di unità presente nello slot di un'unità di controllo.
FPGA_VERSION	Restituisce la versione FPGA dell'unità con numero_unità nel sistema di un'unità di controllo.
SLOT	È un modificatore che specifica il numero di slot dell'unità.

3.1.10 Comandi e funzioni di sistema

Nome	Descrizione
\$ (DATI ESADECIMALI)	Assegna un numero esadecimale a una variabile.
AXIS	Imposta l'asse per un comando, la lettura dei parametri degli assi o l'assegnazione a un asse particolare.
BASICERROR	È utilizzato per eseguire una specifica routine quando si verifica un errore in un comando BASIC.
CLEAR	Cancella tutte le variabili globali e locali dal task corrente.
CLEAR_BIT	Cancella il bit specificato della variabile VR specificata.
CLEAR_PARAMS	Riporta ogni variabile e ogni parametro archiviato nella memoria flash EPROM alle impostazioni predefinite.

Nome	Descrizione
CONSTANT	Dichiara una costante da utilizzare in un programma BASIC.
DATE\$	Stampa la data corrente come stringa.
DAY\$	Stampa il giorno corrente come stringa.
DEVICENET	Configura TJ1-DRT (unità slave DeviceNet) per lo scambio di dati o restituisce lo stato di scambio dati di TJ1-DRT.
ETHERNET	Legge e imposta vari parametri della porta Ethernet di TJ1-MC__.
EX	Reimposta l'unità di controllo.
FLAG	Imposta e legge un insieme di 32 bit.
FLAGS	Legge e imposta FLAGS come blocco.
FREE	Restituisce la quantità di memoria disponibile.
GLOBAL	Dichiara un riferimento a una delle variabili VR.
HALT	Arresta tutti i programmi al momento in esecuzione.
INITIALISE	Imposta tutti gli assi e i parametri sui valori predefiniti.
INVERT_IN	Inverte i canali d'ingresso 0 – 31 del software.
INVERTER_COMMAND	Legge l'I/O e cancella l'allarme dell'inverter di frequenza.
INVERTER_READ	Legge parametro, allarme, velocità e coppia di riferimento dell'inverter di frequenza.
INVERTER_WRITE	Scriva parametro, velocità e coppia di riferimento dell'inverter di frequenza.
LIST_GLOBAL	Mostra tutte le variabili GLOBAL e CONSTANT.
LOCK	Impedisce che i programmi vengano visualizzati o modificati.
PROFIBUS	Configura TJ1-PRT (unità slave PROFIBUS-DP) per lo scambio di dati di I/O con il master e restituisce lo stato di TJ1-PRT.
READ_BIT	Restituisce il valore del bit specificato nella variabile VR specificata.
RESET	Reimposta tutte le variabili locali di un task.
SCOPE	Programma la memorizzazione automatica da parte del sistema di un massimo di 4 parametri sulla matrice di variabili della memoria TABLE per ogni periodo di campionamento.

Nome	Descrizione
SET_BIT	Imposta su uno il valore del bit specificato nella variabile VR specificata.
TABLE	Scriva e legge i dati nella matrice di variabili della memoria TABLE.
TABLEVALUES	Restituisce una lista di valori della memoria TABLE.
TIME\$	Stampa l'ora corrente come stringa.
TRIGGER	Avvia un comando SCOPE precedentemente impostato.
VR	Scriva e legge i dati nelle variabili globali (VR).
VRSTRING	Riunisce i valori della memoria VR per consentirne la stampa come stringa.
WA	Sospende l'esecuzione del programma per il numero di millisecondi specificato.
WAIT IDLE	Sospende l'esecuzione del programma fino a quando l'asse di base non ha concluso il movimento corrente e tutti i movimenti contenuti nel buffer.
WAIT LOADED	Sospende l'esecuzione del programma fino a quando nel buffer non vi sono più movimenti da eseguire per l'asse di base, a parte quelli attualmente in esecuzione.
WAIT UNTIL	Valuta ripetutamente la condizione fino a quando non è TRUE .

3.1.11 Parametri di sistema

Nome	Descrizione
AIN	Contiene il valore del canale analogico.
AOUT	Contiene il valore del canale analogico.
BATTERY_LOW	Restituisce la condizione corrente della batteria.
CHECKSUM	Contiene il checksum per i programmi della RAM.
COMMSERROR	Contiene tutti gli errori di comunicazione che si sono verificati dall'ultima volta che il sistema è stato inizializzato.
CONTROL	Contiene il tipo di TJ1-MC__ del sistema.

Nome	Descrizione
D_ZONE_MAX	Controlla l'uscita DAC insieme al valore di errore di inseguimento.
D_ZONE_MIN	Controlla l'uscita DAC insieme al valore di errore di inseguimento.
DATE	Imposta o restituisce la data corrente contenuta nell'orologio calendario.
DAY	Imposta o restituisce il giorno corrente.
DISPLAY	Determina i canali di I/O da visualizzare sui LED del pannello frontale.
ERROR_AXIS	Contiene il numero dell'asse che ha causato l'errore di movimento.
FRAME	Specifica il frame operativo per le trasformazioni del frame.
LAST_AXIS	Contiene il numero dell'ultimo asse elaborato dal sistema.
MOTION_ERROR	Contiene un flag di errore per gli errori di movimento degli assi.
NAIO	Restituisce il numero di canali analogici collegati al bus MECHATROLINK-II.
NEG_OFFSET	Consente di attribuire un offset negativo al segnale DAC proveniente dall'anello servo.
NIO	Contiene il numero di entrate e uscite collegate al sistema.
POWER_UP	Determina se i programmi debbano essere letti dalla memoria flash EPROM all'avvio o al ripristino.
POS_OFFSET	Applica un offset positivo al segnale DAC proveniente dall'anello servo.
SCOPE_POS	Contiene la posizione corrente all'interno della memoria TABLE dove il comando SCOPE sta attualmente memorizzando il primo parametro.
SERVO_PERIOD	Imposta il periodo del ciclo di servoazionamento di TJ1-MC__.
SYSTEM_ERROR	Contiene gli errori di sistema dopo l'ultima inizializzazione.
TIME	Restituisce l'ora corrente contenuta nell'orologio calendario.
TSIZE	Restituisce le dimensioni della tabella definita correntemente.
VERSION	Restituisce il numero di versione del firmware dell'unità di controllo assi.
WDOG	L'interruttore software che abilita i servoazionamenti.

3.1.12 Comandi e parametri dei task

Nome	Descrizione
ERROR_LINE	Contiene il numero della riga che ha provocato l'ultimo errore del programma BASIC.
PMOVE	Contiene lo stato dei buffer dei task.
PROC	Consente l'accesso al parametro di un processo particolare.
PROC_STATUS	Restituisce lo stato del processo specificato.
PROCNUMBER	Contiene il numero del task in cui il programma attualmente selezionato viene eseguito.
RUN_ERROR	Contiene il numero dell'ultimo errore BASIC che si è verificato nel task specificato.
TICKS	Contiene il conteggio corrente degli impulsi di clock del task.

3.2 Tutti i comandi BASIC

3.2.1 + (Addizione)

Tipo	Funzione matematica
Sintassi	espressione1 + espressione2
Descrizione	L'operatore + somma due espressioni.
Argomenti	• espressione1 Qualsiasi espressione valida in BASIC. • espressione2 Qualsiasi espressione valida in BASIC.
Esempio	risultato = 4 + 3 Assegna il valore 7 alla variabile risultato .
Vedere anche	N/D

3.2.2 - (Sottrazione)

Tipo	Funzione matematica
Sintassi	espressione1 - espressione2
Descrizione	L'operatore - sottrae espressione2 da espressione1 .
Argomenti	• espressione1 Qualsiasi espressione valida in BASIC. • espressione2 Qualsiasi espressione valida in BASIC.
Esempio	risultato = 10 - 2 Assegna il valore 8 alla variabile risultato .
Vedere anche	N/D

3.2.3 * (Moltiplicazione)

Tipo	Funzione matematica
Sintassi	espressione1 * espressione2
Descrizione	L'operatore * moltiplica due espressioni.
Argomenti	• espressione1 Qualsiasi espressione valida in BASIC. • espressione2 Qualsiasi espressione valida in BASIC.
Esempio	risultato = 3 * 7 Assegna il valore 21 alla variabile risultato .
Vedere anche	N/D

3.2.4 / (Divisione)

Tipo	Funzione matematica
Sintassi	espressione1 / espressione2
Descrizione	L'operatore / divide espressione1 per espressione2 .
Argomenti	• espressione1 Qualsiasi espressione valida in BASIC. • espressione2 Qualsiasi espressione valida in BASIC.
Esempio	risultato = 11 / 4 Assegna il valore 2.75 alla variabile risultato .
Vedere anche	N/D

3.2.5 ^ (Potenza)

Tipo	Funzione matematica
Sintassi	espressione1 ^ espressione2
Descrizione	L'esponente ^ eleva espressione_1 alla potenza di espressione_2 . Questa operazione utilizza algoritmi a virgola mobile e potrebbe produrre alcune piccole deviazioni nei calcoli con numeri interi.
Argomenti	• espressione_1 Espressione in BASIC. • espressione_2 Espressione in BASIC.
Esempio	risultato = 2^5 Questo assegna il valore 32 alla variabile risultato .
Vedere anche	N/D

3.2.6 = (Uguale a)

Tipo	Funzione matematica
Sintassi	espressione1 = espressione2
Descrizione	L'operatore = restituisce TRUE se espressione1 è uguale a espressione2 , in caso contrario restituisce FALSE .
Argomenti	• espressione1 Qualsiasi espressione valida in BASIC. • espressione2 Qualsiasi espressione valida in BASIC.
Esempio	IF a = 10 THEN GOTO etichetta1 Se la variabile a contiene un valore uguale a 10, l'esecuzione del programma prosegue all'etichetta etichetta1 . In caso contrario, l'esecuzione del programma prosegue con l'istruzione successiva.
Vedere anche	N/D

3.2.7 = (Assegnazione)

Tipo	Funzione matematica
Sintassi	variabile = espressione
Descrizione	L'operatore = assegna il valore dell'espressione alla variabile.
Argomenti	• variabile Un nome di variabile. • espressione Qualsiasi espressione valida in BASIC.
Esempio	var = 18 Assegna il valore 18 alla variabile var .
Vedere anche	N/D

3.2.8 <> (Diverso da)

Tipo	Funzione matematica
Sintassi	espressione1 <> espressione2
Descrizione	L'operatore <> restituisce TRUE se espressione1 è diversa da espressione2 , in caso contrario restituisce FALSE .
Argomenti	• espressione1 Qualsiasi espressione valida in BASIC. • espressione2 Qualsiasi espressione valida in BASIC.
Esempio	IF a <> 10 THEN GOTO etichetta1 Se la variabile a contiene un valore non uguale a 10, l'esecuzione del programma prosegue all'etichetta etichetta1 . In caso contrario, l'esecuzione del programma prosegue con l'istruzione successiva.
Vedere anche	N/D

3.2.9 > (Maggiore di)

Tipo	Funzione matematica
Sintassi	espressione1 > espressione2
Descrizione	L'operatore > restituisce TRUE se espressione1 è maggiore di espressione2 , in caso contrario restituisce FALSE .
Argomenti	• espressione1 Qualsiasi espressione valida in BASIC. • espressione2 Qualsiasi espressione valida in BASIC.
Esempio	IF a > 10 THEN GOTO etichetta1 Se la variabile a contiene un valore maggiore di 10, l'esecuzione del programma prosegue all'etichetta etichetta1 . In caso contrario, l'esecuzione del programma prosegue con l'istruzione successiva.
Vedere anche	N/D

3.2.10 >= (Maggiore di o uguale a)

Tipo	Funzione matematica
Sintassi	espressione1 >= espressione2
Descrizione	L'operatore >= restituisce TRUE se espressione1 è maggiore o uguale a espressione2 , in caso contrario restituisce FALSE .
Argomenti	• espressione1 Qualsiasi espressione valida in BASIC. • espressione2 Qualsiasi espressione valida in BASIC.
Esempio	IF a >=10 THEN GOTO etichetta1 Se la variabile a contiene un valore maggiore o uguale a 10, l'esecuzione del programma prosegue all'etichetta etichetta1 . In caso contrario, l'esecuzione del programma prosegue con l'istruzione successiva.
Vedere anche	N/D

3.2.11 < (Minore di)

Tipo	Funzione matematica
Sintassi	espressione1 < espressione2
Descrizione	L'operatore < restituisce TRUE se espressione1 è minore di espressione2 , in caso contrario restituisce FALSE .
Argomenti	• espressione1 Qualsiasi espressione valida in BASIC. • espressione2 Qualsiasi espressione valida in BASIC.
Esempio	IF a < 10 THEN GOTO etichetta1 Se la variabile a contiene un valore minore di 10, l'esecuzione del programma prosegue all'etichetta etichetta1 . In caso contrario, l'esecuzione del programma prosegue con l'istruzione successiva.
Vedere anche	N/D

3.2.12 <= (Minore o uguale a)

Tipo	Funzione matematica
Sintassi	espressione1 <= espressione2
Descrizione	L'operatore <= restituisce TRUE se espressione1 è minore o uguale a espressione2 , in caso contrario restituisce FALSE .
Argomenti	• espressione1 Qualsiasi espressione valida in BASIC. • espressione2 Qualsiasi espressione valida in BASIC.
Esempio	IF a <= 10 THEN GOTO etichetta1 Se la variabile a contiene un valore minore o uguale a 10, l'esecuzione del programma prosegue all'etichetta etichetta1 . In caso contrario, l'esecuzione del programma prosegue con l'istruzione successiva.
Vedere anche	N/D

3.2.13 \$ (Dati esadecimali)

Tipo	Comando di sistema
Sintassi	\$num_esadecimale
Descrizione	Il comando \$ rende il numero che segue un numero esadecimale.
Argomenti	num_esadecimale Un numero esadecimale (costituito dai caratteri 0 – 9 e A – F). num_esadecimale è compreso tra 0 e FFFFFF.
Esempio	<pre>>>TABLE(0,\$F,\$ABCD) >>print TABLE(0),TABLE(1) 15.0000 43981.0000</pre>
Vedere anche	HEX (PRINT)

3.2.14 ' (Campo di commento)

Tipo	Comando di programma
Sintassi	'
Descrizione	' contrassegna tutto ciò che segue su una riga come commento e non come codice di programma. Il commento non viene eseguito quando il programma è in esecuzione. È possibile utilizzare ' all'inizio di una riga o dopo un'istruzione valida.
Argomenti	N/D
Esempio	<pre>' Questa riga non è stampata PRINT "Avvio"</pre>
Vedere anche	N/D

3.2.15 : (Separatore di istruzioni)

Tipo	Comando di programma
Sintassi	:
Descrizione	Il separatore di istruzioni : divide più istruzioni in BASIC presenti su una singola riga. È possibile utilizzarlo sulla riga di comando e nei programmi.
Argomenti	N/D
Esempio	PRINT "QUESTA RIGA": GET low : PRINT "FA TRE COSE"
Vedere anche	N/D

3.2.16

Tipo	Carattere speciale
Sintassi	#
Descrizione	Il simbolo # viene utilizzato per specificare un canale di comunicazione da usare per comandi di ingresso/uscita seriali. Nota: i canali di comunicazione maggiori di 3 saranno utilizzati solo se è in esecuzione il software Trajexia Tools.
Argomenti	N/D
Esempio	<pre>PRINT #1,"RS232" PRINT #2,"Porta 2"</pre>
Esempio	<pre>IF KEY #1 THEN GET #1,k</pre> Controllare il tastierino sulla porta RS232
Vedere anche	N/D

3.2.17 ABS

Tipo	Funzione matematica
Sintassi	ABS(espressione)
Descrizione	La funzione ABS restituisce il valore assoluto di un'espressione.
Argomenti	espressione Qualsiasi espressione valida in BASIC.
Esempio	IF ABS(A) > 100 THEN PRINT "A non rientra nell'intervallo -100... 100"
Vedere anche	N/D

3.2.18 ACC

Tipo	Comando degli assi
Sintassi	ACC(velocità)
Descrizione	Imposta contemporaneamente accelerazione e decelerazione. Questo comando fornisce un metodo rapido per impostare sia ACCEL sia DECEL . Si consiglia di impostare la velocità di accelerazione e quella di decelerazione con i parametri ACCEL e DECEL per gli assi.
Argomenti	velocità La velocità di accelerazione in unità/s². È possibile definire le unità con il parametro UNITS per gli assi.
Esempio	ACC(100) Imposta ACCEL e DECEL su 100 unità/s ² .
Vedere anche	ACCEL, DECEL, UNITS

3.2.19 ACCEL

Tipo	Parametro degli assi
Sintassi	ACCEL = espressione
Descrizione	Il parametro degli assi ACCEL contiene la velocità di accelerazione degli assi. La velocità è impostata su unità/s ² . Il parametro può avere qualsiasi valore positivo, compreso lo zero.
Argomenti	N/D
Esempio	BASE(0) ACCEL = 100 ' Imposta la velocità di accelerazione PRINT "Velocità di accelerazione: ";ACCEL;" mm/s/s" ACCEL AXIS(2) = 100 ' Imposta la velocità di accelerazione per l'asse (2)
Vedere anche	ACCEL, DECEL, UNITS

3.2.20 ACOS

Tipo	Funzione matematica
Sintassi	ACOS(espressione)
Descrizione	La funzione ACOS restituisce l'arcocoseno dell'espressione. Il valore dell'espressione deve essere compreso tra -1 e 1. Il risultato in radianti è compreso tra 0 and PI. I valori d'ingresso fuori intervallo restituiranno 0.
Argomenti	espressione Qualsiasi espressione valida in BASIC.
Esempio	>> PRINT ACOS(-1) 3.1416
Vedere anche	N/D

3.2.21 ADD_DAC

Tipo	Comando degli assi
Sintassi	ADD_DAC(asse)
Descrizione	Il comando ADD_DAC può fornire un doppio controllo di retroazione permettendo che un encoder secondario venga utilizzato sul servoaasse. Il comando consente la somma delle uscite di 2 anelli servo determinando la velocità di riferimento per il servoaazionamento. Questo comando è solitamente utilizzato in applicazioni come l'alimentazione a rulli, dove un encoder secondario è richiesto per compensare lo slittamento. Per l'utilizzo di ADD_DAC è necessario collegare i due assi con retroazione fisica a un asse comune sul quale vengono eseguiti i movimenti richiesti. Solitamente questo si ottiene eseguendo i movimenti su uno solo dei due assi e utilizzando ADDAX o CONNECT per raggiungere la posizione richiesta corrispondente (DPOS) per entrambi gli assi. I guadagni dell'anello servo devono essere impostati per entrambi gli assi. Le uscite dell'anello servo sono sommate all'uscita della velocità di riferimento del servoaasse. Per cancellare il collegamento, usare ADD_DAC(-1). ADD_DAC agisce sull'asse di base predefinito (impostato con BASE) a meno che AXIS non venga usato per specificare un asse di base temporaneo. Nota: 1. tenere presente che occorre determinare con precisione per entrambi gli assi i guadagni dell'anello di controllo. Dal momento che vengono utilizzati encoder diversi con risoluzioni diverse, i guadagni non sono identici. 2. Impostare il parametro OUTLIMIT sullo stesso valore per entrambe le sincronizzazioni.
Argomenti	• asse Asse da cui prendere l'uscita della velocità di riferimento da sommare all'asse di base. Impostare l'argomento su -1 per annullare il collegamento e tornare al funzionamento normale.
Esempio	BASE(0) OUTLIMIT AXIS(1) = 15000 ADD_DAC(1) AXIS(0) ADDAX(0) AXIS(1) WDOG = ON SERVO AXIS(0) = ON SERVO AXIS(1) = ON ' Esegue i movimenti sull'asse 0 Questo esempio mostra il controllo del servoaazionamento dell'asse 0 tramite un doppio controllo di retroazione utilizzando sia l'asse 0 che l'asse 1.

Esempio	BASE(0) OUTLIMIT AXIS(1) = 15000 ADD_DAC(1) AXIS(0) ADDAX(0) AXIS(1) WDOG = ON SERVO = OFF S_REF = 0 BASE(1) SERVO = ON ' Esegue i movimenti sull'asse 1 Questo esempio mostra il controllo del servoaazionamento dell'asse 0 utilizzando solamente la retroazione dell'encoder sull'asse 1.

Vedere anche **AXIS, ADDAX, OUTLIMIT**

3.2.22 ADDAX

Tipo	Comando degli assi
Sintassi	ADDAX(asse)
Descrizione	Il comando ADDAX riceve dall'asse sovrapposto i cambiamenti relativi alla posizione richiesta in base a quanto specificato dall'argomento dell'asse e li aggiunge a qualsiasi movimento in corso di esecuzione sull'asse destinatario del comando. Una volta dato il comando ADDAX, il collegamento tra i due assi si mantiene fino a quando non viene interrotto. Utilizzare ADDAX(-1) per annullare il collegamento tra gli assi. ADDAX consente a un asse di eseguire i movimenti specificati per 2 assi collegati. È inoltre possibile combinare più di due assi applicando il comando ADDAX all'asse sovrapposto. ADDAX agisce sull'asse di base predefinito (impostato con BASE) a meno che AXIS non sia usato per specificare un asse di base temporaneo.
Argomenti	• asse L'asse da impostare come asse sovrapposto. Impostare l'argomento su -1 per cancellare il collegamento e tornare al funzionamento normale.

Esempio **FORWARD** ' Imposta un movimento continuo
ADDAX(2) ' Aggiunge l'asse 2 per correzione
REPEAT
 GOSUB getoffset ' Ottiene l'offset da applicare
 MOVE(offset) AXIS(2)
UNTIL IN(2) = ON ' Finché non viene effettuata la correzione
 I pezzi sono collocati su un nastro trasportatore a moto continuo e vengono raccolti più avanti sul nastro. Un sistema di rilevamento indica se un pezzo si trova avanti o indietro rispetto alla sua posizione normale e la distanza da tale posizione.
 In questo esempio si presume che l'asse 0 sia l'asse di base ed esegua un movimento continuo in avanti e che un movimento sovrapposto sull'asse 2 venga utilizzato per applicare gli offset in base all'offset calcolato in una subroutine.

Vedere anche **AXIS**, **OUTLIMIT**



AVVERTENZA

Tenere presente che l'utilizzo di diversi comandi **ADDAX** nel sistema può provocare un anello pericoloso quando, ad esempio, un asse è collegato a un altro e viceversa. Una simile evenienza potrebbe causare instabilità nel sistema.

3.2.23 ADDAX_AXIS

Tipo Parametro degli assi (sola lettura)
 Sintassi **ADDAX_AXIS**
 Descrizione Il parametro degli assi **ADDAX_AXIS** restituisce il numero dell'asse a cui l'asse di base è correntemente collegato mediante il comando **ADDAX**.
 Argomenti N/D
 Esempio **>>BASE(0)**
>>ADDAX(2)
>> PRINT ADDAX_AXIS
2.0000

Vedere anche **ADDAX**, **AXIS**

3.2.24 AIN

Tipo Parametro di sistema
 Sintassi **AIN(canale_analogico)**
 Descrizione I canali di ingresso analogici a +/-10V si ottengono collegando i moduli JEPMC-AN2900 del bus MECHATROLINK-II.
 Nota: anche se il valore di ingresso analogico dovrebbe essere sempre positivo, viene comunque controllato per garantire che sia maggiore di zero. Questo è per tenere conto di eventuali disturbi del segnale di ingresso che potrebbero rendere il valore negativo e causare un errore; una velocità negativa, infatti, non è valida per alcun tipo di movimento, eccetto **FORWARD** o **REVERSE**.
 Argomenti **canale_analogico**.
 Canale di ingresso analogico numero 0.31
 Esempio **MOVE(-5000)**
REPEAT
 a=AIN(1)
 IF a<0 THEN a=0
 SPEED=a*0.25
UNTIL MTYPE=0
 La velocità di una linea di produzione dipende dalla velocità di immissione del materiale su quella linea. Il caricamento del materiale avviene tramite un anello a bassa velocità dotato di un sensore d'altezza a ultrasuoni. La gamma d'uscita del sensore a ultrasuoni va da 0 a 4 V; quando l'uscita è pari a 4 V, l'anello si trova alla massima estensione.

Vedere anche N/D

3.2.25 AND

Tipo Operazione matematica
 Sintassi **espressione1 AND espressione2**

Descrizione	L'operatore AND esegue la funzione logica AND sui bit corrispondenti dei numeri interi di due espressioni valide in BASIC. La funzione logica AND tra due bit viene definita come segue: 0 AND 0 = 0 0 AND 1 = 0 1 AND 0 = 0 1 AND 1 = 1
Argomenti	• espressione1 Qualsiasi espressione valida in BASIC. • espressione2 Qualsiasi espressione valida in BASIC.
Esempio	VR(0) = 10 AND (2.1*9) Le operazioni tra parentesi vengono calcolate per prime, ma solo il numero intero del risultato, cioè 18, è utilizzato per l'operazione AND. Pertanto questa espressione equivale alla seguente: VR(0) = 10 AND 18 AND è un operatore di bit e quindi l'azione binaria si svolge come segue: 01010 AND 10010 = 00010 Pertanto, VR(0) conterrà il valore 2.
Esempio	IF MPOS AXIS(0) > 0 AND MPOS AXIS(1) > 0 THEN GOTO ciclo1
Vedere anche	N/D

3.2.26 AOUT

Tipo	Parametro di sistema
Sintassi	AOUT(canale_analogico)
Descrizione	Il comando imposta il valore dei canali di uscita analogici +/-10 V disponibili collegando i moduli JEPMC-AN2910 del bus MECHATROLINK-II. La gamma dei valori è pari a [-32000, 32000] per una tensione di [-10 V, 10 V].
Argomenti	• canale_analogico. Canale di uscita analogico numero 0.31
Esempio	Nessun esempio.
Vedere anche	N/D

3.2.27 ASIN

Tipo	Funzione matematica
Sintassi	ASIN(espressione)
Descrizione	La funzione ASIN restituisce l'arcoseno dell'argomento. L'argomento deve avere un valore compreso tra -1 e 1. Il risultato in radianti è compreso tra -PI/2 e PI/2. I valori d'ingresso non compresi in questo intervallo restituiscono 0.
Argomenti	• espressione Qualsiasi espressione valida in BASIC.
Esempio	>> PRINT ASIN(-1) -1.5708
Vedere anche	N/D

3.2.28 ATAN

Tipo	Funzione matematica
Sintassi	ATAN(espressione)
Descrizione	La funzione ATAN restituisce l'arcotangente dell'argomento. espressione può avere un qualsiasi valore. Il risultato è espresso in radianti ed è compreso tra -PI/2 e PI/2.
Argomenti	• espressione Qualsiasi espressione valida in BASIC.
Esempio	>> PRINT ATAN(1) 0.7854
Vedere anche	N/D

3.2.29 ATAN2

Tipo	Funzione matematica
Sintassi	ATAN2(espressione1,espressione2)

Descrizione La funzione **ATAN2** restituisce l'arcotangente del numero complesso diverso da zero (**espressione2**, **espressione1**), che è equivalente all'angolo compreso tra un punto con coordinate (**espressione1**, **espressione2**) e l'asse x. Se **espressione2** >= 0, il risultato è uguale al valore di **ATAN(espressione1 / espressione2)**. Il risultato in radianti sarà compreso tra -PI e PI.

Argomenti

- **espressione1**
Qualsiasi espressione valida in BASIC.
- **espressione2**
Qualsiasi espressione valida in BASIC.

Esempio >> **PRINT ATAN2(0,1)**
0.0000

Vedere anche N/D

3.2.30 ATYPE

Tipo Parametro degli assi

Sintassi **ATYPE = valore**

Descrizione Il parametro degli assi **ATYPE** imposta il tipo di asse per un asse specifico. I valori validi dipendono dal modulo TJ1 a cui è collegato il servozionamento che controlla l'asse. Vedere la tabella riportata di seguito. I parametri **ATYPE** sono impostati dal sistema all'avvio. Per gli assi controllati dai servozionamenti collegati al sistema attraverso il bus MECHATROLINK-II, il valore predefinito di **ATYPE** è 41 (velocità Mechatro). Per gli assi controllati dai servozionamenti collegati al sistema attraverso TJ1-FL02, il valore predefinito di **ATYPE** è 44 (servo dell'asse flessibile).

Argomenti N/D

Esempio **ATYPE AXIS(1) = 45**
Questo comando imposterà l'asse 1 come asse di uscita encoder dell'asse flessibile.

Vedere anche **AXIS**

AXIS tipo	ATYPE valore	Unità TJ1 applicabile
Virtuale	0	Tutte
Posizione Mechatro	40	TJ1-ML__ (modulo master MECHATROLINK-II)
Velocità Mechatro	41	TJ1-ML__
Coppia Mechatro	42	TJ1-ML__
Uscita motore passo-passo asse flessibile	43	TJ1-FL02
Servo dell'asse flessibile	44	TJ1-FL02
Uscita encoder asse flessibile	45	TJ1-FL02
Tamagawa assoluto asse flessibile	46	TJ1-FL02
EnDat assoluto asse flessibile	47	TJ1-FL02
SSI assoluto asse flessibile	48	TJ1-FL02

3.2.31 AUTORUN

Tipo Comando di programma

Sintassi **AUTORUN**

Descrizione Il comando **AUTORUN** avvia tutti i programmi che sono stati impostati per essere eseguiti all'avvio.

Argomenti N/D

Esempio Nessun esempio.

Vedere anche **RUNTYPE**

3.2.32 AXIS

Tipo Comando di sistema

Sintassi **AXIS(numero_asse)**

Descrizione Il modificatore **AXIS** imposta l'asse per un solo comando di movimento o una sola lettura/scrittura di parametri relativi a un asse specifico. **AXIS** risulta valido solo per la riga di comando o di programma in cui è programmato. Usare il comando **BASE** per cambiare l'asse di base per tutte le righe di comando che seguono.

Argomenti • **numero_asse**
Qualsiasi espressione valida in BASIC che specifichi il numero di asse.

Esempio **BASE(0)**
PRINT VP_SPEED AXIS(2)

Esempio **MOVE(300) AXIS(0)**

Esempio **REPDIST AXIS(1) = 100**

Vedere anche **BASE**

3.2.33 AXIS_DISPLAY

Tipo Parametro degli assi

Sintassi **AXIS_DISPLAY = valore**

Descrizione Il parametro degli assi **AXIS_DISPLAY** consente la visualizzazione di diversi dati sui LED del coperchio frontale di TJ1-FL02. I LED su cui agisce questa impostazione di parametro sono i due LED gialli che mostrano lo stato dell'asse. All'avvio, il valore predefinito di questo parametro è 0 per tutti gli assi. I valori validi sono riportati nella tabella di seguito.

Argomenti N/D

Esempio **AXIS_DISPLAY AXIS(2) = 2**
Questo comando visualizza lo stato di OUT 0 e OUT 1 allocati all'asse 2.

Vedere anche N/D

AXIS_DISPLAY valore	0	1	2	3
A0	REG 0	AUX IN	OUT 0	ENCODER A
A1	REG 1	ENCODER Z	OUT 1	ENCODER B
B0	REG 0	AUX IN	OUT 0	ENCODER A
B1	REG 1	ENCODER Z	OUT 1	ENCODER B

3.2.34 AXIS_ENABLE

Tipo Parametro degli assi

Sintassi **AXIS_ENABLE = ON/OFF**

Descrizione Il parametro degli assi **AXIS_ENABLE** viene utilizzato per attivare o disattivare un particolare asse indipendentemente dagli altri. Questo parametro può essere impostato come attivo o non attivo per ciascun asse singolarmente. Il valore predefinito all'avvio è "attivo" per ogni asse. Un asse verrà attivato se per quell'asse sia **AXIS_ENABLE** che **WDOG** sono "attivi". Impostando **AXIS_ENABLE** su "non attivo" per gli assi MECHATROLINK-II, l'uscita del servoazionamento verso il motore sarà disattivata. Impostando **AXIS_ENABLE** su "non attivo" per il servosasse dell'asse flessibile, entrambe le tensioni di uscita saranno portate a 0. Impostando **AXIS_ENABLE** su "non attivo" per l'uscita motore passo-passo e l'uscita encoder dell'asse flessibile, la generazione degli impulsi sulle uscite verrà bloccata.

Argomenti N/D

Esempio **AXIS_ENABLE AXIS(3) = OFF**
Questo comando disattiverà l'asse 3 indipendentemente dagli altri assi del sistema.

Vedere anche **AXIS_DISABLE_GROUP**

3.2.35 AXISSTATUS

Tipo Parametro degli assi (sola lettura)

Sintassi **AXISSTATUS**

Descrizione Il parametro degli assi **AXISSTATUS** contiene lo stato dell'asse. Le definizioni del parametro degli assi **AXISSTATUS** sono riportate nella tabella di seguito. Il parametro **AXISSTATUS** viene utilizzato per la gestione degli errori di movimento dell'unità.

Argomenti N/D

Esempio **IF (AXISSTATUS AND 16)>0 THEN PRINT "Entro il limite di avanzamento"**

Vedere anche **AXIS_ERRORMASK**

Numero bit	Descrizione	Valore	Carattere (usato in Trajexia Tools)
0	–	1	–
1	Avviso di errore di inseguimento	2	w
2	Errore di comunicazione del servozionamento	4	a
3	Allarme del servozionamento	8	m
4	Limite di avanzamento	16	f
5	Limite di marcia indietro	32	r
6	Ricerca origine dati	64	d
7	Ingresso della sospensione dell'alimentazione	128	h
8	Limite di errore di inseguimento	256	e
9	Limite software di avanzamento	512	x
10	Limite software di marcia indietro	1024	y
11	Annullamento del movimento in corso	2048	c
12	Velocità eccessiva dell'uscita encoder	4096	o

3.2.36 B_SPLINE

Tipo Comando degli assi

Sintassi **B_SPLINE(tipo, dati_ingresso, numero_ingresso, dati_uscita, n. espansione)**

Descrizione Espande un profilo esistente registrato nella memoria TABLE a un'altra area della memoria TABLE, utilizzando la funzione matematica B-Spline mediante un fattore di espansione configurabile.
L'utilizzo ideale di questo comando si ha quando il profilo **CAM** sorgente non è sufficientemente elaborato e richiede l'estrapolazione in un maggior numero di punti.

Argomenti

- **tipo**
Riservato a un'espansione futura. Impostare sempre su 1.
- **dati_ingresso**
Posizione della memoria TABLE dove è registrato il profilo sorgente.
- **numero_ingresso**
Numero di punti nel profilo sorgente.
- **dati_uscita**
Posizione della memoria TABLE dove sarà memorizzato il profilo espanso.
- **rapporto_espansione**
Il coefficiente di espansione: se il profilo di sorgente è di 100 punti e il **rapporto_espansione** è impostato su 10, il profilo risultante sarà di 1000 punti (100 * 10).

Esempio Nessun esempio.

Vedere anche N/D

3.2.37 BASE

Tipo Comando degli assi

Sintassi **BASE**
BASE(asse_1 [,asse_2 [, asse_3 [, asse_4 [, asse_...]]]])
BA
BA(asse_1 [,asse_2 [, asse_3 [, asse_4 [, asse_...]]]])

Descrizione Il comando **BASE** viene utilizzato per impostare l'asse di base predefinito o per impostare un insieme specifico di sequenza di assi. Tutti i comandi di movimento e i parametri d'asse successivi saranno applicati all'asse di base o a un insieme specifico di sequenza di assi, salvo il caso in cui il comando **AXIS** venga utilizzato per specificare un asse di base temporaneo. L'asse di base rimane valido fino a quando non viene cambiato nuovamente con il comando **BASE**.

Ogni processo BASIC può avere un insieme di assi proprio e ciascun programma ne può impostare uno indipendentemente. Usare il modificatore **PROC** per accedere al parametro di un determinato task.

Il comando **BASE** di raggruppamento in sequenza può essere impostato stabilendo esplicitamente l'ordine degli assi. Questo ordine viene utilizzato a fini di interpolazione nei movimenti multiasse lineari e circolari. All'avvio ovvero quando un programma inizia ad essere eseguito su un task, l'impostazione predefinita dell'insieme degli assi di base è (0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15). Il comando **BASE** senza alcun argomento restituisce il raggruppamento corrente in sequenza di base.

Nota: se il comando **BASE** non specifica tutti gli assi, il comando **BASE** "assegnerà" automaticamente i valori rimanenti. Per prima cosa assegnerà un valore a ogni asse rimasto superiore all'ultimo valore dichiarato, quindi ne assegnerà uno a ogni asse rimasto in sequenza.

Pertanto **BASE(2,6,10)** imposterà la matrice interna di 16 assi su: **2,6,10,11,12,13,14,15,0,1,3,4,5,7,8,9**.

Argomenti Il comando può supportare fino a 16 argomenti.

- **asse_i**
Numero dell'asse impostato come asse di base e qualsiasi asse successivo nella sequenza del gruppo per i movimenti a più assi.

Esempio

BASE(1)
UNITS = 2000 ' Imposta il fattore di conversione delle unità per l'asse 1
SPEED = 100 ' Imposta la velocità per l'asse 1
ACCEL = 5000 ' Imposta la velocità di accelerazione per l'asse 1
BASE(2)
UNITS = 2000 ' Imposta il fattore di conversione delle unità per l'asse 2
SPEED = 125 ' Imposta la velocità per l'asse 2
ACCEL = 10000 ' Imposta la velocità di accelerazione per l'asse 2
 È possibile programmare la velocità, l'accelerazione e altri parametri di ciascun asse.

Esempio **BASE(0)**
MOVE(100,-23.1,1250)
 In questo esempio, gli assi 0, 1 e 2 si porteranno nella posizione specificata alla velocità e all'accelerazione impostate per l'asse 0. **BASE(0)** imposta l'asse 0 come asse di base; questa operazione determina quali sono i tre assi utilizzati da **MOVE** e la velocità e accelerazione degli stessi.

Esempio **>>BASE(0,2,1)**
 Sulla riga di comando la sequenza del gruppo di base può essere visualizzata digitando **BASE**.

Esempio **>> RUN "PROGRAM",3**
>> BASE PROC(3)(0,2,1)
 Utilizzare il modificatore **PROC** per mostrare la sequenza del gruppo di base di un determinato task.

Esempio **>>BASE(2)**
>> PRINT BASE
2.0000
 La stampa di **BASE** restituirà l'asse di base correntemente selezionato.

Vedere anche **AXIS**

3.2.38 BASICERROR

Tipo Comando di sistema

Sintassi **BASICERROR**

Descrizione Il comando **BASICERROR** può essere utilizzato per eseguire una routine quando si verifica un errore di run-time in un programma. **BASICERROR** può essere utilizzato solo all'interno di un comando **ON... GOSUB** o **ON... GOTO**. L'esecuzione di questo comando nel programma BASIC è richiesta una sola volta. Se vengono utilizzati più comandi, risulta valido solo quello che viene eseguito per ultimo.

Argomenti N/D

Esempio **ON BASICERROR GOTO routine_errore**

```
...
nessun_errore = 1
STOP
routine_errore:
IF nessun_errore = 0 THEN
PRINT "L'errore";RUN_ERROR[0];
PRINT " si è verificato alla riga ";ERROR_LINE[0]
ENDIF
STOP
```

In questo esempio, se si verifica un errore in un comando BASIC, verrà eseguita la routine di errore.

La presenza dell'istruzione **IF** impedisce che il programma entri in routine d'errore quando viene arrestato normalmente.

Vedere anche **ERROR_LINE**, **ON**, **RUN_ERROR**.

3.2.39 BATTERY_LOW

Tipo Parametro di sistema (sola lettura)

Sintassi **BATTERY_LOW**

Descrizione Questo parametro restituisce lo stato corrente della batteria.
Se **BATTERY_LOW=ON** occorre sostituire la batteria.
Se **BATTERY_LOW=OFF** la batteria è sufficientemente carica.

Argomenti N/D

Esempio Nessun esempio.

Vedere anche N/D

3.2.40 BREAK_RESET

Tipo Comando di sistema

Sintassi **BREAK_RESET "nome_programma"**

Descrizione Utilizzato da Trajexia Tools per rimuovere tutti i punti di interruzione dal programma specificato.

Argomenti

- **nome_programma**
Il nome del programma da cui si desiderano rimuovere tutti i punti di interruzione.

Esempio **BREAK_RESET "sempliceprova"**
Rimuoverà tutti i punti di interruzione dal programma **sempliceprova**.

Vedere anche N/D

3.2.41 CAM

Tipo Comando degli assi

Sintassi **CAM(punto_iniziale, punto_finale, moltiplicatore_tabella, distanza)**

Descrizione Il comando **CAM** viene utilizzato per generare il movimento di un asse secondo un profilo di posizione memorizzato nella matrice delle variabili della memoria TABLE. I valori della memoria TABLE costituiscono posizioni assolute rispetto al punto d'inizio e sono specificati nei fronti dell'encoder. La matrice della memoria TABLE viene specificata con il comando **TABLE**. Il movimento può essere definito tramite un qualsiasi numero di punti compreso tra 2 e 64000. TJ1-MC__ si sposta in continuazione tra i valori della memoria TABLE per permettere a diversi di punti di definire un profilo scorrevole. È possibile eseguire simultaneamente due o più comandi **CAM** utilizzando valori simili o sovrapposti presenti nella matrice della memoria TABLE. Il profilo della memoria TABLE viene percorso trasversalmente una sola volta. **CAM** richiede che l'elemento di avvio presente nella matrice della memoria TABLE abbia come valore zero. L'argomento distanza, assieme ai parametri **SPEED** e **ACCEL**, determina la velocità di spostamento attraverso la matrice della memoria TABLE. Notare che per poter seguire esattamente il profilo **CAM**, il parametro dell'asse **ACCEL** deve essere almeno 1000 volte superiore al parametro **SPEED**. **CAM** agisce sull'asse di base predefinito (impostato con **BASE**) a meno che **AXIS** non sia usato per specificare un asse di base temporaneo.

Argomenti

- **punto_iniziale**
L'indirizzo del primo elemento da utilizzare nella matrice della memoria TABLE.
La matrice della memoria TABLE, potendo specificare il punto di avvio, è in grado di mantenere più di un profilo e/o altre informazioni.
- **punto_finale**
L'indirizzo dell'elemento finale nella matrice della memoria TABLE.
- **moltiplicatore_tabella**
Il valore del moltiplicatore di tabella che viene utilizzato per scalare i valori registrati nella memoria TABLE. Dal momento che i valori della memoria TABLE sono specificati nei fronti dell'encoder, è possibile utilizzare questo argomento per impostare, ad esempio, i valori del fattore di conversione delle unità (impostato attraverso il parametro **UNITS**).
- **distanza**
Fattore espresso in unità utente che controlla la velocità di spostamento attraverso la Tabella. Il tempo impiegato per eseguire **CAM** dipende dalla velocità corrente dell'asse e da questa distanza. Supponiamo ad esempio che il sistema sia programmato in mm, la velocità impostata su 10 mm/s e l'accelerazione abbastanza elevata. Se la distanza specificata è di 100 mm, **CAM** ci metterà 10 secondi ad eseguirla. Il parametro **SPEED** dell'asse di base consente la modifica della velocità di spostamento quando si utilizza il movimento **CAM**.

Nota: quando il comando **CAM** è in esecuzione, il parametro **ENDMOVE** è impostato sulla fine dell'ultimo movimento.

Esempio

Supponiamo che un movimento debba seguire l'equazione di posizione $t(x) = x \cdot 25 + 10000(1 - \cos(x))$. In questo esempio, x è espresso in gradi. L'esempio riguarda una memoria TABLE che genera una semplice oscillazione cui è sovrapposta una velocità costante. Il codice di seguito può essere utilizzato per caricare la memoria TABLE e attraversarla a ciclo continuo.

GOSUB tabellacam

anello:

CAM(1,19,1,200)

GOTO anello

La subroutine **tabellacam** carica i dati della tabella di seguito nella matrice della memoria TABLE.

Vedere anche **ACCEL, AXIS, CAMBOX, SPEED, TABLE**.

Posizione nella memoria TABLE	Grado	Valore
1	0	0
2	20	1103
3	40	3340
4	60	6500
5	80	10263
6	100	14236
7	120	18000
8	140	21160
9	160	23396
10	180	24500
11	200	24396
12	220	23160
13	240	21000
14	260	18236
15	280	15263
16	300	12500
17	320	10340
18	340	9103
19	360	9000

3.2.42 CAMBOX

Tipo Comando degli assi

Sintassi **CAMBOX(punto_iniziale, punto_finale, moltiplicatore_tabella, distanza_collegamento, asse_collegamento [, opzione_collegamento [, posizione_collegamento]])**

Descrizione Il comando **CAMBOX** viene utilizzato per generare il movimento di un asse secondo un profilo di posizione memorizzato nella matrice delle variabili della memoria TABLE. Il movimento è collegato al movimento misurato di un altro asse, formando così una trasmissione software a variazione continua. I valori della memoria TABLE costituiscono posizioni assolute rispetto al punto d'inizio e sono specificati nei fronti dell'encoder.

La matrice della memoria TABLE viene specificata con il comando **TABLE**. Il movimento può essere definito attraverso un numero qualsiasi di punti compresi tra 2 e 64000. La matrice della memoria TABLE, potendo specificare il punto di avvio, è in grado di mantenere più di un profilo e/o altre informazioni. TJ1-MC__ si sposta in continuazione tra i valori della memoria TABLE per permettere a un numero di punti di definire un profilo scorrevole. È possibile eseguire simultaneamente due o più comandi **CAMBOX** utilizzando valori simili o sovrapposti presenti nella matrice della memoria TABLE. Il comando **CAMBOX** richiede che l'elemento iniziale della memoria TABLE abbia valore zero. Notare inoltre che il comando **CAMBOX** consente l'attraversamento in avanti e indietro della memoria TABLE, in base alla direzione dell'asse principale.

L'argomento **opzione_collegamento** può essere utilizzato sia per specificare diverse opzioni d'avvio per il comando sia per specificare un **CAM** continuo. Se per esempio l'**opzione_collegamento** viene impostata su 4, **CAMBOX** funziona come un **CAM** "fisico".

CAMBOX agisce sull'asse di base predefinito (impostato con **BASE**) a meno che **AXIS** non sia usato per specificare un asse di base temporaneo. Nota: durante l'esecuzione di **CAMBOX**, il parametro **ENDMOVE** è impostato sulla fine dell'ultimo movimento e il parametro degli assi **REMAIN** contiene la distanza rimanente sull'asse di collegamento.

Argomenti

- **punto_iniziale**
L'indirizzo del primo elemento da utilizzare nella matrice della memoria TABLE.
- **punto_finale**
L'indirizzo dell'elemento finale nella matrice della memoria TABLE.
- **moltiplicatore_tabella**
Il valore del moltiplicatore di tabella che viene utilizzato per scalare i valori registrati nella memoria TABLE. Dal momento che i valori della memoria TABLE sono specificati nei fronti dell'encoder, è possibile utilizzare questo argomento per impostare, ad esempio, i valori del fattore di conversione delle unità (impostato attraverso il parametro UNITS).
- **distanza_collegamento**
La distanza, espressa in unità utente, che l'asse di collegamento deve coprire per completare il movimento di uscita specificato. La distanza di collegamento deve essere specificata come distanza positiva.
- **asse_collegamento**
L'asse a cui ci si deve collegare.
- **opzione_collegamento**
Vedere tabella di seguito.
- **posizione_collegamento**
La posizione assoluta da dove **CAMBOX** ha inizio quando l'**opzione_collegamento** è impostata su 2.

Esempio Nessun esempio.

Vedere anche • **AXIS, CAM, REP_OPTION, TABLE**

valore opzione_collegamento	Descrizione
1	Il collegamento inizia quando si verifica un evento di registrazione sull'asse di collegamento.
2	Il collegamento inizia da una posizione assoluta dell'asse di collegamento (vedere posizione_collegamento).
4	CAMBOX si ripete automaticamente e nelle due direzioni. Questa opzione viene cancellata impostando il bit 1 del parametro REP_OPTION (REP_OPTION = REP_OPTION OR 2).
5	Combinazione delle opzioni 1 e 4.
6	Combinazione delle opzioni 2 e 4.

3.2.43 CANCEL

Tipo	Comando degli assi
Sintassi	CANCEL [(1)] CA [(1)]
Descrizione	Il comando CANCEL cancella il movimento in corso di esecuzione su un asse. I movimenti dotati di profilo di velocità (FORWARD, REVERSE, MOVE, MOVEABS, MOVECIRC, MHELICAL e MOVEMODIFY) subiscono una decelerazione in base alla velocità di decelerazione impostata nel parametro DECEL e vengono infine arrestati. Gli altri movimenti vengono arrestati subito. Il comando CANCEL cancella il contenuto del buffer del movimento corrente (MTYPE). Il comando CANCEL(1) cancella il contenuto del buffer dei movimenti successivi (NTYPE) senza influire sul movimento corrente presente nel buffer MTYPE. CANCEL agisce sull'asse di base predefinito (impostato con BASE) a meno che AXIS non sia usato per specificare un asse di base temporaneo. Nota: • CANCEL cancella soltanto il movimento in corso di esecuzione. Se il buffer contiene altri movimenti, questi vengono caricati successivamente. • Durante il processo di decelerazione del movimento corrente, eventuali CANCEL aggiuntivi sono ignorati. • CANCEL(1) cancella soltanto il movimento attualmente presente nel buffer. Qualsiasi movimento presente nei buffer del task e indicato dalla variabile PMOVE può essere caricato nel buffer non appena viene cancellato dal buffer il movimento presente in quel momento.
Argomenti	N/D
Esempio	FORWARD WA(10000) CANCEL
Esempio	MOVE(1000) MOVEABS(3000) CANCEL ' Cancellato il movimento a 3000 e imposta il movimento a 4000. MOVEABS(4000) Notare che in questo caso il comando MOVEMODIFY è una soluzione migliore per modificare i punti finali di un movimento.
Vedere anche	AXIS , MTYPE , NTYPE , PMOVE , RAPIDSTOP

3.2.44 CHECKSUM

Tipo	Parametro di sistema (sola lettura)
Sintassi	CHECKSUM
Descrizione	Il parametro CHECKSUM contiene il checksum per i programmi presenti nella RAM. All'avvio, il checksum viene ricalcolato e messo a confronto con il valore registrato in precedenza. Se il checksum è errato, il programma non viene eseguito.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	N/D

3.2.45 CHR

Tipo	Comando di I/O
Sintassi	CHR(x)
Descrizione	Il comando CHR viene utilizzato per inviare singoli caratteri ASCII designati da un numero. In alcune versioni di BASIC PRINT CHR(x) ; equivale a PUT(x) .
Argomenti	• x Espressione in BASIC.
Esempio	>>PRINT CHR(65); A
Vedere anche	N/D

3.2.46 CLEAR

Tipo	Comando di sistema
Sintassi	CLEAR
Descrizione	Il comando CLEAR reimposta su 0 tutte le variabili VR globali. Quando viene usato in un programma, reimposta su 0 anche le variabili locali del task corrente.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	• RESET, VR

3.2.47 CLEAR_BIT

Tipo	Comando di sistema
Sintassi	CLEAR_BIT(numero_bit, numero_vr)
Descrizione	Il comando CLEAR_BIT reimposta su 0 il bit specificato nella variabile VR specificata. Gli altri bit della variabile mantengono il proprio valore.
Argomenti	• numero_bit Il numero del bit da reimpostare. Intervallo: 0 – 23. • numero_vr Numero della variabile VR per la quale il bit viene reimpostato. Intervallo: 0 – 1023.
Esempio	Nessun esempio.
Vedere anche	READ_BIT, SET_BIT, VR.

3.2.48 CLEAR_PARAMS

Tipo	Comando di sistema
Sintassi	CLEAR_PARAMS
Descrizione	Riporta ogni variabile e ogni parametro archiviato nella memoria flash EPROM alle impostazioni predefinite. CLEAR_PARAM non può essere eseguito se l'unità di controllo è bloccata.

Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	N/D

3.2.49 CLOSE_WIN

Tipo	Parametro degli assi
Sintassi	CLOSE_WIN CW
Descrizione	Il parametro degli assi CLOSE_WIN definisce l'estremità della finestra all'interno o all'esterno della quale è prevista la presenza di la tacca di registrazione. Il valore è espresso in unità utente.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	AXIS, OPEN_WIN, REGIST, UNITS.

3.2.50 CLUTCH_RATE

Tipo	Parametro degli assi
Sintassi	CLUTCH_RATE
Descrizione	Il parametro degli assi CLUTCH_RATE definisce il cambiamento del rapporto di connessione quando viene utilizzato il comando CONNECT. La velocità è definita come quantità di cambiamento al secondo. Il valore predefinito è impostato su un valore elevato (1000000) per garantire compatibilità con le unità TJ1-MC__ precedenti. Nota: l'operazione che utilizza CLUTCH_RATE non è deterministica quanto alla posizione. Se necessario, utilizzare invece il comando MOVELINK per evitare inutili differenze di fase tra asse di base e assi collegati.
Argomenti	N/D
Esempio	CLUTCH_RATE = 4 Questa impostazione implica che dando il comando CONNECT(4,1) ci vuole un secondo per portare a termine il collegamento.
Vedere anche	AXIS, CONNECT, MOVELINK.

3.2.51 COMMSERROR

Tipo	Parametro di sistema (sola lettura)
Sintassi	COMMSERROR
Descrizione	Il parametro COMMSERROR contiene tutti gli errori di comunicazione che si sono verificati dall'ultima volta che il sistema è stato inizializzato. I bit di COMMSERROR sono forniti nella tabella di seguito.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	N/D

Bit	Descrizione	Ubicazione dell'errore
8	Porta 1 pronta alla ricezione dati	Porta seriale 1
9	Porta 1 ricezione overrun	Porta seriale 1
10	Porta 1 errore di parità	Porta seriale 1
11	Porta 1 ricezione errore di contorno	Porta seriale 1
12	Porta 2 pronta alla ricezione dati	Porta seriale 2
13	Porta 2 ricezione overrun	Porta seriale 2
14	Porta 2 errore di parità	Porta seriale 2
15	Porta 2 ricezione errore di contorno	Porta seriale 2

3.2.52 COMMSTYPE

Tipo	Parametro di slot
Sintassi	COMMSTYPE SLOT(numero_unità)
Descrizione	Questo parametro restituisce il tipo di unità di un'unità di controllo. La tabella di seguito elenca i valori restituiti.
Argomenti	numero_unità I numeri delle unità vanno da 0 a 6; 0 rappresenta la prima unità sulla destra rispetto a TJ1-MC__.
Esempio	Nessun esempio.
Vedere anche	N/D

Valore restituito	Descrizione
0	Unità inutilizzata
31	TJ1-ML__
33	TJ1-FL02
34	TJ1-PRT
35	TJ1-DRT

3.2.53 COMPILE

Tipo	Comando di programma
Sintassi	COMPILE
Descrizione	Il comando COMPILE forza la compilazione del programma corrente in codice intermedio. I programmi sono compilati automaticamente dal software del sistema prima dell'esecuzione oppure quando viene selezionato un altro programma.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	N/D

3.2.54 CONNECT

Tipo	Comando degli assi
Sintassi	CONNECT(rapporto, asse_motore) CO(rapporto, asse_motore)
Descrizione	Il comando CONNECT collega la posizione richiesta dell'asse di base ai movimenti misurati dell'asse specificato da asse_motore al fine di creare una trasmissione elettronica. Il rapporto può essere cambiato in qualsiasi momento eseguendo un altro comando CONNECT sullo stesso asse. Per cambiare l'asse motore, il comando CONNECT deve essere prima cancellato. Se il comando CONNECT rinvia ad assi motori differenti, sarà ignorato. Il comando CONNECT può essere cancellato con un comando CANCEL o RAPIDSTOP. Il parametro degli assi CLUTCH_RATE può essere utilizzato per impostare una velocità specifica di modifica della connessione. CONNECT agisce sull'asse di base predefinito (impostato con BASE) a meno che AXIS non sia usato per specificare un asse di base temporaneo.
Argomenti	• rapporto Il rapporto di connessione della trasmissione. Il rapporto è specificato come rapporto del margine dell'encoder (non come unità). Contiene il numero di margini che l'asse di base deve coprire col proprio movimento per ogni incremento di margine dell'asse motore. Il valore del rapporto ha una risoluzione frazionaria a sedici bit e può avere un valore sia positivo che negativo. • asse_motore L'asse principale che azionerà l'asse di base.
Esempio	In un alimentatore a pressione, una rotella deve ruotare una velocità pari a un quarto della velocità di un encoder montato sul nastro trasportatore in entrata. La rotella è cablata all'asse 0. Un canale di ingresso controlla gli impulsi dell'encoder provenienti dal nastro trasportatore, formando l'asse 1. È possibile usare il codice di seguito: <pre> BASE(1) SERVO = OFF ' Questo asse viene utilizzato per monitorare il nastro trasportatore BASE(0) SERVO = ON CONNECT(0.25,1) </pre>
Vedere anche	AXIS, CANCEL, CLUTCH_RATE, CONNECT, RAPIDSTOP.

3.2.55 CONSTANT

Tipo	Comando di sistema
Sintassi	CONSTANT "nome", valore
Descrizione	Dichiara il nome come costante da utilizzare sia all'interno del programma contenente la definizione CONSTANT, sia all'interno di tutti gli altri programmi presenti nel progetto Trajexia Tools. Nota: prima che il nome venga usato in altri programmi è necessario assicurarsi che sia eseguito il programma contenente la definizione CONSTANT. Inoltre, solo quel programma deve essere in esecuzione nel momento in cui viene eseguita la definizione CONSTANT: in caso contrario verrà visualizzato l'errore di programma e il programma sarà bloccato quando si tenterà di eseguire il comando. Per un avvio rapido, il programma dovrebbe altresì essere l'unico processo in esecuzione al momento dell'accensione. Una volta effettuata, la dichiarazione CONSTANT rimane attiva fino al primo ripristino di TJ1-MC__ tramite spegnimento e riaccensione oppure fino a quando non viene eseguito il comando EX. È ammessa la dichiarazione fino a un massimo di 128 CONSTANT.
Argomenti	• nome Qualsiasi nome definito dall'utente contenente caratteri alfanumerici minuscoli o sottolineature. • valore Il valore assegnato a nome.
Esempio	<pre> CONSTANT "nak", \$15 CONSTANT "pulsante_inizio", 5 IF IN(pulsante_inizio)=ON THEN OP(led1, ON) IF tasto_car=nak THEN GOSUB ness_conf_ricevuta </pre>
Vedere anche	N/D

3.2.56 CONTROL

Tipo	Parametro di sistema (sola lettura)
Sintassi	CONTROL
Descrizione	Il parametro CONTROL contiene il tipo di TJ1-MC__ presente nel sistema. Il valore di questo parametro di sistema per TJ1-MC__ è 262.

Argomenti N/D
 Esempio Nessun esempio.
 Vedere anche N/D

3.2.57 COPY

Tipo Comando di programma
 Sintassi **COPY nome_programma nuovo_nome_programma**
 Descrizione Il comando **COPY** copia un programma esistente dall'unità di controllo a un nuovo programma con il nome specificato. Il nome del programma può essere specificato senza virgolette.
 Nota: questo comando è implementato per un terminale offline (VT100). All'interno di Trajexia Tools gli utenti possono selezionare il comando dal menu **Program**.
 Argomenti

- nome_programma**
Nome del programma da copiare.
- nuovo_nome_programma**
Nome da usare per il nuovo programma.

 Esempio **>> COPY "prog" "nuovoprogram"**
 Vedere anche **DEL, NEW, RENAME**.

3.2.58 COS

Tipo Funzione matematica
 Sintassi **COS(espressione)**
 Descrizione La funzione **COS** restituisce il coseno dell'espressione. I valori di ingresso sono espressi in radianti e possono avere qualsiasi valore. Il valore del risultato sarà compreso tra -1 e 1.
 Argomenti

- espressione**
Qualsiasi espressione valida in BASIC.

 Esempio **>> PRINT COS(0)**
1.0000
 Vedere anche N/D

3.2.59 CREEP

Tipo Parametro degli assi
 Sintassi **CREEP**
 Descrizione Il parametro degli assi **CREEP** contiene la velocità di slittamento per l'asse. La velocità di slittamento viene utilizzata per la fase lenta di una sequenza di ricerca dell'origine. **CREEP** può avere qualsiasi valore positivo, compreso lo 0. La velocità di slittamento viene inserita in unità utilizzando il fattore di conversione delle unità **UNITS**. Ad esempio, se il fattore di conversione delle unità è impostato sul numero di fronti encoder/pollici, la velocità viene impostata in pollici.
 Argomenti N/D
 Esempio **BASE(2)**
CREEP = 10
SPEED = 500
DATUM(4)
CREEP AXIS(1) = 10
SPEED AXIS(1) = 500
DATUM(4) AXIS(1)
 Vedere anche **AXIS, DATUM, UNITS**.

3.2.60 D_GAIN

Tipo Parametro degli assi
 Sintassi **D_GAIN**
 Descrizione Il parametro degli assi **D_GAIN** contiene il guadagno derivativo dell'asse. Il contributo dell'uscita derivativa viene calcolato moltiplicando il cambiamento nell'errore di inseguimento per il **D_GAIN**. Il valore predefinito è 0. Per produrre una risposta più graduale e utilizzare un guadagno proporzionale maggiore, aggiungere a un sistema il guadagno derivativo. Valori elevati possono provocare oscillazioni.
 Nota: il guadagno servoazionamento deve essere cambiato solo quando **SERVO** è disattivato.
 Argomenti N/D

Esempio Nessun esempio.

Vedere anche • **AXIS, I_GAIN, OV_GAIN, P_GAIN, VFF_GAIN.**

3.2.61 D_ZONE_MAX

Tipo Parametro di sistema

Sintassi **D_ZONE_MAX=valore**

Descrizione Questo parametro funziona congiuntamente a **D_ZONE_MIN** per ridurre a zero l'uscita DAC quando il movimento richiesto è completo e le grandezze dell'errore di inseguimento sono inferiori al valore **D_ZONE_MIN**. L'anello servo viene riattivato non appena l'errore di inseguimento supera il valore **D_ZONE_MAX** oppure quando viene avviato un nuovo movimento.

Argomenti N/D

Esempio **D_ZONE_MIN=3**
D_ZONE_MAX=10

Con questi 2 parametri impostati come mostrato qui sopra, l'uscita DAC viene ridotta a zero allorché il movimento è completo e l'errore d'inseguimento scende sotto a 3. Se un movimento viene riavviato o il valore dell'errore d'inseguimento supera 10, l'anello servo viene riattivato.

Vedere anche **D_ZONE_MIN**.

3.2.62 D_ZONE_MIN

Tipo Parametro di sistema

Sintassi **D_ZONE_MIN=valore**

Descrizione Questo parametro funziona congiuntamente a **D_ZONE_MAX** per ridurre a zero l'uscita DAC quando il movimento richiesto è completo e le grandezze dell'errore d'inseguimento sono inferiori al valore **D_ZONE_MIN**. L'anello servo viene riattivato non appena l'errore di inseguimento supera il valore **D_ZONE_MAX** oppure quando viene avviato un nuovo movimento.

Argomenti N/D

Esempio **D_ZONE_MIN=3**

D_ZONE_MAX=10

Con questi 2 parametri impostati come mostrato qui sopra, l'uscita DAC viene ridotta a zero allorché il movimento è completo e l'errore di inseguimento scende sotto a 3. Se un movimento viene riavviato o il valore dell'errore di inseguimento supera 10, l'anello servo viene riattivato.

Vedere anche **D_ZONE_MAX**.

3.2.63 DAC

Vedere **S_REF**.

3.2.64 DAC_OUT

Vedere **S_REF_OUT**.

3.2.65 DAC_SCALE

Tipo Parametro degli assi

Sintassi **DAC_SCALE**

Descrizione Questo parametro ha 2 scopi:

1. al momento dell'accensione, viene impostato su 16 negli assi incorporati del sistema. Ciò scala i valori applicati alla maggiore risoluzione **DAC**, rendendo i guadagni richiesti per l'asse simili a quelli richiesti per le altre unità di controllo.
2. È possibile impostare **DAC_SCALE** su un valore negativo (–16) per invertire la polarità del segnale DAC di uscita. Quando il servoazionamento non è attivo, la grandezza di **DAC_SCALE** non è importante, poiché la tensione applicata è controllata dal parametro **DAC**. La polarità resta comunque invertita da **DAC_SCALE**.

Argomenti N/D

Esempio **DAC_SCALE AXIS(3)=-16**

Vedere anche **DAC, S_REF**.

3.2.66 DATE

Tipo	Parametro di sistema
Sintassi	DATE
Descrizione	Restituisce o imposta la data corrente contenuta nell'orologio calendario di Trajexia. Il numero può essere inserito in formato GG:MM:AA o GG:MM:AAAA.
Argomenti	N/D
Esempio	DATE=20:10:05 oppure DATE=20:10:2005
Esempio	>>PRINT DATE 36956 Stampa il numero che rappresenta il giorno corrente. La cifra rappresenta il numero di giorni dal 1° gennaio 1900, laddove il 1° gen. 1900 è pari a 1.
Vedere anche	N/D

3.2.67 DATE\$

Tipo	Comando di sistema
Sintassi	DATE\$
Descrizione	Invia alla porta di stampa la data corrente GG/MM/AA come stringa. L'anno viene identificato da 2 cifre.
Argomenti	N/D
Esempio	PRINT #1,DATE\$ Stampa la data in base al formato, ad esempio: 20/10/05
Vedere anche	N/D

3.2.68 DATUM

Tipo	Comando degli assi
Sintassi	DATUM(sequenza)
Descrizione	Il comando DATUM unifica 6 ricerche di origine per collocare un asse in una posizione assoluta, reimpostando anche gli errori seguenti: DATUM utilizza per la ricerca dell'origine sia la velocità di slittamento che la velocità richiesta. All'interno delle sequenze, la velocità di slittamento è impostata attraverso il parametro degli assi CREEP mentre la velocità richiesta è impostata attraverso il parametro degli assi SPEED . Il numero d'ingresso del selettore del dato, utilizzato per le sequenze da 3 a 6, viene impostato dal parametro DATUM_IN . DATUM agisce sull'asse di base predefinito (impostato con BASE) a meno che AXIS non sia usato per specificare un asse di base temporaneo. Nota: l'ingresso d'origine impostato con il parametro DATUM_IN è active low, il che significa che il selettore d'origine viene impostato quando l'ingresso è spento. Gli ingressi limite della sospensione dell'alimentazione, del jog all'indietro, del jog in avanti, dell'avanzamento e della retromarcia sono anch'essi active low. Gli ingressi active low sono utilizzati per consentire un cablaggio fail-safe.
Argomenti	• sequenza Vedere tabella di seguito.
Esempio	Nessun esempio.
Vedere anche	ACCEL, AXIS, AXISSTATUS, CREEP, DATUM_IN, DECEL, MOTION_ERROR, SPEED.

valore di sequenza	Descrizione
0	Il comando DATUM(0) cancella l'errore di movimento. La posizione misurata correntemente viene impostata come posizione richiesta (ciò risulta particolarmente utile sugli assi del motore passo-passo con verifica della posizione). DATUM(0) cancella anche l'errore di inseguimento che ha superato la condizione FE_LIMIT nel registro AXISSTATUS per TUTTI gli assi. Imposta su zero i seguenti bit presenti in AXISSTATUS: Bit 1: avviso di errore di inseguimento. Bit 2: errore di comunicazione del driver remoto. Bit 3: errore del driver remoto. Bit 8: limite di errore di inseguimento superato. Bit 11: annullamento del movimento in corso. Notare che lo stato non può essere cancellato se la causa del problema è ancora presente.
1	L'asse si sposta in avanti alla velocità di slittamento fino a quando non incontra il marker Z. La posizione richiesta viene quindi reimpostata su 0 e la posizione misurata viene corretta per mantenere l'errore di inseguimento.
2	L'asse si sposta indietro alla velocità di slittamento fino a quando non incontra il marker Z. La posizione richiesta viene quindi reimpostata su 0 e la posizione misurata viene corretta per mantenere l'errore di inseguimento.
3	L'asse si sposta in avanti alla velocità richiesta fino a quando il selettore del dato non viene raggiunto. L'asse quindi si sposta indietro alla velocità di slittamento fino a quando il selettore del dato non viene reimpostato. La posizione richiesta viene quindi reimpostata su 0 e la posizione misurata viene corretta così da mantenere l'errore di inseguimento.
4	L'asse si sposta indietro alla velocità richiesta fino a quando il selettore del dato non viene raggiunto. L'asse quindi si sposta in avanti alla velocità di slittamento fino a quando il selettore del dato non viene reimpostato. La posizione richiesta viene quindi reimpostata su 0 e la posizione misurata viene corretta così da mantenere l'errore di inseguimento.

valore di sequenza	Descrizione
5	L'asse si sposta in avanti alla velocità richiesta fino a quando il selettore del dato non viene raggiunto. L'asse quindi inverte la propria direzione e si sposta indietro alla velocità di slittamento fino a quando il selettore del dato non viene reimpostato. L'asse continua a spostarsi alla velocità di slittamento fino a quando non incontra il marker Z dell'encoder. La posizione richiesta viene quindi reimpostata su 0 e la posizione misurata viene corretta così da mantenere l'errore di inseguimento.
6	L'asse si sposta indietro alla velocità richiesta fino a quando il selettore del dato non viene raggiunto. L'asse quindi si sposta in avanti alla velocità di slittamento fino a quando il selettore del dato non viene reimpostato. L'asse continua a spostarsi alla velocità di slittamento fino a quando non incontra il marker Z dell'encoder. La posizione richiesta viene quindi reimpostata su 0 e la posizione misurata viene corretta così da mantenere l'errore di inseguimento.

3.2.69 DATUM_IN

Tipo Parametro degli assi

Sintassi **DATUM_IN**
DAT_IN

Descrizione Il parametro degli assi **DATUM_IN** contiene il numero di ingresso da usare come ingresso del selettore del dato per il comando **DATUM**. L'intervallo d'ingresso valido è compreso tra 0 e 31. I valori da 0 a 15 rappresentano gli ingressi fisicamente presenti sul connettore di I/O di TJ1-MC__ e sono comuni a tutti gli assi. I valori da 16 a 27 rappresentano ingressi software che possono essere liberamente utilizzati da programmi e comandi quali IN e OP. Sono inoltre valori comuni a tutti gli assi. I valori da 28 a 31 vengono direttamente mappati agli ingressi del servozionamento presenti sul connettore CN1 e sono univocamente associati a ogni asse. La mappatura degli ingressi del servozionamento agli ingressi da 28 a 31 dipende dall'impostazione del parametro di servozionamento Pn81E. L'impostazione consigliata è Pn81E = 0x4321, con la seguente mappatura:
 Nota: l'ingresso d'origine è active low, il che significa che il selettore d'origine viene impostato quando l'ingresso è spento. L'ingresso limite della sospensione dell'alimentazione, del jog all'indietro, del jog in avanti, dell'avanzamento e della retromarcia è anch'esso active low. Gli ingressi active low sono utilizzati per consentire un cablaggio fail-safe.

- Sigma II
- ingresso 28: CN1-40
 - ingresso 29: CN1-41
 - ingresso 30: CN1-42
 - ingresso 31: CN1-43

- Sigma III
- ingresso 28: CN1-13
 - ingresso 29: CN1-7
 - ingresso 30: CN1-8
 - ingresso 31: CN1-9

- Junma
- ingresso 26: CN1-2
 - ingresso 27: CN1-1

Per maggiori informazioni sull'impostazione del parametro del driver Pn81E, consultare il manuale del servozionamento. Come impostazione predefinita, il parametro è impostato su -1 e nessun ingresso è selezionato.

Argomenti N/D

Esempio **DATUM_IN AXIS(0) = 5**

Vedere anche **AXIS, DATUM**.

3.2.70 DAY

Tipo Parametro di sistema

Sintassi **DAY**

Descrizione Restituisce il giorno corrente come numero da 0 a 6, dove domenica corrisponde a 0. **DAY** può essere impostato tramite assegnazione.

Argomenti N/D

Esempio
>>DAY=3
>>? DAY
3.0000

Vedere anche N/D

3.2.71 DAY\$

Tipo Comando di sistema

Sintassi **DAY\$**

Descrizione Stampa il giorno corrente come stringa.

Argomenti N/D

Esempio
>>DAY=3
>>? DAY\$
Mercoledì

Vedere anche N/D

3.2.72 DECEL

Tipo Parametro degli assi

Sintassi **DECEL**

Descrizione Il parametro degli assi **DECEL** contiene la velocità di decelerazione degli assi. La velocità è impostata su unità/s². Il parametro può avere qualsiasi valore positivo, compreso lo 0.

Argomenti N/D

Esempio
DECEL = 100 ' Imposta la velocità di decelerazione
PRINT " La velocità di decelerazione è di ";DECEL;" mm/s/s"

Vedere anche **ACCEL, AXIS, UNITS**.

3.2.73 DEFPOS

Tipo Comando degli assi

Sintassi
DEFPOS(pos_1 [, pos_2 [, pos_3 [, pos_4 [, ...]]]])
DP(pos_1 [, pos_2 [, pos_3 [, pos_4 [, ...]]]])

Descrizione	Il comando DEFPOS definisce la posizione richiesta corrente (DPOS) come nuova posizione assoluta. La posizione misurata (MPOS) sarà modificata di conseguenza per mantenere l'errore di inseguimento. DEFPOS viene normalmente utilizzato dopo una sequenza di ricerca dell'origine (vedere comando DATUM), poiché il comando imposta la posizione corrente su 0. DEFPOS può essere utilizzato in qualsiasi momento. In alternativa, è possibile utilizzare anche il parametro degli assi OFFPOS. Questo parametro può essere utilizzato per eseguire una regolazione relativa della posizione corrente. DEFPOS agisce sull'asse di base predefinito (impostato con BASE) a meno che AXIS non sia usato per specificare un asse di base temporaneo. Nota: i cambiamenti apportati alla posizione dell'asse utilizzando DEFPOS o OFFPOS vengono apportati al primo aggiornamento del servozionamento. Questo potrebbe causare problemi se un movimento viene iniziato nello stesso ciclo di servozionamento dei comandi DEFPOS o OFFPOS. L'esempio di seguito mostra come usare il parametro OFFPOS per evitare questo problema. Con la seguente programmazione, i comandi DEFPOS vengono convertiti internamente in offset di posizione OFFPOS, fornendo così un modo semplice di evitare il problema: DEFPOS(100): WAIT UNTIL OFFPOS = 0: MOVEABS(0)
Argomenti	Il comando può supportare fino a 16 argomenti. pos_i La posizione assoluta per l'asse (base+i) espresso in unità utente. Fare riferimento al comando BASE per il raggruppamento degli assi.
Esempio	BASE(2) DATUM(5) BASE(1) DATUM(4) WAIT IDLE DEFPOS(-1000,-3500) L'ultima riga definisce la posizione corrente con (-1000, -3500) espresso in unità utente. La posizione corrente sarebbe stata reimpostata su (0,0) dai due comandi DATUM.
Vedere anche	AXIS, DATUM, DPOS, OFFPOS, MPOS, UNITS.

3.2.74 DEL

Tipo	Comando di programma
Sintassi	DEL [nome_programma] RM [nome_programma]
Descrizione	Il comando DEL cancella un programma dall'unità di controllo. Usando il comando DEL senza il nome di un programma, si cancella il programma correntemente selezionato (con il comando SELECT). Il nome del programma può inoltre essere specificato senza virgolette. DEL ALL cancella tutti i programmi. È possibile usare DEL anche per cancellare la memoria TABLE: DEL "TABLE". Il nome "TABLE" va scritto tra virgolette. Nota: questo comando è implementato per un terminale offline (VT100). All'interno di Trajexia Tools gli utenti possono selezionare il comando dal menu Program.
Argomenti	nome_programma Nome del programma da cancellare.
Esempio	>> DEL vecchioprogramma
Vedere anche	COPY, NEW, RENAME, SELECT, TABLE.

3.2.75 DEMAND_EDGES

Tipo	Parametro degli assi (sola lettura)
Sintassi	DEMAND_EDGES
Descrizione	Il parametro degli assi DEMAND_EDGES contiene il valore corrente del parametro degli assi DPOS espresso in unità del margine dell'encoder.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	AXIS, DPOS.

3.2.76 DEVICENET

Tipo Comando di sistema

Sintassi **DEVICENET(numero_unità, 2, 1, VR_inizio_uscite, numero_uscite, VR_inizio_ingressi, numero_ingressi)**
DEVICENET(numero_unità, 4, 0)

Descrizione La funzione 2 di **DEVICENET** configura TJ1-DRT per lo scambio di dati con il modulo master DeviceNet e definisce le aree della memoria VR dove si verifica lo scambio di I/O. La funzione 4 di **DEVICENET** restituisce lo stato di scambio di dati di TJ1-DRT. Vedere la tabella per una descrizione dei bit contenuti nei canali di stato dello scambio di dati.

Argomenti

- **numero_unità**
Specifica il numero d'unità di TJ1-DRT nel Trajexia.
- **VR_inizio_uscite**
L'indirizzo iniziale nella memoria VR dell'unità di controllo dove si trovano i dati di uscita del master DeviceNet.
- **numero_uscite**
Il numero di parole in uscita dal master DeviceNet della memoria VR.
- **VR_inizio_ingressi**
L'indirizzo iniziale nella memoria VR dell'unità di controllo dove si trovano i dati di ingresso per il master DeviceNet.
- **numero_ingressi**
Il numero di parole d'ingresso nel master DeviceNet della memoria VR.

Esempio **DEVICENET (0,2,1,10,16,150,31)**

In questo esempio, TJ1-DRT è configurato per scambiare dati con il master DeviceNet attraverso 16 parole di uscita (ricevute dal master) situate tra VR(10) e VR(25) e 31 parole d'ingresso (inviate al master) situate tra VR(150) e VR(180).

Vedere anche N/D

Bit	Valore	Descrizione
0	0	DEVICENET (numero_unità, 2, ...) non ancora eseguito
	1	DEVICENET (numero_unità, 2, ...) eseguito senza errori
1	0	Nessun collegamento di I/O DeviceNet
	1	Collegamento di I/O DeviceNet in funzione
2	0	Le variabili VR dell'intervallo dati di uscita sono state aggiornate
	1	Le variabili VR dell'intervallo dati di uscita non sono state ancora aggiornate
3	0	Le dimensioni del collegamento di I/O DeviceNet corrispondono al comando DEVICENET (numero_unità, 2,...)
	1	Le dimensioni del collegamento di I/O DeviceNet non corrispondono ancora al comando DEVICENET (numero_unità, 2,...)
4 – 7	0	Sempre zero
8	0	Alimentazione di rete OK
	1	Guasto nell'alimentazione di rete
9	0	Non si è verificato nessun BUSOFF
	1	Si è verificato un BUSOFF
10	0	Nessun errore di duplicazione dell'indirizzo del nodo
	1	Errore di duplicazione dell'indirizzo del nodo

3.2.77 DIR

Tipo	Comando di programma
Sintassi	DIR LS
Descrizione	Il comando DIR mostra un elenco dei programmi contenuti nell'unità di controllo, le dimensioni della memoria e il RUNTYPE . DIR mostra anche le dimensioni della memoria disponibile, la modalità all'avvio e i programmi dell'unità di controllo correntemente selezionati.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	FREE , POWER_UP , PROCESS , RUNTYPE , SELECT .

3.2.78 DISABLE_GROUP

Tipo	Comando degli assi
Sintassi	DISABLE_GROUP(-1) DISABLE_GROUP(asse_1 [, asse_2 [, ...]])
Descrizione	Questo comando viene impiegato per creare un elenco di assi da sottoporre a disabilitazione di errori. Qualora venga creato un gruppo di assi, se un errore si verifica anche solo su un asse , AXIS_ENABLE e SERVO saranno disattivati su tutti gli assi. È possibile creare gruppi multipli, sebbene un asse non possa appartenere a più di un gruppo. È possibile cancellare tutti i raggruppamenti usando DISABLE_GROUP(-1) .
Argomenti	asse_i Espressione in BASIC il cui risultato è un numero di asse.

Esempio	DISABLE_GROUP(-1) DISABLE_GROUP(0,1,2,3) DISABLE_GROUP(4,5,6,7) WDOG=ON STOP abilita_b: FOR as=4 TO 7 AXIS_ENABLE AXIS(as)=ON NEXT as In un sistema a 8 assi, è necessario che gli assi 4 – 7 continuino a funzionare se gli assi 0 – 3 contengono un errore e viceversa. Gli assi vengono raggruppati usando DISABLE_GROUP . Nota: da usare solo con MECHATROLINK-II.

Vedere anche N/D

3.2.79 DISPLAY

Tipo	Parametro di sistema
Sintassi	DISPLAY=valore
Descrizione	Determina i canali di I/O da visualizzare sui LED del pannello frontale. Il parametro DISPLAY può essere utilizzato per selezionare l'insieme di I/O che deve essere visualizzato. I valori sono indicati nella tabella di seguito.
Argomenti	N/D
Esempio	DISPLAY=5 Mostra le uscite 8 – 15.
Vedere anche	N/D

valore	Descrizione
0	Ingressi da 0 a 7 (predefiniti)
1	Ingressi da 8 a 15
2	Ingressi da 16 a 23

valore	Descrizione
3	Ingressi da 24 a 31
4	Uscite da 0 a 7 (non utilizzate su Trajexia)
5	Uscite da 8 a 15
6	Uscite da 16 a 23
7	Uscite da 24 a 31

3.2.80 DPOS

Tipo	Parametro degli assi (sola lettura)
Sintassi	DPOS
Descrizione	Il parametro degli assi DPOS contiene la posizione richiesta che viene generata dai comandi di movimento del controllo servo, espressa in unità utente. Quando l'unità di controllo è in modalità ad anello aperto (SERVO=OFF), la posizione misurata (MPOS) viene copiata nel DPOS per mantenere l'errore di inseguimento a 0. L'intervallo della posizione richiesta è controllato attraverso i parametri degli assi REP_DIST e REP_OPTION. Il valore può essere regolato senza eseguire alcun movimento, utilizzando il comando DEFPOS o il parametro degli assi OFFPOS. All'avvio, DPOS viene reimpostato su 0.
Argomenti	N/D
Esempio	<pre>>> PRINT DPOS AXIS(0) 34.0000</pre> La riga qui sopra restituirà la posizione richiesta espressa in unità utente.
Vedere anche	AXIS, DPOS, DEFPOS, DEMAND_EDGES, FE, MPOS, REP_DIST, REP_OPTION, OFFPOS, UNITS.

3.2.81 DRIVE_ALARM

Tipo	Comando degli assi
Sintassi	DRIVE_ALARM(VR)
Descrizione	La funzione DRIVE_ALARM legge l'allarme corrente del servozionamento che è collegato al Trajexia attraverso MECHATROLINK-II. In caso di esecuzione riuscita, il comando restituisce -1 e registra il valore nella posizione della memoria VR specificata dal parametro VR. Se invece non è possibile eseguire il comando, viene restituito il valore 0. Il comando viene eseguito sul driver dell'asse di base impostato da BASE. L'asse di base può essere cambiato con il modificatore AXIS, alla stessa maniera di tutti gli altri comandi e parametri degli assi. Siccome questo comando attende la risposta dall'asse, l'esecuzione del comando può essere lenta e variare nel tempo. Se si desidera una risposta rapida, si consiglia di non utilizzare questo comando.
Argomenti	• VR In caso di esecuzione riuscita, il valore dell'allarme viene memorizzato all'indirizzo VR.
Esempio	<pre>IF NOT DRIVE_ALARM(10) AXIS(2) THEN PRINT "La lettura dell'allarme del servozionamento non è riuscita" ELSE IF VR(10) = 0 THEN PRINT "Servozionamento in buone condizioni" ELSE PRINT "Codice allarme servo: "; VR(10) ENDIF ENDIF</pre> In questo esempio, viene letto un allarme del servozionamento che aziona l'asse 2, dopodiché vengono presentate all'utente le informazioni relative.
Vedere anche	N/D

3.2.82 DRIVE_CLEAR

Tipo Comando degli assi

Sintassi **DRIVE_CLEAR**

Descrizione Il comando **DRIVE_CLEAR** cancella lo stato di allarme del servozionamento collegato attraverso il bus MECHATROLINK-II. Questo comando non è in grado di cancellare tutti gli stati di allarme possibili. Alcuni allarmi possono essere cancellati solo interrompendo l'alimentazione (sia di TJ1-MC___ che del servozionamento) e quindi riattivandola nuovamente.

Argomenti N/D

Esempio Nessun esempio.

Vedere anche **DRIVE_STATUS**.

Argomenti N/D

Esempio **DRIVE_CONTROL AXIS(2) = 256**

In questo esempio, OUT 0 è attivato per l'asse 2, collegato usando TJ1-FL02.

Vedere anche N/D

Codice	Descrizione
2	Errore seguente (questo è il vero FE quando viene utilizzato ATYPE=40)
8	Velocità di retroazione (con Atype=41 Unità=velocità massima/40000000H, con altre unità Atype= unità di riferimento/s)
9	Velocità di comando (stesse unità della velocità di retroazione)
10	Velocità di riferimento (stesse unità della velocità di retroazione)
11	Coppia (forza) di riferimento (con Atype=42 Unità=Coppia max/40000000H, con altre unità Atype=% al di sopra della coppia nominale)
14	Monitor selezionato con Pn813.0 Utile per controllare i monitor servozionamento (Unxxx)
15	Monitor selezionato con Pn813.1 Utile per controllare i monitor servozionamento (Unxxx)



Attenzione

Quando si esegue questo comando, assicurarsi che nessuna consolle di programmazione o software per personal computer sia collegato al servozionamento. In caso contrario, il task del programma sarà messo in pausa fino a quando l'altro dispositivo non verrà rimosso dal servozionamento.

3.2.83 DRIVE_CONTROL

Tipo Parametro degli assi

Sintassi **DRIVE_CONTROL**

Descrizione Questo parametro, quando viene applicato all'asse controllato dal servozionamento collegato al sistema attraverso il bus MECHATROLINK-II, seleziona i dati da controllare attraverso il **DRIVE_MONITOR** in base alla tabella di seguito. Se il servozionamento di un asse è collegato utilizzando TJ1-FL02, questo parametro imposta le uscite di TJ1-FL02. Per attivare OUT 0 su di un asse, impostare il bit 8 di questo parametro. Per attivare OUT 1 su di un asse, impostare il bit 9 di questo parametro. Tenere presente che le stesse uscite sono utilizzate dal comando **HW_PSWITCH**.

3.2.84 DRIVE_INPUTS

Tipo	Parametro degli assi
Sintassi	DRIVE_INPUTS
Descrizione	Questo parametro controlla lo stato degli ingressi del servozionamento collegato attraverso il bus MECHATROLINK-II. Il valore del parametro viene aggiornato a ciascun ciclo SERVO_PERIOD . Si tratta di un operatore bitwise, elencato nella tabella di seguito. L'impostazione consigliata è: Pn81E=4321 e Pn511=654x.
Argomenti	N/D
Esempio	In questo operatore, tutti gli ingressi possono essere controllati come segue (servozionamento Sigma-II): CN1-40 DRIVE_INPUTS bit 12 CN1-41 DRIVE_INPUTS bit 13 CN1-42 DRIVE_INPUTS bit 14 CN1-43 DRIVE_INPUTS bit 15 CN1-44 DRIVE_INPUTS bit 06 CN1-45 DRIVE_INPUTS bit 07 CN1-46 DRIVE_INPUTS bit 08
Esempio	In questo operatore, tutti gli ingressi possono essere controllati come segue (servozionamento Junma): CN1-1 DRIVE_INPUTS bit 6 CN1-2 DRIVE_INPUTS bit 2 CN1-3 DRIVE_INPUTS bit 1 CN1-4 DRIVE_INPUTS bit 0
Vedere anche	N/D

Numero di bit	Descrizione Sigma-II	Descrizione Junma
0	P_OT	P_OT
1	N_OT	N_OT
2	Segnale DEC (selezionato con Pn511.0)	/DEC
3	Fase A Encoder	N/C

Numero di bit	Descrizione Sigma-II	Descrizione Junma
4	Fase B Encoder	N/C
5	Fase C Encoder	N/C
6	Segnale EXT1 (selezionato con Pn511.1)	/EXT1
7	Segnale EXT2 (selezionato con Pn511.2)	N/C
8	Segnale EXT3 (selezionato con Pn511.3)	N/C
9	BRK Uscita freno	/BK
10	Riservato	E_STP
11	Riservato	N/C
12	IO12 (segnale di ingresso CN1 selezionato in Pn81E.0)	N/C
13	IO13 (segnale di ingresso CN1 selezionato in Pn81E.1)	N/C
14	IO14 (segnale di ingresso CN1 selezionato in Pn81E.2)	N/C
15	IO15 (segnale di ingresso CN1 selezionato in Pn81E.3)	N/C

3.2.85 DRIVE_MONITOR

Tipo	Parametro degli assi
Sintassi	DRIVE_MONITOR
Descrizione	Questo parametro contiene i dati controllati del servozionamento collegato al sistema attraverso il bus MECHATROLINK-II. I dati da controllare vengono selezionati utilizzando DRIVE_CONTROL e possono essere visualizzati nell'ambito di Trajexia Tools oppure utilizzati all'interno di un programma. I dati controllati sono aggiornati a ogni SERVO_PERIOD .
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	N/D

3.2.86 DRIVE_READ

Tipo	Comando degli assi
Sintassi	DRIVE_READ(parametro,dimensioni,VR)
Descrizione	La funzione DRIVE_READ legge il parametro del servozionamento specificato e collegato al Trajexia attraverso il bus MECHATROLINK-II. In caso di esecuzione riuscita, questo comando restituisce -1 e colloca il valore letto nella posizione della memoria VR specificata dal parametro VR. Se invece non è possibile eseguire il comando, viene restituito il valore 0. Il comando viene eseguito sul driver dell'asse di base impostato con BASE. Può essere cambiato usando il modificatore AXIS, alla stessa maniera di tutti gli altri comandi e parametri degli assi. Nota: siccome questo comando attende la risposta dall'asse, la sua esecuzione è lenta e varia nel tempo. Si consiglia di non usare questo comando assieme ad altri comandi che richiedono una rapida esecuzione. Nota: l'esecuzione di un comando DRIVE_READ disabilita temporaneamente il display sul pannello frontale del servozionamento. Nota: DRIVE_READ restituisce -1 in caso di riuscita. Restituisce -1 anche quando non viene letto nessun parametro, se il numero del parametro non esiste o è di dimensioni errate.
Argomenti	• parametro Il numero di parametro che deve essere letto. Notare che i numeri dei parametri sono esadecimali. È possibile reperire il formato dei dati nel manuale del servozionamento. • dimensioni Per la maggior parte dei parametri, le dimensioni sono solitamente di 2 byte. Alcuni parametri speciali possono essere lunghi 4 byte. Le dimensioni di ciascun parametro possono essere reperite nel manuale del servozionamento. • VR L'indirizzo VR dove il parametro letto viene memorizzato in seguito ad una esecuzione riuscita.
Esempio	<pre>IF DRIVE_READ(\$100,2,1) THEN PRINT "Il guadagno dell'anello di velocità è: ";VR(1) ELSE PRINT "Impossibile leggere il guadagno dell'anello di velocità" ENDIF</pre>
Vedere anche	DRIVE_WRITE, HEX, \$ (DATI ESADECIMALI).



Attenzione

Quando si esegue questo comando, assicurarsi che nessuna consolle di programmazione o software per personal computer sia collegato al servozionamento. In caso contrario, il task del programma sarà messo in pausa fino a quando l'altro dispositivo non verrà rimosso dal servozionamento.

3.2.87 DRIVE_RESET

Tipo	Comando degli assi
Sintassi	DRIVE_RESET
Descrizione	Il comando DRIVE_RESET reimposta lo stato di allarme del servozionamento collegato attraverso il bus MECHATROLINK-II.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	N/D



Attenzione

Quando si esegue questo comando, assicurarsi che nessuna consolle di programmazione o software per personal computer sia collegato al servozionamento. In caso contrario, il task del programma sarà messo in pausa fino a quando l'altro dispositivo non verrà rimosso dal servozionamento.

3.2.88 DRIVE_STATUS

Tipo Parametro degli assi (sola lettura)

Sintassi **DRIVE_STATUS**

Descrizione Per gli assi MECHATROLINK-II, questo parametro viene impostato a partire dal campo STATUS del frame di comunicazione di MECHATROLINK-II e viene aggiornato a ogni ciclo di servoazionamento. Questi bit sono visibili nella finestra di configurazione Intelligent Drives di Trajexia Tools e possono essere utilizzati all'interno dei programmi. La tabella di seguito fornisce una spiegazione per ciascun bit. (Nota: sono elencati solo i bit relativi a MECHATROLINK-II.)

Per una spiegazione dettagliata di questi bit di stato, consultare il manuale MECHATROLINK-II.

Per gli assi definiti come assi flessibili, questo parametro conserva lo stato della registrazione e gli ingressi ausiliari, nonché la selezione della registrazione. Una seconda tabella di seguito fornisce una spiegazione per ciascun bit. (Nota: sono elencati solo i bit relativi all'asse flessibile.)

Argomenti N/D

Esempio **PRINT DRIVE_STATUS AXIS(4)**

Questo comando stampa il valore corrente del **DRIVE_STATUS** per l'asse(4).

Esempio **BASE(3)**

ATYPE = 44

IF (DRIVE_STATUS AND 32)= 32 THEN

PRINT "L'ingresso REG 0 è ON per asse(3)"

ENDIF

Vedere anche **AXIS, MARK, MARKB, REGIST.**

Bit	Descrizione (MECHATROLINK-II)
4	Alimentazione attiva
5	Blocco macchina
6	Posizione iniziale
7	Alla posizione/velocità
8	Invio completato
9	Limite di coppia
10	Blocco completato
11	Entro l'intervallo/limite di velocità

Bit	Descrizione (asse flessibile)
0	MARK
1	MARKB
2	Valore corrente selezionato REG 0
3	Valore corrente selezionato REG 1
4	Valore corrente AUX IN
5	Valore corrente REG 0
6	Valore corrente REG 1

Bit	Descrizione (MECHATROLINK-II)
0	Allarme
1	Avviso
2	Pronto
3	Servoazionamento attivo

3.2.89 DRIVE_WRITE

Tipo	Comando degli assi
Sintassi	DRIVE_WRITE (parametro, dimensioni, valore [,modalità])
Descrizione	La funzione DRIVE_WRITE scrive il parametro del servozionamento specificato e collegato al Trajexia attraverso il bus MECHATROLINK-II. In caso di esecuzione riuscita, questo comando restituisce -1. Se invece non è possibile eseguire il comando, viene restituito il valore 0. Il comando viene eseguito sul driver dell'asse di base impostato con BASE. Può essere cambiato usando il modificatore AXIS, alla stessa maniera degli altri comandi e parametri degli assi. Per alcuni parametri, è necessario interrompere e riattivare l'alimentazione del driver. Il comando DRIVE_RESET può essere utilizzato a questo scopo. Nota: siccome questo comando attende la risposta dall'asse, la sua esecuzione è lenta e varia nel tempo. Si consiglia di non usare questo comando assieme ad altri comandi che richiedono una rapida esecuzione. Nota: l'esecuzione di un comando DRIVE_WRITE disabilita temporaneamente il display sul pannello frontale del servozionamento. Nota: DRIVE_WRITE restituisce -1 in caso di riuscita. Restituisce -1 anche quando non viene letto nessun parametro, se il numero del parametro non esiste o è di dimensioni errate.
Argomenti	• parametro Il numero di parametro da scrivere. Notare che i numeri dei parametri sono esadecimali. Per il formato dei dati, consultare il manuale del servozionamento. • dimensioni Per la maggior parte dei parametri, le dimensioni sono solitamente di 2 byte. Alcuni parametri speciali possono essere lunghi 4 byte. Le dimensioni di ciascun parametro possono essere reperite nel manuale del servozionamento. • valore Il valore che deve essere scritto nel parametro del driver. • modalità La modalità di scrittura. Valori possibili: 0 (od omesso) – scrittura e memorizzazione nella RAM; 1 – scrittura e memorizzazione nella memoria EPROM.

Esempio	<pre>IF DRIVE_WRITE(\$100,2,90) THEN PRINT "Il nuovo guadagno dell'anello di velocità è: 90" ELSE PRINT "Impossibile scrivere nella RAM il guadagno dell'anello di velocità" ENDIF</pre>
Vedere anche	• DRIVE_READ, DRIVE_RESET, \$ (DATI ESADECIMALI)



Attenzione

Quando si esegue questo comando, assicurarsi che nessuna consolle di programmazione o software per personal computer sia collegato al servozionamento. In caso contrario, il task del programma sarà messo in pausa fino a quando l'altro dispositivo non verrà rimosso dal servozionamento.

3.2.90 EDIT

Tipo	Comando di programma
Sintassi	EDIT [numero_riga] ED [numero_riga]
Descrizione	Il comando EDIT avvia l'editor a schermo incorporato, consentendo la modifica di un programma contenuto nell'unità di controllo attraverso un terminale VT100. Il programma correntemente selezionato sarà modificato. I comandi dell'editor sono i seguenti: questo comando è implementato per un terminale offline (VT100). All'interno di Trajexia Tools, gli utenti possono selezionare il comando dal menu Program. • Uscire dall'editor: [CTRL] K e D • Cancellare riga: [CTRL] Y
Argomenti	• numero_riga Il numero della riga dove iniziare la modifica.
Esempio	Nessun esempio.
Vedere anche	SELECT .

3.2.91 ELSE

Vedere IF..THEN..ELSE..ENDIF.

3.2.92 ELSEIF

Vedere IF..THEN..ELSE..ENDIF.

3.2.93 ENCODER

Tipo	Parametro degli assi (sola lettura)
Sintassi	ENCODER
Descrizione	Il parametro degli assi ENCODER contiene una copia non elaborata dell'encoder. Il parametro degli assi MPOS contiene la posizione misurata, calcolata automaticamente a partire dal valore dell' ENCODER , che consente overflow e offset.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	AXIS , MPOS .

3.2.94 ENCODER_BITS

Tipo	Parametro degli assi
Sintassi	ENCODER_BITS = valore
Descrizione	Questo parametro degli assi configura l'interfaccia per il numero di bit dell'encoder relativi agli assi flessibili assoluti dell'encoder EnDat e SSI. Il parametro può essere applicato agli assi con valore ATYPE 47 e 48. Se applicato a un asse flessibile dell'encoder assoluto EnDat, i bit da 0 a 7 del parametro devono essere impostati sul numero totale dei bit dell'encoder. I bit da 8 a 14 devono essere impostati sul numero di bit multigiro da usare. Se applicato a un asse flessibile dell'encoder assoluto SSI, i bit da 0 a 5 del parametro devono essere impostati sul numero dei bit dell'encoder. Il bit 6 deve essere 1 in caso di operazione binaria o 0 in caso di codice Gray. Nota: usando un asse flessibile assoluto dell'encoder, è essenziale impostare questo parametro per l'asse prima di impostare ATYPE . Argomenti N/D

Esempio	ENCODER_BITS = 25 + (256 * 12) ATYPE = 47 In questo esempio viene utilizzato un encoder EnDat a 25 bit dotato di 12 bit per i valori multigiro e 13 bit per ciascuna rivoluzione.
Esempio	ENCODER_BITS = 12 + (64 * 1) ATYPE = 48 In questo esempio viene utilizzato un encoder SSI a 12 bit (4096 posizioni per ciascuna rivoluzione), dotato di un'uscita di tipo binario.
Vedere anche	AXIS .

3.2.95 ENCODER_CONTROL

Tipo	Parametro degli assi
Sintassi	ENCODER_CONTROL = valore
Descrizione	Il parametro ENCODER_CONTROL è applicabile unicamente a un asse flessibile assoluto EnDat con valore ATYPE 47. Il parametro controlla la modalità usata dall'encoder EnDat per restituirne la posizione. È possibile impostare l'encoder affinché restituisca la propria posizione ciclicamente o impostarlo su una modalità di lettura/scrittura del parametro. L'impostazione predefinita dopo l'inizializzazione è la modalità di restituzione ciclica della posizione. Per maggiori informazioni, vedere le specifiche dell'interfaccia dell'encoder assoluto EnDat.
Argomenti	N/D
Esempio	ENCODER_CONTROL AXIS(1) = 0 Questo comando imposta la modalità di restituzione ciclica della posizione.
Esempio	ENCODER_CONTROL AXIS(1) = 1 Questo comando imposta il parametro di modalità di lettura/scrittura.
Vedere anche	AXIS , ENCODER , ENCODER_BITS .

3.2.96 ENCODER_ID

Tipo	Parametro degli assi (sola lettura)
Sintassi	ENCODER_ID
Descrizione	Questo parametro restituisce il valore ID di un encoder assoluto per l'asse. Questo parametro è applicabile solo all'asse flessibile assoluto Tamagawa con valore ATYPE 46. Restituisce il parametro ENID dall'encoder, che è impostato su 17. Per maggiori informazioni, vedere le caratteristiche dell'interfaccia dell'encoder assoluto Tamagawa. Se applicato a un asse con valore ATYPE diverso da 46, questo parametro restituisce uno 0.
Argomenti	N/D
Esempio	>>PRINT ENCODER_ID AXIS (1) 17.0000 Questo comando stampa il valore ID dell'encoder assoluto per l'asse 1.
Vedere anche	AXIS , ENCODER , ENCODER_BITS .

3.2.97 ENCODER_RATIO

Tipo	Parametro degli assi
Sintassi	ENCODER_RATIO(denominatore,numeratore)
Descrizione	Consente al conteggio dell'encoder in entrata di essere scalato tramite un numero non intero, utilizzando l'equazione: MPOS = (numeratore)/(denominatore) x ingresso fronti dell'encoder Contrariamente ai parametri UNITS , ENCODER_RATIO influenza sia MOVECIRC che CAMBOX . Nota: i rapporti alti devono essere evitati, poiché provocano una perdita di risoluzione oppure riducono considerevolmente l'uniformità del movimento. Il conteggio effettivo e fisico dell'encoder corrisponde alla risoluzione BASIC dell'asse e l'utilizzo di questo comando potrebbe ridurre la capacità dell'unità di controllo di raggiungere con precisione tutte le posizioni. Nota: ENCODER_RATIO non sostituisce UNITS . ENCODER_RATIO deve essere utilizzato solo quando è assolutamente necessario. In tutti gli altri casi, per scalare gli assi utilizzare UNITS .

Argomenti	• denominatore Un numero compreso tra 0 e 16777215, usato per definire il denominatore dell'equazione qui sopra. • numeratore Un numero compreso tra 0 e 16777215, usato per definire il numeratore dell'equazione qui sopra.
Esempio	ENCODER_RATIO(8192,7200) UNITS=20 Un tavolo girevole è dotato di un servomotore collegato direttamente al suo centro di rotazione. Un encoder è montato sul lato posteriore del servomotore e restituisce un valore di 8192 conteggi per giro. L'applicazione richiede che il tavolo venga calibrato in gradi, ma in maniera tale per cui un grado corrisponda a un numero intero di conteggi.
Vedere anche	N/D

3.2.98 ENCODER_READ

Tipo	Comando degli assi
Sintassi	ENCODER_READ(indirizzo)
Descrizione	Il comando ENCODER_READ è applicabile unicamente a un asse flessibile assoluto EnDat con valore ATYPE 47. Il parametro restituisce un parametro encoder a 16 bit memorizzato in un indirizzo specifico. I bit da 8 a 15 dell'indirizzo sono le impostazioni campo EnDat MRS mentre i bit da 0 a 7 corrispondono all'offset interno al blocco EnDat MRS. Se si verifica un errore CRC, questo comando restituirà -1. Per maggiori informazioni, vedere le caratteristiche dell'interfaccia dell'encoder assoluto EnDat.
Argomenti	• indirizzo Specifica quale campo EnDat MRS deve essere letto.
Esempio	VR(100) = ENCODER_READ(\$A10D) AXIS(7) Questo comando legge il numero dei bit dell'encoder e colloca quel valore nell'area di memoria VR(10).
Vedere anche	AXIS , ENCODER , ENCODER_BITS .

3.2.99 ENCODER_STATUS

Tipo	Parametro degli assi (sola lettura)
Sintassi	ENCODER_STATUS
Descrizione	Questo parametro restituisce lo stato dell'encoder assoluto. Questo parametro può essere applicato solo all'asse flessibile assoluto Tamagawa con valore ATYPE 46. Restituisce sia il campo di stato SF sia il campo di errore dell'encoder ALMC. Il campo SF si trova nei bit da 0 a 7, mentre il campo ALMC è nei bit da 8 a 15. Per maggiori informazioni, vedere le caratteristiche dell'interfaccia dell'encoder assoluto Tamagawa. Se applicato a un asse con valore ATYPE diverso da 46, questo parametro restituisce uno 0.
Argomenti	N/D
Esempio	PRINT (ENCODER_STATUS AXIS (1) AND 255) Questo comando stampa il campo SF dell'encoder assoluto Tamagawa per l'asse 1.
Vedere anche	AXIS, ENCODER, ENCODER_BITS.

3.2.100 ENCODER_TURNS

Tipo	Parametro degli assi (sola lettura)
Sintassi	ENCODER_TURNS
Descrizione	Il parametro ENCODER_TURNS restituisce il numero del conteggio multigiro proveniente dall'encoder. Può essere applicato solo all'asse flessibile assoluto Tamagawa con valore ATYPE 46 e all'asse flessibile assoluto con valore ATYPE 47. I dati multigiro non sono automaticamente applicati al parametro degli assi MPOS dopo l'inizializzazione. Il programmatore dell'applicazione deve applicare il parametro a partire dal programma utilizzando, se necessario, il comando OFFPOS o DEFPOS . Se applicato a un asse con valore ATYPE diverso da 46 o 47, questo parametro restituisce uno 0.
Argomenti	N/D

Esempio **PRINT ENCODER_TURNS AXIS (1)**
Questo comando stampa il conteggio multigiro dell'encoder assoluto per l'asse 1.

Vedere anche **AXIS, ENCODER, ENCODER_BITS.**

3.2.101 ENCODER_WRITE

Tipo	Comando degli assi
Sintassi	ENCODER_WRITE(indirizzo, valore)
Descrizione	Il comando ENCODER_WRITE è applicabile unicamente a un asse flessibile assoluto EnDat con valore ATYPE 47. Il comando scrive su un parametro encoder specificato dall'indirizzo. I bit da 8 a 15 dell'indirizzo sono le impostazioni campo EnDat MRS mentre i bit da 0 a 7 corrispondono all'offset interno al blocco EnDat MRS. Se si verifica un errore CRC, questo comando restituirà 0. La scrittura di 0 all'indirizzo svolge la funzione di ripristino dell'encoder. Per maggiori informazioni, vedere le caratteristiche dell'interfaccia dell'encoder assoluto EnDat. Per poter scrivere un parametro dell'encoder con questo comando, il parametro ENCODER_WRITE deve essere impostato su 1, modalità di lettura/scrittura del parametro encoder.
Argomenti	• indirizzo Specifica quale campo EnDat MRS deve essere scritto. • valore Espressione in BASIC.
Esempio	Nessun esempio.
Vedere anche	AXIS, ENCODER, ENCODER_BITS, ENCODER_CONTROL.

3.2.102 ENDIF

Vedere **IF..THEN..ELSE..ENDIF.**

3.2.103 ENDMOVE

Tipo	Parametro degli assi
Sintassi	ENDMOVE
Descrizione	Il parametro degli assi ENDMOVE contiene la posizione finale del movimento corrente espresso in unità utente. Se il parametro degli assi SERVO è attivo, diventa possibile scrivere il parametro ENDMOVE per produrre un cambiamento passo-passo della posizione richiesta (DPOS). Note: dal momento che la posizione misurata non viene cambiata all'inizio, occorre tenere in considerazione il limite dell'errore di inseguimento (FE_LIMIT). Se il cambiamento della posizione richiesta è eccessivo, il limite sarà superato.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	AXIS, DPOS, FE_LIMIT, UNITS.

3.2.104 EPROM

Tipo	Comando di programma
Sintassi	EPROM
Descrizione	Il comando EPROM registra nella memoria RAM di TJ1-MC__ alimentata a batteria i programmi BASIC contenuti nella memoria flash EPROM. Il parametro di sistema POWER_UP controlla la copia nella RAM dei programmi registrati nella memoria flash EPROM. Nota: in Trajexia Tools questo comando viene presentato come pulsante sul pannello di controllo. Vi sono anche pagine pop-up che chiedono se si desiderano scrivere i dati del programma nella memoria flash.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	POWER_UP, RUNTYPE.

3.2.105 ERROR_AXIS

Tipo	Parametro di sistema (sola lettura)
Sintassi	ERROR_AXIS
Descrizione	Il parametro degli assi ERROR_AXIS contiene il numero dell'asse che ha provocato l'errore di movimento. Un errore di movimento si verifica quando lo stato AXISSTATUS per uno degli assi corrisponde all'impostazione ERRORMASK . In questo caso il selettore di attivazione (WDOG) viene disattivato, il parametro MOTION_ERROR prende il valore 1 e il parametro ERROR_AXIS prende il numero del primo asse contenente l'errore.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	AXISSTATUS, ERRORMASK, MOTION_ERROR, WDOG.

3.2.106 ERROR_LINE

Tipo	Parametro dei task (sola lettura)
Sintassi	ERROR_LINE
Descrizione	Il parametro ERROR_LINE contiene il numero della riga che ha causato l'ultimo errore BASIC di run-time nel task di programma. Tale valore è valido solo quando il parametro BASICERROR è TRUE . Ciascun task dispone di un parametro ERROR_LINE proprio. Usare il modificatore PROC per accedere al parametro di un determinato task. Senza PROC viene preso in considerazione il task corrente.
Argomenti	N/D
Esempio	>> PRINT ERROR_LINE PROC(4) 23.0000
Vedere anche	BASICERROR, PROC, RUN_ERROR.

3.2.107 ERRORMASK

Tipo	Parametro degli assi
Sintassi	ERRORMASK
Descrizione	Il parametro degli assi ERRORMASK contiene un valore di maschera che viene elaborato un bit alla volta con il parametro degli assi AXISSTATUS attraverso l'operatore AND, a ciascun ciclo di servoazionamento per stabilire se si è verificato un errore di movimento. Quando si verifica un errore, il selettore di attivazione (WDOG) viene spento, il parametro MOTION_ERROR prende il valore 1 e il parametro ERROR_AXIS prende il numero del primo asse contenente l'errore. Per conoscere lo stato delle assegnazioni dei bit, controllare il parametro AXISVALUES. L'impostazione predefinita di ERRORMASK è 268.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	AXIS , AXISSTATUS , MOTION_ERROR , WDOG .

**Attenzione**

È compito dell'utente definire in quali casi viene generato un errore di movimento. Per un funzionamento sicuro si consiglia vivamente di generare un errore di movimento in tutti i casi in cui l'errore di inseguimento supera il limite. Per fare ciò, impostare il bit 8 di **ERRORMASK**

3.2.108 ETHERNET

Tipo	Comando di sistema
Sintassi	ETHERNET(funzione, numero_unità, parametro [,valori])
Descrizione	Il comando ETHERNET viene utilizzato per leggere e impostare certe funzioni della comunicazione Ethernet. Il comando ETHERNET deve essere inserito nella riga di comando attraverso la porta seriale 0, quando Trajexia Tools è in modalità scollegata. Nota: per attivare i nuovi parametri è necessario riavviare Trajexia.
Argomenti	• funzione 0 = lettura, 1 = scrittura. • numero_unità -1. • parametro 0 = indirizzo IP; 2 = subnet mask; 3 = indirizzo MAC; 8 = gateway; 11 = cache ARP (sola lettura). • valori Il parametro richiesto per una scrittura.
Esempio	ETHERNET(1,-1,0,192,200,185,2) Impostare l'indirizzo IP di Trajexia su 192.200.185.002.
Vedere anche	N/D

3.2.109 EX

Tipo	Comando di sistema
Sintassi	EX[(opzione)]
Descrizione	Ripristina l'unità di controllo come se venisse riaccesa. Il comando EX esegue due tipi di ripristino diversi. EX senza argomento ovvero EX(0) esegue il ripristino del software dell'unità di controllo. EX(1) esegue il ripristino dell'hardware dell'unità di controllo
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	N/D

3.2.110 EXP

Tipo	Funzione matematica
Sintassi	EXP(espressione)
Descrizione	La funzione EXP restituisce il valore esponenziale dell'espressione.
Argomenti	espressione Qualsiasi espressione valida in BASIC.
Esempio	>> PRINT EXP(1.0) 2.7183
Vedere anche	N/D

3.2.111 FALSE

Tipo	Costante (sola lettura)
Sintassi	FALSE
Descrizione	La costante FALSE restituisce il valore numerico 0.
Argomenti	N/D
Esempio	test: res = IN(0) OR IN(2) IF res = FALSE THEN PRINT "Gli ingressi non sono attivi" ENDIF
Vedere anche	N/D

3.2.112 FAST_JOG

Tipo	Parametro degli assi
Sintassi	FAST_JOG
Descrizione	Il parametro degli assi FAST_JOG contiene il numero d'ingresso da utilizzare come ingresso di jog rapido. Il numero può essere compreso tra 0 e 7. Dal momento che l'impostazione predefinita del parametro è -1, nessun ingresso viene selezionato. L'ingresso di jog rapido regola la velocità di jog tra due velocità. Se viene impostato l'ingresso di jog rapido, per il funzionamento intermittente sarà utilizzata la velocità data dal parametro degli assi SPEED. Se non viene impostato l'ingresso di jog rapido, sarà utilizzata la velocità data dal parametro degli assi JOGSPEED. Nota: questo ingresso è active low.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	AXIS, FWD_JOG, JOGSPEED, REV_JOG, SPEED.

3.2.113 FASTDEC

Tipo	Parametro degli assi
Sintassi	FASTDEC
Descrizione	Il valore predefinito è zero. Se viene specificato un FASTDEC diverso da zero, l'asse rallenterà fino a zero alla velocità di decelerazione precisata quando raggiunge il finecorsa o la posizione stabilita.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	N/D

3.2.114 FE

Tipo	Parametro degli assi (sola lettura)
Sintassi	FE
Descrizione	Il parametro degli assi FE contiene l'errore di posizione espresso in unità utente. Questo viene calcolato sottraendo alla posizione richiesta (parametro asse DPOS) la posizione misurata (parametro asse MPOS). Il valore dell'errore di inseguimento può essere verificato utilizzando i parametri degli assi FE_LIMIT e FE_RANGE .
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	AXIS , DPOS , FE_LIMIT , FE_RANGE , MPOS , UNITS .

3.2.115 FE_LATCH

Tipo	Parametro degli assi (sola lettura)
Sintassi	FE_LATCH
Descrizione	Contiene il valore iniziale FE che ha spinto l'asse a mettere l'unità di controllo in MOTION_ERROR . Il valore viene impostato solo quando FE supera FE_LIMIT e il parametro SERVO è stato impostato su 0. FE_LATCH viene reimpostato su 0 quando il parametro degli assi SERVO viene riportato a 1.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	N/D

3.2.116 FE_LIMIT

Tipo	Parametro degli assi
Sintassi	FE_LIMIT
Descrizione	Il parametro degli assi FE_LIMIT contiene il limite massimo ammesso per l'errore di inseguimento espresso in unità utente. Quando viene superato, il bit 8 del parametro degli assi AXISSTATUS viene impostato. Se il parametro ERRORMASK è stato impostato correttamente, viene generato un errore di movimento. Questo limite viene utilizzato come protezione contro varie condizioni di errore, ad esempio i blocchi meccanici, la perdita di retroazione dell'encoder, ecc.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	AXIS , AXISSTATUS , ERRORMASK , FE , FE_RANGE , UNITS .

3.2.117 FE_LIMIT_MODE

Tipo	Parametro degli assi
Sintassi	FE_LIMIT_MODE=valore
Descrizione	Quando questo parametro è impostato su 0, l'asse provocherà immediatamente un MOTION_ERROR non appena FE supera il valore FE_LIMIT . Se FE_LIMIT_MODE è impostato su 1, l'asse genererà soltanto un MOTION_ERROR qualora FE superasse FE_LIMIT nel corso di 2 tempi di servoazionamento consecutivi. Ciò significa che se FE_LIMIT viene superato durante un solo ciclo di servoazionamento, l'evento sarà ignorato. Il valore predefinito di FE_LIMIT_MODE è 0.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	N/D

3.2.118 FE_RANGE

Tipo	Parametro degli assi
Sintassi	FE_RANGE
Descrizione	Il parametro degli assi FE_RANGE contiene il limite dell'intervallo d'avviso di errore di inseguimento espresso in unità utente. Quando su un servoaasse l'errore di inseguimento supera questo valore, il bit 1 presente sul parametro degli assi AXISSTATUS viene attivato. Questo intervallo viene utilizzato come primo indicatore che una condizione di errore è presente nell'applicazione (confrontare FE_LIMIT).
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	AXIS , AXISSTATUS , ERRORMASK , FE , UNITS .

3.2.119 FHOLD_IN

Tipo	Parametro degli assi
Sintassi	FHOLD_IN FH_IN
Descrizione	Il parametro degli assi FHOLD_IN contiene il numero d'ingresso da utilizzare come ingresso per la sospensione dell'alimentazione. L'intervallo d'ingresso valido è compreso tra 0 e 31. I valori da 0 a 15 rappresentano gli ingressi fisicamente presenti sul connettore di I/O di TJ1-MC__ e sono comuni a tutti gli assi. I valori da 16 a 27 rappresentano gli ingressi del software che possono essere liberamente utilizzati da programmi e comandi quali IN e OP. Sono inoltre valori comuni a tutti gli assi. I valori da 28 a 31 vengono direttamente mappati agli ingressi del servoazionamento presenti sul connettore CN1 e sono univocamente associati a ogni asse. La mappatura degli ingressi del servoazionamento agli ingressi da 28 a 31 dipende dall'impostazione del parametro di servoazionamento Pn81E. L'impostazione consigliata è Pn81E = 0x4321, con la seguente mappatura: Come impostazione predefinita, il parametro è impostato su -1 e nessun ingresso è selezionato. Nota: questo ingresso è active low.
Sigma II	• ingresso 28: CN1-40 • ingresso 29: CN1-41 • ingresso 30: CN1-42 • ingresso 31: CN1-43
Sigma III	• ingresso 28: CN1-13 • ingresso 29: CN1-7 • ingresso 30: CN1-8 • ingresso 31: CN1-9
Junma	• ingresso 26: CN1-2 • ingresso 27: CN1-1

Per maggiori informazioni su come impostare il parametro di servoazionamento Pn81E, consultare il Manuale del servoazionamento. Come impostazione predefinita, il parametro è impostato su -1 e nessun ingresso è selezionato.

Se un numero d'ingresso viene impostato e l'ingresso della sospensione dell'alimentazione passa da non impostato a impostato, la velocità di movimento dell'asse cambia e prende il valore impostato nel parametro degli assi **FHSPEED**. Il movimento corrente non viene cancellato. Inoltre, il bit 7 del parametro **AXISSTATUS** viene impostato. Quando l'ingresso si ripristina, qualsiasi movimento in corso al momento in cui l'ingresso era stato impostato tornerà alla velocità programmata.

Nota: questa funzione agisce soltanto sui movimenti controllati dalla velocità. I movimenti che non sono controllati dalla velocità (**CAMBOX**, **CONNECT** e **MOVELINK**) non vengono influenzati.

Argomenti N/D

Esempio Nessun esempio.

Vedere anche **AXIS**, **AXISSTATUS**, **UNITS**.

3.2.120 FHSPEED

Tipo Parametro degli assi

Sintassi **FHSPEED**

Descrizione Il parametro degli assi **FHSPEED** contiene la velocità di sospensione dell'alimentazione. Questo parametro può essere impostato su un valore espresso in unità utente/s, velocità alla quale si muove l'asse quando si attiva l'ingresso di sospensione dell'alimentazione. Il movimento corrente non viene cancellato. **FHSPEED** può avere qualsiasi valore positivo, incluso lo 0. Il valore predefinito è 0.

Nota: questa funzione agisce soltanto sui movimenti controllati dalla velocità. I movimenti che non sono controllati dalla velocità (**CAMBOX**, **CONNECT** e **MOVELINK**) non vengono influenzati.

Argomenti N/D

Esempio Nessun esempio.

Vedere anche **AXIS**, **AXISSTATUS**, **FHOLD_IN**, **UNITS**.

3.2.121 FINS_COMMS

Tipo Comando di comunicazione

Sintassi **FINS_COMMS(tipo, rete, nodo, unità, area_remota, offset_remoto, lunghezza, area_locale, offset_locale, timeout [, ip1, ip2, ip3, ip4])**

Descrizione FINS (Factory Interface Network Service) è un protocollo di comunicazione proprietario di OMRON. Una parte di questo protocollo è stata integrata in Trajexia. Il protocollo FINS è stato integrato con l'intenzione di abilitare una comunicazione senza soluzione di continuità con altri dispositivi OMRON (PLC, HMI, ecc.) e altri software (CX-Drive, CX-Server, ecc.). Per maggiori informazioni sul protocollo di comunicazione FINS, consultare la sezione 4.2.4 e il Communication Commands Reference Manual, num. cat. W342-E1, sezioni 3 e 5.

Trajexia dispone di funzionalità client di FINS incorporate ed è pertanto in grado di avviare le comunicazioni FINS con dispositivi slave di FINS, utilizzando **FINS_COMMS**. Entrambi i comandi FINS 0101 (lettura di memoria) e FINS 0102 (scrittura di memoria) sono attivi. Il comando FINS 0101 consente la lettura della memoria da altri dispositivi dotati di funzionalità server FINS. FINS 0102 può essere utilizzato per scrivere dati su altri dispositivi dotati di funzionalità server FINS.

Questo comando restituisce uno dei seguenti valori, in base al risultato dell'esecuzione:

- 1: il comando è stato eseguito con successo.
- 0: il comando non è riuscito.
- 1: richiesta non inviata perché il client o il protocollo FINS è occupato.
- 2: uno o più dei parametri richiesti non sono validi.
- 3: area di memoria sorgente non valida.
- 4: la richiesta è stata inviata, ma il server remoto non ha risposto entro il periodo di timeout.
- 5: il server remoto ha inviato un codice di risposta di errore.

Argomenti

- **tipo**
Il tipo di comando FINS. 0 significa FINS 0101, lettura della memoria del server remoto FINS. 1 significa FINS 0102, scrittura nella memoria del server remoto.
- **rete**
La rete di destinazione. Per maggiori informazioni, consultare il Communication Commands Reference Manual, num. cat W342-E1, sezione 3.
- **nodo**
Il nodo del server FINS di destinazione. Per maggiori informazioni, consultare il Communication Commands Reference Manual, num. cat W342-E1, sezione 3.
- **unità**
Il numero di unità del server FINS di destinazione. Per maggiori informazioni, consultare il Communication Commands Reference Manual, num. cat W342-E1, sezione 3.
- **area_remota**
L'area di memoria d'accesso del server FINS di destinazione. Intervallo: 128..255. Notare che se la destinazione è un altro Trajexia, quest'area deve essere rappresentata da uno dei seguenti valori: 0xB0: valore memoria VR a numero intero; 0x82: valore memoria TABLE a numero intero; 0xC2: valore memoria TABLE con numero a virgola mobile.
- **offset_remoto**
L'offset di memoria presente sul server FINS di destinazione. Intervallo: 0..65535. Notare che se il sistema di destinazione è un altro Trajexia, questo intervallo verrà ulteriormente limitato agli indirizzi massimi della memoria TABLE o della memoria VR.
- **lunghezza**
Il numero degli elementi che devono essere trasferiti. L'intervallo dipende dalla lunghezza del frame FINS e dalla capacità del client e dei server remoti. L'intervallo per un Trajexia è compreso tra i numeri interi 1 e 700 o tra i numeri a virgola variabile da 1 a 350.
- **area_locale**
L'area di memoria locale (sorgente). Notare che se la destinazione è un altro Trajexia, quest'area deve essere rappresentata da uno dei seguenti valori: 0x00: valore memoria VR a numero intero; 0x01: valore memoria TABLE a numero intero; 0x02: valore memoria TABLE con numero a virgola mobile.

- **offset_locale**
L'offset del primo valore dell'area di memoria locale (sorgente). L'intervallo dipende dalle dimensioni della matrice della memoria VR o TABLE e dal valore dell'argomento "lunghezza".
- **timeout**
Il numero di millisecondi di attesa di una risposta dal server FINS di destinazione che devono trascorrere prima del timeout.
- **IP1, IP2, IP3, IP4**
Parametri opzionali che definiscono l'indirizzo IP del server remoto (di destinazione). Questi argomenti devono essere utilizzati se Trajexia e il server di destinazione FINS non appartengono alla stessa rete.

Esempio

Un Trajexia e un sistema OMRON CJ1 PLC con modulo Ethernet CJ1W-ETN11 sono collegati alla stessa rete. L'indirizzo IP del Trajexia è 192.168.0.5. L'indirizzo IP del modulo Ethernet PLC è 192.168.0.12. Quando viene eseguito il comando **FINS_COMMS(0,0,12,0,\$82,1000,20,0,500,5000,192,168,0,12)**, vengono lette 20 parole (**lunghezza=20**) dall'area di memoria PLC di DM (**area_remota=\$82**), a partire da DM1000 (**offset_remoto=1000**), e vengono scritte nella memoria VR di Trajexia in formato di numero intero (**area_locale=0**), a partire da VR(500) (**offset_locale=500**). Quindi, i valori nella memoria PLC compresi tra DM1000 e DM1019 vengono collocati nella memoria di Trajexia tra VR(500) e VR(519). Il timeout viene impostato a 5 secondi. Quando viene eseguito il comando **FINS_COMMS(1,0,12,0,\$80,50,10,0,300,3000,192,168,0,12)**, vengono scritte 10 parole (**lunghezza=10**) della memoria VR di Trajexia (**area_locale=0**), a partire da VR(300) (**offset_locale=300**), come numeri interi nell'area CIO della memoria PLC (**area_remota=\$80**), a partire da CIO50 (**offset_remoto=50**). Quindi, i valori nella memoria PLC compresi tra DM1000 e DM1019 vengono posizionati nella memoria PLC da CIO50 a CIO59. Il timeout viene impostato a 3 secondi.

Vedere anche N/D

3.2.122 FLAG

Tipo	Comando di sistema
Sintassi	FLAG(numero_flag [,valore])
Descrizione	Il comando FLAG viene utilizzato per impostare e leggere un insieme di 32 bit flag. Il comando FLAG può essere utilizzato con uno o due parametri. Quando è specificato un parametro, viene restituito lo stato del bit flag precisato. Quando sono specificati due parametri, il flag precisato viene impostato sul valore del secondo parametro. Il comando FLAG viene fornito per favorire la compatibilità con unità di controllo precedenti e non è consigliato per i nuovi programmi.
Argomenti	• numero_flag Il numero di flag è un valore compreso tra 0 e 31. • valore Se specificato, questo è lo stato su cui impostare il flag precisato, cioè "attivo" o "non attivo". Può anche essere scritto come 1 o 0.
Esempio	FLAG(27,ON) Imposta il bit flag 27 su "on".
Vedere anche	N/D

3.2.123 FLAGS

Tipo	Comando di sistema
Sintassi	FLAGS([valore])
Descrizione	Legge e imposta FLAGS come blocco. Il comando FLAGS viene fornito per favorire la compatibilità con unità di controllo precedenti e non è consigliato per i nuovi programmi. I flag a 32 bit possono essere letti con FLAGS e impostati con FLAGS(valore) .
Argomenti	• valore L'equivalente decimale dello schema dei bit su cui i flag devono essere impostati. Vedere tabella di seguito.
Esempio	FLAGS(146) ' 2 + 16 + 128 Imposta i flag 1,4 e 7 su "attivo", tutti gli altri su "non attivo".
Esempio	IF (FLAGS AND 8) <>0 THEN GOSUB qualche parte Verifica se il flag 3 è impostato.
Vedere anche	N/D

Numero di bit	Valore decimale
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

3.2.124 FOR..TO..STEP..NEXT

Tipo	Comando di controllo del programma.
Sintassi	FOR <i>variabile</i> = <i>inizio</i> TO <i>fine</i> [STEP <i>incremento</i>] comandi NEXT <i>variabile</i>
Descrizione	L'anello FOR ... NEXT consente la ripetizione per un dato numero di volte del segmento di programma compreso tra le istruzioni FOR e NEXT. L'ingresso in questo anello produce l'inizializzazione della variabile sul valore di "inizio"; quindi viene eseguito il blocco di comandi. Quando viene raggiunto il comando NEXT, la variabile subisce l'incremento specificato dopo STEP. Il valore STEP può essere sia positivo che negativo; se omesso, viene considerato pari a 1. Fintanto che la variabile è minore o uguale a "fine", il blocco di comandi viene eseguito ripetutamente fino a quando la variabile non diventa maggiore di "fine"; a quel punto, l'esecuzione del programma proseguirà dopo NEXT. Nota: le istruzioni FOR ... NEXT possono essere nidificate fino a un massimo di 8 livelli in un programma BASIC.
Argomenti	• variabile Espressione in BASIC. • inizio Espressione in BASIC. • fine Espressione in BASIC. • incremento Espressione in BASIC. • comandi Uno o più comandi BASIC.
Esempio	FOR numop = 8 TO 13 OP(numop,ON) NEXT numop Questo anello attiva le uscite dalla 8 alla 13.

Esempio

```
anello:
FOR dist = 5 TO -5 STEP -0.25
  MOVEABS(dist)
  GOSUB pick_up
NEXT dist
```

L'incremento **STEP** può essere sia positivo che negativo.

Esempio

```
anello1:
FOR I1 = 1 TO 8
anello2:
FOR I2 = 1 TO 6
  MOVEABS(I1*100,I2*100)
  GOSUB 1000
NEXT I2
NEXT I1
```

Le istruzioni **FOR...TO...STEP...NEXT** possono essere nidificate (fino a un massimo di 8 livelli), a condizione che i comandi interni **FOR** e **NEXT** siano entrambi inclusi nell'anello esterno **FOR...TO...STEP...NEXT**.

Vedere anche **REPEAT..UNTIL**, **WHILE..WEND**.

3.2.125 FORWARD

Tipo	Comando degli assi
Sintassi	FORWARD FO
Descrizione	Il comando FORWARD sposta in avanti un asse costantemente alla velocità impostata nel parametro degli assi SPEED . La velocità di accelerazione è definita dal parametro degli assi ACCEL . FORWARD agisce sull'asse di base predefinito (impostato con BASE) a meno che AXIS non sia usato per specificare un asse di base temporaneo. Nota: il movimento di avanzamento può essere arrestato eseguendo i comandi CANCEL o RAPIDSTOP oppure raggiungendo il limite di avanzamento.
Argomenti	N/D
Esempio	avvio: FORWARD WAIT UNTIL IN(0) = ON ' Aspetta il segnale di arresto CANCEL
Vedere anche	AXIS, CANCEL, RAPIDSTOP, REVERSE, UNITS.

3.2.126 FPGA_VERSION

Tipo	Parametro di slot
Sintassi	FPGA_VERSION SLOT(numero_unità)
Descrizione	Questo parametro restituisce la versione FPGA dell'unità attraverso il numero_unità del sistema di un'unità di controllo.
Argomenti	numero_unità I numeri delle unità vanno da -1 a 6 e includono lo 0; -1 rappresenta TJ1-MC__ e 0 la prima unità sulla destra rispetto a TJ1-MC__.
Esempio	N/D
Vedere anche	N/D

3.2.127 FRAC

Tipo	Funzione matematica
Sintassi	FRAC(espressione)
Descrizione	La funzione FRAC restituisce la parte frazionaria dell'espressione.
Argomenti	espressione Qualsiasi espressione valida in BASIC.
Esempio	>> PRINT FRAC(1.234) 0.2340
Vedere anche	N/D

3.2.128 FRAME

Tipo	Parametro di sistema
Sintassi	FRAME=valore
Descrizione	Viene utilizzato per specificare in quale frame occorre operare quando si utilizzano le trasformazioni dei frame. Le trasformazioni dei frame vengono utilizzate per consentire la specificazione dei movimenti che, in un ambito di riferimento coordinato a più assi, non presentano una corrispondenza univoca con i vari assi. Un esempio può essere il braccio robot SCARA con assi giuntati. Affinché l'estremità del braccio del robot possa eseguire movimenti in linea retta su un piano cartesiano, i motori devono muoversi seguendo un modello determinato dalla geometria dei robot. Le trasformazioni dei frame necessarie per eseguire funzioni di questo tipo, devono essere compilate a partire da una sorgente di linguaggio C e caricate nel software dell'unità di controllo del sistema. Contattare OMRON se si richiede un simile intervento. È possibile specificare il sistema della macchina con più frame diversi. Il FRAME correntemente attivo viene specificato con il parametro di sistema FRAME . Il FRAME predefinito è 0 che corrisponde a una trasformazione singola univoca.
Argomenti	N/D
Esempio	FRAME=1
Vedere anche	N/D

3.2.129 FREE

Tipo	Funzione di sistema
Sintassi	FREE
Descrizione	La funzione FREE restituisce la quantità rimanente di memoria disponibile per i programmi utente e gli elementi della matrice della memoria TABLE. Nota: ogni riga occupa un minimo di 4 caratteri (byte) in memoria, vale a dire la lunghezza della riga interessata, la lunghezza della riga precedente, il numero di spazi iniziali e un singolo token di comando. I comandi addizionali necessitano di un byte per token; la maggior parte degli altri dati vengono conservati come ASCII. TJ1-MC__ compila i programmi prima che vengano eseguiti; ciò significa che è necessaria una memoria doppia per potere eseguire un programma.
Argomenti	N/D
Esempio	>> PRINT FREE 47104.0000
Vedere anche	N/D

3.2.130 FS_LIMIT

Tipo	Parametro degli assi
Sintassi	FS_LIMIT FSLIMIT
Descrizione	Il parametro degli assi FS_LIMIT contiene la posizione assoluta del limite software di avanzamento espresso in unità utente. Un limite software per il movimento di avanzamento può essere impostato a partire dal programma che controlla la gamma di utilizzo del macchinario. Una volta raggiunto il limite, TJ1-MC__ decelera fino a raggiungere lo 0, quindi cancella il movimento. Il bit 9 del parametro degli assi AXISSTATUS viene attivato fintantoché la posizione dell'asse rimane superiore a FS_LIMIT .
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	AXIS, AXISSTATUS, UNITS.

3.2.131 FWD_IN

Tipo	Parametro degli assi
Sintassi	FWD_IN
Descrizione	Il parametro degli assi FWD_IN contiene il numero d'ingresso da utilizzare come ingresso del limite di avanzamento. L'intervallo d'ingresso valido è compreso tra 0 e 31. I valori da 0 a 15 rappresentano gli ingressi fisicamente presenti sul connettore di I/O di TJ1-MC__ e sono comuni a tutti gli assi. I valori da 16 a 27 rappresentano ingressi software che possono essere liberamente utilizzati da programmi e comandi quali IN e OP. Sono inoltre valori comuni a tutti gli assi. I valori da 28 a 31 vengono direttamente mappati agli ingressi di servoazionamento presenti sul connettore CN1 e sono univocamente associati a ogni asse. La mappatura degli ingressi del servoazionamento agli ingressi da 28 a 31 dipende dall'impostazione del parametro di servoazionamento Pn81E. L'impostazione consigliata è Pn81E = 0x4321, con la seguente mappatura: Se viene impostato un numero d'ingresso e il limite viene raggiunto, qualsiasi movimento di avanzamento su quell'asse verrà arrestato. Inoltre sarà impostato il bit 4 di AXISSTATUS . Nota: questo ingresso è active low.
Sigma II	- ingresso 28: CN1-40 - ingresso 29: CN1-41 - ingresso 30: CN1-42 - ingresso 31: CN1-43
Sigma III	- ingresso 28: CN1-13 - ingresso 29: CN1-7 - ingresso 30: CN1-8 - ingresso 31: CN1-9
Junma	- ingresso 26: CN1-2 - ingresso 27: CN1-1
	Per maggiori informazioni su come impostare il parametro di servoazionamento Pn81E, consultare il Manuale del servoazionamento. Come impostazione predefinita, il parametro è impostato su -1 e nessun ingresso è selezionato.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	AXIS, AXISSTATUS REV_IN.

3.2.132 FWD_JOG

Tipo	Parametro degli assi
Sintassi	FWD_JOG
Descrizione	Il parametro degli assi FWD_JOG contiene il numero d'ingresso da utilizzare come ingresso del jog di avanzamento L'ingresso può essere impostato tra 0 e 7. Come impostazione predefinita, il parametro è impostato su -1 e nessun ingresso è selezionato. Nota: questo ingresso è active low.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	AXIS, FAST_JOG, JOGSPEED, REV_JOG.

Esempio **GET#5, k**
Questa riga memorizza in **k** il carattere ASCII ricevuto sul canale della porta 5 di Trajexia Tools.

Vedere anche **INDEVICE INDEVICE, INPUT, KEY, LINPUT**

Numero del dispositivo d'ingresso	Descrizione
0	Porta di programmazione 0
1	Porta seriale 1 di RS-232C
2	Porta seriale 2 di RS-422A/485
5	Canale utente 5 della porta 0 di Trajexia Tools
6	Canale utente 6 della porta 0 di Trajexia Tools
7	Canale utente 7 della porta 0 di Trajexia Tools

3.2.133 GET

Tipo	Comando di I/O
Sintassi	GET [n.,] variabile
Descrizione	Il comando GET assegna a una variabile il codice ASCII di un carattere ricevuto. Se il buffer della porta seriale è vuoto, l'esecuzione del programma è sospesa fino a quando non viene ricevuto un carattere. I canali da 5 a 7 sono canali logici che vengono sovrapposti alla porta di programmazione 0 quando sono utilizzati i Trajexia Tools. Nota: il canale 0 è riservato al collegamento con Trajexia Tools e/o l'interfaccia della riga di comando. Tenere presente che questo canale potrebbe dare dei problemi relativamente a questa funzione.
Argomenti	n Il dispositivo d'ingresso specificato. Quando questo argomento è omesso, viene utilizzata la porta specificata da INDEVICE. Vedere tabella di seguito. variabile Il nome della variabile che riceve il codice ASCII.

3.2.134 GLOBAL

Tipo	Comando di sistema
Sintassi	GLOBAL "nome", numero_vr
Descrizione	Dichiara il nome come riferimento a una delle variabili globali VR. Il nome può quindi essere utilizzato sia all'interno del programma contenente la definizione GLOBAL, sia all'interno di tutti gli altri programmi presenti nel progetto Trajexia Tools 2. Nota: prima che il nome venga usato in altri programmi è necessario assicurarsi che sia eseguito il programma contenente la definizione GLOBAL. Inoltre, solo quel programma deve essere in esecuzione nel momento in cui viene eseguita la definizione GLOBAL: in caso contrario verrà visualizzato l'errore di programma e il programma sarà bloccato quando si tenterà di eseguire il comando. Per un avvio rapido, il programma dovrebbe altresì essere l'unico processo in esecuzione al momento dell'accensione. Una volta effettuata, la dichiarazione GLOBAL rimane attiva fino al primo ripristino di TJ1-MC__ tramite spegnimento e riaccensione oppure fino a quando non viene eseguito il comando EX. Nei programmi che utilizzano il comando GLOBAL definito, il nome ha lo stesso significato di VR(numero_vr). Non utilizzare la sintassi: VR(nome). È ammessa la dichiarazione fino a un massimo di 128 GLOBAL.
Argomenti	• nome Qualsiasi nome definito dall'utente contenente caratteri alfanumerici minuscoli o sottolineature. • numero_vr Il numero della memoria VR da associare a nome.
Esempio	GLOBAL "passo_vite",12 GLOBAL "rapporto1",534 rapporto1 = 3.56 passo_vite = 23.0 PRINT passo_vite, rapporto1
Vedere anche	N/D

3.2.135 GOSUB..RETURN

Tipo	Comando di controllo del programma.
Sintassi	GOSUB etichetta ... RETURN
Descrizione	La struttura GOSUB consente il passaggio a una subroutine. GOSUB memorizza la posizione della riga dopo il comando GOSUB e poi passa all'etichetta specificata. Quando viene raggiunta l'istruzione RETURN, l'esecuzione del programma torna alla posizione memorizzata. Nota: le subroutine di ogni task possono essere nidificate fino a un massimo di 8 livelli.
Argomenti	• etichetta Un'etichetta valida presente nel programma. Un'etichetta non valida provocherà un errore di compilazione prima dell'esecuzione. Le etichette possono essere stringhe di caratteri di qualsiasi lunghezza, anche se solo i primi 15 sono significativi.
Esempio	principale: GOSUB routine GOTO principale routine: PRINT "Posizione misurata=";MPOS;CHR(13); RETURN
Vedere anche	GOTO

3.2.136 GOTO

Tipo	Comando di controllo del programma.
Sintassi	GOTO etichetta
Descrizione	La struttura GOTO consente di effettuare un salto nell'esecuzione di un programma. GOTO fa passare l'esecuzione del programma direttamente alla riga contenente l'etichetta.

Argomenti • **etichetta**
Un'etichetta valida presente nel programma. Un'etichetta non valida provocherà un errore di compilazione prima dell'esecuzione. Le etichette possono essere stringhe di caratteri di qualsiasi lunghezza, anche se solo i primi 15 sono significativi.

Esempio **anello:**
PRINT "Posizione misurata = ";MPOS;CHR(13);
GOTO anello

Vedere anche **GOSUB..RETURN**

3.2.137 HALT

Tipo Comando di sistema

Sintassi **HALT**

Descrizione Il comando **HALT** arresta tutti i task di programma in corso di esecuzione. Il comando può essere usato sia sulla riga di comando che nei programmi. Il comando **STOP** può essere usato per arrestare un singolo task di programma.

Argomenti N/D

Esempio Nessun esempio.

Vedere anche **PROCESS, STOP.**

3.2.138 HEX

Tipo Comando di I/O

Sintassi **HEX**

Descrizione Questo comando viene utilizzato in un'istruzione di stampa per produrre un numero in formato esadecimale.

Argomenti N/D

Esempio **PRINT#5,HEX(IN(8,16))**

Vedere anche N/D

3.2.139 HLM_COMMAND

Tipo Comando di comunicazione

Sintassi **HLM_COMMAND(comando, porta [, nodo [, area/modalità_mc [, offset_mc]])**

Descrizione Il comando **HLM_COMMAND** esegue un'operazione specifica del comando Host Link su uno o su tutti gli slave Host Link presenti sulla porta selezionata. L'esecuzione del programma è sospesa fino a quando non viene ricevuta la stringa di risposta o non è trascorso il tempo di timeout. Il tempo di timeout viene specificato utilizzando il parametro **HLM_TIMEOUT**. Lo stato del trasferimento può essere controllato tramite il parametro **HLM_STATUS**.

Note:

- quando si utilizza **HLM_READ**, assicurarsi di impostare il protocollo del master Host Link attraverso il comando **SETCOM**.
- I comandi del master Host Link devono essere eseguiti da un solo task di programma per evitare problemi di temporizzazione con il multitasking.

Argomenti • **comando**
La selezione dell'operazione di Host Link che deve essere eseguita. Vedere la prima tabella di seguito.

• **porta**
La porta seriale specificata. 1 = porta seriale 1 di RS-232C; 2 = porta seriale 2 di RS-422A.

• **nodo** (per **HLM_MREAD**, **HLM_TEST**, **HLM_ABORT** e **HLM_STWR**)
Il numero di nodo slave a cui inviare il comando Host Link. Intervallo: [0, 31].

• **modalità** (per **HLM_STWR**)
La modalità operativa specificata per la CPU. 0 = modalità PROGRAM; 2 = modalità MONITOR; 3 = modalità RUN.

• **area_mc** (per **HLM_MREAD**)
Selezione della memoria di TJ1-MC__ da cui leggere i dati da inviare. Vedere la seconda tabella di seguito.

• **offset_mc** (per **HLM_MREAD**)
L'indirizzo dell'area di memoria di TJ1-MC__ specificata da cui leggere. Intervallo per le variabili della memoria VR: [0, 1023]. Intervallo per le variabili della memoria TABLE: [0, 63999].

Esempio **HLM_COMMAND(HLM_MREAD,1,12,MC_VR,233)**
 Questo comando legge il codice del modello della CPU dello slave Host Link con indirizzo nodo 12 collegato alla porta RS-232C. Il risultato viene scritto in VR(233).
 Se lo slave collegato è un C200HX PC, la VR(233) conterrà il valore 12 (esadecimale) dopo un'esecuzione riuscita.

Esempio **HLM_COMMAND(HLM_TEST,2,23)**
PRINT HLM_STATUS PORT(2)
 Questo comando verificherà la comunicazione di Host Link con lo slave Host Link (nodo 23) collegato alla porta RS-422A.
 Se il parametro **HLM_STATUS** contiene il valore 0, la comunicazione è funzionale.

Esempio **HLM_COMMAND(HLM_INIT,2)**
HLM_COMMAND(HLM_ABORT,2,4)
 Questi due comandi eseguono le operazioni **INITIALIZE** e **ABORT** di Host Link sulla porta 2 dell'RS-422A. Il numero di nodo dello slave è 4.

Esempio **HLM_COMMAND(HLM_STWR,2,0,2)**
 Quando è necessario scrivere dati su un PC utilizzando Host Link, la CPU non può essere in modalità RUN. È possibile impostarla su modalità MONITOR utilizzando il comando **HLM_COMMAND**. L'indirizzo nodo dello slave è 0 e lo slave è collegato alla porta RS-232C.

Vedere anche **HLM_READ**, **HLM_COMMAND**, **HLM_STATUS**, **HLM_TIMEOUT**, **HLS_NODE**, **HLM_WRITE**, **SETCOM**.

valore di comando	Descrizione
HLM_MREAD (o valore 0)	Esegue il comando per PC MODEL READ (MM) di Host Link che legge il codice del modello della CPU. Il risultato viene scritto nella variabile TJ1-MC__ specificata da area_mc e offset_mc .
HLM_TEST (o valore 1)	Esegue il comando TEST (TS) di Host Link per verificare che la comunicazione sia corretta, inviando la stringa "MCW151 TEST STRING" e controllando la stringa ripetuta. Per conoscere il risultato, verificare il parametro HLM_STATUS .
HLM_ABORT (o valore 2)	Esegue il comando ABORT (XZ) di Host Link per interrompere il comando di Host Link in corso di esecuzione. Il comando ABORT non riceve una risposta.

valore di comando	Descrizione
HLM_INIT (o valore 3)	Esegue il comando INITIALIZE (**) di Host Link per inizializzare la procedura di controllo della trasmissione di tutte le unità slave.
HLM_STWR (o valore 4)	Esegue il comando STATUS WRITE (SC) di Host Link per cambiare la modalità operativa della CPU.

valore area_mc	Area dati
MC_TABLE (o valore 8)	Matrice di variabili della memoria TABLE
MC_VR (o valore 9)	Matrice di variabili globale (VR)

3.2.140 HLM_READ

Tipo Comando di comunicazione

Sintassi **HLM_READ(porta, nodo, area_pc, offset_pc, lunghezza, area_mc, offset_mc)**

Descrizione Il comando **HLM_READ** legge i dati da uno slave Host Link inviando alla porta seriale una stringa di comando Host Link contenente il nodo dello slave specificato. I dati ricevuti in risposta vengono scritti in variabili della memoria VR o della memoria TABLE. Ogni parola contenuta nei dati viene trasferita su una sola variabile. La lunghezza massima dei dati è di 30 parole (trasferimento di frame singolo).
 L'esecuzione del programma è sospesa fino a quando non viene ricevuta la stringa di risposta o non è trascorso il tempo di timeout. Il tempo di timeout viene specificato utilizzando il parametro **HLM_TIMEOUT**. Lo stato del trasferimento può essere controllato tramite il parametro **HLM_STATUS**.
Note:

- quando si utilizza **HLM_READ**, assicurarsi di impostare il protocollo del master Host Link attraverso il comando **SETCOM**.
- I comandi del master Host Link devono essere eseguiti da un solo task di programma per evitare problemi di temporizzazione con il multitasking.

- Argomenti
- **porta**
La porta seriale specificata. 1 = porta seriale 1 di RS-232C; 2 = porta seriale 2 di RS-422A.
 - **nodo**
Il numero di nodo slave a cui inviare il comando Host Link. Intervallo: [0, 31].
 - **area_pc**
Selezione della memoria PC per il comando Host link. Vedere la prima tabella di seguito.
 - **offset_pc**
L'indirizzo dell'area di memoria PC specificata da cui leggere. Intervallo: [0, 9999].
 - **lunghezza**
Il numero di parole contenute nei dati che devono essere trasferite. Intervallo: [1, 30].
 - **area_mc**
Selezione della memoria di TJ1-MC__ da cui leggere i dati da inviare. Vedere la seconda tabella di seguito.
 - **offset_mc**
L'indirizzo dell'area di memoria di TJ1-MC__ specificata su cui scrivere. Intervallo per le variabili della memoria VR: [0, 1023]. Intervallo per le variabili della memoria TABLE: [0, 63999].

Esempio **HLM_READ(2,17,PLC_DM,120,20,MC_TABLE,4000)**
Questo esempio mostra come leggere 20 parole dagli indirizzi 120 – 139 dell'area DM del PC trasferendoli agli indirizzi 4000 – 4019 della memoria TABLE di TJ1-MC__. Il PC dispone dell'indirizzo nodo slave 17 ed è collegato alla porta RS-422A.

Vedere anche **HLM_COMMAND, HLM_STATUS, HLM_TIMEOUT, HLS_NODE, HLM_WRITE, SETCOM.**

valore area_pc	Area dati	Comando Host Link
PLC_DM (o valore 0)	Area DM (memoria dati)	RD
PLC_IR (o valore 1)	Area CIO/IR	RR

valore area_pc	Area dati	Comando Host Link
PLC_LR (o valore 2)	Area LR (data link)	RL
PLC_HR (o valore 3)	Area HR (ritenzione)	RH
PLC_AR (o valore 4)	Area AR (ausiliaria)	RJ
PLC_EM (o valore 6)	Area EM (memoria estesa)	RE

valore area_mc	Area dati
MC_TABLE (o valore 8)	Matrice di variabili della memoria TABLE
MC_VR (o valore 9)	Matrice di variabili globale (VR)

3.2.141 HLM_STATUS

Tipo Parametro di comunicazione

Sintassi **HLM_STATUS PORT(n)**

Descrizione Il parametro **HLM_STATUS** contiene lo stato dell'ultimo comando master Host Link inviato alla porta specificata. Il parametro indica lo stato dei comandi **HLM_READ**, **HLM_WRITE** e **HLM_COMMAND**. I bit di stato sono definiti nella tabella di seguito.

Il parametro **HLM_STATUS** avrà valore 0 se non si sono verificati problemi. Nel caso in cui il valore del parametro fosse diverso da 0, occorre programmare l'azione appropriata nel programma utente in BASIC, come ad esempio un nuovo tentativo o un arresto di emergenza.

Ogni porta dispone di un parametro **HLM_STATUS**. Il modificatore **PORT** è richiesto per specificare la porta.

Argomenti

- **n**
La porta seriale specificata. 1 = porta seriale 1 di RS-232C; 2 = porta seriale 2 di RS-422A

Esempio >> **HLM_WRITE(1,28,PLC_EM,50,25,MC_VR,200)**
 >> **PRINT HEX(HLM_STATUS PORT(1))**
1
 Apparentemente la CPU si trova in modalità RUN e non accetta l'operazione di scrittura.

Esempio >> **HLM_COMMAND(HLM_TEST,2,0)**
 >> **PRINT HLM_STATUS PORT(2)**
256.0000
 Si è verificato un errore di timeout.

Vedere anche **HLM_READ, HLM_COMMAND, HLM_TIMEOUT, HLS_NODE, HLM_WRITE, SETCOM.**

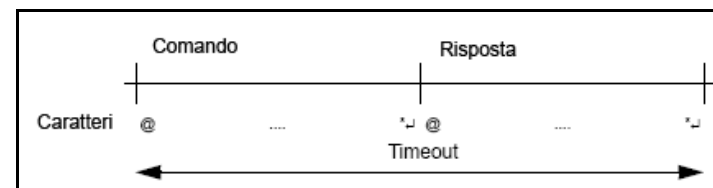
Bit	Nome	Descrizione
0 – 7	Codice di fine	Il codice di fine può essere un codice di fine definito dallo slave di Host Link (problemi nella stringa di comando inviata) oppure un codice di fine definito a causa di un problema riscontrato dal master Host Link (problemi sulla stringa di risposta ricevuta).
8	Errore di timeout	Un errore di timeout si verifica se non viene ricevuta alcuna risposta entro il tempo di timeout. Questo significa che si è interrotta la comunicazione.
9	Comando non riconosciuto	Questo stato indica che lo slave non ha riconosciuto il comando e ha restituito una risposta IC.

3.2.142 HLM_TIMEOUT

Tipo Parametro di comunicazione

Sintassi **HLM_TIMEOUT**

Descrizione Il parametro **HLM_TIMEOUT** specifica il tempo di timeout fissato per il protocollo del master Host Link per entrambe le porte seriali. Un errore di timeout si verifica quando il tempo necessario per inviare il comando e ricevere la risposta dallo slave supera quello specificato con questo parametro.
 Il parametro si applica ai comandi **HLM_READ, HLM_WRITE** e **HLM_COMMAND**. Il parametro **HLM_TIMEOUT** viene specificato nei tempi di servozionamento.



Argomenti N/D

Esempio >> **HLM_TIMEOUT=2000**

Considerare che il ciclo di servozionamento di TJ1-MC__ è impostato su 500 ms (SERVO_PERIOD=500). Per entrambe le porte seriali il tempo di timeout del master Host Link è stato impostato su 1 s.

Vedere anche **HLM_READ, HLM_COMMAND, HLM_STATUS, HLS_NODE, HLM_WRITE, SETCOM SERVO_PERIOD.**

3.2.143 HLM_WRITE

Tipo Comando di comunicazione

Sintassi **HLM_WRITE(porta, nodo, area_pc, offset_pc, lunghezza, area_mc, offset_mc)**

Descrizione Il comando **HLM_WRITE** scrive i dati di TJ1-MC__ sullo slave Host Link inviando alla porta seriale una stringa di comando Host Link contenente il nodo dello slave specificato. I dati ricevuti in risposta vengono scritti prendendoli dalle variabili della memoria VR o della memoria TABLE. Ogni variabile definisce la parola o i dati che saranno trasferiti. La lunghezza massima dei dati è di 29 parole (trasferimento di frame singolo). L'esecuzione del programma è sospesa fino a quando non viene ricevuta la stringa di risposta o non è trascorso il tempo di timeout. Il tempo di timeout viene specificato utilizzando il parametro **HLM_TIMEOUT**. Lo stato del trasferimento può essere controllato tramite il parametro **HLM_STATUS**.
Note:

- quando si utilizza **HLM_WRITE**, assicurarsi di impostare il protocollo del master Host Link attraverso il comando **SETCOM**.
- I comandi del master Host Link devono essere eseguiti da un solo task di programma per evitare problemi di temporizzazione con il multitasking.

Argomenti

- **porta**
La porta seriale specificata. 1 = porta seriale 1 di RS-232C; 2 = porta seriale 2 di RS-422A
- **nodo**
Il numero di nodo slave a cui inviare il comando Host Link. Intervallo: [0, 31].
- **area_pc**
Selezione della memoria PC per il comando Host link. Vedere la prima tabella di seguito.
- **offset_pc**
L'indirizzo dell'area di memoria PC specificata su cui scrivere. Intervallo: [0, 9999].
- **lunghezza**
Il numero di parole contenute nei dati che devono essere trasferite. Intervallo: [1, 29].
- **area_mc**
Selezione della memoria di TJ1-MC__ da cui leggere i dati da inviare. Vedere la seconda tabella di seguito
- **offset_mc**
L'indirizzo dell'area di memoria di TJ1-MC__ specificata da cui leggere. Intervallo per le variabili della memoria VR: [0, 1023]. Intervallo per le variabili della memoria TABLE: [0, 63999].

Esempio

HLM_WRITE(1,28,PLC_EM,50,25,MC_VR,200)

Questo esempio mostra come scrivere 25 parole trasferendole dagli indirizzi 200 – 224 della memoria VR di TJ1-MC__ agli indirizzi 50 – 74 dell'area EM del PC. Il PC dispone dell'indirizzo nodo slave 28 ed è collegato alla porta RS-232C.

Vedere anche **HLM_READ, HLM_COMMAND, HLM_STATUS, HLM_TIMEOUT, HLS_NODE, SETCOM.**

valore area_pc	Area dati	Comando Host Link
PLC_DM (o valore 0)	Area DM (memoria dati)	RD
PLC_IR (o valore 1)	Area CIO/IR	RR
PLC_LR (o valore 2)	Area LR (data link)	RL

valore area_pc	Area dati	Comando Host Link
PLC_HR (o valore 3)	Area HR (ritenzione)	RH
PLC_AR (o valore 4)	Area AR (ausiliaria)	RJ
PLC_EM (o valore 6)	Area EM (memoria estesa)	RE

valore area_mc	Area dati
MC_TABLE (o valore 8)	Matrice di variabili della memoria TABLE
MC_VR (o valore 9)	Matrice di variabili globale (VR)

3.2.144 HLS_NODE

Tipo	Parametro di comunicazione
Sintassi	HLS_NODE
Descrizione	Il parametro HLS_NODE definisce il numero di modulo dello slave per il protocollo slave Host Link. TJ1-MC__ risponde solamente alle stringhe di comando di master Host Link con il numero di modulo specificato in questo parametro. L'intervallo valido per questo parametro è [0, 31]. Il valore predefinito è 0.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	N/D

3.2.145 (HW_PSWITCH)

Tipo	Comando degli assi
Sintassi	HW_PSWITCH(modalità, direzione, statousc, inizio_tabella, fine_tabella)

Descrizione Il comando **HW_PSWITCH** attiva l'uscita OUT 0 per l'asse quando viene raggiunta la posizione misurata dell'asse predefinito e disattiva l'uscita quando viene raggiunta un'altra posizione misurata. Le posizioni sono definite come sequenza della memoria TABLE in un intervallo compreso tra **inizio_tabella** e **fine_tabella** e quando viene eseguito il comando **HW_PSWITCH** vengono memorizzate nella coda FIFO. Questo comando può essere applicato solo agli assi definiti come assi flessibili con valori **ATYPE** 43, 44 e 45. Il comando può essere utilizzato con 1 e fino a tutti e 5 i parametri. Per disabilitare l'interruttore o cancellare la coda FIFO, è sufficiente un solo parametro. Tutti e cinque i parametri sono necessari per attivare l'interruttore. Dopo aver caricato la FIFO e aver attuato la sequenza di posizioni presenti nella FIFO stessa, qualora fosse necessario eseguire nuovamente la stessa sequenza la FIFO deve essere cancellata prima di eseguire il comando **HW_PSWITCH** con gli stessi parametri.

Argomenti	• modalità 0 = disabilitazione interruttore; 1 = accensione e caricamento FIFO; 2 = cancellazione FIFO. • direzione 0 = decrescente; 1 = crescente. • statousc Lo stato dell'uscita da impostare, su "attiva" o "non attiva", nella prima posizione della FIFO. • inizio_tabella Indirizzo della memoria TABLE dove ha inizio la sequenza. • fine_tabella Indirizzo della memoria TABLE dove finisce la sequenza.

Esempio **HW_PSWITCH(1, 1, ON, 21, 50)**
 Questo comando carica la FIFO con 30 posizioni, registrate nella memoria TABLE a partire da **TABLE(21)** in ordine crescente. Quando la posizione memorizzata in **TABLE(21)** viene raggiunta, l'uscita OUT 0 viene impostata su "attiva" e poi, a mano a mano che vengono raggiunte le posizioni successive della sequenza, alternativamente su "non attiva" e "attiva" fino a quando non si arriva alla posizione memorizzata in **TABLE(50)**.

Esempio **HW_PSWITCH(0)**
 Questo comando disattiva l'interruttore se era stato attivato in precedenza, ma non cancella la coda FIFO.

Esempio **HW_PSWITCH(2)**
 Questo comando cancella la coda FIFO, se è stata caricata in precedenza.

Vedere anche **AXIS**

3.2.146 I_GAIN

Tipo Parametro degli assi
Sintassi **I_GAIN**
Descrizione Il parametro **I_GAIN** contiene il guadagno integrale per l'asse. Il contributo integrale d'uscita viene calcolato moltiplicando la somma degli errori seguenti per il valore del parametro **I_GAIN**. Il valore predefinito è 0. L'aggiunta di un guadagno integrale a un sistema di servozionamento ne riduce l'errore di posizionamento quando è fermo o quando si muove con regolarità. Ciò può produrre o aumentare la sovraelongazione e l'oscillazione ed è pertanto adatto solo per quei sistemi che funzionano a velocità costante e con accelerazioni lente. Nota: per evitare qualsiasi instabilità i guadagni dei servozionamenti vanno cambiati solamente quando **SERVO** non è attivo.

Argomenti N/D

Esempio Nessun esempio.

Vedere anche **D_GAIN, I_GAIN, OV_GAIN, P_GAIN, VFF_GAIN**.

3.2.147 IDLE

Vedere **WAIT IDLE**.

3.2.148 IEEE_IN

Tipo Funzione matematica

Sintassi **IEEE_IN(byte0,byte1,byte2,byte3)**

Descrizione La funzione **IEEE_IN** restituisce il numero a virgola mobile rappresentato da 4 byte che solitamente sono stati ricevuti tramite un collegamento di comunicazione.

Argomenti

- **byte0** – byte3
 Qualsiasi combinazione di valori a 8 bit rappresentante un numero a virgola mobile IEEE valido.

Esempio **VR(20) = IEEE_IN(b0,b1,b2,b3)**

Vedere anche N/D

3.2.149 IEEE_OUT

Tipo Funzione matematica

Sintassi **byte_n = IEEE_OUT(valore, n)**

Descrizione La funzione **IEEE_OUT** restituisce un singolo byte in formato IEEE estratto dal valore a virgola mobile per trasmetterlo tramite un sistema bus. Normalmente la funzione viene chiamata 4 volte, per estrarre ogni volta un byte. Nota: il byte 0 è il byte alto del formato a virgola mobile IEEE a 32 bit.

Argomenti

- **valore**
 Qualsiasi parametro o variabile a virgola mobile in BASIC.
- **n**
 Il numero di byte (0 – 3) da estrarre.

Esempio
V=MPOS AXIS(2)
byte0 = IEEE_OUT(V, 0)
byte1 = IEEE_OUT(V, 1)
byte2 = IEEE_OUT(V, 2)
byte3 = IEEE_OUT(V, 3)

Vedere anche N/D

3.2.150 IF..THEN..ELSE..ENDIF

Tipo	Comando di controllo del programma.
Sintassi	IF condizione_1 THEN comandi { ELSEIF condizione_i THEN comandi} [ELSE comandi] ENDIF IF condizione_1 THEN comandi
Descrizione	Questa struttura controlla il flusso del programma in base ai risultati della condizione. Se la condizione è TRUE i comandi successivi a THEN e fino a ELSEIF , ELSE o ENDIF vengono eseguiti. Se la condizione è FALSE e il comando della sottostruttura successiva ELSEIF è TRUE , i comandi di tale sottostruttura vengono eseguiti. Se tutte le condizioni sono FALSE , i comandi successivi a ELSE vengono eseguiti oppure il programma riprende l'esecuzione alla riga successiva a ENDIF qualora non sia incluso un ELSE . ENDIF viene utilizzato per segnare la fine del blocco condizionale. Nota: le sequenze IF...THEN...ELSE...ENDIF possono essere nidificate senza limiti. Per una struttura IF...THEN a più righe, nessuna istruzione deve seguire THEN . Una struttura a una sola riga non deve utilizzare ENDIF .
Argomenti	condizione_i Un'espressione logica. comandi Uno o più comandi BASIC.
Esempio	IF MPOS > (0.22 * VR(0)) THEN GOTO lunghezza_eccedente
Esempio	IF IN(0) = ON THEN conteggio = conteggio + 1 PRINT "COUNTS = ";conteggio esito_negativo = 0 ELSE esito_negativo = esito_negativo + 1 ENDIF
Esempio	IF IN(arresto)=ON THEN OP(8,ON) VR(flag_ciclo)=0 ELSEIF IN(avvio_ciclo)=ON THEN VR(flag_ciclo)=1 ELSEIF IN(fase1)=ON THEN VR(flag_ciclo)=99 ENDIF

Esempio

```

IF car_tasto=$31 THEN
  GOSUB car_1
ELSEIF car_tasto=$32 THEN
  GOSUB car_2
ELSEIF car_tasto=$33 THEN
  GOSUB car_3
ELSE
  PRINT "Carattere sconosciuto"
ENDIF

```

Vedere anche N/D

3.2.151 IN

Tipo	Funzione di I/O
Sintassi	IN(numero_ingresso [,numero_ingresso_finale]) IN
Descrizione	La funzione IN restituisce il valore degli ingressi digitali. IN(numero_ingresso, numero_ingresso_finale) restituisce la somma binaria del gruppo di ingressi. La differenza tra i due argomenti deve essere minore di 24. IN(numero_ingresso) con un valore di numero_ingresso minore di 32 restituisce il valore del canale particolare. IN (senza argomenti) restituisce la somma binaria dei primi 24 ingressi (ad esempio IN(0,23)).
Argomenti	numero_ingresso: Il numero dell'ingresso per il quale deve essere restituito un valore. Valore: un numero intero compreso tra 0 e 31. numero_ingresso_finale Il numero dell'ultimo ingresso per il quale deve essere restituito un valore. Valore: un numero intero compreso tra 0 e 31.

Esempio Le righe di seguito possono essere utilizzate per spostare, in base a un determinato fattore, la posizione impostata su un selettore rotativo. Il selettore rotativo è collegato agli ingressi 4, 5, 6 e 7 e fornisce l'uscita espressa in BCD.

anellomovimento:

MOVEABS(IN(4,7)*1.5467)

WAIT IDLE

GOTO anellomovimento

Il comando **MOVEABS** viene costruito come segue:

Fase 1: **IN(4,7)** ottiene un numero compreso tra 0 e 15.

Fase 2: il numero viene moltiplicato per 1,5467 così da ottenere la distanza richiesta.

Fase 3: viene eseguito un movimento assoluto fino a questa posizione.

Esempio In questo esempio viene testato un singolo ingresso:

test:

WAIT UNTIL IN(4)=ON ' Il nastro trasportatore è in posizione quando è ON
GOSUB posizione

Vedere anche **OP**.

3.2.152 INDEVICE

Tipo Parametro di I/O

Sintassi **INDEVICE**

Descrizione Il parametro **INDEVICE** definisce il dispositivo di ingresso predefinito. Il dispositivo viene selezionato per i comandi d'ingresso quando l'opzione **#n** è omessa. Il parametro **INDEVICE** è specifico di un task. I valori supportati sono elencati nella tabella di seguito.

Argomenti N/D

Esempio Nessun esempio.

Vedere anche **GETGET, INPUT, LINPUT, KEY**.

Valore	Descrizione
0	Porta di programmazione 0 (predefinita)
1	Porta seriale 1 di RS-232C
2	Porta seriale 2 di RS-422A/485
5	Canale utente 5 della porta 0 di Trajexia Tools
6	Canale utente 6 della porta 0 di Trajexia Tools
7	Canale utente 7 della porta 0 di Trajexia Tools

3.2.153 INITIALISE

Tipo Comando di sistema

Sintassi **INITIALISE**

Descrizione Imposta tutti i parametri degli assi, di sistema e di processo sui rispettivi valori predefiniti. I parametri sono inoltre reimpostati ogni volta che l'unità di controllo viene accesa o quando viene eseguito un comando **EX** (reset software). In Trajexia Tools, il menu **Reset the controller...** sotto il menu **Controller** esegue un'operazione equivalente a quella di un comando **EX**.

Argomenti N/D

Esempio Nessun esempio.

Vedere anche • **EX**

3.2.154 INPUT

Tipo	Comando di I/O
Sintassi	INPUT [#n], variabile { , variabile }
Descrizione	Il comando INPUT assegna alle variabili specificate valori numerici di stringa d'ingresso. È possibile richiedere più valori di stringa d'ingresso su una riga separati da virgole oppure su più righe separati da un ritorno a capo. L'esecuzione del programma è messa in pausa fino a quando la stringa non viene terminata tramite ritorno a capo dopo l'assegnazione dell'ultima variabile. Se la stringa non è valida, l'utente viene informato da un messaggio di errore e il task è ripetuto. Non vi è alcun limite alla quantità massima di ingressi su una riga, tranne la lunghezza della riga stessa. I canali da 5 a 7 sono canali logici che vengono sovrapposti alla porta di programmazione 0 di RS-232C quando sono utilizzati i Trajexia Tools. Nota: il canale 0 è riservato al collegamento con Trajexia Tools e/o l'interfaccia della riga di comando. Tenere presente che questo canale potrebbe dare dei problemi relativamente a questa funzione.
Argomenti	• n Il dispositivo d'ingresso specificato. Quando questo argomento è omesso, viene utilizzata la porta specificata da INDEVICE. • variabile La variabile su cui scrivere.
Esempio	Considerare il seguente programma per ricevere dati dal terminale. <pre>INPUT#5, num PRINT#5, "BATCH COUNT=";num[0]</pre> Una possibile risposta sul terminale potrebbe essere: <pre>123<CR> BATCH COUNT=123</pre>
Vedere anche	INDEVICE, GET, LINPUT, KEY

3.2.155 INT

Tipo	Funzione matematica
Sintassi	INT(espressione)
Descrizione	La funzione INT restituisce il numero intero dell'espressione. Nota: per arrotondare un numero positivo al valore intero più vicino, prendere la funzione INT del valore incrementato di 0,5. In modo analogo, per arrotondare un valore negativo sottrarre 0,5 prima di applicare INT.
Argomenti	• espressione Qualsiasi espressione valida in BASIC.
Esempio	<pre>>> PRINT INT(1.79) 1.0000</pre>
Vedere anche	N/D

3.2.156 INVERT_IN

Tipo	Comando di sistema
Sintassi	INVERT_IN(ingresso,on/off)
Descrizione	Il comando INVERT_IN consente ai canali di ingresso 0..31 di essere invertiti separatamente nel software. Si tratta di una funzionalità importante poiché consente a questi canali di ingresso di essere assegnati all'attivazione di funzioni come la sospensione dell'alimentazione. La funzione INVERT_IN attiva ovvero disattiva l'inversione di un canale. Può essere applicata solo agli ingressi 0..31.
Argomenti	• ingresso Espressione in BASIC.
Esempio	<pre>>>? IN(3) 0.0000 >>INVERT_IN(3,ON) >>? IN(3) 1.0000</pre>
Vedere anche	N/D

3.2.157 INVERT_STEP

Tipo	Parametro degli assi
Sintassi	INVERT_STEP
Descrizione	INVERT_STEP viene utilizzato per collegare un inverter hardware al circuito di uscita impulsi del motore passo-passo. Questo può rivelarsi necessario nel collegamento di alcuni driver del motore passo-passo. La logica elettronica contenuta dal generatore Trajexia di impulsi per il motore passo-passo presume che il fronte FALLING (di discesa) dell'uscita del motore passo-passo sia il fronte attivo che produce il movimento del motore. Tale funzionamento è appropriato per la maggior parte dei driver del motore passo-passo. L'impostazione INVERT_STEP=ON di fatto trasforma il fronte RISING (di salita) del segnale del motore passo-passo nel fronte attivo. Laddove necessario, INVERT_STEP deve essere impostato prima di abilitare l'unità di controllo con WDOG=ON. L'impostazione predefinita non è attiva. Nota: se l'impostazione è errata, un motore passo-passo può rimanere sfasato di un passo al cambiamento di direzione.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	N/D

3.2.158 INVERTER_COMMAND

Tipo	Comando di sistema
Sintassi	INVERTER_COMMAND(modulo, stazione, 7, segnali_operazione) INVERTER_COMMAND(modulo, stazione, 1, numero_allarme)
Descrizione	INVERTER_COMMAND controlla gli ingressi e cancella l'allarme dell'inverter di frequenza collegato al sistema attraverso il bus MECHATROLINK-II. Esistono due funzioni INVERTER_COMMAND: 1: cancella un allarme. 7: controlla i segnali di funzionamento. Per utilizzare un inverter tramite MECHATROLINK-II, occorre far passare il comando e il riferimento attraverso l'opzione di comunicazione: - Inverter MV/V7: N3=3; N4=9 - Inverter F7/G7: B1-01=3; B1-02=3. Assicurarsi che il firmware dell'inverter supporti la scheda MECHATROLINK-II. Il comando restituisce -1 se viene eseguito con successo oppure 0 se non riesce. Il comando inviato all'inverter corrisponde ai bit elencati nella tabella di seguito.
Argomenti	modulo Numero di TJ1-ML__ a cui l'inverter è collegato. stazione Il numero di stazione MECHATROLINK-II dell'inverter. numero_allarme Il numero dell'allarme. Vedere il manuale dell'inverter. segnali_funzionamento Un valore bitwise per controllare i segnali di funzionamento. Vedere tabella di seguito.
Esempio	Nessun esempio.
Vedere anche	N/D

Bit	Valore	Comando	Descrizione
0	Esadecimale	1	Marcia avanti
1	Esadecimale	2	Marcia indietro
2	Esadecimale	4	Ingresso 3 multifunzione dell'inverter
3	Esadecimale	8	Ingresso 4 multifunzione dell'inverter
4	Esadecimale	10	Ingresso 5 multifunzione dell'inverter
5	Esadecimale	20	Ingresso 6 multifunzione dell'inverter
6	Esadecimale	40	Ingresso 7 multifunzione dell'inverter
7	Esadecimale	80	Ingresso 8 multifunzione dell'inverter (solo G7)
8	Esadecimale	100	Errore esterno
9	Esadecimale	200	Ripristino dopo errore
10	Esadecimale	400	Ingresso 9 multifunzione dell'inverter (solo G7)
11	Esadecimale	800	Ingresso 10 multifunzione dell'inverter (solo G7)
12	Esadecimale	1000	Ingresso 11 multifunzione dell'inverter (solo G7)
13	Esadecimale	2000	Ingresso 12 multifunzione dell'inverter (solo G7)
14	Esadecimale	4000	Cancellazione dei dati dello storico degli errori
15	Esadecimale	8000	Comando di BB esterno

3.2.159 INVERTER_READ

Tipo Comando di sistema

Sintassi **INVERTER_READ(modulo, stazione, 0, numero_parametro, dimensioni_parametro, VR)**
INVERTER_READ(modulo, stazione, 1, numero_allarme, VR)
INVERTER_READ(modulo, stazione, 2, VR)
INVERTER_READ(modulo, stazione, 3, VR)
INVERTER_READ(modulo, stazione, 4, da, lunghezza, VR)

Descrizione **INVERTER_READ** legge il parametro, la velocità di riferimento, la coppia di riferimento o l'allarme dall'inverter di frequenza collegato al sistema attraverso il bus MECHATROLINK-II.

Esistono cinque funzioni **INVERTER_READ**:

- 0: legge un parametro dell'inverter.
- 1: legge l'allarme dell'inverter.
- 2: legge la velocità di riferimento.
- 3: legge la coppia di riferimento.
- 4: legge gli ingressi dell'inverter.

Per utilizzare un inverter tramite MECHATROLINK-II, occorre far passare il comando e il riferimento attraverso l'opzione di comunicazione:

- Inverter MV/V7: N3=3; N4=9
- Inverter F7/G7: B1-01=3; B1-02=3.

Assicurarsi che il firmware dell'inverter supporti la scheda MECHATROLINK-II.

Il comando restituisce 1 se viene eseguito con successo oppure 0 se non riesce. L'eventuale risultato viene restituito nella VR selezionata.

- Argomenti
- **modulo**
Numero di TJ1-ML__ a cui l'inverter è collegato.
 - **stazione**
Il numero di stazione MECHATROLINK-II dell'inverter.
 - **numero_parametro**
Il numero di parametro da leggere. Vedere il manuale dell'inverter.
 - **dimensioni_parametro**
Le dimensioni del parametro da leggere: 2 o 4 byte. Vedere il manuale dell'inverter.
 - **VR**
L'indirizzo della memoria VR di TJ-MC__ dove vengono registrate le informazioni lette. Quando la funzione è **4**, il risultato viene restituito come valore bitwise. Vedere tabella di seguito.
 - **numero_allarme**
Il numero dell'allarme da leggere. Vedere il manuale dell'inverter.
 - **da**
L'indirizzo di partenza dell'ingresso da leggere.
 - **lunghezza**
La lunghezza dell'ingresso da leggere.

Esempio Nessun esempio.

Vedere anche N/D

Bit	Valore	Comando	Descrizione
0	Esadecimale	1	Marcia avanti
1	Esadecimale	2	Marcia indietro
2	Esadecimale	4	Ingresso 3 multifunzione dell'inverter
3	Esadecimale	8	Ingresso 4 multifunzione dell'inverter
4	Esadecimale	10	Ingresso 5 multifunzione dell'inverter
5	Esadecimale	20	Ingresso 6 multifunzione dell'inverter
6	Esadecimale	40	Ingresso 7 multifunzione dell'inverter
8	Esadecimale	100	Errore esterno
9	Esadecimale	200	Ripristino dopo errore
14	Esadecimale	4000	Cancellazione dei dati dello storico degli errori
15	Esadecimale	8000	Comando di BB esterno

3.2.160 INVERTER_WRITE

Tipo Comando di sistema

Sintassi **INVERTER_WRITE(modulo, stazione,0, numero_parametro, dimensioni_parametro, VR, modalità)**
INVERTER_WRITE(modulo, stazione, 2, valore)
INVERTER_WRITE(modulo, stazione, 3, valore)

Descrizione **INVERTER_WRITE** scrive il parametro, la velocità di riferimento o la coppia di riferimento ottenendoli dall'inverter di frequenza collegato al sistema attraverso il bus MECHATROLINK-II.

Esistono tre funzioni **INVERTER_WRITE**:

- 0: scrive un parametro dell'inverter.
- 2: scrive la velocità di riferimento.
- 3: scrive la coppia di riferimento.

Per utilizzare un inverter tramite MECHATROLINK-II, occorre far passare il comando e il riferimento attraverso l'opzione di comunicazione:

- Inverter MV/V7: N3=3; N4=9
- Inverter F7/G7: B1-01=3; B1-02=3.

Assicurarsi che il firmware dell'inverter supporti la scheda MECHATROLINK-II. Il comando restituisce -1 se viene eseguito con successo oppure 0 se non riesce. L'eventuale risultato viene restituito nella VR selezionata.

- Argomenti
- **modulo**
Numero di TJ1-ML__ a cui l'inverter è collegato.
 - **stazione**
Numero di stazione MECHATROLINK-II dell'inverter
 - **numero_parametro**
Il numero di parametro da scrivere. Vedere il manuale dell'inverter.
 - **dimensioni_parametro**
Le dimensioni del parametro da scrivere: 2 o 4 byte. Vedere il manuale dell'inverter.
 - **VR**
L'indirizzo nella memoria VR di TJ1-MC__ dove si trova il nuovo valore del parametro.
 - **modalità**
0 = sola scrittura; 1= scrittura e accesso; 2 = scrittura e config.
 - **valore**
Il nuovo valore che viene scritto.

Esempio Nessun esempio.

Vedere anche N/D



Qualora fosse necessario trasferire molti parametri contemporaneamente, il modo più efficiente è utilizzare MODE 0 per tutti i parametri e MODE 1 per l'ultimo. L'esecuzione di MODE 0 è più rapida dell'esecuzione di MODE 1.

3.2.161 JOGSPEED

- Tipo Parametro degli assi
- Sintassi **JOGSPEED**
- Descrizione Il parametro **JOGSPEED** imposta la velocità di jog di un asse espressa in unità utente. Un jog viene eseguito quando è stato dichiarato un ingresso di jog per un asse e tale ingresso è basso. Per ogni asse sono disponibili un ingresso di jog di avanzamento e un ingresso di jog di marcia indietro, impostati rispettivamente da **FWD_JOG** e **REV_JOG**. È possibile controllare la velocità del jog tramite l'ingresso **FAST_JOG**.
- Argomenti N/D
- Esempio Nessun esempio.
- Vedere anche **AXIS AXIS, FAST_JOG, FWD_JOG, REV_JOG, UNITS.**

3.2.162 KEY

- Tipo Parametro di I/O
- Sintassi **KEY [#n]**
- Descrizione Il parametro **KEY** restituisce **TRUE** o **FALSE** in base all'avvenuta o mancata ricezione di un carattere da parte del buffer della porta seriale. Il risultato **TRUE** genera un ripristino se il carattere viene letto con il comando **GET**. I canali da 5 a 7 sono canali logici che vengono sovrapposti alla porta di programmazione 0 quando sono utilizzati i Trajexia Tools. Nota: il canale 0 è riservato al collegamento con Trajexia Tools e/o l'interfaccia della riga di comando. Tenere presente che questo canale potrebbe dare dei problemi relativamente a questa funzione.
- Argomenti
- **n**
Il dispositivo d'ingresso specificato. Quando questo argomento è omesso, viene utilizzata la porta specificata da **INDEVICE**. Vedere tabella di seguito.

Esempio **WAIT UNTIL KEY#1**

GET#1, k

Tenere presente che se si intende utilizzare **KEY#1** all'interno di un'equazione, può essere necessario mettere l'istruzione tra parentesi, in questo caso: **WAIT UNTIL (KEY#1)=TRUE**.

Vedere anche • **GET**

Valore	Dispositivo di ingresso
0	Porta di programmazione 0
1	Porta seriale 1 di RS-232C
2	Porta seriale 2 di RS-422A/485
5	Canale utente 5 della porta 0 di Trajexia Tools
6	Canale utente 6 della porta 0 di Trajexia Tools
7	Canale utente 7 della porta 0 di Trajexia Tools

3.2.163 LAST_AXIS

Tipo Parametro di sistema

Sintassi **LAST_AXIS** (sola lettura)

Descrizione Il parametro **LAST_AXIS** contiene il numero dell'ultimo asse elaborato al sistema.
La maggior parte dei sistemi non utilizza tutti gli assi disponibili. Sarebbe pertanto una perdita di tempo creare un task per i movimenti inattivi di tutti gli assi che non vengono utilizzati. Per evitare ciò almeno in parte, TJ1-MC__ assegna task di movimento agli assi da 0 a **LAST_AXIS**, laddove **LAST_AXIS** corrisponde al numero dell'asse più alto per il quale è stato elaborato un comando **AXIS** o **BASE**.

Argomenti N/D

Esempio Nessun esempio.

Vedere anche **AXIS**, **BASE**.

3.2.164 LINKAX

Tipo Parametro degli assi (sola lettura)

Sintassi **LINKAX**

Descrizione Restituisce il numero dell'asse al quale l'asse è collegato durante un qualsiasi movimento collegato. Un movimento collegato è un movimento in cui la posizione richiesta è una funzione di un altro asse.

Argomenti N/D

Esempio Nessun esempio.

Vedere anche **CONNECT**, **CAMBOX**, **MOVELINK**.

3.2.165 LINPUT

Tipo Comando di I/O

Sintassi **LINPUT** [#n ,] **variabile_vr**

Descrizione Il comando **LINPUT** assegna il codice ASCII dei caratteri a una matrice di variabili, partendo dalla variabile VR specificata. L'esecuzione del programma è sospesa fino a quando la stringa non viene terminata tramite un ritorno a capo, anch'esso memorizzato. La stringa non viene ripetuta dall'unità di controllo.
I canali da 5 a 7 sono canali logici che vengono sovrapposti alla porta di programmazione 0 quando sono utilizzati i Trajexia Tools. Nota: il canale 0 è riservato al collegamento con Trajexia Tools e/o l'interfaccia della riga di comando. Tenere presente che questo canale potrebbe dare problemi per questo comando.

Argomenti

- **n**
Il dispositivo d'ingresso specificato. Quando questo argomento è omissso, viene utilizzata la porta specificata da INDEVICE. Vedere tabella di seguito.
- **variabile_vr**
La prima variabile VR su cui scrivere.

Esempio Considerare la seguente riga di un programma.

LINPUT#5, VR(0)

Immettendo START<CR> si ottiene

VR(0)=83S

VR(1)=84T

VR(2)=65A

VR(3)=82R

VR(4)=84T

VR(5)=13<CR>

Vedere anche

- **GET**
- **INPUT**
- **VR**

Valore	Dispositivo di ingresso
0	Porta di programmazione 0
1	Porta seriale 1 di RS-232C
2	Porta seriale 2 di RS-422A/485
5	Canale utente 5 della porta 0 di Trajexia Tools
6	Canale utente 6 della porta 0 di Trajexia Tools
7	Canale utente 7 della porta 0 di Trajexia Tools

3.2.166 LIST

Tipo Comando di programma (solo per la riga di comando di Trajexia Tools)

Sintassi **LIST ["nome_programma"]**
TYPE ["nome_programma"]

Descrizione Da utilizzare soltanto con la finestra del terminale. **LIST** viene utilizzato solo come comando immediato (riga di comando) e non deve essere utilizzato nei programmi.
 Il comando **LIST** stampa il programma correntemente selezionato o il programma specificato da **nome_programma**. Il nome del programma può essere specificato anche senza virgolette. Se il nome del programma è omissso, viene inserito in elenco quello correntemente selezionato.
 Nota: questo comando è implementato per un terminale offline (VT100). In Trajexia Tools gli utenti possono utilizzare la finestra del terminale.

Argomenti

- **nome_programma**
 Il programma da stampare.

Esempio Nessun esempio.

Vedere anche **SELECT**.

3.2.167 LIST_GLOBAL

Tipo Comando di sistema (solo del terminale)

Sintassi **LIST_GLOBAL**

Descrizione Quando viene eseguito dalla riga di comando (canale 0 del terminale), restituisce tutti i parametri **GLOBAL** e **CONSTANT** correntemente impostati.

Argomenti N/D

Esempio In un'applicazione dove sono stati impostati i seguenti parametri
GLOBAL e CONSTANT:
CONSTANT "taglierina", 23
GLOBAL "nastro trasportatore", 5

```
>>LIST_GLOBAL
VR globale
-----
nastro trasportatore 5
Valore costante
-----
taglierina 23.0000
```

Vedere anche N/D

3.2.168 LN

Tipo Funzione matematica

Sintassi **LN(espressione)**

Descrizione La funzione **LN** restituisce il logaritmo naturale dell'espressione.
Il valore dell'espressione di ingresso deve essere maggiore di 0.

Argomenti

- **espressione**
Qualsiasi espressione valida in BASIC.

Esempio

```
>> PRINT LN(10)
2.3026
```

Vedere anche N/D

3.2.169 LOCK

Tipo Comando di sistema

Sintassi **LOCK(codice)**
UNLOCK(codice)

Descrizione Il comando **LOCK** impedisce che il programma possa essere visualizzato, modificato o cancellato dal personale non al corrente del codice di sicurezza. Il comando **UNLOCK** consente di rimuovere il blocco precedentemente attivato. Il numero di codice può essere un qualsiasi numero intero ed è conservato sotto forma di codice. **LOCK** è sempre un comando immediato e può essere dato solo quando il sistema è **UNLOCKED**.

Argomenti

- **codice**
Qualsiasi numero intero con un massimo di 7 cifre.

Esempio

```
>> LOCK(561234)
Il programma non può essere modificato né visualizzato.
>> UNLOCK(561234)
Il sistema è ora sbloccato.
```

Vedere anche N/D



Attenzione

È indispensabile ricordare il codice di sicurezza poiché è necessario per sbloccare il sistema. Senza il codice di sicurezza non è possibile recuperare il sistema.

3.2.170 MARK

Tipo Parametro degli assi (sola lettura)

Sintassi **MARK**

Descrizione Il parametro **MARK** è impostato su **FALSE** una volta che è stato eseguito il comando **REGIST**; è invece impostato su **TRUE** quando si verifica l'evento di registrazione primario.

Argomenti N/D

Esempio **IF MARK AXIS(1) THEN**
 PRINT "Si è verificato l'evento di registrazione primario per l'asse 1"
 ENDIF

Vedere anche **AXIS, REGIST, REG_POS.**

3.2.171 MARKB

Tipo Parametro degli assi (sola lettura)

Sintassi **MARKB**

Descrizione Il parametro **MARKB** è impostato su **FALSE** una volta che è stato eseguito il comando **REGIST**; è invece impostato su **TRUE** quando si verifica l'evento di registrazione secondario.

Argomenti N/D

Esempio **IF MARKB AXIS(2) THEN**
 PRINT "Si è verificato l'evento di registrazione secondario per l'asse 2"
 ENDIF

Vedere anche **AXIS, REGIST, REG_POSB.**

3.2.172 MECHATROLINK

Tipo Comando di sistema

Sintassi **MECHATROLINK(unità,0)**

Rileva e collega i dispositivi presenti sull'unità **unità** MECHATROLINK-II. È necessario utilizzare questo comando per reimpostare la rete dopo un problema di comunicazione e per rilevare una seconda volta i servosistemi che non sono stati rilevati (ad esempio: quando la lettera A della parola **AXISSTATUS** diventa maiuscola e rossa).

MECHATROLINK(unità,3,VR)

Restituisce il numero di dispositivi MECHATROLINK-II rilevati dopo l'esecuzione di un **MECHATROLINK(unità,0)**. Viene utilizzato dal programma **STARTUP** per verificare che il numero di unità MECHATROLINK-II rilevate corrisponda a quello previsto.

MECHATROLINK(unità,4,stazione,VR)

Restituisce l'indirizzo del dispositivo MECHATROLINK-II presente su numero di "stazione" indicato. Per tutti i dispositivi collegati, i numeri di stazione sono rappresentati da una sequenza 0...x. Se nessun dispositivo è allocato a quella stazione, viene restituito -1. Viene utilizzato dal programma **STARTUP** per verificare che il numero di unità MECHATROLINK-II rilevate corrisponda a quello previsto.

MECHATROLINK(unità,5,stazione,VR)

Legge e azzerà il conteggio erroneo dei messaggi. Un dispositivo MECHATROLINK-II non relativo agli assi non riporta automaticamente un problema di rete; questo comando va quindi utilizzato per testare l'inverter e i moduli di I/O e verificare così che la rete sia funzionante. Questo comando sarà utilizzato nel programma SHELL proposto.

Nota:

- è possibile utilizzare il comando **MECHATROLINK(unità,5,stazione,VR)** per controllare lo stato di un dispositivo durante l'esecuzione di un programma.
 Se il contenuto dell'indirizzo della memoria VR è maggiore di 0, si verifica un errore di comunicazione con il dispositivo e quest'ultimo può funzionare in modo errato. È possibile utilizzare questo comando per arrestare il programma quando si verifica un errore nel dispositivo.

Descrizione Nota: questo comando dispone di due formati, secondo la funzione richiesta: funzioni master e funzioni stazione.

Tutte le funzioni **MECHATROLINK** restituiscono **TRUE** (–1) se l'esecuzione del comando è riuscita oppure **FALSE** (0) se non è riuscita.

Le funzioni si dividono in 2 categorie: le funzioni **MASTER** che agiscono su un'unità e le funzioni **STATION** che agiscono sullo specifico **indirizzo_stazione** di una data unità.

Tutte le funzioni che recuperano un valore, lo memorizzano nella variabile **VR** indicata nell'ultimo parametro. Se il parametro ha valore –1, esso viene stampato sulla porta della riga di comando.

Note:

- se un comando MECHATROLINK-II non riesce, la stazione MECHATROLINK-II entrerà nello stato di allerta/allarme. Tutti i comandi successivi restituiranno tale stato di allerta/allarme, anche se il comando viene eseguito correttamente.
- È possibile cancellare lo stato di allerta/allarme solo tramite il comando **ALM_CLR**.
- Non esistono sottocomandi **ALM_CLR**; pertanto per inviare il comando **ALM_CLR** occorre entrare in modalità di messa a punto.

Argomenti N/D

Esempio Nessun esempio.

Vedere anche N/D

3.2.173 MERGE

Tipo Parametro degli assi

Sintassi **MERGE**

Descrizione Il parametro **MERGE** è un interruttore software che può essere utilizzato per abilitare o disabilitare l'integrazione di movimenti consecutivi. Se il parametro **MERGE** è attivo e il movimento successivo è già nel buffer dei movimenti successivi (**NTYPE**), la velocità dell'asse non scenderà a 0 e l'asse caricherà il movimento successivo, integrandolo perfettamente. L'impostazione predefinita del parametro **MERGE** è "non attivo".

È compito del programmatore assicurarsi che l'integrazione si inserisca logicamente nel contesto. Ad esempio, l'integrazione di un movimento in avanti con un movimento all'indietro provocherebbe un istantaneo cambiamento di direzione.

Il parametro **MERGE** funziona solo se le seguenti condizioni sono tutte vere:

1. L'integrazione è possibile solo tra movimenti dotati di profilo di velocità, vale a dire **MOVE**, **MOVEABS**, **MOVECIRC**, **MHELICAL**, **REVERSE**, **FORWARD** e **MOVEMODIFY**. Questi movimenti non possono essere integrati con i movimenti collegati **CONNECT**, **MOVELINK** e **CAMBOX**.
2. È presente un movimento nel buffer dei movimenti successivi (**NTYPE**).
3. Il gruppo dell'asse non cambia per i movimenti a più assi.

Quando vengono integrati movimenti a più assi, è sufficiente impostare il parametro degli assi **MERGE** solo sull'asse di base.

Nota: se i movimenti sono brevi, occorre impostare una velocità di decelerazione elevata per evitare che TJ1-MC__ decelerino in previsione della fine del movimento presente sul buffer.

Argomenti N/D

Esempio **MERGE = OFF** ' Decelera alla fine di ogni movimento
MERGE = ON ' Se possibile, i movimenti verranno integrati

Vedere anche **AXIS**.

3.2.174 MHELICAL

Tipo	Comando degli assi
Sintassi	MHELICAL(fine1, fine2, centro1, centro2, direzione, distanza3) MH(fine1, fine2, centro1, centro2, direzione, distanza3)
Descrizione	Esegue un movimento elicoidale, ovvero, muove 2 assi ortogonali in modo tale da produrre un arco nel punto dove si trova lo strumento, con un movimento lineare simultaneo su un terzo asse. I primi 5 parametri sono simili a quelli di un comando MOVECIRC() . Il sesto parametro definisce il movimento lineare simultaneo. Fine1 e centro1 si trovano sull'asse BASE corrente. Fine2 e centro2 si trovano sull'asse successivo. Le prime 4 distanze e il sesto parametro vengono scalate in base al fattore di conversione delle unità corrente per ogni asse.
Argomenti	• fine1 Posizione finale che deve essere raggiunta sull'asse BASE. • fine2 Posizione finale che deve essere raggiunta sull'asse successivo della matrice BASE. • centro1 Posizione dell'asse BASE attorno al quale si svolge il movimento. • centro2 Posizione dell'asse successivo della matrice BASE attorno al quale si svolge il movimento. • direzione La direzione è un interruttore software che determina se l'arco viene interpolato in direzione oraria o antioraria. Il parametro è impostato su 0 o 1. Vedere MOVECIRC. • distanza3 La distanza, espressa in unità utente, che deve essere percorsa sul terzo asse della matrice BASE.
Esempio	Nessun esempio.
Vedere anche	MOVECIRC .

3.2.175 MOD

Tipo	Funzione matematica
Sintassi	espressione1 MOD espressione2
Descrizione	La funzione MOD restituisce il modulo espressione2 dell' espressione1 . Questa funzione prende il numero intero di un qualsiasi ingresso non intero.
Argomenti	• espressione1 Qualsiasi espressione valida in BASIC. • espressione2 Qualsiasi espressione valida in BASIC.
Esempio	>> PRINT 122 MOD 13 5.0000
Vedere anche	N/D

3.2.176 MOTION_ERROR

Tipo	Parametro di sistema (sola lettura)
Sintassi	MOTION_ERROR
Descrizione	Il parametro MOTION_ERROR contiene uno schema dei bit che mostra gli assi contenenti un errore di movimento. Ad esempio, se gli assi 2 e 6 contengono un errore di movimento, il valore di MOTION_ERROR sarà 68 (4+64). Un errore di movimento si verifica quando lo stato AXISSTATUS di un asse coincide con l'impostazione ERRORMASK. In questo caso il selettore di attivazione (WDOG) viene disattivato, MOTION_ERROR prende uno schema dei bit che mostra tutti gli assi contenenti l'errore di movimento e il parametro ERROR_AXIS prende il numero del primo asse contenente l'errore. È possibile cancellare un errore di movimento eseguendo un comando DATUM(0).
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	AXIS , AXISSTATUS , DATUM , ERROR_AXIS , ERRORMASK , WDOG .

3.2.177 MOVE

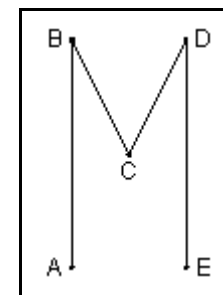
Tipo	Comando degli assi
Sintassi	MOVE(distanza_1 [, distanza_2 [, distanza_3 [, distanza_4 [, ...]]]]) MO(distanza_1 [, distanza_2 [, distanza_3 [, distanza_4 [, ...]]]])
Descrizione	Il comando MOVE sposta alla velocità, accelerazione e decelerazione richieste uno o più assi nella posizione specificata, come incremento rispetto alla posizione corrente. Nei movimenti a più assi il movimento viene interpolato e la velocità, accelerazione e decelerazione vengono prese dall'asse di base. Le distanze specificate vengono scalate utilizzando il fattore di conversione delle unità contenuto nel parametro assi UNITS. Ad esempio, se un asse ha 4.000 fronti di encoder/mm, il numero di unità per quell'asse deve essere impostato su 4000 e MOVE(12.5) produrrà un movimento di 12,5 mm. MOVE agisce sul gruppo di assi di base predefinito (impostato con BASE) a meno che AXIS non sia usato per specificare un asse di base temporaneo. L'argomento distanza_1 viene applicato all'asse di base, distanza_2 viene applicato all'asse successivo, ecc. Cambiando l'asse tra i singoli comandi MOVE è possibile ottenere un movimento a più assi, non sincronizzato e non interpolato. Attivando il parametro degli assi MERGE, è possibile integrare movimenti incrementali ottenendo un profilo di movimenti a posizionamento continuo. Nel caso di un movimento di 2 assi, le velocità individuali vengono calcolate mediante le equazioni indicate di seguito. Dato il comando MOVE(x₁,x₂) e il profilo di velocità v_p calcolato a partire dai parametri SPEED, ACCEL e DECEL ricavati dall'asse di base e dalla distanza totale tra i diversi assi $L = \text{SQR}(x_1^2 + x_2^2)$. La velocità individuale v_i dell'asse i in qualsiasi istante del movimento viene calcolata come segue: $v_i = (x_i * v_p) / L$.
Argomenti	Il comando può supportare fino a 16 argomenti. distanza_i La distanza che deve percorrere ogni asse i espressa in unità utente a partire dall'asse di base.

Esempio Un sistema sta funzionando con un fattore di conversione delle unità pari a 1 e dispone di un encoder a 1000 righe. È quindi necessario utilizzare il seguente comando per eseguire 10 movimenti rotatori sul motore. (Un encoder a 1000 righe genera 4000 fronti/giro).
MOVE(40000)

Esempio **MOVE(10) AXIS(0)**
MOVE(10) AXIS(1)
MOVE(10) AXIS(2)

In questo esempio, gli assi 0, 1 e 2 vengono mossi indipendentemente (senza interpolazione). Ogni asse si muove in base alla propria velocità programmata e agli altri parametri degli assi.

Esempio Una stampante di diagrammi X-Y può scrivere un testo in qualsiasi punto del suo spazio operativo. I singoli caratteri sono definiti come una sequenza di movimenti in relazione a un punto di partenza; in questo modo i medesimi comandi possono essere utilizzati a prescindere dalla posizione specificata. La subroutine di comando per la lettera M potrebbe essere la seguente:
MOVE(0,12) ' A -> B
MOVE(3,-6) ' B -> C
MOVE(3,6) ' C -> D
MOVE(0,-12) ' D -> E



Vedere anche **AXIS**, **MOVEABS**, **UNITS**.

3.2.178 MOVEABS

Tipo	Comando degli assi
Sintassi	MOVEABS(distanza_1 [, distanza_2 [, distanza_3 [, distanza_4 [, ...]]]) MA(distanza_1 [, distanza_2 [, distanza_3 [, distanza_4 [, ...]]])
Descrizione	Il comando MOVEABS sposta alla velocità, accelerazione e decelerazione richieste uno o più assi nella posizione specificata come posizione assoluta, cioè con riferimento all'origine. Nei movimenti a più assi il movimento viene interpolato e la velocità, accelerazione e decelerazione vengono prese dall'asse di base. Le distanze specificate vengono scalate utilizzando il fattore di conversione delle unità contenuto nel parametro degli assi UNITS. Ad esempio, se un asse ha 4.000 fronti di encoder/mm, il numero di unità per quell'asse viene impostato su 4000 e MOVEABS(12.5) si sposterà a un punto che si trova a 12,5 mm dall'origine. MOVEABS agisce sul gruppo di assi di base predefinito (impostato con BASE) a meno che AXIS non sia usato per specificare un asse di base temporaneo. L'argomento distanza_1 viene applicato all'asse di base, distanza_2 viene applicato all'asse successivo, ecc. Cambiando l'asse tra i singoli comandi MOVE è possibile ottenere un movimento a più assi, non sincronizzato e non interpolato. Attivando il parametro degli assi MERGE, è possibile integrare movimenti assoluti ottenendo un profilo di movimenti a posizionamento continuo. Nel caso di un movimento di 2 assi, le velocità individuali vengono calcolate mediante le equazioni indicate di seguito. Dato il comando MOVE(ax₁,ax₂), la posizione corrente (ay₁,ay₂) e il profilo di velocità v_p calcolati a partire dai parametri SPEED, ACCEL e DECEL ricavati dell'asse di base e dalla distanza totale tra i diversi assi $L = \text{SQR}(x_1^2 + x_2^2)$, dove $x_i = ax_i - ay_i$. La velocità individuale dell'asse in un qualsiasi istante del movimento viene calcolata come $v_i = (x_i \times v_p) / L$.
Argomenti	Il comando può supportare fino a 16 argomenti. distanza_i La posizione a cui muovere ogni asse <i>i</i> espressa in unità utente, a partire dall'asse di base.

Esempio **MOVEABS(20,350)**
Una stampante di diagrammi X-Y dispone di un portapenna rotante la cui posizione viene fissata in base all'origine del diagrammatore. Per cambiare penna, un movimento assoluto verso la posizione del portapenna rotante individuerà l'obiettivo, a prescindere dalla posizione segnata quando il comando viene eseguito.

Esempio Un pallet consiste in una griglia 6x8 sulla quale un'imballatrice posiziona bombole del gas a 85 mm l'una dall'altra. Le bombole sono prelevate da un punto fisso. La prima posizione sul pallet viene definita come posizione (0,0) utilizzando il comando DEFPOS. La parte di programma che posiziona le bombole sul pallet è come segue:

```

xanello:
  FOR x = 0 TO 5
yanello:
  FOR y = 0 TO 7
    MOVEABS(-340,-516.5) ' Spostarsi fino al punto di prelievo
    GOSUB prelievo ' Va alla subroutine di prelievo
    PRINT "MOVE TO POSITION: ";x*6+y+1
    MOVEABS(x*85,y*85)
    GOSUB deposito ' Va alla subroutine di deposito
  NEXT y
NEXT x

```

Vedere anche **AXIS**, **MOVE**, **MOVEABS**, **UNITS**.

3.2.179 MOVECIRC

Tipo Comando degli assi

Sintassi **MOVECIRC(fine_1,fine_2,centro_1,centro_2,direzione)**
MC(fine_1,fine_2,centro_1,centro_2,direzione)

Descrizione Il comando **MOVECIRC** interpola 2 assi ortogonali producendo un arco di circonferenza. Il tracciato del movimento è determinato dai 5 argomenti, che sono incrementali a partire dalla posizione corrente.

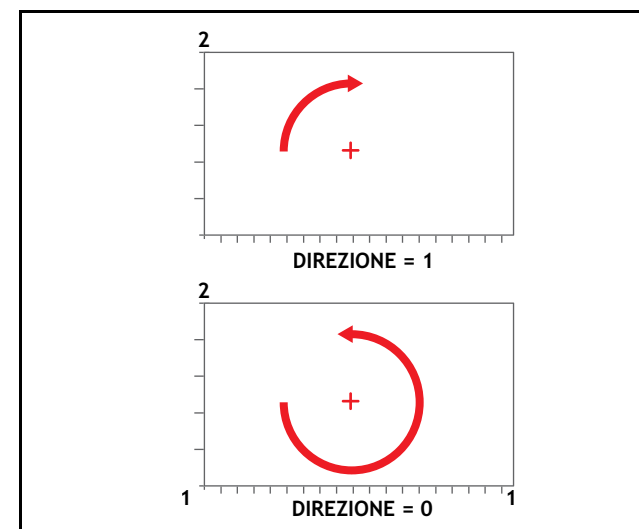
Gli argomenti **fine_1** e **centro_1** si applicano all'asse di base mentre **fine_2** e **centro_2** si applicano all'asse seguente. Tutti gli argomenti sono espressi nelle unità utente di ciascun asse. La velocità di spostamento lungo l'arco di circonferenza è stabilito dai parametri **SPEED**, **ACCEL** e **DECEL** dell'asse di base.

MOVECIRC agisce sul gruppo di assi di base predefinito (impostato con **BASE**) a meno che **AXIS** non sia usato per specificare un asse di base temporaneo.

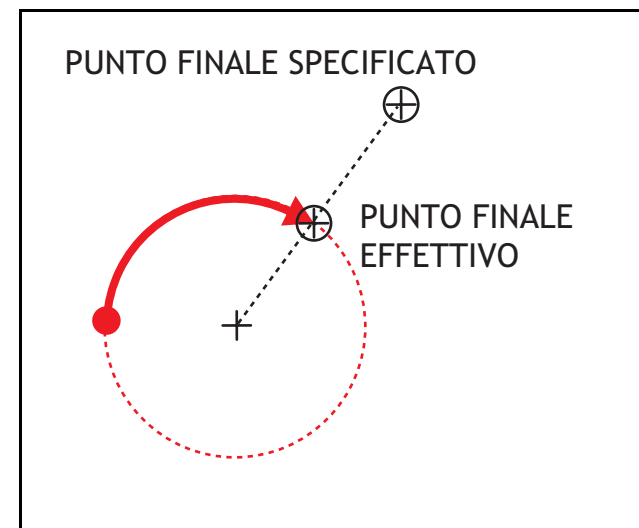
Affinché **MOVECIRC** venga eseguito correttamente, i due assi che si spostano lungo l'arco di circonferenza devono avere lo stesso numero di impulsi dell'encoder per ogni tratto di distanza lineare tra un asse e l'altro. Se così non è, in molti casi è possibile regolare le scale dell'encoder tramite il parametro degli assi **ENCODER_RATIO**.

Argomenti

- **fine_1**
La posizione finale dell'asse di base.
- **fine_2**
La posizione finale dell'asse successivo.
- **centro_1**
La posizione attorno alla quale deve muoversi l'asse di base.
- **centro_2**
La posizione attorno alla quale deve muoversi l'asse successivo.
- **direzione**
Un interruttore software che determina se l'arco viene interpolato in direzione oraria o antioraria. Valore: 0 o 1.
Se i due assi coinvolti nel movimento formano un asse sul lato destro, impostare la direzione su 0 per produrre un movimento positivo attorno al terzo (presumibilmente immaginario) asse ortogonale. Se i due assi coinvolti nel movimento formano un asse sul lato sinistro, impostare la direzione su 1 per produrre un movimento negativo attorno al terzo (presumibilmente immaginario) asse ortogonale. Vedere tabella di seguito.

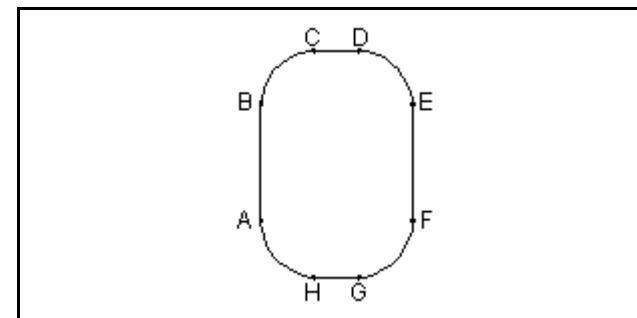


Nota: Il comando **MOVECIRC** calcola il raggio, l'angolo totale di rotazione dal centro e il punto finale. Se il punto finale non si trova sul percorso calcolato, il movimento termina semplicemente nel punto finale calcolato e non nel punto finale specificato. È responsabilità del programmatore assicurarsi che i due punti corrispondano a punti corretti di un cerchio.



Esempio Di seguito, una possibile sequenza di comandi per tracciare la lettera O:

```
MOVE(0,6) ' Sposta A -> B
MOVECIRC(3,3,3,0,1) ' Sposta B -> C
MOVE(2,0) ' Sposta C -> D
MOVECIRC(3,-3,0,-3,1) ' Sposta D -> E
MOVE(0,-6) ' Sposta E -> F
MOVECIRC(-3,-3,-3,0,1) ' Sposta F -> G
MOVE(-2,0) ' Sposta G -> H
MOVECIRC(-3,3,0,3,1) ' Sposta H -> A
```



Vedere anche **AXIS, ENCODER_RATIO, UNITS**

Direzione	Asse lato destro	Asse lato sinistro
1	Negativo	Positivo
0	Positivo	Negativo

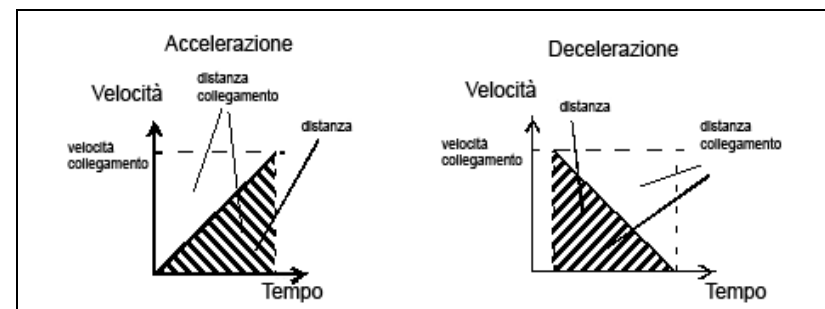
3.2.180 MOVELINK

Argomenti

Tipo Comando degli assi

Sintassi **MOVELINK**(distanza, distanza_collegamento, accelerazione_collegamento, decelerazione_collegamento, asse_collegamento [, opzione_collegamento [, posizione_collegamento]])
ML(distanza, distanza_collegamento, accelerazione_collegamento, decelerazione_collegamento, asse_collegamento [, opzione_collegamento [, posizione_collegamento]])

Descrizione Il comando **MOVELINK** crea un movimento lineare dell'asse di base collegato tramite trasmissione software alla posizione misurata di un asse di collegamento. L'asse di collegamento può spostarsi in entrambe le direzioni per azionare il movimento di uscita. I parametri mostrano la distanza coperta dall'asse di base per una data distanza percorsa dall'asse di collegamento (**distanza_collegamento**). La distanza dell'asse di collegamento è suddivisa in base alle tre fasi di movimento dell'asse di base. Le tre fasi sono l'accelerazione, la velocità costante e la decelerazione. Le distanze di accelerazione e decelerazione dell'asse di collegamento sono specificate dai parametri **accelerazione_collegamento** e **decelerazione_collegamento**. La distanza della velocità costante del collegamento viene derivata dalla distanza totale coperta dall'asse di collegamento e da questi due parametri. È possibile dividere le tre fasi in comandi **MOVELINK** separati o sommarli in un unico comando. Tenere in considerazione le seguenti due regole quando si imposta il comando **MOVELINK**.
 Regola 1: in una fase di accelerazione o di decelerazione con velocità corrispondenti, il parametro **distanza_collegamento** deve essere il doppio della distanza. Vedere la figura.
 Regola 2: in una fase di velocità costante con velocità corrispondenti, i due assi percorrono la stessa distanza; pertanto la distanza percorsa deve essere uguale a **distanza_collegamento**.
MOVELINK agisce sul gruppo di assi di base predefinito (impostato con **BASE**) a meno che **AXIS** non sia usato per specificare un asse di base temporaneo. L'asse impostato per **asse_collegamento** aziona l'asse di base.
 Nota: se la somma di **accelerazione_collegamento** e **decelerazione_collegamento** è maggiore di **distanza_collegamento**, entrambe vengono proporzionalmente ridotte per essere pari a **distanza_collegamento**.



- **distanza**
La distanza incrementale espressa in unità utente, risultante dal movimento **distanza_collegamento** misurato sull'asse di collegamento.
- **distanza_collegamento**
La distanza incrementale positiva, espressa in unità utente, che deve essere misurata sull'asse di collegamento per produrre sull'asse di base il movimento che coprirà la distanza.
- **accelerazione_collegamento**
La distanza incrementale positiva, espressa in unità utente, lungo la quale l'asse di base accelererà.
- **decelerazione_collegamento**
La distanza incrementale positiva, espressa in unità utente, lungo la quale l'asse di base decelererà.
Nota: se la somma del parametro 3 e del parametro 4 è maggiore del parametro 2, entrambi vengono proporzionalmente ridotti fino a quando tale somma non è pari al parametro 2.
- **asse_collegamento**
L'asse a cui ci si deve collegare.
- **opzione_collegamento**
Vedere tabella di seguito.
- **posizione_collegamento**
La posizione assoluta da dove **MOVELINK** inizierà quando l'**opzione_collegamento** è impostata su 2.

Nota: il comando utilizza i fattori **BASE()** e **AXIS()** e il fattore di conversione delle unità in maniera analoga ad altri comandi **MOVE**.

Nota: l'asse di "collegamento" può spostarsi in entrambe le direzioni per azionare il movimento di uscita. Le distanze di collegamento specificate sono sempre positive.

Esempio Una lama semovente taglia un rotolo di carta ogni 160 m, spostandosi alla stessa velocità della carta. La capacità di spostamento della lama è pari a 1,2 m; in questo esempio, viene usato 1 m di tale capacità. La distanza della carta è misurata da un encoder; il fattore di conversione delle unità è impostato per restituire unità espresse in metri su entrambi gli assi. L'asse 1 è l'asse di collegamento.

MOVELINK(0,150,0,0,1) ' attende la distanza
MOVELINK(0.4,0.8,0.8,0,1) ' accelera
MOVELINK(0.6,1.0,0,0.8,1) ' raggiunge la velocità prevista e poi decelera
WAIT UNTIL NTYPE=0 ' attende fino a quando non inizia l'ultimo movimento
OP(8,ON) ' attiva la taglierina
MOVELINK(-1,8,2,0.5,0.5,1) ' torna indietro

In questo programma, TJ1-MC__ attende lo svolgimento di 150 m alla prima riga. Quando questa distanza viene raggiunta, la lama accelera fino a raggiungere la velocità della carta, la segue a quella stessa velocità, quindi decelera fino ad arrestarsi entro 1 m dal finecorsa. Questo movimento viene specificato utilizzando due comandi **MOVELINK** separati. Il programma quindi attende che il buffer dei movimenti successivi si azzeri **NTYPE=0**. Questo indica che la fase di accelerazione è completa. Le distanze dell'asse di collegamento (**distanza_collegamento**) contenute nei comandi **MOVELINK** sono 150, 0,8, 1,0 e 8,2 che, sommate, danno 160 m. Per garantire che la velocità e la posizione della taglierina e della carta corrispondano durante la fase di taglio, gli argomenti del comando **MOVELINK** devono essere corretti. Suggerimento: considerare separatamente le fasi di accelerazione, velocità costante e decelerazione. Come si è detto, in una fase di accelerazione o di decelerazione con velocità corrispondenti, il parametro **distanza_collegamento** deve essere il doppio della distanza. È possibile specificare entrambe le fasi come:

MOVELINK(0.4,0.8,0.8,0,1) ' Questo spostamento è interamente accel
MOVELINK(0.4,0.8,0,0.8,1) ' Questo spostamento è interamente decel

In una fase di velocità costante con velocità corrispondenti, i due assi percorrono la stessa distanza; pertanto la distanza percorsa deve essere uguale alla distanza di collegamento. È possibile specificare la fase di velocità costante come segue:

MOVELINK(0.2,0.2,0,0,1) ' Questa è interamente velocità costante

Il comando **MOVELINK** consente l'aggiunta delle tre sezioni sommando per ciascuna fase **distanza**, **distanza_collegamento**, **accelerazione_collegamento** e **decelerazione_collegamento**, producendo così il seguente comando.

MOVELINK(1,1.8,0.8,0.8,1)

Nel programma qui sopra, la fase di accelerazione è programmata separatamente per permettere a un'azione di essere eseguita alla fine della fase di accelerazione.

MOVELINK(0.4,0.8,0.8,0,1)

MOVELINK(0.6,1.0,0,0.8,1)

Vedere anche **AXIS, UNITS, REP_OPTION**.

Opzione di collegamento	Descrizione
1	Il collegamento inizia quando si verifica un evento di registrazione sull'asse di collegamento.
2	Il collegamento inizia da una posizione assoluta dell'asse di collegamento (vedere posizione_collegamento).
4	MOVELINK si ripete automaticamente e nelle due direzioni. Questa opzione viene cancellata impostando il bit 1 del parametro REP_OPTION (cioè REP_OPTION = REP_OPTION OR 2).
5	Combinazione delle opzioni 1 e 4.
6	Combinazione delle opzioni 2 e 4.

3.2.181 MOVEMODIFY

Tipo	Comando degli assi
Sintassi	MOVEMODIFY(posizione) MM(posizione)
Descrizione	Il comando MOVEMODIFY cambia la posizione finale assoluta del movimento lineare corrente ad asse singolo (MOVE o MOVEABS). Se non vi è nessun movimento corrente o se il movimento corrente non è un movimento lineare, MOVEMODIFY viene trattato come un comando MOVEABS . Il parametro ENDMOVE conterrà la posizione finale del movimento corrente espresso in unità utente. MOVEMODIFY agisce sull'asse di base predefinito (impostato con BASE) a meno che AXIS non sia usato per specificare un asse di base temporaneo.
Argomenti	posizione La posizione assoluta da impostare come nuova posizione finale del movimento.
Esempio	Nessun esempio.
Vedere anche	AXIS, MOVE, MOVEABS, UNITS .

3.2.182 MPOS

Tipo	Parametro degli assi (sola lettura)
Sintassi	MPOS
Descrizione	Il parametro MPOS è la posizione misurata dell'asse derivata dall'encoder ed espressa in unità utente. È possibile impostare questo parametro utilizzando il comando DEFPOS . Per spostare il punto di origine è inoltre possibile utilizzare il parametro degli assi OFFPOS . All'avvio, MPOS viene reimpostato su 0. L'intervallo della posizione misurata è controllato attraverso i parametri degli assi REP_DIST e REP_OPTION .
Argomenti	N/D
Esempio	WAIT UNTIL MPOS >= 1250 SPEED = 2.5
Vedere anche	UNITS, AXIS, DEFPOS, ENCODER, FE, OFFPOS, REP_DIST, REP_OPTION, UNITS.

3.2.183 MSPEED

Tipo	Parametro degli assi (sola lettura)
Sintassi	MSPEED
Descrizione	Il parametro MSPEED contiene la velocità misurata espressa in unità/s. Viene calcolato prendendo il cambiamento della posizione misurato durante l'ultimo ciclo di servoazionamento (espresso in unità utente) e dividendolo per il ciclo di servoazionamento (espresso in secondi). Il periodo di servoazionamento viene impostato con il parametro SERVO_PERIOD . MSPEED rappresenta un'immagine della velocità e delle fluttuazioni significative che possono verificarsi specialmente alle basse velocità. Se si richiede un valore stabile alle basse velocità, può essere utile generare una media delle diverse letture.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	AXIS, SERVO_PERIOD, VP_SPEED, UNITS.

3.2.184 MTYPE

Tipo	Parametro degli assi (sola lettura)
Sintassi	MTYPE
Descrizione	Il parametro MTYPE contiene il tipo di movimento attualmente in esecuzione. I valori possibili sono elencati nella tabella di seguito. È possibile utilizzare MTYPE per determinare se un movimento è terminato o se c'è stata una transizione da un tipo di movimento a un altro. Se un movimento non è inattivo ciò non significa necessariamente che l'asse si stia effettivamente muovendo. Potrebbe avere eseguito un movimento solo in parte e trovarsi in quel momento ad avere velocità 0 oppure potrebbe essere in fase di interpolazione con un altro asse ed essere quindi fermo.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	AXIS, NTYPE.

Numero del movimento	Tipo di movimento
0	IDLE (nessun movimento)
1	MOVE
2	MOVEABS
3	MHELICAL
4	MOVECIRC
5	MOVEMODIFY
10	FORWARD
11	REVERSE
12	DATUM
13	CAM
14	JOG_FORWARD fare riferimento a FWD_JOG
15	JOG_REVERSE fare riferimento a REV_JOG

Numero del movimento	Tipo di movimento
20	CAMBOX
21	CONNECT
22	MOVELINK

3.2.185 NAO

Tipo	Parametro di sistema (sola lettura)
Sintassi	NAO
Descrizione	Questo parametro restituisce il numero di canali di ingresso analogici collegati al bus di espansione MECHATROLINK-II. Ad esempio, TJ1-MC__ restituisce 8 se vi sono due unità AN2900 collegate, poiché ognuna dispone di 4 canali di ingresso analogici.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	N/D

3.2.186 NEG_OFFSET

Tipo	Parametro di sistema
Sintassi	NEG_OFFSET=valore
Descrizione	Per il funzionamento piezoelettrico. Consente a un offset negativo di essere applicato al segnale di uscita DAC proveniente dall'anello servo. L'offset viene applicato dopo la funzione DAC_SCALE . Un offset di 327 rappresenta un offset di 0,1 volt. Si consiglia di usare un offset pari al 65 – 70% del valore necessario per far muovere lo stadio in una situazione ad anello aperto.
Argomenti	valore Espressione in BASIC.
Esempio	Nessun esempio.
Vedere anche	N/D

3.2.187 NEW

Tipo	Comando di programma
Sintassi	NEW ["nome_programma"]
Descrizione	Il comando NEW elimina dal programma tutte le righe relative all'unità di controllo. Usando il comando NEW senza il nome di un programma, si cancella il programma correntemente selezionato (con il comando SELECT). Il nome del programma può essere specificato anche senza virgolette. NEW ALL cancella tutti i programmi. È possibile usare questo comando anche per cancellare la memoria TABLE. NEW "TABLE" Il nome "TABLE" va scritto tra virgolette. Nota: questo comando è implementato per un terminale offline (VT100). All'interno di Trajexia Tools gli utenti possono selezionare il comando dal menu Program.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	COPY, DEL, RENAME, SELECT, TABLE

3.2.188 NEXT

Vedere **FOR..TO..STEP..NEXT**.

3.2.189 NIO

Tipo	Parametro di sistema
Sintassi	NIO
Descrizione	Restituisce il numero di ingressi/uscite installati sul sistema o collegati al bus di espansione MECHATROLINK-II. Se TJ-MC__ non dispone di MECHATROLINK-II, l'I/O restituisce NIO=32 . Gli ingressi incorporati sono i canali da 0 a 15. Le uscite incorporate sono i canali da 8 a 15. I canali da 16 a 27 possono essere utilizzati come I/O "virtuali" collegati tra di loro. I canali d'ingresso da 28 a 31 sono riservati per consentire a ciascun asse di utilizzare i canali di ingresso del driver MECHATROLINK-II per le funzioni di controllo degli assi.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	N/D

3.2.190 NOT

Tipo	Operazione matematica
Sintassi	NOT espressione
Descrizione	L'operatore NOT esegue la funzione logica NOT su tutti bit dei numeri interi dell'espressione. La funzione logica NOT tra due bit viene definita nella tabella di seguito.
Argomenti	espressione Qualsiasi espressione valida in BASIC.
Esempio	>> PRINT 7 AND NOT 1 6.0000
Vedere anche	N/D

Bit	Risultato
0	1
1	0

3.2.191 NTYPE

Tipo	Parametro degli assi (sola lettura)
Sintassi	NTYPE
Descrizione	Il parametro NTYPE contiene il tipo del movimento presente nel buffer dei movimenti successivi. Una volta terminato il movimento corrente, sarà eseguito il movimento presente nel buffer NTYPE . I valori sono gli stessi del parametro degli assi MTYPE . NTYPE viene azzerato dal comando CANCEL(1) .
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	AXIS, MTYPE .

3.2.192 OFF

Tipo	Costante (sola lettura)
Sintassi	OFF
Descrizione	La costante OFF restituisce il valore numerico 0.
Argomenti	N/D
Esempio	OP (leva,OFF) Questa riga imposta l'output denominato "leva" su off.
Vedere anche	N/D

3.2.193 OFFPOS

Tipo	Parametro degli assi
Sintassi	OFFPOS
Descrizione	Il parametro OFFPOS contiene un offset che sarà applicato alla posizione richiesta (DPOS) senza influenzare il movimento in nessun altro modo. La posizione misurata sarà modificata di conseguenza per mantenere l'errore di inseguimento. OFFPOS regola in maniera efficace la posizione 0 dell'asse. Il valore impostato nel parametro OFFPOS viene reimpostato su 0 dal sistema quando viene caricato l'offset. Nota: l'offset viene applicato in corrispondenza del ciclo di servozionamento successivo. È ammessa l'esecuzione di altri comandi prima del ciclo di servozionamento successivo. Assicurarsi che tali comandi non presumano che si sia verificato uno spostamento di posizione. È possibile ottenere questo risultato utilizzando l'istruzione WAIT UNTIL (vedere esempio).
Argomenti	N/D
Esempio	Le righe di seguito definiscono la posizione attualmente richiesta come 0. OFFPOS = -DPOS WAIT UNTIL OFFPOS = 0 ' Aspetta fino a quando non viene applicata Questo esempio è equivalente a DEFPOS(0).
Vedere anche	AXIS, DEFPOS, DPOS, MPOS, UNITS .

3.2.194 ON

Tipo	Costante (sola lettura)
Sintassi	ON
Descrizione	La costante ON restituisce il valore numerico 1.
Argomenti	N/D
Esempio	OP (leva,ON) Questa riga imposta l'output denominato "leva" su ON .
Vedere anche	N/D

3.2.195 ON.. GOSUB

Tipo	Comando di controllo del programma.
Sintassi	ON espressione GOSUB etichetta { , etichetta }
Descrizione	Le strutture ON...GOSUB e ON...GOTO consentono un conditional jump. L'espressione intera è utilizzata per selezionare un'etichetta dall'elenco. Se l'espressione ha valore 1 viene utilizzata la prima etichetta, se ha valore 2 viene utilizzata la seconda etichetta e così via. Una volta che l'etichetta è selezionata, la subroutine GOSUB esegue il passaggio a quella etichetta. Nota: se l'espressione non è valida, non viene eseguito nessun passaggio.
Argomenti	• espressione Qualsiasi espressione valida in BASIC. • etichetta Qualsiasi etichetta valida presente nel programma.
Esempio	REPEAT GET#5,car UNTIL 1<=car e car<=3 ON car GOSUB motore, arresto, cambio
Vedere anche	GOSUB..RETURN, GOTO.

3.2.196 ON.. GOTO

Tipo	Comando di controllo del programma.
Sintassi	ON espressione GOTO etichetta[,etichetta[,...]]
Descrizione	L'espressione viene valutata, dopodiché la parte intera viene utilizzata per selezionare un'etichetta dall'elenco. Se l'espressione ha valore 1 viene utilizzata la prima etichetta, se ha valore 2 viene utilizzata la seconda etichetta e così via. Se il valore dell'espressione è minore di 1 o maggiore del numero di etichette, si verifica un errore. Una volta che l'etichetta è selezionata, la subroutine GOTO esegue il passaggio a quella etichetta.
Argomenti	• espressione Qualsiasi espressione valida in BASIC. • etichetta Qualsiasi etichetta valida presente nel programma.
Esempio	REPEAT GET #1,car UNTIL 1<=car e car<=3 ON car GOTO motore, arresto, cambio
Vedere anche	N/D

3.2.197 OP

Tipo	Comando di I/O
Sintassi	OP(numero_uscita, valore) OP(schema_binario) OP

- Descrizione Il comando **OP** imposta una o più uscite oppure restituisce lo stato delle prime 24 uscite. **OP** può avere tre forme differenti, secondo il numero degli argomenti.
- Il comando **OP(numero_uscita, valore)** imposta un solo canale di uscita. L'intervallo del **numero_uscita** è compreso tra 8 e 256, mentre **valore** è il valore che deve essere inviato, cioè 0 o 1.
 - Il comando **OP(schema_binario)** imposta lo schema binario sulle 24 uscite, secondo il valore impostato da **schema_binario**.
 - La funzione **OP (senza argomenti)** restituisce lo stato delle prime 24 uscite. Ciò consente di impostare più uscite senza corrompere le altre che non devono essere cambiate.

Nota: le prime 8 uscite (da 0 a 7) non esistono fisicamente su TJ1-MC___. Non è possibile scrivere su di esse e restituiranno sempre 0.

- Argomenti
- numero_uscita**
Il numero dell'uscita da impostare.
 - valore**
Il valore che deve essere inviato, cioè "attivo" o "non attivo". Tutti i valori diversi da 0 sono considerati "attivi".
 - schema_binario**
L'intero equivalente allo schema binario che deve essere inviato.

Esempio **OP(12,1)**
OP(12,ON)
Queste due righe sono equivalenti.

Esempio **OP(18*256)**
Questa riga imposta lo schema dei bit 10010 sulle prime 5 uscite fisiche; le uscite dalla 13 alla 17 vengono azzerate. Dal momento che le uscite dalla 0 alla 7 non esistono, per impostare le prime uscite disponibili lo schema dei bit viene spostato di 8 bit eseguendo una moltiplicazione per 256.

Esempio **VR(0) = OP**
VR(0) = VR(0) AND 65280
OP(VR(0))
Questa routine imposta le uscite dalla 8 alla 15 su **ON** (attivo) e tutte le altre su "off" (non attivo).
È possibile scrivere il programma precedente anche in questo modo:
OP(OP AND 65280)

Esempio **val = 8 ' Il valore da impostare**
maschera = OP AND NOT(15*256) ' Ottiene lo stato e la maschera correnti
OP(maschera OR val*256) ' Imposta val su OP(8) su OP(11)
Questa routine imposta il valore **val** sulle uscite dalla 8 alla 11 senza influenzare le altre uscite tramite il mascheramento.

Vedere anche **IN**.

3.2.198 OPEN_WIN

- Tipo Parametro degli assi
- Sintassi **OPEN_WIN**
OW
- Descrizione Il parametro degli assi **OPEN_WIN** definisce l'inizio della finestra all'interno o all'esterno della quale è prevista la presenza di un evento di registrazione. Il valore è espresso in unità utente.
- Argomenti N/D
- Esempio Nessun esempio.
- Vedere anche **CLOSE_WIN, REGIST, UNITS**.

3.2.199 OR

- Tipo Operazione matematica
- Sintassi **espressione1 OR espressione2**
- Descrizione L'operatore **OR** esegue la funzione logica **OR** tra i bit corrispondenti dei numeri interi di due espressioni valide in BASIC.
La funzione logica **OR** tra due bit viene definita nella tabella di seguito.
- Argomenti
- espressione1**
Qualsiasi espressione valida in BASIC.
 - espressione2**
Qualsiasi espressione valida in BASIC.

Esempio **Esempio 1:**
risultato = 10 OR (2.1*9)
 Le operazioni tra parentesi vengono calcolate per prime, ma solo il numero intero del risultato, cioè 18, è utilizzato per l'operazione.
 Pertanto questa espressione equivale alla seguente:
risultato = 10 OR 18

Pertanto, "risultato" conterrà il valore 26.

Esempio 2:

Esempio **risultato = 10 OR 18**
OR è un operatore di bit e quindi l'azione binaria che interviene è:
01010 OR 10010 = 11010

Esempio **IF KEY OR VR(0) = 2 THEN GOTO etichetta**

Vedere anche N/D

Bit 1	Bit 2	Risultato
0	0	0
0	1	1
1	0	1
1	1	1

3.2.200 OUTDEVICE

Tipo Parametro di I/O

Sintassi **OUTDEVICE**

Descrizione Il parametro **OUTDEVICE** definisce il dispositivo di uscita predefinito. Il dispositivo viene selezionato per il comando **PRINT** quando l'opzione **#n** è omessa. Il parametro **OUTDEVICE** è specifico di un task. I valori supportati sono elencati nella tabella di seguito.

Argomenti N/D

Esempio Nessun esempio.

Vedere anche **PRINT**.

Valore	Descrizione
0	Porta di programmazione 0 (predefinita)
1	Porta seriale 1 di RS-232C
2	Porta seriale 2 di RS-422A/485
5	Canale utente 5 della porta 0 di Trajexia Tools
6	Canale utente 6 della porta 0 di Trajexia Tools
7	Canale utente 7 della porta 0 di Trajexia Tools

3.2.201 OUTLIMIT

Tipo Parametro degli assi

Sintassi **OUTLIMIT**

Descrizione Il limite di uscita circonda l'uscita richiesta, proveniente da un servomotore, a un valore inferiore a quello massimo. Il valore necessario varia in base all'uscita richiesta massima possibile. Se la tensione di uscita è generata da un DAC a 16 bit, un valore **OUTLIMIT** di 32767 produrrà l'intera gamma di tensione +/-10 V. La richiesta massima su un asse della velocità MECHATROLINK-II è di 32 bit.

Argomenti N/D

Esempio Nessun esempio.

Vedere anche **AXIS, S_REF, S_REF_OUT, SERVO**.

3.2.202 OV_GAIN

Tipo Parametro degli assi

Sintassi **OV_GAIN**

Descrizione	Il parametro OV_GAIN contiene il guadagno della velocità di uscita. Il contributo alla velocità d'uscita viene calcolato moltiplicando il cambio della posizione misurata per il valore del parametro OV_GAIN. Il valore predefinito è 0. Aggiungere un guadagno di velocità a un sistema equivale, dal punto di vista meccanico, ad aggiungere damping. Ciò può produrre una reazione più morbida e consentire l'uso di un guadagno proporzionale più alto di quello che potrebbe essere altrimenti usato; ciò nondimeno produce anche un maggior numero di errori seguenti. Valori elevati possono provocare oscillazioni e generare un'elevata quantità di errori seguenti. Nota: per evitare qualsiasi instabilità i guadagni dei servoazionamenti vanno cambiati solamente quando SERVO non è attivo.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	D_GAIN, I_GAIN, P_GAIN, VFF_GAIN.

3.2.203 P_GAIN

Tipo	Parametro degli assi
Sintassi	P_GAIN
Descrizione	Il parametro P_GAIN contiene il guadagno proporzionale. Il contributo proporzionale d'uscita viene calcolato moltiplicando l'errore di inseguimento per il valore del parametro P_GAIN. Il valore predefinito di P_GAIN per l'asse Mechatro Speed (ATYPE = 41) è 131072. Il valore predefinito del servo dell'asse flessibile (ATYPE = 44) è 1.0. Il guadagno proporzionale imposta la rigidità della risposta del servo. I valori troppo alti provocano oscillazioni. I valori troppo bassi producono errori seguenti di grosse dimensioni. Nota: per evitare qualsiasi instabilità i guadagni dei servoazionamenti vanno cambiati solamente quando SERVO non è attivo.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	D_GAIN, I_GAIN, OV_GAIN, VFF_GAIN.

3.2.204 PI

Tipo	Costante (sola lettura)
Sintassi	PI
Descrizione	La costante PI restituisce il valore numerico 3.1416.
Argomenti	N/D
Esempio	circonf = 100 PRINT "Raggio = ";circonf/(2*PI)
Vedere anche	N/D

3.2.205 PMOVE

Tipo	Parametro dei task (sola lettura)
Sintassi	PMOVE
Descrizione	Il parametro PMOVE contiene lo stato dei buffer dei task. Il parametro restituisce TRUE se i buffer dei task sono pieni e FALSE se sono vuoti. Quando il task esegue un comando di movimento, esso carica le informazioni relative nei buffer di movimento dei task. I buffer possono contenere un'unica istruzione di movimento per qualsiasi gruppo di assi. PMOVE viene impostato su TRUE quando il caricamento dei buffer è completato. Quando si verifica l'interruzione successiva del servoazionamento, il generatore di movimento carica le informazioni relative nel buffer dei movimenti successivi (NTYPE) dell'asse richiesto, se disponibili. Una volta completato questo secondo trasferimento, PMOVE viene portato a 0 fino a quando un altro movimento non viene eseguito nell'ambito del task. Ciascun task dispone di un parametro PMOVE proprio. Usare il modificatore PROC per accedere al parametro di un determinato task. Senza PROC viene preso in considerazione il task corrente.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	NTYPE, PROC.

3.2.206 POS_OFFSET

Tipo	Parametro di sistema
Sintassi	POS_OFFSET=valore
Descrizione	Per il funzionamento piezoelettrico. Questa parola chiave consente a un offset positivo di essere applicato al segnale di uscita DAC proveniente dall'anello servo. L'offset viene applicato dopo la funzione DAC_SCALE . Un offset di 327 rappresenta un offset di 0,1 volt. Si consiglia di usare un offset pari al 65 – 70% del valore necessario per far muovere lo stadio in una situazione ad anello aperto.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	N/D

3.2.207 POWER_UP

Tipo	Parametro di sistema
Sintassi	POWER_UP
Descrizione	Questo parametro viene utilizzato per determinare se all'avvio o al ripristino del software (EX) devono essere letti i programmi dalla memoria flash EPROM. Sono possibili due valori: 0: utilizza i programmi contenuti nella RAM 1 alimentata a batteria; 1: copia nella RAM i programmi contenuti nella memoria flash EPROM dell'unità di controllo. I programmi da eseguire all'avvio sono selezionati individualmente tramite il comando RUNTYPE . Note: POWER_UP è sempre un comando immediato e non può pertanto essere incluso in programmi. - Questo valore viene solitamente impostato da Trajexia Tools.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	EPROM

3.2.208 PRINT

Tipo	Comando di I/O
Sintassi	PRINT [#n,] espressione { , espressione } ? [#n,] espressione { , espressione }
Descrizione	Il comando PRINT invia una serie di caratteri alle porte seriali. PRINT può inviare parametri, stringhe ASCII fisse e caratteri ASCII singoli. Tramite PRINT #n , è possibile selezionare una qualsiasi porta verso la quale inviare le informazioni. È possibile collocare sulla stessa riga più elementi da stampare, separati da una virgola o da un punto e virgola. Un separatore a virgola all'interno del comando di stampa colloca una tabulazione tra gli elementi stampati. Il separatore a punto e virgola stampa l'elemento successivo senza interporre spazi tra gli elementi stampati. La larghezza del campo in cui è stampato un numero può essere impostata con l'uso di [w,x] dopo il numero da stampare. L'ampiezza della colonna è data da w mentre x fornisce il numero di posizioni decimali. Se si usa solo il parametro [x], viene utilizzata l'ampiezza predefinita e specificato il numero di posizioni decimali che devono essere stampate. All'interno del campo, i numeri sono allineati sulla destra e i caratteri iniziali non utilizzati sono sostituiti da spazi. Se il numero è troppo lungo, il campo viene riempito di asterischi per indicare che lo spazio è insufficiente per visualizzare il numero. L'ampiezza massima ammessa per il campo è di 127 caratteri. Per stampare un singolo carattere ASCII, è possibile usare il comando barra rovesciata \.
Argomenti	n Il dispositivo di uscita specificato. Quando questo argomento è omissivo, viene utilizzata la porta specificata da OUTDEVICE. Vedere tabella di seguito. espressione L'espressione da stampare.
Esempio	PRINT "Le LETTERE MAIUSCOLE e le lettere minuscole POSSONO ESSERE STAMPATE"
Esempio	Se VR(1) = 6 e variab = 1,5, viene stampato quando segue: PRINT 123.45,VR(1)-variab 123.4500 4.5000

- Esempio **lunghezza:**
PRINT "DISTANCE = ";mpos
DISTANCE = 123.0000
 In questo esempio viene utilizzato il separatore a punto e virgola.
 Nessuna tabulazione porta quindi alla colonna successiva, il che consente al programmatore maggiore libertà sul collocamento degli elementi di stampa.
- Esempio **PRINT VR(1)[4,1];variab[6,2]**
6.0 1.50
- Esempio **parametri:**
PRINT "DISTANCE = ";mpos[0]; " SPEED = ";v[2];
DISTANCE = 123 SPEED = 12.34
- Esempio **PRINT "ITEM ";totale" OF ";limite;CHR(13);**
- Esempio **>> PRINT HEX(15),HEX(-2)**
F FFFFA
- Vedere anche **\$ (DATI ESADECIMALI), OUTDEVICE.**

Valore	Descrizione
0	Porta di programmazione 0 (predefinita)
1	Porta seriale 1 di RS-232C
2	Porta seriale 2 di RS-422A/485
5	Canale utente 5 della porta 0 di Trajexia Tools
6	Canale utente 6 della porta 0 di Trajexia Tools
7	Canale utente 7 della porta 0 di Trajexia Tools

3.2.209 PROC

- Tipo Comando dei task
- Sintassi **PROC(numero_task)**
- Descrizione Il modificatore **PROC** consente la lettura o la scrittura del parametro di processo di un particolare processo. Se omissso, viene preso in considerazione il task corrente.
- Argomenti
- numero_task**
Il numero di task cui accedere.
- Esempio **WAIT UNTIL PMOVE PROC(3)=0**
- Vedere anche N/D

3.2.210 PROC_STATUS

- Tipo Parametro dei task
- Sintassi **PROC_STATUS**
- Descrizione Il parametro **PROC_STATUS** restituisce lo stato del processo o del task specificato. Il parametro viene usato con il modificatore **PROC** e può restituire i valori elencati nella tabella di seguito.
- Argomenti N/D
- Esempio **WAIT UNTIL PROC_STATUS PROC(3)=0**
- Vedere anche **PROCNUMBER, PROC.**

Valore	Descrizione
0	Processo arrestato
1	Processo in corso di esecuzione
2	Processo in corso di esecuzione passo-passo
3	Processo messo in pausa

3.2.211 PROCESS

Tipo	Comando di programma
Sintassi	PROCESS
Descrizione	Il comando PROCESS restituisce l'elenco degli stati di tutti i task in esecuzione con il rispettivo numero di task.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	HALT, RUN, STOP.

3.2.212 PROCNUMBER

Tipo	Parametro dei task (sola lettura)
Sintassi	PROCNUMBER
Descrizione	Il parametro PROCNUMBER contiene il numero del task in cui il programma attualmente selezionato viene eseguito. PROCNUMBER è spesso richiesto quando più copie di un programma sono eseguite su task diversi.
Argomenti	N/D
Esempio	MOVE(lunghezza) AXIS(PROCNUMBER)
Vedere anche	PROC_STATUS, PROC.

3.2.213 PROFIBUS

Tipo	Comando di sistema
Sintassi	PROFIBUS(numero_unità, 2, 1, VR_inizio_uscite, numero_uscite, VR_inizio_ingressi, numero_ingressi) PROFIBUS(numero_unità,4,0)
Descrizione	La funzione 2 di PROFIBUS configura TJ1-PRT per lo scambio di dati con il modulo master PROFIBUS-DP e definisce le aree della memoria VR dove si verifica lo scambio di I/O. La funzione 4 di PROFIBUS restituisce lo stato di scambio di dati di TJ1-PRT. Vedere la tabella per una descrizione dei bit contenuti nei canali di stato dello scambio di dati.
Argomenti	• numero_unità Specifica il numero d'unità di TJ1-PRT nel Trajexia. • VR_inizio_uscite L'indirizzo iniziale nella memoria VR dell'unità di controllo dove si trovano i dati di uscita del master PROFIBUS-DP. • numero_uscite Il numero di canali di uscita dal master PROFIBUS-DP della memoria VR. • VR_inizio_ingressi L'indirizzo iniziale nella memoria VR dell'unità di controllo dove si trovano i dati di ingresso per il master PROFIBUS-DP. • numero_ingressi Il numero di canali di ingresso nel master PROFIBUS-DP della memoria VR.
Esempio	PROFIBUS (0,2,1,10,16,150,31) In questo esempio, TJ1-PRT è configurato per scambiare dati con il master PROFIBUS-DP attraverso 16 canali di uscita (ricevuti dal master) situati tra VR(10) e VR(25) e 31 canali d'ingresso (inviati al master) situati tra VR(150) e VR(180).
Vedere anche	N/D

Bit	Valore	Descrizione
0	0	Impossibile effettuare la configurazione dello scambio di dati di I/O
	1	Configurazione dello scambio di dati di I/O riuscita
1	0	Dati di I/O non disponibili
	1	Dati di I/O disponibili
2	0	Scambio dati attivo in modalità OPERATE
	1	Scambio dati attivo in modalità CLEAR

3.2.214 PSWITCH

Tipo Comando di I/O

Sintassi **PSWITCH(interruttore, abilita [, asse, numero_uscita, stato_uscita, posizione_impostata, posizione_ripristinata])**

Descrizione Il comando **PSWITCH** attiva un'uscita quando viene raggiunta una posizione predefinita e disattiva l'uscita quando viene raggiunta una seconda posizione. Le posizioni sono specificate come posizioni assolute misurate. Esistono 16 selettori di posizione, ciascuno dei quali può essere assegnato a un asse qualsiasi. Ciascun selettore dispone di posizioni on ("acceso") e off ("spento") e numero di uscita esclusivi. Il comando può essere utilizzato con 2 degli argomenti o con tutti e 7. È possibile disabilitare un selettore utilizzando solo 2 argomenti. I **PSWITCH** vengono calcolati per ogni ciclo di servozionamento e il risultato restituito viene applicato all'hardware. Il tempo di risposta è pertanto pari a circa 1 ciclo di servozionamento. Nota: un'uscita può rimanere attiva se era già attiva al momento in cui il comando **PSWITCH** è stato spento. Per attivare o disattivare un'uscita, è possibile utilizzare il comando **OP** come segue:
PSWITCH(2,OFF) OP(14,OFF) ' Spegne (OFF) il pswitch che controlla OP 14
 Nota: gli interruttori fisici utilizzati con **PSWITCH** non sono interruttori hardware a funzionamento rapido; l'apertura e la chiusura dell'interruttore vengono pertanto effettuate dal software e ciò può produrre dei piccoli ritardi di funzionamento. Gli interruttori hardware a funzionamento rapido possono essere utilizzati solo con gli assi collegati tramite TJ1-FL02. Utilizzare il comando **HW_PSWITCH**.

Argomenti

- **interruttore**
Il numero dell'interruttore. Intervallo: [0,15].
- **abilita**
L'abilitazione dell'interruttore. Intervallo: [on, off].
- **asse**
Il numero dell'asse che fornisce l'ingresso della posizione.
- **numero_uscita**
L'uscita fisica da impostare. Intervallo: [8,31].
- **stato_uscita**
Lo stato dell'uscita. Intervallo: [on, off].
- **posizione_impostata**
La posizione assoluta espressa in unità utente, alla quale viene impostata l'uscita.
- **posizione_ripristinata**
La posizione assoluta espressa in unità utente, alla quale viene ripristinata l'uscita.

Esempio Un albero rotante è dotato di un interruttore a camme che deve essere cambiato per la lavorazione di pezzi di diverse dimensioni. Sull'albero è presente anche un interruttore di prossimità che indica il TDC della macchina. Se si utilizza una camma meccanica, il passaggio da un tipo lavoro a un altro richiede un elevato dispendio di tempo. La cosa può essere resa più facile utilizzando PSWITCH come interruttore a camme software. L'interruttore di prossimità è collegato all'ingresso 7 mentre l'uscita è l'uscita 11. L'albero viene controllato dall'asse 0. Il motore dispone di un encoder a 900 impulsi per rivoluzione. L'uscita deve essere attiva a partire da 80 unità. **PSWITCH** utilizza il fattore di conversione delle unità per consentire l'impostazione delle posizioni in unità idonee. Come prima cosa, occorre calcolare e impostare il fattore di conversione delle unità. Ciascun impulso su un encoder fornisce quattro margini per il conteggio di TJ1-MC___. Vi sono pertanto 3.600 fronti/giro ovvero 10 margini/grado. Impostando il fattore di conversione delle unità su 10, risulta possibile lavorare in gradi. Successivamente occorre determinare un valore per tutti gli argomenti **PSWITCH**.

sw: il numero dell'interruttore che può essere qualsiasi interruttore inutilizzato. In questo esempio, viene utilizzato il numero 0.

en: per funzionare, l'interruttore deve essere abilitato; impostare questo argomento su 1.

asse: l'albero è controllato dall'asse 0.

opno: l'uscita posta sotto controllo è l'uscita 11.

opst: l'uscita deve essere attiva; impostare pertanto su 1.

setpos: l'uscita viene generata a 80 unità.

rspos: l'uscita deve essere attiva per un periodo di 120 unità.

Tutti gli elementi di cui sopra possono essere integrati nelle righe di codice BASIC illustrate di seguito:

interruttore:

```

UNITS AXIS(0) = 10 ' Imposta il fattore di conversione delle unità
REPDIST = 360
REP_OPTION = ON
PSWITCH(0,ON,0,11,ON,80,200)

```

Questo programma utilizza la distanza di ripetizione impostata a 360 gradi e l'opzione di ripetizione attivata, affinché la posizione dell'asse venga mantenuta tra 0 e 360 gradi.

Vedere anche **(HW_PSWITCH), OP, UNITS**.

3.2.215 RAPIDSTOP

Tipo	Comando degli assi
Sintassi	RAPIDSTOP RS
Descrizione	Il comando RAPIDSTOP annulla su tutti gli assi il movimento corrente proveniente dal buffer del movimento corrente (MTYPE). I movimenti dettati dai comandi dei movimenti dotati di profilo di velocità (MOVE, MOVEABS, MOVEMODIFY, FORWARD, REVERSE, MOVECIRC e MHELICAL) subiranno una decelerazione fino all'arresto, alla velocità di decelerazione impostata dal parametro DECEL . I movimenti degli altri comandi saranno immediatamente arrestati. Note: RAPIDSTOP cancella soltanto i movimenti in corso di esecuzione. Se nei buffer dei movimenti successivi (NTYPE) o nei buffer dei task sono contenuti altri movimenti saranno caricati a quel punto. - Durante il processo di decelerazione dei movimenti correnti, eventuali RAPIDSTOP aggiuntivi saranno ignorati.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	CANCEL, MTYPE, NTYPE .

3.2.216 READ_BIT

Tipo	Comando di sistema
Sintassi	READ_BIT(numero_bit, numero_vr)
Descrizione	Il comando READ_BIT restituisce il valore del bit specificato nella variabile VR specificata, vale a dire 0 oppure 1.
Argomenti	numero_bit Il numero del bit da leggere. Intervallo: [0,23]. numero_vr Il numero della variabile VR per la quale il bit è letto. Intervallo: [0,1023].
Esempio	Nessun esempio.
Vedere anche	CLEAR_BIT, SET_BIT .

3.2.217 REG_POS

Tipo	Parametro degli assi (sola lettura)
Sintassi	REG_POS
Descrizione	Il parametro REG_POS memorizza, in unità utente, la posizione in cui si è verificato l'evento di registrazione primario.
Argomenti	N/D
Esempio	PRINT REG_POS AXIS(2) Questa istruzione stampa la posizione memorizzata per l'asse 2, espressa in unità utente.
Vedere anche	AXIS, MARK, REGIST.

3.2.218 REG_POSB

Tipo	Parametro degli assi (sola lettura)
Sintassi	REG_POSB
Descrizione	Il parametro REG_POSB memorizza, in unità utente, la posizione in cui si è verificato l'evento di registrazione secondario.
Argomenti	N/D
Esempio	PRINT REG_POSB AXIS(2) Questa istruzione stampa la posizione memorizzata per l'asse 2, espressa in unità utente.
Vedere anche	AXIS, MARKB, REGIST.

3.2.219 REGIST

Tipo	Comando degli assi
Sintassi	REGIST(modalità)
Descrizione	Il comando REGIST imposta l'operazione di registrazione. Il comando acquisisce la posizione di un asse quando viene individuato un segnale di registrazione. Con un TJ1-FL02 l'acquisizione viene effettuata dall'hardware, per evitare che i ritardi del software possano influenzare la precisione della posizione acquisita. Con un asse MECHATROLINK-II, l'acquisizione viene eseguita dal servozionamento.

Un comando **REGIST** può acquisire due posizioni di registrazione utilizzando ingressi di registrazione separati. Quando si verifica un evento di registrazione primario, il parametro degli assi **MARK** viene impostato su "attivo" e la posizione memorizzata nel parametro degli assi **REG_POS**. Per l'evento di registrazione secondario, il parametro degli assi **MARKB** viene impostato su "attivo" e la posizione memorizzata nel parametro degli assi **REG_POSB**. Questo comando può essere applicato solo agli assi definiti come assi flessibili con valori **ATYPE** 43, 44 e 45.

La registrazione MECHATROLINK-II può essere eseguita utilizzando il marker Z dell'encoder oppure gli ingressi di registrazione esterna EXT1, EXT2 o EXT3. A differenza di quanto accade con gli assi flessibili, è possibile l'acquisizione di una sola registrazione. Quando si verifica un evento di registrazione, il parametro degli assi **MARK** viene impostato su "attivo" e la posizione memorizzata nel parametro degli assi **REG_POS**. Il comando **REGIST** consente la memorizzazione di un evento di registrazione sullo specifico ingresso di registrazione. Quando si verifica un evento di registrazione, il parametro degli assi **MARK** viene impostato su "attivo" e la posizione memorizzata nel parametro degli assi **REG_POS**.

I segnali di registrazione EXT1, EXT2 e EXT3 devono essere allocati agli ingressi CN1 tramite il parametro del driver Pn511. Ad esempio, Pn511=654x imposta il collegamento EXT1 su CN1 pin44, il collegamento EXT2 su CN1 pin45 e il collegamento EXT3 su CN1 pin46.

La tabella di seguito mostra come configurare separatamente gli ingressi esterni. Nota: per configurare i segnali di registrazione EXT1, EXT2 e EXT3 vengono utilizzati rispettivamente i numeri di parametro Pn511.1, Pn511.2 e Pn511.3. Pn511.0 non è utilizzato. Per maggiori informazioni, fare riferimento al manuale dell'utente del servozionamento.

Segnale di registrazione	Numero del parametro	Valore del parametro	Descrizione
EXT 1	Pn511.1	Da 0 a 3	Non utilizzato
		4	Ingresso da CN1 pin44 (fronte di salita)
		5	Ingresso da CN1 pin45 (fronte di salita).
		6	Ingresso da CN1 pin46 (fronte di salita).
		7	Segnale sempre OFF (spento).
		8	Segnale sempre ON (acceso).
		Da 9 a C	Non utilizzato
		D	Ingresso da CN1 pin44 (fronte di discesa).
		E	Ingresso da CN1 pin45 (fronte di discesa).
		F	Ingresso da CN1 pin46 (fronte di discesa).
EXT 2	Pn511.2	Come per EXT 1	Come per EXT 1
EXT 3	Pn511.3	Come per EXT 1	Come per EXT 1

Una funzione di windowing inclusivo consente la registrazione solo all'interno di una finestra specifica di posizioni degli assi. Grazie a questa funzione di windowing, gli eventi di registrazione sono ignorati se la posizione misurata dell'asse non è maggiore del parametro degli assi **OPEN_WIN** e minore del parametro **CLOSE_WIN**.

Una funzione di windowing esclusivo consente la registrazione solo al di fuori di una finestra specifica di posizioni degli assi. Grazie a questa funzione di windowing, gli eventi di registrazione sono ignorati se la posizione misurata dell'asse non è minore del parametro degli assi **OPEN_WIN** e maggiore del parametro **CLOSE_WIN**.

Argomenti

- modalità**

Il parametro della modalità specifica l'ingresso e l'evento di registrazione da usare, nonché il fronte del segnale in corrispondenza del quale avviene l'evento di registrazione. Il parametro della modalità specifica anche l'utilizzo della funzione di windowing e del filtering.

Il parametro "modalità" agisce diversamente in MECHATROLINK-II e nell'asse flessibile. Le funzioni per ciascun bit del parametro "modalità" sono descritte nelle tabelle di seguito.

Esempio **REGIST(4 + 1) AXIS (1)**

Questo comando seleziona l'evento di registrazione primario che si verifica sul fronte di salita del segnale di ingresso REG 0 per l'asse 1.

Esempio **REGIST(48+64+128+512+1024) AXIS(2)**

Questo comando seleziona l'evento di registrazione secondario che si verifica sul fronte di discesa del segnale di ingresso AUX IN con funzioni di windowing esclusivo e filtering per l'asse 2.

Vedere anche **AXIS, MARK, MARKB, REG_POS, REG_POSB, OPEN_WIN, CLOSE_WIN**.

Bit	Funzione (asse flessibile)
1, 0	Una registrazione primaria si verifica per: 00: Marker Z dell'encoder 01: Ingresso REG 0 10: Ingresso REG 1 11: Ingresso AUX IN
2	Impostare questo bit per utilizzare un evento di registrazione primario
3	Un evento di registrazione primario si verifica in corrispondenza con il segnale: 0: fronte di salita 1: fronte di discesa
5, 4	Una registrazione secondaria si verifica per: 00: marker Z dell'encoder 01: ingresso REG 0 10: ingresso REG 1 11: ingresso AUX IN
6	Impostare questo bit per utilizzare un evento di registrazione secondario
7	Un evento di registrazione secondario si verifica in corrispondenza con il segnale: 0: fronte di salita 1: fronte di discesa
9, 8	Scelta della funzione di windowing: 00: Nessun windowing 01: windowing inclusivo 10: windowing esclusivo 11: windowing esclusivo
10	Impostare questo bit per utilizzare la funzione di filtering

Bit	Funzione (MECHATROLINK-II)
1, 0	Una registrazione primaria si verifica per: 00: marker Z dell'encoder 01: ingresso EXT1 (CN1 pin programmato con Pn511.1) 10: ingresso EXT2 (CN1 pin programmato con Pn511.2) 11: ingresso EXT3 (CN1 pin programmato con Pn511.3)
2 – 7	Non utilizzato
9, 8	Scelta della funzione di windowing: 00: Nessun windowing 01: windowing inclusivo 10: windowing inclusivo 11: windowing esclusivo
10	Non utilizzato

3.2.220 REMAIN

Tipo	Parametro degli assi (sola lettura)
Sintassi	REMAIN
Descrizione	Il parametro REMAIN contiene la distanza rimanente fino alla fine del movimento corrente. Controllando questo parametro è possibile vedere il grado di completamento del movimento. REMAIN viene definito in unità utente.
Argomenti	N/D
Esempio	Per abbassare la velocità a 5 mm dal termine di un movimento. avvio: SPEED = 10 MOVE(45) WAIT UNTIL REMAIN < 5 SPEED = 1 WAIT IDLE
Vedere anche	AXIS, UNITS

3.2.221 REMOTE_ERROR

Tipo	Parametro degli assi
Sintassi	REMOTE_ERROR
Descrizione	Restituisce il numero di errori sul collegamento di comunicazione digitale di un driver.
Argomenti	N/D
Esempio	>>PRINT REMOTE_ERROR 1.0000
Vedere anche	N/D

3.2.222 RENAME

Tipo	Comando di programma
Sintassi	RENAME "vecchio_nome_programma" "nuovo_nome_programma"
Descrizione	Il comando RENAME cambia il nome di un programma nella directory TJ1-MC__. I nomi dei programmi possono essere specificati anche senza virgolette. Nota: questo comando è implementato per un terminale offline (VT100). All'interno di Trajexia Tools gli utenti possono selezionare il comando dal menu Program .
Argomenti	vecchio_nome_programma Il nome corrente del programma. nuovo_nome_programma Il nuovo nome del programma.
Esempio	RENAME "car" "voiture"
Vedere anche	COPY, DEL, NEW.

3.2.223 REP_DIST

Tipo	Parametro degli assi
Sintassi	REP_DIST
Descrizione	Il parametro REP_DIST contiene la distanza ripetuta, vale a dire l'intervallo di movimento ammesso per un asse prima che la posizione richiesta (DPOS) e la posizione misurata (MPOS) siano corrette. REP_DIST viene definito in unità utente. L'intervallo esatto è controllato dal parametro REP_OPTION. Il parametro REP_DIST può avere qualsiasi valore positivo diverso da 0. Quando la posizione misurata ha raggiunto il proprio limite, TJ1-MC__ regola le posizioni assolute senza influenzare il movimento in corso o l'algoritmo del servozionamento. Notare che la posizione richiesta può ricadere fuori dell'intervallo perché la posizione misurata viene utilizzata per avviare la regolazione. Per ogni evento (DEFPOS, OFFPOS, MOVEABS, MOVEMODIFY) che definisce una posizione fuori dell'intervallo, la posizione finale sarà ridefinita all'interno dell'intervallo. Il valore predefinito per tutti gli assi è 5000000.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	AXIS , DPOS , MPOS , REP_OPTION , UNITS .

3.2.224 REP_OPTION

Tipo	Parametro degli assi
Sintassi	REP_OPTION
Descrizione	Il parametro REP_OPTION controlla l'applicazione del parametro degli assi REP_DIST nonché l'opzione di ripetizione dei comandi degli assi CAMBOX e MOVELINK. Il valore predefinito è 0. Vedere tabella di seguito.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	AXIS , CAMBOX , MOVELINK , REP_DIST .

Bit	Descrizione
0	L'intervallo della distanza ripetuta è controllato dal bit 0 del parametro REP_OPTION. - Se il bit 0 di REP_OPTION è "non attivo", l'intervallo delle posizioni richieste e misurate sarà compreso tra -REP_DIST e REP_DIST. - Se il bit 0 di REP_OPTION è "attivo", l'intervallo delle posizioni richieste e misurate sarà compreso tra 0 e REP_DIST.
1	L'opzione di ripetizione automatica dei comandi CAMBOX e MOVELINK è controllata dal bit 1 del parametro REP_OPTION. L'impostazione del bit è "attivo" per richiedere al software di sistema di terminare l'opzione di ripetizione automatica. Dopo avere impostato l'opzione su "non attivo", il software di sistema azzera automaticamente il bit 1 di REP_OPTION.

3.2.225 REPEAT..UNTIL

Tipo	Comando di controllo del programma.
Sintassi	REPEAT comandi UNTIL condizione

Descrizione	La struttura REPEAT ... UNTIL consente la ripetizione per un dato numero di volte del segmento di programma compreso tra REPEAT e l'istruzione UNTIL , fino a quando la condizione non diventa TRUE . Nota: la struttura REPEAT ... UNTIL può essere nidificata indefinitamente.
Argomenti	• comandi Qualsiasi insieme valido di comandi BASIC • condizione Qualsiasi valida espressione logica in BASIC
Esempio	Un nastro trasportatore deve indicizzare 100 mm alla velocità di 1000 mm/s, attendere 0,5 s e quindi ripetere il ciclo fino a quando un contatore esterno non induce l'arresto attivando l'ingresso 4. ciclo: SPEED = 1000 REPEAT MOVE(100) WAIT IDLE WA(500) UNTIL IN(4) = ON
Vedere anche	FOR..TO..STEP..NEXT , WHILE..WEND .

3.2.226 RESET

Tipo	Comando di sistema
Sintassi	RESET
Descrizione	Il comando RESET imposta su 0 il valore di tutte le variabili locali del task BASIC corrente.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	CLEAR .

3.2.227 RETURN

Vedere **GOSUB..RETURN**.

3.2.228 REV_IN

Tipo	Parametro degli assi
Sintassi	REV_IN
Descrizione	Il parametro degli assi REV_IN contiene il numero d'ingresso da utilizzare come ingresso del limite di marcia indietro. Il numero può essere impostato tra 0 e 7 e su 19. L'intervallo d'ingresso valido è compreso tra 0 e 31. I valori tra 0 e 15 rappresentano gli ingressi fisicamente presenti sul connettore di I/O di TJ1-MC__ e sono comuni a tutti gli assi. I valori da 16 a 27 rappresentano ingressi software che possono essere liberamente utilizzati da programmi e comandi quali IN e OP. Sono inoltre valori comuni a tutti gli assi. I valori da 28 a 31 vengono direttamente mappati agli ingressi del servozionamento presenti sul connettore CN1 e sono univocamente associati a ogni asse. La mappatura degli ingressi del servozionamento agli ingressi da 28 a 31 dipende dall'impostazione del parametro di servozionamento Pn81E. L'impostazione consigliata è Pn81E = 0x4321, con la seguente mappatura. Se viene impostato un numero d'ingresso e il limite viene raggiunto, qualsiasi movimento di marcia indietro su quell'asse verrà arrestato. Inoltre sarà impostato il bit 5 del parametro degli assi AXISSTATUS . Nota: questo ingresso è active low.
Sigma II	• ingresso 28: CN1-40 • ingresso 29: CN1-41 • ingresso 30: CN1-42 • ingresso 31: CN1-43
Sigma III	• ingresso 28: CN1-13 • ingresso 29: CN1-7 • ingresso 30: CN1-8 • ingresso 31: CN1-9
Junma	• ingresso 26: CN1-2 • ingresso 27: CN1-1
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	AXIS , AXISSTATUS , FWD_IN .

3.2.229 REV_JOG

Tipo	Parametro degli assi
Sintassi	REV_JOG
Descrizione	Il parametro degli assi REV_JOG contiene il numero d'ingresso da utilizzare come ingresso di marcia indietro del jog. L'ingresso può essere compreso tra 0 e 7. Come impostazione predefinita, il parametro è impostato su -1 e nessun ingresso è selezionato. Note: questo ingresso è active low.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	AXIS, FAST_JOG, FWD_JOG, JOGSPEED, UNITS.

3.2.230 REVERSE

Tipo	Comando degli assi
Sintassi	REVERSE RE
Descrizione	Il comando REVERSE sposta indietro un asse costantemente alla velocità impostata nel parametro degli assi SPEED . La velocità di accelerazione è definita dal parametro degli assi ACCEL . REVERSE agisce sull'asse di base predefinito (impostato con BASE) a meno che AXIS non sia usato per specificare un asse di base temporaneo. Nota: il movimento di marcia indietro può essere arrestato eseguendo i comandi CANCEL o RAPIDSTOP oppure raggiungendo il limite di marcia indietro, inibizione o di ritorno all'origine.
Argomenti	N/D
Esempio	indietro: REVERSE WAIT UNTIL IN(0) = ON ' Aspetta il segnale di arresto CANCEL
Vedere anche	AXIS, CANCEL, FORWARD, RAPIDSTOP.

3.2.231 RS_LIMIT

Tipo	Parametro degli assi
Sintassi	RS_LIMIT RSLIMIT
Descrizione	Il parametro degli assi RS_LIMIT contiene la posizione assoluta del limite software di marcia indietro espresso in unità utente. Un limite software per il movimento di marcia indietro può essere impostato a partire dal programma per controllare la gamma di utilizzo del macchinario. Una volta raggiunto il limite, TJ1-MC__ decelera fino a raggiungere lo 0, quindi cancella il movimento. Il bit 10 del parametro degli assi AXISSTATUS viene attivato fintantoché la posizione dell'asse rimane minore di/inferiore a RS_LIMIT .
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	AXIS, FS_LIMIT, UNITS.

3.2.232 RUN

Tipo	Comando di programma
Sintassi	RUN ["nome_programma" [, numero_task]]
Descrizione	Il comando RUN esegue in TJ1-MC__ il programma specificato con nome_programma . RUN seguito dalla specificazione del nome del programma produrrà l'esecuzione del programma correntemente selezionato. Il nome del programma può essere specificato anche senza virgolette. Indicando il numero di task si specifica il numero di task su cui il programma sarà eseguito. Omettendo il numero di task, il programma viene eseguito sul numero di task più elevato disponibile. È possibile includere RUN all'interno di un programma per eseguire un altro programma. Nota: l'esecuzione prosegue fino a quando non si verifica una delle seguenti condizioni: • Non ci sono più righe da eseguire. • Sulla riga di comando viene digitato HALT, arrestando tutti i programmi. • Sulla riga di comando viene digitato STOP, arrestando un singolo programma. • Nel programma si incontra il comando STOP. • Si incontra un errore di run-time.

Argomenti	nome_programma Qualsiasi nome di programma valido. numero_task Qualsiasi numero di task valido. Intervallo: [1,14].
Esempio	<pre>>> SELECT "PROGRAM" PROGRAM selezionato >> RUN</pre> Questo esempio esegue il programma correntemente selezionato.
Esempio	<pre>RUN "salsiccia"</pre> Questo esempio esegue il programma denominato salsiccia.
Esempio	<pre>RUN "salsiccia",3</pre> Questo esempio esegue il programma denominato salsiccia sul task 3.
Vedere anche	HALT, STOP.

3.2.233 RUN_ERROR

Tipo	Parametro dei task (sola lettura)
Sintassi	RUN_ERROR
Descrizione	Il parametro RUN_ERROR contiene il numero dell'ultimo errore BASIC di run-time che si è verificato nel task specificato. Ciascun task dispone di un parametro RUN_ERROR proprio. Usare il modificatore PROC per accedere al parametro di un determinato task. Senza PROC viene preso in considerazione il task corrente.
Argomenti	N/D
Esempio	<pre>>> PRINT RUN_ERROR PROC(5) 9.0000</pre>
Vedere anche	BASICERROR, ERROR_LINE, PROC.

3.2.234 RUNTYPE

Tipo	Comando di programma
Sintassi	RUNTYPE "nome_programma", esecuzione_automatica [, numero_task]
Descrizione	Il comando RUNTYPE determina se il programma specificato da nome_programma viene eseguito automaticamente all'avvio e, in caso affermativo, su quale task. Il numero di task è opzionale: omettendolo, il programma viene eseguito sul numero di task più elevato disponibile. Eseguendo un comando DIR viene visualizzato lo stato RUNTYPE corrente di ciascun programma. Se un programma contiene errori di compilazione, nessun programma sarà eseguito all'avvio. Per impostare il comando RUNTYPE tramite Trajexia Tools, selezionare Set Power-up mode dal menu Program. Nota: l'esecuzione del comando EPROM è richiesta per la registrazione delle nuove impostazioni RUNTYPE nella memoria flash, poiché in caso contrario tutte le nuove impostazioni andranno perse al momento dello spegnimento.
Argomenti	nome_programma Il nome del programma il cui RUNTYPE viene impostato. esecuzione_automatica 0 = esecuzione manuale su comando; 1 = esecuzione automatica all'avvio. Tutti i valori diversi da 0 sono considerati come 1. numero_task Il numero del task su cui eseguire il programma. Intervallo: [1, 14].
Esempio	<pre>>> RUNTYPE nomeprog,1,3</pre> Questa riga imposta l'esecuzione automatica all'avvio del programma nomeprog sul task 3.
Esempio	<pre>>> RUNTYPE nomeprog,0</pre> Questa riga imposta l'esecuzione manuale del programma nomeprog.
Vedere anche	AUTORUN, EPROM, EX.

3.2.235 S_REF

Tipo	Parametro degli assi
Sintassi	DAC S_REF
Descrizione	Questo parametro contiene il valore della velocità di riferimento che viene applicata direttamente al servozionamento quando l'asse è in modalità ad anello aperto (SERVO=OFF). L'intervallo di questo parametro è definito dal numero di bit disponibili. Per gli assi MECHATROLINK, S_REF prende 32 bit, pertanto l'intervallo disponibile è [-2147483648, 2147483648], pari a un intervallo di tensione [-10 V, 10 V]. Per gli assi flessibili, S_REF prende 16 bit, pertanto l'intervallo disponibile è [-32768, 32767], pari a un intervallo di tensione [-10 V, 10 V]. Questi intervalli possono essere limitati utilizzando il parametro OUTLIMIT. La velocità di riferimento effettiva dipende dal servomotore. Per determinare la velocità di riferimento in giri al minuto (RPM), moltiplicare il valore del parametro per il valore del parametro S_RATE. Il valore correntemente utilizzato dal driver può essere letto utilizzando il parametro degli assi S_REF_OUT.
Argomenti	N/D
Esempio	WDOG = ON SERVO = OFF onda quadra: S_REF AXIS(0) = 2000 WA(250) S_REF AXIS(0) = -2000 WA(250) GOTO onda quadra Queste righe possono essere usate per forzare un'onda quadra di movimento positivo e negativo di circa 500 ms sull'asse 0.
Vedere anche	AXIS, S_REF_OUT, OUTLIMIT, SERVO.

3.2.236 S_REF_OUT

Tipo	Parametro degli assi (sola lettura)
Sintassi	DAC_OUT S_REF_OUT
Descrizione	Il parametro S_REF_OUT contiene il valore della velocità di riferimento che viene applicata al servozionamento sia in modalità ad anello chiuso che in modalità ad anello aperto. In modalità ad anello chiuso (SERVO=ON), l'algoritmo di controllo del movimento emetterà un segnale per la velocità di riferimento, determinato dalle impostazioni del guadagno di controllo e dall'errore di inseguimento. La posizione del servomotore viene determinata utilizzando i comandi degli assi. In modalità ad anello aperto (SERVO=OFF), il segnale della velocità di riferimento è determinato dal parametro degli assi S_REF. La velocità di riferimento effettiva dipende dal servomotore. Per determinare la velocità di riferimento in giri al minuto (RPM), moltiplicare il valore del parametro S_REF per il valore del parametro S_RATE.
Argomenti	N/D
Esempio	>> PRINT S_REF_OUT AXIS(0) 288.0000
Vedere anche	AXIS, S_REF, OUTLIMIT, SERVO.

3.2.237 SCOPE

Tipo	Comando di sistema
Sintassi	SCOPE(controllo, periodo, inizio_tabella, fine_tabella, P0 [, P1 [, P2 [, P3]])
Descrizione	Il comando SCOPE programma la memorizzazione automatica da parte del sistema di un massimo di 4 parametri per ogni periodo di campionamento. La memorizzazione dei dati inizia non appena viene eseguito il comando TRIGGER. Il periodo di campionamento può essere un qualsiasi multiplo del ciclo di servoazionamento. I parametri sono memorizzati nella matrice della memoria TABLE e possono essere successivamente letti e trasmessi a un computer e visualizzati sull'oscilloscopio di Trajexia Tools oppure scritti in un file attraverso l'opzione Create Table file del menu File, per essere sottoposti a ulteriore analisi. È possibile leggere la posizione corrente nella memoria TABLE del primo parametro, che è scritto da SCOPE, sul parametro SCOPE_POS. Note: 1.Quando esegue la funzione di oscilloscopio, Trajexia Tools utilizza il comando SCOPE. 2.Per ridurre al minimo il tempo richiesto per scrivere i dati in tempo reale, il comando SCOPE scrive dati grezzi nella matrice della memoria TABLE. Ad esempio - a)I parametri sono espressi in fronti dell'encoder (al secondo) e pertanto non sono compensati per il fattore di conversione UNITS. - b)Il parametro MSPEED è espresso in cambiamenti nei fronti dell'encoder per ciclo di servoazionamento. 3.Applicazioni come i comandi CAM, CAMBOX e SCOPE utilizzano tutte la stessa memoria TABLE come area dati.

Argomenti	• controllo Impostare su "attivo" o "non attivo" per controllare l'esecuzione del comando SCOPE. Se attivo, il comando SCOPE è pronto ad essere eseguito non appena viene eseguito il comando TRIGGER. • periodo Il numero di tempi di servoazionamento tra un campionamento di dati e l'altro. • inizio_tabella L'indirizzo del primo elemento da cui iniziare a memorizzare i dati nella matrice della memoria TABLE. • fine_tabella L'indirizzo dell'ultimo elemento da utilizzare nella matrice della memoria TABLE. • P0 Primo parametro da memorizzare. • P1 Secondo parametro da memorizzare (opzionale). • P2 Terzo parametro da memorizzare (opzionale). • P3 Quarto parametro da memorizzare (opzionale).
Esempio	SCOPE(ON,10,0,1000,MPOS AXIS(1),DPOS AXIS(1)) Questo esempio programma la funzione SCOPE per memorizzare il parametro MPOS per l'asse 1 e il parametro DPOS per l'asse 1 ogni 10 cicli di servoazionamento. Il parametro MPOS viene memorizzato nelle posizioni da 0 a 499 della memoria TABLE; i parametri DPOS, invece, nelle posizioni da 500 a 999. La funzione SCOPE torna all'inizio e comincia a memorizzare, salvo il caso in cui non venga arrestata. Il campionamento non ha inizio fino a quando non viene eseguito il comando TRIGGER.
Esempio	SCOPE(OFF) Questa riga disattiva la funzione SCOPE.
Vedere anche	SCOPE_POS , TABLE , TRIGGER .

3.2.238 SCOPE_POS

Tipo	Parametro di sistema (sola lettura)
Sintassi	SCOPE_POS
Descrizione	Il parametro SCOPE_POS contiene la posizione corrente all'interno della memoria TABLE dove il comando SCOPE sta attualmente memorizzando il primo parametro.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	SCOPE .

3.2.239 SELECT

Tipo	Comando di programma
Sintassi	SELECT "nome_programma"
Descrizione	Il comando SELECT specifica qual è il programma corrente da sottoporre a modifica, esecuzione, elenco ecc. SELECT crea un nuovo programma se il nome inserito non esiste. Il nome del programma può essere specificato anche senza virgolette. Quando un programma viene selezionato, i comandi COMPILE, DEL, EDIT, LIST, NEW, RUN, STEPLINE, STOP e TROFF diventano applicabili a quel programma, a meno che un altro programma non sia specificato nella riga di comando. Quando viene selezionato un altro programma, il programma selezionato precedentemente viene compilato. Il programma selezionato non può esser cambiato quando un programma è in corso di esecuzione. Nota: questo comando è implementato per un terminale offline (VT100). Trajexia Tools seleziona automaticamente i programmi selezionati sull'elenco del pannello di controllo.
Argomenti	N/D
Esempio	<pre>>> SELECT "PROGRAM" PROGRAM selezionato >> RUN</pre>
Vedere anche	COMPILE , DEL , EDIT , LIST , NEW , RUN , STEPLINE , STOP , TROFF .

3.2.240 SERVO

Tipo	Parametro degli assi
Sintassi	SERVO
Descrizione	Il parametro SERVO determina se l'asse di base viene azionato dal controllo servo (SERVO=ON) o in modalità ad anello aperto (SERVO=OFF). In modalità ad anello chiuso, l'algoritmo di controllo del movimento emetterà un segnale per la velocità di riferimento, determinato dalle impostazioni del guadagno di controllo e dall'errore di inseguimento. La posizione del servomotore viene determinata utilizzando i comandi degli assi. In modalità ad anello aperto, il segnale della velocità di riferimento è determinato nella sua totalità dal parametro degli assi S_REF.
Argomenti	N/D
Esempio	SERVO AXIS(0) = ON ' L'asse 0 è azionato dal controllo servo SERVO AXIS(1) = OFF ' L'asse 1 è azionato in modalità ad anello aperto
Vedere anche	AXIS , FE_LIMIT , S_REF , S_REF_OUT , WDOG .

3.2.241 SERVO_PERIOD

Tipo	Parametro di sistema
Sintassi	SERVO_PERIOD
Descrizione	Il parametro SERVO_PERIOD imposta il ciclo di servoazionamento di TJ1-MC__. La temporizzazione dell'esecuzione dei task di programma e l'aggiornamento dei dati di controllo e dell'I/O dell'unità dipendono tutti da questa impostazione. Il parametro è definito in microsecondi. TJ1-MC__ può essere impostato su cicli di servoazionamento di 0,5, 1,0 o 2,0 ms. Vedere tabella di seguito.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	DRIVE_RESET .

Valore	Descrizione
500	0,5 ms
1000	1,0 ms
2000	2,0 ms

**Attenzione**

Quando il parametro è stato impostato, occorre operare un'interruzione dell'alimentazione o un reset software (utilizzando **EX**) sull'intero sistema. Disattendere queste precauzioni potrebbe essere causa di anomalie.

3.2.242 SET_BIT

Tipo	Comando di sistema
Sintassi	SET_BIT(numero_bit, numero_vr)
Descrizione	Il comando SET_BIT imposta su 1 il valore del bit specificato nella variabile VR. Gli altri bit della variabile mantengono i rispettivi valori.
Argomenti	• numero_bit Il numero del bit da impostare. Intervallo: [0,23]. • numero_vr Il numero della variabile VR per la quale il bit è impostato. Intervallo: [0,1023].
Esempio	Nessun esempio.
Vedere anche	CLEAR_BIT, READ_BIT, VR.

3.2.243 SETCOM

Tipo	Comando di comunicazione
Sintassi	SETCOM(velocità_trasmissione, bit_dati, bit_stop, parità, numero_porta, modalità)
Descrizione	Il comando SETCOM imposta la comunicazione seriale per le porte seriali. Il comando abilita i protocolli Host Link oppure definisce la comunicazione per uso generico. L'impostazione predefinita prevede sulle porte seriali una velocità di 9600 baud, 7 bit di dati, 2 bit di stop, parità pari e XON/XOFF per consentire la comunicazione ad uso generico. Queste impostazioni predefinite vengono recuperate all'avvio.
Argomenti	• velocità_trasmissione 1200, 2400, 4800, 9600, 19200, 38400 • bit_dati 7, 8 • bit_stop 1, 2 • parità 0 = nessuna; 1 = dispari; 2 = pari. • numero_porta Vedere la prima tabella di seguito. • modalità Selezionare una delle modalità elencate nella seconda tabella di seguito per le porte seriali 1 e 2.
Esempio	Nessun esempio.
Vedere anche	N/D

valore numero_porta	Descrizione
1	Porta seriale 1 di RS-232C
2	Porta seriale 2 di RS-422A/485

Modalità	Descrizione
0	Comunicazione per uso generico (nessun meccanismo XON/XOFF)
5	Protocollo slave Host Link
6	Protocollo master Host Link

3.2.244 SGN

Tipo	Funzione matematica
Sintassi	SGN(espressione)
Descrizione	La funzione SGN restituisce il segno di un numero. Per i valori positivi (incluso lo 0) restituisce il valore 1 mentre per quelli negativi il valore -1.
Argomenti	espressione Qualsiasi espressione valida in BASIC.
Esempio	>> PRINT SGN(-1.2) -1.0000
Vedere anche	N/D

3.2.245 SIN

Tipo	Funzione matematica
Sintassi	SIN(espressione)
Descrizione	La funzione SIN restituisce il seno dell'espressione. I valori di ingresso sono espressi in radianti e possono avere qualsiasi valore. Il valore del risultato sarà compreso tra -1 e 1.
Argomenti	espressione Qualsiasi espressione valida in BASIC.
Esempio	>> PRINT SIN(PI/2) 1.0000
Vedere anche	N/D

3.2.246 SLOT

Tipo	Modificatore di slot
Sintassi	SLOT
Descrizione	Il modificatore specifica il numero di unità per un parametro, ad esempio COMMSTYPE . I numeri di unità Trajexia vanno da 0 a 6.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	N/D

3.2.247 SPEED

Tipo	Parametro degli assi
Sintassi	SPEED
Descrizione	Il parametro SPEED contiene la velocità richiesta espressa in unità/s. Può avere qualsiasi valore positivo (incluso lo 0). La velocità richiesta è la velocità massima per i comandi dei movimenti dotati di profilo di velocità.
Argomenti	N/D
Esempio	SPEED = 1000 PRINT "Imposta la velocità = ";SPEED
Vedere anche	ACCEL, AXIS, DATUM, DECEL, FORWARD, MOVE, MOVEABS, MOVECIRC, MOVEMODIFY, REVERSE, UNITS.

3.2.248 SQR

Tipo	Funzione matematica
Sintassi	SQR(espressione)
Descrizione	La funzione SQR restituisce la radice quadrata dell'espressione. L'espressione deve avere un valore positivo (incluso lo 0).
Argomenti	espressione Qualsiasi espressione valida in BASIC.
Esempio	>> PRINT SQR(4) 2.0000
Vedere anche	N/D

3.2.249 SRAMP

Tipo	Parametro degli assi
Sintassi	SRAMP
Descrizione	Il parametro degli assi SRAMP contiene il fattore della curva a S. Il fattore della curva a S controlla la quantità di arrotondamento applicato ai profili trapezoidali. Un valore pari a 0 non imposta alcun arrotondamento. Un valore pari a 10 imposta un arrotondamento massimo. Il valore predefinito del parametro è 0. SRAMP viene applicato ai comandi FORWARD, MOVE, MOVEABS, MOVECIRC, MHELICAL e REVERSE. Note: - l'utilizzo di curve a S aumenta il tempo richiesto da un movimento per essere completato. - Il fattore della curva a S non deve essere cambiato mentre un movimento è in corso.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	AXIS .

3.2.250 STEP

Vedere **FOR..TO..STEP..NEXT**.

3.2.251 STEP_RATIO

Tipo	Comando degli assi
Sintassi	STEP_RATIO(conteggio_uscita, conteggio_dpos)
Descrizione	Questo comando imposta un rapporto per l'uscita del motore passo-passo dell'asse. Per ciascun periodo di servoazionamento, la funzione STEP_RATIO elabora il numero dei passi prima che questo raggiunga l'uscita impulsi del motore passo-passo. Conteggio degli impulsi in uscita = (numeratore)/(denominatore) * MPOS. STEP_RATIO influenza sia MOVECIRC che CAMBOX. Note: - la funzione STEP_RATIO opera prima del fattore di divisione per 16 dell'asse del motore passo-passo. - I rapporti alti devono essere evitati, poiché provocano una perdita di risoluzione oppure riducono considerevolmente l'uniformità del movimento. Le dimensioni effettive e fisiche del passo x 16 corrispondono alla risoluzione BASIC dell'asse e l'utilizzo di questo comando potrebbe ridurre la capacità dell'unità di controllo di raggiungere con precisione tutte le posizioni. STEP_RATIO non sostituisce UNITS. Astenersi dall'utilizzare STEP_RATIO per rimuovere il fattore x16 dall'asse del motore passo-passo: in caso contrario, infatti, si rende carente il controllo della frequenza dei passi.
Argomenti	denominatore Un numero intero compreso tra 0 e 16777215, usato per definire il denominatore dell'equazione qui sopra. numeratore Un numero intero compreso tra 0 e 16777215, usato per definire il numeratore dell'equazione qui sopra.

Esempio Due assi sono impostati come X e Y, ma i passi degli assi, in mm, non sono gli stessi. Per mantenere costante la velocità di posizionamento e consentire il corretto funzionamento di MOVECIRC, i movimenti interpolati richiedono valori UNITS identici su entrambi gli assi. L'asse con la risoluzione più bassa viene modificato e portato ad avere la risoluzione di posizionamento più alta per mantenere il miglior livello di precisione per entrambi gli assi.

```
' Asse 0: 500 conteggi/mm (31,25 passi/mm)
' Asse 1: 800 conteggi/mm (50,00 passi/mm)
BASE(0)
STEP_RATIO(500,800)
UNITS = 800
BASE(1)
UNITS = 800
```

Vedere anche N/D

3.2.252 STEPLINE

Tipo Comando di programma

Sintassi **STEPLINE** ["nome_programma" [, numero_task]]

Descrizione Il comando **STEPLINE** esegue una linea (cioè un "passo") del programma specificato dal **nome_programma**. Il nome del programma può essere specificato anche senza virgolette. Se il comando **STEPLINE** viene eseguito senza il nome del programma sulla riga di comando, il programma correntemente selezionato sarà eseguito in modalità passo-passo. Se il comando **STEPLINE** viene eseguito senza il nome del programma all'interno di un programma, quel programma sarà eseguito in modalità passo-passo.

Se il programma viene specificato, tutte le occorrenze del programma saranno eseguite in modalità passo-passo. Un nuovo task viene avviato quando nessuna copia del programma è in esecuzione. Se anche il task è specificato, solo la copia del programma in esecuzione sul task specificato viene eseguita in modalità passo-passo. Se nessuna copia del programma è in esecuzione sul task specificato, ne viene avviata una.

Argomenti

- **nome_programma**
Il nome del programma da eseguire in modalità passo-passo.
- **numero_task**
Il numero del task con il programma da eseguire in modalità passo-passo. Intervallo: [1,14].

Esempio >> **STEPLINE "nastro trasportatore"**

Esempio >> **STEPLINE "matematica",2**

Vedere anche **RUN, SELECT, STOP, TROFF, TRON.**

3.2.253 STOP

Tipo Comando di programma

Sintassi **STOP** ["nome_programma" [, numero_task]]

Descrizione Il comando **STOP** arresta l'esecuzione del programma specificato nel **nome_programma**. Se il nome del programma è omesso, viene arrestato il programma correntemente selezionato. Il nome del programma può essere specificato anche senza virgolette. In caso di esecuzione multipla di un singolo programma su task diversi, è possibile utilizzare l'argomento **numero_task** per specificare il task che deve essere arrestato.

Argomenti

- **nome_programma**
Il nome del programma da arrestare.
- **numero_task**
Il numero del task con il programma da arrestare. Intervallo: [1,14].

Esempio >> **STOP nomeprog**

Esempio In questo esempio le righe da "etichetta" in giù non vengono eseguite.

```
STOP
etichetta:
PRINT var
RETURN
```

Vedere anche **HALT, RUN, SELECT.**

3.2.254 SYSTEM_ERROR

Tipo	Parametro di sistema (sola lettura)
Sintassi	SYSTEM_ERROR
Descrizione	Il parametro SYSTEM_ERROR contiene gli errori di sistema che si sono verificati nel sistema TJ1 dall'ultima volta che è stato inizializzato. I bit del parametro SYSTEM_ERROR sono forniti nella tabella di seguito.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	N/D

Bit	Descrizione
0	Errore BASIC
1	Errore di batteria scarica
2 – 7	Riservato a un uso futuro
8	Errore nell'unità di configurazione (qualsiasi unità del sistema)
9	Errore nel dispositivo di configurazione (qualsiasi dispositivo del sistema)
10 – 15	Riservato a un uso futuro
16	Errore dovuto alla perdita di un'unità (qualsiasi unità del sistema)
17	Terminatore non installato
18	Errore dovuto alla perdita di un dispositivo (qualsiasi dispositivo del sistema)

3.2.255 T_REF

Tipo	Parametro degli assi
Sintassi	T_REF DAC
Descrizione	Il parametro T_REF contiene il valore della coppia di riferimento che sarà applicata al servomotore. L'intervallo di questo parametro è definito dal numero di bit disponibili. Per gli assi MECHATROLINK, T_REF prende 32 bit, pertanto l'intervallo disponibile è [–2147483648, 2147483648], pari a un intervallo di tensione [–10 V, 10 V]. Per gli assi flessibili, T_REF prende 16 bit, pertanto l'intervallo disponibile è [–32768, 32767], pari a un intervallo di tensione [–10 V, 10 V]. Questi intervalli possono essere limitati utilizzando il parametro OUTLIMIT . La coppia di riferimento effettiva dipende dal servomotore.
Argomenti	N/D
Esempio	T_REF AXIS(0)=1000
Vedere anche	AXIS , S_REF .

3.2.256 TABLE

Tipo Comando di sistema

Sintassi **TABLE(indirizzo, valore {, valore})**
TABLE(indirizzo)

Descrizione Il comando **TABLE** carica e legge i dati su e dalla matrice della memoria TABLE. La memoria TABLE ha una lunghezza massima di 64000 elementi. I valori della memoria TABLE sono numeri a virgola mobile con frazioni. La memoria TABLE può essere utilizzata anche per contenere informazioni, in alternativa alle variabili. Il comando **TABLE** ha due forme.

- **TABLE(indirizzo, valore{, valore})** scrive una sequenza di valori nella matrice della memoria TABLE. La posizione dell'elemento è specificata dall'indirizzo. La sequenza può avere una lunghezza massima di 20 elementi.
- **TABLE(indirizzo)** restituisce il valore della memoria TABLE a quella voce.

Un valore della memoria TABLE può essere di sola lettura se un valore con quel numero o un numero più alto è stato scritto in precedenza nella memoria. Ad esempio, se la posizione più alta della memoria TABLE in cui è stato scritto un valore è la numero 1000, qualora si cercasse di stampare **TABLE(1001)** si otterrebbe un messaggio d'errore. Le dimensioni totali della memoria TABLE sono indicate dal parametro **TSIZE**. Notare che questo valore supera di uno l'indirizzo dell'elemento definito più alto. È possibile cancellare la memoria TABLE utilizzando **DEL "TABLE"** o **NEW "TABLE"** sulla riga di comando.

Note:

- Applicazioni come i comandi **CAM**, **CAMBOX** e **SCOPE** in Trajexia Tools utilizzano tutte la stessa memoria TABLE come area dati. Non utilizzare lo stesso intervallo di area dati per scopi diversi.
- Tutti i vari task in esecuzione possono accedere ai dati TABLE e VR. Per evitare i problemi causati dall'imprevista scrittura di una stessa variabile globale da parte di due task di programma, si consiglia di scrivere i programmi in maniera tale che la scrittura della variabile globale sia fatta da un solo programma alla volta.
- I dati TABLE e VR contenuti nella RAM vanno persi al momento dello spegnimento.

- Argomenti
- **indirizzo**
La prima posizione della memoria TABLE da leggere o scrivere. Intervallo: [0,63999]
 - **valore**
Il valore che deve essere scritto nella posizione specificata e nelle posizioni successive.

Esempio **TABLE(100,0,120,250,370,470,530,550)**
Questa riga carica una memoria TABLE interna come quella riportata di seguito.

Esempio Questa riga stampa il valore presente alla posizione 1000.
>> PRINT TABLE(1000)

Vedere anche **CAM**, **CAMBOX**, **DEL**, **NEW**, **SCOPE**, **TSIZE**, **VR**.

Voce della memoria TABLE	Valore
100	0
101	120
102	250
103	370
104	470
105	530
106	550

3.2.257 TABLEVALUES

Tipo Comando di sistema

Sintassi **TABLEVALUES(indirizzo, numero_di_punti, formato)**

Descrizione Restituisce un elenco di punti della memoria TABLE iniziando dal numero specificato. Attualmente è supportato un solo formato, vale a dire testo delimitato da virgole.
Nota: **TABLEVALUES** viene fornito principalmente per consentire a Trajexia Tools di accedere rapidamente agli insiemi di valori della memoria TABLE.

Argomenti	• indirizzo Numero del primo punto che deve essere restituito • numero_di_punti Numero totale dei punti che devono essere restituiti • formato Formato dell'elenco

Esempio Nessun esempio.

Vedere anche N/D

3.2.258 TAN

Tipo	Funzione matematica
Sintassi	TAN(espressione)
Descrizione	La funzione TAN restituisce la tangente dell'espressione. L'espressione si presume espressa in radianti.
Argomenti	• espressione Qualsiasi espressione valida in BASIC.
Esempio	>> PRINT TAN(PI/4) 1.0000

Vedere anche N/D

3.2.259 THEN

Vedere IF..THEN..ELSE..ENDIF.

3.2.260 TICKS

Tipo	Parametro dei task
Sintassi	TICKS
Descrizione	Il parametro TICKS contiene il conteggio corrente degli impulsi di clock del task. TICKS è un contatore a 32 bit il cui valore scende a ogni ciclo di servozionamento. TICKS consente sia la lettura che la scrittura, Può essere usato per misurare i tempi dei cicli, aggiungere ritardi, ecc. Ciascun task dispone di un parametro TICKS proprio. Usare il modificatore PROC per accedere al parametro di un determinato task. Senza PROC viene preso in considerazione il task corrente.

Argomenti	N/D
Esempio	ritardo: TICKS = 3000 OP(9,ON) test: IF TICKS <= 0 THEN OP(9,OFF) ELSE GOTO test ENDIF

Vedere anche N/D

3.2.261 TIME

Tipo	Parametro di sistema
Sintassi	TIME
Descrizione	Restituisce l'ora contenuta nell'orologio calendario. Il tempo restituito è espresso nel numero di secondi trascorsi dopo la mezzanotte (00:00:00).
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	N/D

3.2.262 TIME\$

Tipo	Comando di sistema
Sintassi	TIME\$
Descrizione	Stampa il tempo definito dall'orologio calendario come una stringa in formato 24 ore.
Argomenti	N/D
Esempio	>>? TIME\$ 14/39/02
Vedere anche	N/D

3.2.263 TO

Vedere **FOR..TO..STEP..NEXT**.

3.2.264 TRANS_DPOS

Tipo	Parametro degli assi (sola lettura)
Sintassi	TRANS_DPOS
Descrizione	La posizione richiesta per l'asse all'uscita della trasformazione del contorno. TRANS_DPOS è normalmente uguale a DPOS su ciascun asse. La trasformazione del contorno è quindi pari a 1:1 per ciascun asse. Per la configurazione di alcuni macchinari, ad esempio i bracci meccanici o le macchine con movimenti parassiti sugli assi, può risultare utile installare una trasformazione del contorno diversa da 1:1. Le trasformazioni dei contorni devono essere scritte appositamente in linguaggio C e scaricate nell'unità di controllo. Se si desiderano installare delle trasformazioni del contorno, è essenziale contattare OMRON.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	FRAME .

3.2.265 TRIGGER

Tipo	Comando di sistema
Sintassi	TRIGGER
Descrizione	Il comando TRIGGER avvia un comando SCOPE precedentemente impostato. Nota: Trajexia Tools utilizza TRIGGER automaticamente per la propria funzione di oscilloscopio.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	SCOPE .

3.2.266 TROFF

Tipo	Comando di programma
Sintassi	TROFF ["nome_programma"]
Descrizione	Il comando TROFF sospende una registrazione alla riga corrente e riprende l'esecuzione normale del programma specificato con nome_programma . Il nome del programma può essere specificato anche senza virgolette. Se il nome del programma è omissso, viene preso in considerazione quello selezionato.
Argomenti	• nome_programma Il nome del programma per cui deve essere sospesa la registrazione.
Esempio	>> TROFF "righe"
Vedere anche	SELECT , TRON .

3.2.267 TRON

Tipo	Comando di programma
Sintassi	TRON
Descrizione	Il comando TRON crea un'interruzione in un programma che ne sospende l'esecuzione alla riga che segue il comando TRON . Il programma può quindi essere eseguito, ad esempio, una riga per volta utilizzando il comando STEPLINE . Note: • L'esecuzione del programma può essere ripresa senza dover utilizzare il comando STEPLINE, eseguendo il comando TROFF. • La modalità di registrazione può essere arrestata attraverso un comando STOP o HALT. • Trajexia Tools evidenzia le righe che contengono TRON nelle finestre Edit e Debug.
Argomenti	N/D
Esempio	TRON MOVE(0,10) MOVE(10,0) TRON MOVE(0,-10) MOVE(-10,0)
Vedere anche	SELECT , TROFF .

3.2.268 TRUE

Tipo	Costante (sola lettura)
Sintassi	TRUE
Descrizione	La costante TRUE restituisce il valore numerico -1.
Argomenti	N/D
Esempio	test: t = IN(0) AND IN(2) IF t = TRUE THEN PRINT "Gli ingressi sono attivi" ENDIF
Vedere anche	N/D

3.2.269 TSIZE

Tipo	Parametro di sistema (sola lettura)
Sintassi	TSIZE
Descrizione	Il parametro TSIZE restituisce le dimensioni della matrice della memoria TABLE, che supera di un punto l'elemento più alto definito nella memoria TABLE. TSIZE viene reimpostato su 0 quando la matrice della memoria TABLE viene cancellata utilizzando sulla riga di comando DEL "TABLE" o NEW "TABLE" .
Argomenti	N/D
Esempio	L'esempio di seguito presume che nella matrice della memoria TABLE non sia stata scritta alcuna posizione più alta di 1000. >> TABLE(1000,3400) >> PRINT TSIZE 1001.0000
Vedere anche	DEL , NEW , TABLE .

3.2.270 UNITS

Tipo	Parametro degli assi
Sintassi	UNITS
Descrizione	Il parametro degli assi UNITS contiene il fattore di conversione delle unità. Il fattore di conversione delle unità consente all'utente di definire un'unità che gli risulta più comoda, ad esempio m, mm o giri motore, specificando la quantità di fronti dell'encoder per unità utente. I parametri dell'asse, come ad esempio la velocità, l'accelerazione, la decelerazione e i comandi dell'asse, sono specificati nelle unità utente così definite. Nota: il parametro UNITS può essere qualsiasi valore diverso da zero, ma si consiglia di progettare i sistemi con un numero intero di impulsi encoder per unità utente. Per mantenere le stesse dinamiche all'interno del sistema, qualora si proceda alla modifica del parametro UNITS vengono influenzati tutti i parametri degli assi che da UNITS dipendono.
Argomenti	N/D
Esempio	Una vite conduttrice ha un passo di 5 mm e un encoder da 1000 impulsi/giro. Le unità devono essere impostate per consentire ai movimenti di essere specificati in mm. I 1.000 impulsi/giro generano $1.000 \times 4 = 4.000$ fronti/giro. Un giro è pari a 5 mm. Pertanto vi sono $4.000/5 = 800$ fronti/mm. UNITS viene quindi impostato come segue. >> UNITS = 1000*4/5

Vedere anche **AXIS, ENCODER_RATIO**.

3.2.271 UNLOCK

Vedere **LOCK**.

3.2.272 UNTIL

Vedere **REPEAT..UNTIL**.

3.2.273 VERIFY

Tipo	Parametro degli assi
Sintassi	VERIFY
Descrizione	Il parametro di verifica dell'asse viene utilizzato per selezionare diverse modalità operative su un asse dell'encoder del motore passo-passo. • VERIFY=OFF Il circuito di conteggio dell'encoder è collegato ai segnali hardware STEP e DIRECTION, consentendone il conteggio come segnali encoder. Ciò risulta particolarmente utile per la registrazione perché il circuito di registrazione può a quel punto agire su un asse del motore passo-passo. • VERIFY=ON Il circuito dell'encoder è collegato al segnale Z degli ingressi esterni A, B. Nota: quando in TJ1-FL02 VERIFY=OFF, il circuito di conteggio dell'encoder è configurato per accettare i segnali STEP e DIRECTION cablati agli ingressi A e B dell'encoder. Se VERIFY=ON, il circuito dell'encoder è configurato per il consueto ingresso di quadratura. Assicurarsi che la tensione degli ingressi dell'encoder non superi i 5 volt.
Argomenti	N/D
Esempio	VERIFY AXIS(3)=ON
Vedere anche	N/D

3.2.274 VERSION

Tipo	Parametro di sistema (sola lettura)
Sintassi	VERSION
Descrizione	Il parametro VERSION restituisce il numero di versione corrente del firmware del sistema correntemente installato su TJ1-MC__.
Argomenti	N/D
Esempio	>> PRINT VERSION 1.6100
Vedere anche	N/D

3.2.275 VFF_GAIN

Tipo	Parametro degli assi
Sintassi	VFF_GAIN
Descrizione	Il parametro VFF_GAIN contiene il guadagno di velocità per il feed-forward. Il contributo alla velocità d'uscita per il feed-forward viene calcolato moltiplicando il cambio della posizione richiesta per il valore del parametro VFF_GAIN. Il valore predefinito è 0. L'aggiunta di un guadagno di velocità per il feed-forward a un sistema diminuisce l'errore di inseguimento durante un movimento, incrementando l'uscita in proporzione alla velocità. Nota: per evitare qualsiasi instabilità i guadagni dei servozionamenti vanno cambiati solamente quando SERVO non è attivo.
Argomenti	N/D
Esempio	Nessun esempio.
Vedere anche	D_GAIN, I_GAIN, OV_GAIN, P_GAIN.

3.2.276 VP_SPEED

Tipo	Parametro degli assi (sola lettura)
Sintassi	VP_SPEED
Descrizione	Il parametro VP_SPEED contiene la velocità del profilo di velocità espresso in unità utente/s. La velocità del profilo di velocità è una velocità interna che viene aumentata o diminuita secondo i profili attribuiti al movimento.
Argomenti	N/D
Esempio	' Aspetta fino a quando non viene raggiunta la velocità del comando MOVE(100) WAIT UNTIL SPEED = VP_SPEED
Vedere anche	AXIS, MSPEED, UNITS.

3.2.277 VR

Tipo	Comando di sistema
Sintassi	VR(indirizzo)
Descrizione	Il comando VR legge o scrive il valore di una variabile globale (VR). Queste variabili VR contengono numeri reali e possono essere facilmente utilizzate come elemento o come matrice di elementi. TJ1-MC__ dispone in totale di 1024 variabili VR. Nella programmazione in BASIC è possibile utilizzare le variabili VR per diversi scopi. Le variabili VR sono condivise globalmente dai task e possono essere utilizzate per la comunicazione e tra un task e l'altro. Note: - tutti i vari task in esecuzione possono accedere ai dati TABLE e VR. Per evitare i problemi causati dall'imprevista scrittura di una stessa variabile globale da parte di due task di programma, si consiglia di scrivere i programmi in maniera tale che la scrittura della variabile globale sia fatta da un solo programma alla volta. - I dati TABLE e VR contenuti nella RAM vanno persi al momento dello spegnimento.
Argomenti	indirizzo L'indirizzo della variabile VR. Intervallo: [0,1023].
Esempio	Nell'esempio seguente, il valore 1.2555 viene assegnato alla variabile VR 15. La variabile locale val è utilizzata per dare localmente un nome alla variabile globale: val = 15 VR(val) = 1.2555

Esempio Un gantry di trasferimento dispone di 10 posizioni in una riga. Ciascuna posizione può, in qualsiasi momento, essere piena o vuota. Le variabili da VR(101) a VR(110) sono utilizzate per tenere una matrice di dieci 1 e/o 0 per indicare le posizioni occupate (1) oppure vacanti (0). Il portale deposita il carico nella prima posizione libera. Il programma che ottiene tale risultato sarebbe, in parte, come segue:

movimentop:
MOVEABS(115) ' Si porta alla prima posizione di deposito del carico
FOR VR(0) = 101 TO 110
IF (VR(VR(0)) = 0) THEN GOSUB carico
MOVE(200) ' 200 lascia spazi vuoti tra le posizioni
NEXT VR(0)
PRINT "Tutte le posizioni sono occupate"
WAIT UNTIL IN(3) = ON
GOTO movimentop

carico: ' Deposita il carico nella posizione prevista e lo segna in matrice
OP(15,OFF)
VR(VR(0)) = 1
RETURN

Le variabili usufruiscono del backup di una batteria, quindi il programma in esame potrebbe essere concepito per memorizzare lo stato della macchina quando l'alimentazione è assente. Sarebbe naturalmente necessario fornire un mezzo di ripristino totale in seguito all'intervento manuale.

Esempio **anello: ' Assegna VR(65) a VR(0) moltiplicato per la posizione misurata dell'asse 1**
VR(65) = VR(0)*MPOS AXIS(1)
PRINT VR(65)
GOTO anello

Vedere anche **CLEAR_BIT, READ_BIT, SET_BIT, TABLE.**

3.2.278 VRSTRING

Tipo Comando di sistema

Sintassi **VRSTRING(inizio_vr)**

Descrizione Combina il contenuto di una matrice di variabili VR() per consentirne la stampa come stringa di testo. Tutti i caratteri stampabili vengono stampati e la stringa termina al primo carattere nullo trovato. (cioè una VR(n) che contiene 0).

Argomenti

- inizio_vr**
numero della prima VR() nella matrice di caratteri.

Esempio **PRINT #5,VRSTRING(100)**

Vedere anche N/D

3.2.279 WA

Tipo Comando di sistema

Sintassi **WA(tempo)**

Descrizione Il comando **WA** sospende l'esecuzione del programma per il numero di millisecondi specificati da "tempo". Il comando può essere utilizzato solo all'interno di un programma.

Argomenti

- tempo**
Il numero di millisecondi durante i quali l'esecuzione del programma rimane sospesa.

Esempio Le righe di seguito attivano l'uscita 7 due secondi dopo aver disattivato l'uscita 1.
OP(1,OFF)
WA(2000)
OP(7,ON)

Vedere anche N/D

3.2.280 WAIT IDLE

Tipo	Comando di sistema
Sintassi	WAIT IDLE
Descrizione	Il comando WAIT IDLE sospende l'esecuzione del programma fino a quando l'asse di base non ha concluso il movimento corrente e tutti i movimenti contenuti nel buffer. Il comando può essere utilizzato solo all'interno di un programma. WAIT IDLE agisce sull'asse di base predefinito (impostato con BASE) a meno che AXIS non venga usato per specificare un asse di base temporaneo. Nota: l'esecuzione di WAIT IDLE non necessariamente significa che l'asse sarà stazionario in un sistema a servomotore.
Argomenti	N/D
Esempio	MOVE(1000) WAIT IDLE PRINT "Movimento concluso" L'istruzione di stampa viene eseguita al termine del movimento.
Esempio	MOVE(1000) WAIT UNTIL MTYPE=0 PRINT "Movimento terminato" La maggior parte delle volte, l'istruzione di stampa viene eseguita PRIMA dell'inizio del movimento e a volte quando il movimento è finito.
Spiegazione	Il funzionamento dei programmi di movimento e della sequenza di movimento è parallelo e non sincronizzato. Un ciclo completo può verificarsi prima che il movimento sia caricato nel buffer. Il programma esegue MOVE(1000), ma il movimento non viene caricato nel buffer fino a quando non inizia la "sequenza di movimento" successiva; ecco perché quando si verifica MTYPE=0, si ottiene 0: è perché il movimento NON È ANCORA INIZIATO, non perché è finito.
Vedere anche	AXIS , WAIT LOADED .



Nota:

WAIT IDLE è un comando specificamente concepito per aspettare fino a quando il movimento precedente non è finito e pertanto gestisce il ritardo dal momento in cui nel programma il comando precedente viene eseguito fino al momento in cui il comando viene correttamente caricato nel buffer di movimento.

3.2.281 WAIT LOADED

Tipo	Comando di sistema
Sintassi	WAIT LOADED
Descrizione	Il comando WAIT LOADED sospende l'esecuzione del programma fino a quando nel buffer non vi sono più movimenti da eseguire per l'asse di base, a parte quelli attualmente in esecuzione. Il comando può essere utilizzato solo all'interno di un programma. Ciò risulta utile per l'attivazione di eventi all'inizio di un movimento oppure alla fine, quando i movimenti multipli sono riuniti nel buffer. WAIT_LOADED agisce sull'asse di base predefinito (impostato con BASE) a meno che AXIS non venga usato per specificare un asse di base temporaneo.
Argomenti	N/D
Esempio	' Attiva l'uscita 8 all'inizio di MOVE(500) e la disattiva alla fine MOVE(800) MOVE(500) WAIT LOADED OP(8,ON) MOVE(400) WAIT LOADED OP(8,OFF)
Vedere anche	AXIS , WAIT IDLE

3.2.282 WAIT UNTIL

Tipo	Comando di sistema
Sintassi	WAIT UNTIL condizione
Descrizione	Il comando WAIT UNTIL valuta ripetutamente la condizione fino a quando non è TRUE. Dopodiché l'esecuzione del programma prosegue. Il comando può essere utilizzato solo all'interno di un programma.
Argomenti	condizione Qualsiasi valida espressione logica in BASIC.

Esempio In questo esempio, il programma aspetta fino a quando la posizione misurata sull'asse 0 non supera 150, dopodiché inizia un movimento sull'asse 1.

WAIT UNTIL MPOS AXIS(0)>150
MOVE(100) AXIS(1)

Esempio Il livello di complessità delle espressioni valutate non ha limiti, a condizione che tali espressioni seguano la sintassi BASIC, ad esempio:

WAIT UNTIL DPOS AXIS(2) <= 0 OR IN(1) = ON

Con questa riga si genera un'attesa fino a quando la posizione richiesta per l'asse 2 non risulta minore di o uguale a 0 o l'ingresso 1 è attivo.

Vedere anche N/D

3.2.283 WDOG

Tipo Parametro di sistema

Sintassi **WDOG**

Descrizione Il parametro **WDOG** contiene l'interruttore software di abilitazione del servozionamento che utilizza il comando di ingresso **RUN** (servozionamento attivo). Il servozionamento abilitato controlla il servomotore in base ai valori della velocità e della coppia di riferimento. **WDOG** può essere attivato e disattivato dal controllo dei programmi, alla riga di comando e al pulsante di controllo di Trajexia Tools. Il servozionamento viene automaticamente disabilitato quando si verifica un **MOTION_ERROR**. Un errore di movimento si verifica quando lo stato **AXISSTATUS** per uno degli assi corrisponde all'impostazione **ERRORMASK**. In questo caso l'interruttore software (**WDOG**) viene disattivato, il parametro **MOTION_ERROR** prende il valore 1 e il parametro **ERROR_AXIS** prende il numero del primo asse contenente l'errore. Nota: Trajexia Tools può eseguire automaticamente il parametro **WDOG** quando sul pannello di controllo è selezionato il pulsante **Drives Enable**.

Argomenti N/D

Esempio Nessun esempio.

Vedere anche **AXISSTATUS, ERROR_AXIS, ERRORMASK, MOTION_ERROR, SERVO**.

3.2.284 WHILE..WEND

Tipo Comando di controllo del programma.

Sintassi **WHILE condizione**
comandi
WEND

Descrizione La struttura **WHILE ... WEND** consente la ripetizione per un dato numero di volte del segmento di programma compreso tra **WHILE** e l'istruzione **WEND**, fino a quando la condizione non diventa **TRUE**. In tal caso, l'esecuzione del programma prosegue dopo **WEND**. Nota: è possibile nidificare gli anelli **WHILE ... WEND** illimitatamente.

Argomenti

- **condizione**
Qualsiasi valida espressione logica in BASIC.

Esempio **WHILE IN(12) = OFF**
MOVE(200)
WAIT IDLE
OP(10,OFF)
MOVE(-200)
WAIT IDLE
OP(10,ON)
WEND

Vedere anche **FOR..TO..STEP..NEXT, REPEAT..UNTIL**

3.2.285 XOR

Tipo Operazione matematica

Sintassi **espressione1 XOR espressione2**

Descrizione L'operatore **XOR** esegue la funzione logica **XOR** tra i bit corrispondenti dei numeri interi di due espressioni valide in BASIC.
La funzione logica **XOR** tra due bit viene definita nella tabella di seguito.

Argomenti

- **espressione1**
Qualsiasi espressione valida in BASIC.
- **espressione2**
Qualsiasi espressione valida in BASIC.

Esempio **VR(0)=10 XOR 18**
XOR è un operatore di bit e quindi l'azione binaria che interviene è come segue: **01010 XOR 10010 = 11000**. Il risultato è pertanto 24.

Vedere anche N/D

Bit 1	Bit 2	Risultato
0	0	0
0	1	1
1	0	1
1	1	0

4 Protocolli di comunicazione

4.1 Interfacce disponibili

Le unità Trajexia dispongono delle seguenti interfacce di comunicazione:

Unità	Interfaccia	Protocollo	Commento
TJ1-MC__	Ethernet	Protocollo Trajexia Tools.	Per programmare, controllare ed effettuare il debug del progetto con Trajexia Tools.
		Server FINS	Per comunicare con qualsiasi master FINS, ad esempio PLC, HMI o un personal computer.
		Client FINS	Per comunicare con qualsiasi server FINS, ad esempio PLC o un'altra unità Trajexia.
	Seriale	Master Host Link	Per comunicare con qualsiasi slave Host Link, ad esempio un PLC OMRON.
		Slave Host Link	Per comunicare con qualsiasi master Host Link, generalmente HMI.
		Definito dall'utente	Questo protocollo è creato e gestito utilizzando i comandi BASIC.
TJ1-PRT	PROFIBUS	Slave PROFIBUS DP-V0	Per lo scambio di variabili di canale con qualsiasi master PROFIBUS.
TJ1-DRT	DeviceNet	DeviceNet	Per lo scambio di variabili di canale con qualsiasi master DeviceNet.
TJ1-ML__	MECHATROLINK	MECHATROLINK	Per comunicare con gli slave MECHATROLINK supportati. Questo protocollo risulta trasparente per l'utente.

4.2 Ethernet

TJ1-MC__ è dotato di porta Ethernet standard da 10/100 Mbps. Per collegare TJ1-MC__ a un PC è possibile utilizzare un cavo Ethernet crossover o patch. Per configurare l'interfaccia, impostare i seguenti parametri:

Voce	Valore predefinito	Commento
Indirizzo IP	192.168.0.250	Impostare un indirizzo IP che sia unico all'interno della rete.
Maschera di sottorete	255.255.255.0	Impostare la stessa sottorete utilizzata dal LAN.
Gateway	0.0.0.0	Il gateway è necessario per consentire l'accesso remoto da un altro LAN.

Assicurarsi che l'indirizzo IP del PC e quello di TJ1-MC__ cadano nello stesso intervallo: se l'indirizzo IP di TJ1-MC__ è *aaa.bbb.ccc.ddd*, l'indirizzo IP del PC deve essere *aaa.bbb.ccc.xxx*, laddove *xxx* deve essere compreso tra 000 e 255 ed essere diverso da *ddd*.

Collegando il PC attraverso un hub o uno switch di rete è possibile cambiare l'indirizzo IP di TJ1-MC__ e farlo corrispondere all'indirizzo IP del PC. Per esempio, se l'indirizzo IP del PC è 192.200.185.001, è possibile impostare l'indirizzo IP di TJ1-MC__ su 192.200.185.002.



Nota
TJ1-MC__ non dispone della funzionalità DHCP e pertanto non è in grado di assegnare un indirizzo IP a un PC.

La maschera di sottorete di TJ1-MC__ è generica. Non è necessario che corrisponda alla maschera di sottorete del PC.

Utilizzare il comando **Ethernet** per leggere o scrivere le impostazioni Ethernet. Affinché i cambiamenti diventino effettivi, è necessario spegnere e riaccendere le unità.

Per verificare l'indirizzo IP di TJ1-MC__ è possibile utilizzare la riga di comando di Trajexia Tools e il comando **Ethernet**: è sufficiente digitare il comando **Ethernet(0, -1, 0)** sulla riga di comando per visualizzare su di essa l'indirizzo IP di TJ1-MC__.



Nota
Prima che il cambiamento di indirizzo IP diventi effettivo, è necessario spegnere e poi riaccendere il Trajexia.

4.2.1 Comunicare direttamente con Trajexia dal proprio computer

1. Lasciare immutate le impostazioni Ethernet di Trajexia.
2. Regolare le impostazioni di Trajexia Tools come mostrato in figura.

fig. 1

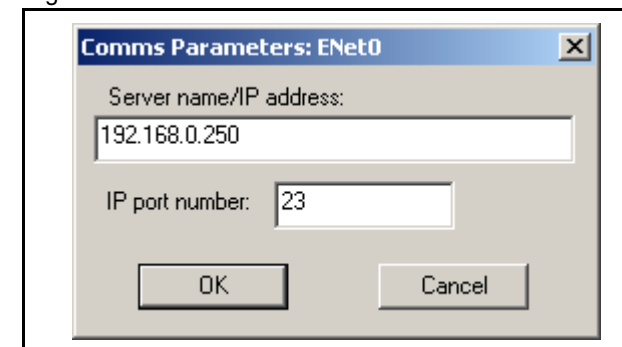
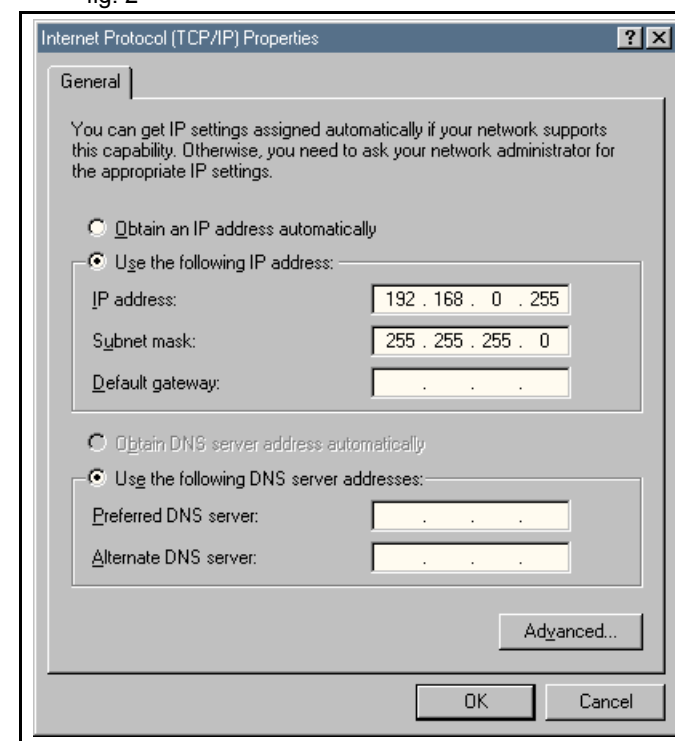


fig. 2



3. Impostare il computer come mostrato in figura.

4.2.2 Comunicare a distanza con Trajexia

Questo esempio mostra come collegarsi a Trajexia locale da un computer in posizione remota. Supponendo che le impostazioni Ethernet del Trajexia siano le seguenti:

- 10.83.50.70 è l'indirizzo IP assegnato a Trajexia.
- 255.255.240.0 è la maschera di sottorete locale.
- 10.83.48.1 è il gateway locale.
- Il server assegna automaticamente un indirizzo IP ai computer.

1. In Trajexia impostare l'indirizzo IP, la maschera di sottorete e il gateway dalla riga di comando della finestra del terminale con:

Ethernet(1,-1,0,10,83,50,70)

Ethernet(1,-1,2,255,255,240,0)

Ethernet(1,-1,8,10,83,48,1)

2. Controllare che le impostazioni IP del Trajexia locale e del computer remoto siano quelle mostrate.

Dopo l'accensione, il display di TJ1-MC__ mostra alternativamente l'indirizzo IP e la maschera di sottorete.

Dopo ogni nuovo collegamento del cavo Ethernet, il display mostra solo l'indirizzo IP.

fig. 3

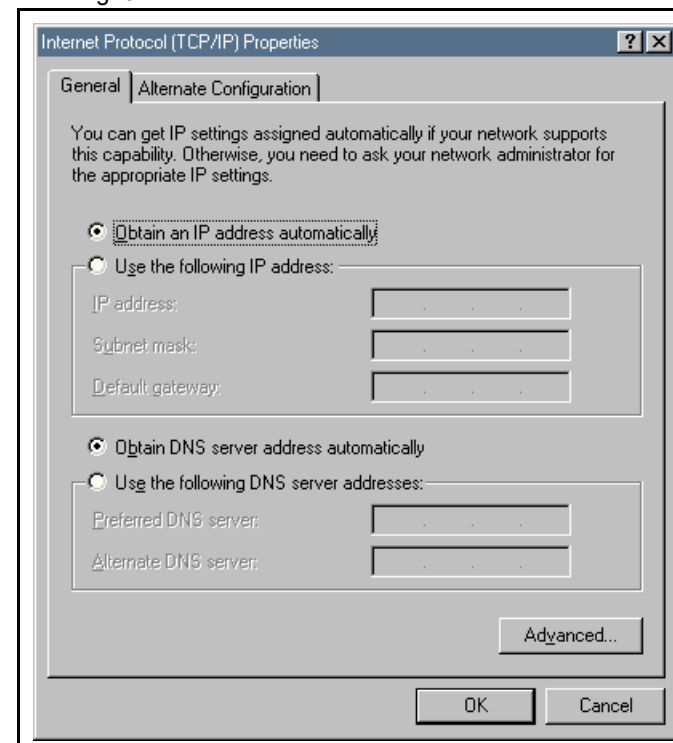
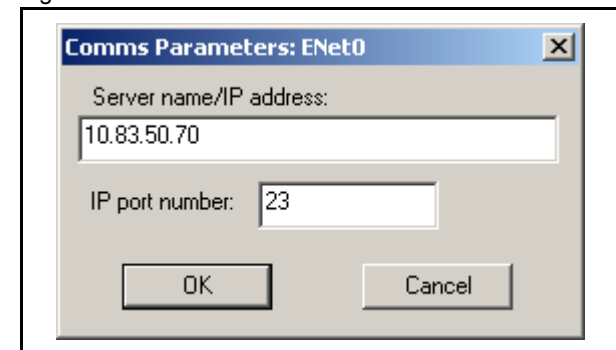


fig. 4



4.2.3 Protocollo Trajexia Tools

Il protocollo Trajexia Tools è utilizzato da Trajexia Tools per programmare, controllare ed effettuare il debug di TJ1-MC__.

Trajexia Tools utilizza un protocollo Telnet. L'impostazione predefinita prevede che questa connessione utilizzi la porta 23. Se questa porta non è accessibile, è possibile cambiare il numero della porta con il comando **Ethernet(1,-1,4,nuova_porta_n)**.

Contrariamente ai comandi Ethernet standard, questo comando diventa immediatamente effettivo una volta eseguito. La porta passa all'impostazione predefinita al momento dell'accensione. Pertanto questo comando deve essere incluso in qualsiasi programma che viene eseguito all'avvio.

Il protocollo Trajexia Tools è esclusivamente un TCP.

4.2.4 Protocollo del server FINS

FINS (Factory Interface Network Service) è un protocollo di comunicazione proprietario di OMRON. Una parte di questo protocollo è integrata in Trajexia. Fare riferimento al Communication Commands Reference Manual (W342-E1).

Il protocollo FINS consente la comunicazione senza soluzione di continuità con altri dispositivi OMRON come ad esempio PLC, HMI e CX-Drive. Il protocollo del server FINS non richiede alcuna impostazione di configurazione.



AVVERTENZA

Dal momento che TJ1-MC__ può comunicare con diverse fonti allo stesso tempo, i comandi di due fonti possono interferire tra di loro.

L'impostazione predefinita prevede che questa connessione utilizzi la porta 9600. Se questa porta non è accessibile, è possibile cambiare porta con il comando **Ethernet(1,-1,12,nuova_porta_n)**.

Contrariamente ai comandi Ethernet standard, questo comando diventa immediatamente effettivo una volta eseguito. La porta passa all'impostazione predefinita al momento dell'accensione. Pertanto questo comando deve essere incluso in qualsiasi programma che viene eseguito all'avvio.

I comandi FINS consentono la comunicazione tra nodi appartenenti a reti diverse: Un dispositivo master FINS può leggere le variabili VR e le variabili della memoria TABLE di Trajexia attraverso i comandi FINS. Questi comandi utilizzano la connessione Ethernet di TJ1-MC__. Il protocollo del server FINS è esclusivamente un UDP.



Nota

La lunghezza massima di un comando FINS su una connessione Ethernet è di 2012 byte.

Trajexia utilizza questi comandi FINS:

- 0101 (lettura di memoria)
- 0102 (scrittura di memoria)

Comando read (lettura)

Il comando **read** di FINS ha il seguente formato:

01	01	..		00	
codice_comando	tipo_var	indirizzo_partenza	fisso	conteggio_elementi	

I parametri possono avere i seguenti valori:

Parametro	Valori (esadecimali)
codice_comando	01 01
tipo_var	• 82 (memoria TABLE in formato di numero intero a 16 bit) • C2 (memoria TABLE in formato a virgola mobile IEEE a 32-bit) • B0 (memoria VR in formato di numero intero a 16 bit)
indirizzo_partenza	0 <= indirizzo_partenza <= numero di variabili - 1 <= FFFF

Parametro	Valori (esadecimali)
conteggio_elementi	1 <= conteggio_elementi <= numero di variabili – indirizzo_partenza

TJ1-MC__ risponde ai seguenti codici:

Condizione	Codice di risposta (esadecimale)	Descrizione
Tutti gli elementi validi	0000	OK
Tipo_var non valido	1101	Nessun tipo di area
Indirizzo_partenza non valido	1103	Errore di designazione dell'intervallo indirizzi
Numero di elementi non valido	1104	Indirizzo fuori intervallo

Se **tipo_var** è 82 o B0 e il codice di risposta è 0000, TJ1-MC__ risponde con:

01	01	00	00			
codice_comando	codice_risposta	word_1	word_2	...		

Se **tipo_var** è C2 e il codice di risposta è 0000, TJ1-MC__ risponde con:

01	01	00	00		
codice_comando	codice_risposta	dword_1	...		



Nota

Le word e dword restituite sono in formato big-endian.

Comando write (scrittura)

Il comando **write** di FINS ha i seguenti formati:

- Se **tipo_var** è 82 o B0:

01	02	..		00			
codice_comando	tipo_var	indirizzo_partenza	fisso	canali_totali	word_1	..	..

- Se **tipo_var** è C2:

01	02	C2		00				..
codice_comando	tipo_var	indirizzo_partenza	fisso	canali_totali	dword_1	..	..	..

- Se **tipo_var** è 30:

01	02	30		00			
codice_comando	tipo_var	indirizzo_partenza	num_bit	bit_totali	bit		

I parametri possono avere i seguenti valori:

Parametro	Valori
codice_comando	01 02
tipo_var	82 (memoria TABLE in formato di numero intero a 16 bit) - C2 (memoria TABLE in formato a virgola mobile IEEE a 32-bit) - B0 (memoria VR in formato di numero intero a 16 bit) 30 (memoria VR in formato bit)
indirizzo_partenza	0 <= indirizzo_partenza <= numero di variabili – 1 <= FFFF
canali_totali	1 <= canali_totali <= dimensioni memoria – indirizzo_partenza + 1
bit_totali	1
bit	00 o 01

TJ1-MC__ risponde ai seguenti codici:

Condizione	Codice di risposta (esadecimale)	Descrizione
Tutti gli elementi validi	0000	OK
Tipo_var non valido	1101	Nessun tipo di area
Indirizzo_partenza non valido	1103	Errore di designazione dell'intervallo indirizzi
Numero_bit non valido	1103	Errore di designazione dell'intervallo indirizzi
Numero di elementi non valido (totali)	1104	Indirizzo fuori intervallo

4.2.5 Protocollo client FINS

Trajexia è in grado di avviare la comunicazione FINS utilizzando il comando BASIC **FINS_COMMS**. Per maggiori dettagli, fare riferimento alla descrizione del comando.

Sia il comando di lettura di memoria (0101) che il comando di scrittura di memoria (0102) sono supportati.

Questa funzionalità risulta utile per comunicare con un PLC OMRON, con un altro Trajexia o con un PC che sta eseguendo software di applicazione per server FINS.

Il comando di lettura di memoria consente di leggere la memoria di altri dispositivi dotati di funzionalità server FINS. Il comando di scrittura della memoria può essere utilizzato per scrivere dati su dispositivi dotati di funzionalità server FINS. Il comando restituisce uno dei seguenti valori, in base al risultato dell'esecuzione:

- 1 Il comando è stato eseguito con successo.
- 0 Il comando non è riuscito.
- 1 La richiesta non è stata inviata perché il client o il protocollo FINS è occupato.
- 2 Uno o più dei parametri richiesti non sono validi.
- 3 Area di memoria sorgente non valida.
- 4 La richiesta è stata inviata, ma il server remoto non ha risposto entro il periodo di timeout.
- 5 Il server remoto ha inviato un codice di risposta di errore.

4.3 Protocollo seriale

TJ1-MC__ è dotato di un connettore DB-9 che contiene due porte seriali:

- Porta 1: RS232
- Porta 2: RS422 or RS485, in base alle impostazioni dei selettori

Per maggiori dettagli, consultare il Trajexia Hardware Reference Manual.

Le due porte sono in grado di supportare entrambi questi protocolli in maniera indipendente:

- Master Host Link
- Slave Host Link
- Protocollo definito dall'utente:



Nota

La porta seriale (porta 1) NON PUÒ essere utilizzata per programmare l'unità.

4.3.1 Master Host Link

Se TJ1-MC__ è il master Host Link è possibile inviare comandi BASIC a uno slave Host Link, ad esempio un PC. Quando si invia un comando BASIC a uno slave Host Link, l'esecuzione del comando BASIC successivo resta in attesa fino a quando lo slave Host Link non invia una risposta.

È possibile utilizzare i seguenti comandi BASIC:

Comando BASIC	Descrizione
HLM_COMMAND	HLM_COMMAND esegue un comando Host Link specifico sullo slave.
HLM_READ	HLM_READ legge i dati dello slave Host Link trasferendoli alla memoria VR o TABLE.
HLM_STATUS	HLM_STATUS fornisce lo stato dell'ultimo comando del master Host Link.
HLM_TIMEOUT	HLM_TIMEOUT definisce il tempo di timeout del master Host Link.
HLM_WRITE	HLM_WRITE scrive i dati sullo slave Host Link trasferendoli dalla memoria VR o TABLE.
SETCOM	SETCOM configura la porta di comunicazione seriale e abilita i protocolli Host Link.

Comandi

I seguenti comandi Host Link sono quelli supportati per il protocollo del master Host Link:

Tipo	Codice intestazione	Nome	Funzione
Memoria di I/O Lettura	RR	CIO AREA READ	Legge il numero specificato di canali, a partire dal canale CIO/IR designato.
	RL	LR AREA READ	Legge il numero specificato di canali, a partire dal canale LR designato.
	RH	HR AREA READ	Legge il numero specificato di canali, a partire dal canale HR designato.
	RD	DM AREA READ	Legge il numero specificato di canali, a partire dal canale DM designato.
	RJ	AR AREA READ	Legge il numero specificato di canali, a partire dal canale AR designato.
	RE	EM AREA READ	Legge il numero specificato di canali, a partire dal canale EM designato.
Memoria di I/O Scrittura	WR	CIO AREA WRITE	Scrive i dati specificati in unità di canale, a partire dal canale CIO/IR designato.
	WL	LR AREA WRITE	Scrive i dati specificati in unità di canale, a partire dal canale LR designato.
	WH	HR AREA WRITE	Scrive i dati specificati in unità di canale, a partire dal canale HR designato.
	WD	DM AREA WRITE	Scrive i dati specificati in unità di canale, a partire dal canale DM designato.
	WJ	AR AREA WRITE	Scrive i dati specificati in unità di canale, a partire dal canale AR designato.
	WE	EM AREA WRITE	Scrive i dati specificati in unità di canale, a partire dal canale EM designato.
Stato CPU	SC	STATUS WRITE	Modifica la modalità operativa della CPU.

Tipo	Codice intestazione	Nome	Funzione
Verifica	TS	TEST	Restituisce, inalterato, un unico blocco che era stato inviato dal master.
Lettura codice modello PC	MM	PC MODEL READ	Legge il codice di modello della CPU.
Elaborazione comunicazioni Host Link	XZ	ABORT (solo comando)	Interrompe l'operazione in corso di esecuzione da parte di un comando Host Link e ritorna allo stato iniziale.
	**	INITIALIZE (solo comando)	Inizializza le procedure di controllo del trasferimento per tutte le unità Host Link.
	IC	Comando non definito (solo risposta)	Questa è la risposta ricevuta qualora il codice di intestazione di un comando non sia valido.



Nota

Il protocollo Host Link supporta solamente i comandi modalità C. Non supporta FINS.

Il protocollo del master Host Link supporta solo i comandi in frame singolo. La tabella di seguito mostra gli usi possibili del protocollo Host Link con i comandi BASIC e per quale modalità operativa della CPU (RUN, MON o PROG) il comando è valido.

Codice intestazione	Nome	Comando BASIC richiesto	RUN	MON	PRG
RR	CIO AREA READ	HLM_READ	Valido	Valido	Valido
RL	LR AREA READ	HLM_READ	Valido	Valido	Valido
RH	HR AREA READ	HLM_READ	Valido	Valido	Valido
RD	DM AREA READ	HLM_READ	Valido	Valido	Valido
RJ	AR AREA READ	HLM_READ	Valido	Valido	Valido
RE	EM AREA READ	HLM_READ	Valido	Valido	Valido

Codice intestazione	Nome	Comando BASIC richiesto	RUN	MON	PRG
WR	CIO AREA WRITE	HLM_WRITE	Non valido	Valido	Valido
WL	LR AREA WRITE	HLM_WRITE	Non valido	Valido	Valido
WH	HR AREA WRITE	HLM_WRITE	Non valido	Valido	Valido
WD	DM AREA WRITE	HLM_WRITE	Non valido	Valido	Valido
WJ	AR AREA WRITE	HLM_WRITE	Non valido	Valido	Valido
WE	EM AREA WRITE	HLM_WRITE	Non valido	Valido	Valido
SC	STATUS CHANGE	HLM_COMMAND	Valido	Valido	Valido
TS	TEST	HLM_COMMAND	Valido	Valido	Valido
MM	PC MODEL READ	HLM_COMMAND	Valido	Valido	Valido
XZ	ABORT (solo comando)	HLM_COMMAND	Valido	Valido	Valido
**	INITIALIZE (solo comando)	HLM_COMMAND	Valido	Valido	Valido
IC	Comando non definito (solo risposta)	–	Valido	Valido	Valido



Attenzione

I comandi del master Host Link devono essere eseguiti da un solo task di programma per evitare problemi di temporizzazione con il multitasking.

**Attenzione**

I comandi del master Host Link forniscono gli strumenti necessari per scambiare dati con lo slave Host Link. Il programma utente deve contenere routine appropriate di risoluzione degli errori per correggere gli eventuali errori di comunicazione ed effettuare nuovi tentativi, se necessario.

Codici di fine

Quelli che seguono sono i codici di fine definiti nel parametro **HLM_STATUS**:

Codice di fine	Descrizione	Causa probabile	Soluzione
\$00	Completamento normale	Non sussiste alcun problema.	N/D
\$01	Non eseguibile in modalità RUN	Il comando inviato non può essere eseguito quando il PC si trova in modalità RUN.	Controllare la relazione tra il comando e la modalità del PC.
\$13	Errore nella FCS	La FCS non è corretta.	Influenza dei disturbi: trasferire nuovamente il comando.
\$14	Errore di formato	- Il formato del comando è errato. - Un comando che non può essere diviso è stato diviso. - La lunghezza del frame è inferiore alla lunghezza minima per il comando applicabile.	Verificare il formato e trasferire nuovamente il comando.
\$15	Errore dei dati nel numero immesso	I dati ricadono fuori dell'intervallo specificato oppure sono troppo lunghi.	Correggere gli argomenti del comando e trasferirlo nuovamente.

Codice di fine	Descrizione	Causa probabile	Soluzione
\$18	Errore nella lunghezza del frame	La lunghezza massima del frame pari a 131 byte è stata superata.	Verificare il comando e trasferirlo nuovamente.
\$19	Non eseguibile	Non si sono ottenuti i diritti di accesso.	Ottenere i diritti di accesso.
\$21	Esecuzione impossibile a causa di un errore della CPU.	Il comando non può essere eseguito poiché si è verificato un errore della CPU.	Riavviare la CPU.
\$100	Timeout ACK dello slave Host Link	–	–
\$200	Errore di indirizzo del comando IC	–	–

Impostazione

Per impostare la porta seriale di TJ1-MC__ per il protocollo del master Host Link è necessario il comando **SETCOM**. Impostare il comando come segue:

SETCOM(velocità trasmissione, bit_dati, bit_stop, parità, porta, 6)

Dopo avere impostato questo comando, è possibile utilizzare i comandi **HLM_READ**, **HLM_WRITE** e **HLM_COMMAND** per leggere e scrivere i dati utilizzando Host Link.

Timeout

Il meccanismo di timeout viene implementato per impedire che il task BASIC rimanga sospeso per un periodo di tempo prolungato a causa di una comunicazione difettosa o inesistente. Il parametro **HLM_TIMEOUT** specifica il periodo di timeout. Tale periodo corrisponde al tempo massimo di attesa di un task di programma per ricevere una risposta dopo avere inviato il comando.

Allo scadere del periodo di timeout, il parametro **HLM_STATUS** registra lo stato del comando mentre il task BASIC continua.

Il parametro **HLM_TIMEOUT** specifica il periodo di timeout per tutti i comandi e per tutte le porte.

Stato

Il parametro **HLM_STATUS** contiene lo stato dell'ultimo comando master Host Link inviato alla porta specificata. Il parametro indica lo stato dei comandi **HLM_READ**, **HLM_WRITE** e **HLM_COMMAND**. I bit di stato sono:

Bit	Nome	Descrizione
0 – 7	Codice di fine	Il codice di fine è: - il codice di fine definito dallo slave Host Link slave, quando un problema si è verificato nella stringa di dati del comando inviato oppure - un codice di fine definito dal master Host Link slave, quando un problema si è verificato nella stringa di dati della risposta ricevuta.
8	Errore di timeout	Un errore di timeout si verifica se non viene ricevuta alcuna risposta entro il periodo di timeout. Questo significa che la comunicazione è interrotta.
9	Comando non riconosciuto	Questo stato indica che lo slave non ha riconosciuto il comando e ha restituito una risposta IC.

Il parametro **HLM_STATUS** avrà valore 0 se non si sono verificati problemi. Nel caso in cui il valore del parametro fosse diverso da 0, occorre programmare nel programma utente in BASIC l'azione appropriata, come ad esempio un nuovo tentativo o un arresto di emergenza. Ogni porta dispone di un parametro **HLM_STATUS**. Il modificatore **PORT** è richiesto per specificare la porta.

Esempi

In questi esempi si presume che sia presente la seguente configurazione:

- Un Trajexia con un TJ1-MC__.
- Uno slave PC, con indirizzo nodo 13.
- Una connessione dalla porta seriale di TJ1-MC__ al PC.
La porta seriale utilizza la comunicazione RS422.

Esempio	Lettura dati dal PC utilizzando HLM_READ .
Codice BASIC	' Imposta il master Host Link per la porta 2 SETCOM(9600,7,2,2,6) ' Indirizzo di origine: CIO/IR 002 ' Quantità di dati: 2 canali ' Indirizzo di destinazione: VR(0) HLM_READ(2,13,PLC_IR,2,2,MC_VR,0)
Comunicazione Host Link	• Da master Host Link a slave Host Link: @13RR0002000242* • Da slave Host Link a master Host Link: @13RR000101010241*
Risultato	• Indirizzo VR = 0: valore = 257.0000 • Indirizzo VR = 1: valore = 258.0000
Esempio	Scrittura dei dati sul PC utilizzando HLM_WRITE .
Codice BASIC	' Indirizzo di origine: TABLE(18) ' Quantità di dati: 2 canali ' Indirizzo di destinazione: LR 014 TABLE(18,\$0701,\$0702) HLM_WRITE(2,13,PLC_LR,14,2,MC_TABLE,18)
Comunicazione Host Link	• Da master Host Link a slave Host Link: @13WL0014070107025F* • Da slave Host Link a master Host Link: @13WL0059*
Risultato	• Indirizzo LR = 0: valore = 701 (esadecimale) • Indirizzo LR = 1: valore = 702 (esadecimale)

Esempio Invio del comando **TS** (test) al PC usando **HLM_COMMAND**.

Codice BASIC **HLM_COMMAND(HLM_TEST,2,13)**

Comunicazione Host Link

- Da master Host Link a slave Host Link:
@13TSMCW151 TEST STRING2A*
- Da slave Host Link a master Host Link:
@13TSMCW151 TEST STRING2A*

Risultato **HLM_STATUS PORT(2) = 0**, che implica una comunicazione corretta.

Esempio Impostazione del PC in modalità MON utilizzando **HLM_COMMAND**.

Codice BASIC **HLM_COMMAND(HLM_STWR,2,13,2)**

Comunicazione Host Link

- Da master Host Link a slave Host Link:
@13SC0250*
- Da slave Host Link a master Host Link:
@13SC0052*

Risultato Il PC funziona in modalità MON. Notare che ciò è necessario per la scrittura dei dati sul PC utilizzando **HLM_WRITE**.

Esempio Lettura del codice modello del PC utilizzando **HLM_COMMAND** (timeout).

Codice BASIC **HLM_TIMEOUT=500**
'Indirizzo di destinazione: VR(100)
HLM_COMMAND(HLM_MREAD,2,13,MC_VR,100)

Comunicazione Host Link

- Da master Host Link a slave Host Link:
@13MM42*
- Da slave Host Link a master Host Link:
nessuna risposta

Risultato Dal momento che il master non ha ricevuto una risposta dal PC, **HLM_STATUS PORT(2)** ha valore 256 (il bit 8 è impostato) dopo 500 cicli di servoazionamento.

4.3.2 Slave Host Link

Se TJ1-MC__ è lo slave Host Link, un master Host Link (ad esempio un terminale programmabile) può leggere e scrivere i dati da e su TJ1-MC__. La mappatura tra lo slave e il master è:

Memoria TJ1-MC__	Mappatura Host Link	Intervallo indirizzi
VR	CIO	Da 0 a 1023
TABLE	DM	Da 0 a 63999

È possibile utilizzare i seguenti comandi BASIC:

Comando BASIC	Descrizione
SETCOM	SETCOM configura la porta di comunicazione seriale e abilita i protocolli Host Link.
HLS_NODE	HLS_NODE definisce il numero di modulo dello slave per il protocollo slave Host Link.
HLS_MODEL	HLS_NODE definisce il codice modello di TJ1-MC__ per il protocollo slave Host Link.

Comandi

I comandi supportati per il protocollo slave Host Link sono riportati nella tabella di seguito. Il protocollo supporta il trasferimento a frame singolo e il trasferimento a più frame.

Tipo	Codice intestazione	Nome	Funzione
Lettura della memoria di I/O	RR	CIO AREA READ	Legge il numero specificato di canali dalla memoria VR, a partire dal canale designato.
	RD	DM AREA READ	Legge il numero specificato di canali dalla memoria TABLE, a partire dal canale designato.

Tipo	Codice intestazione	Nome	Funzione
Scrittura della memoria di I/O	WR	CIO AREA WRITE	Scrive i dati specificati in unità di canale nella memoria VR, a partire dal canale designato.
	WD	DM AREA WRITE	Scrive i dati specificati in unità di canale nella memoria TABLE, a partire dal canale designato.
Verifica	TS	TEST	Restituisce, inalterato, un unico blocco che era stato inviato dal master.
Lettura codice modello del PC	MM	PC MODEL READ	Legge il codice modello di TJ1-MC__ in base a quanto specificato dal parametro HLS_MODEL .
Registrazione e lettura dell'area della memoria di I/O	QQMR	REGISTER I/O MEMORY	Registra nella memoria TABLE di I/O i contenuti dell'effettiva configurazione di I/O.
	QQIR	READ I/O MEMORY	Legge tutti in una volta i canali/bit registrati nella memoria di I/O.
Elaborazione della comunicazione Host Link	XZ	ABORT (solo comando)	Interrompe l'operazione in corso di esecuzione da parte di un comando Host Link e ritorna allo stato iniziale.
	**	INITIALIZE (solo comando)	Inizializza le procedure di controllo del trasferimento per tutte le unità Host Link.
	IC	Comando non definito (solo risposta)	Questa è la risposta ricevuta qualora il codice di intestazione di un comando non sia valido.

Codici di fine

Si tratta dei codici di fine delle risposte che sono restituiti all'interno del frame di risposta:

Codice di fine	Descrizione	Causa probabile	Soluzione
0	Completamento normale	Non sussiste alcun problema.	N/D
13	Errore nella FCS	La FCS non è corretta.	Controllare il metodo di calcolo della FCS. Se c'è stata un'influenza da parte dei disturbi, trasferire nuovamente il comando.
14	Errore di formato	- Il formato del comando è errato. - Un comando che non può essere diviso è stato diviso. - La lunghezza del frame è inferiore alla lunghezza minima per il comando applicabile.	Verificare il formato e trasferire nuovamente il comando.
15	Errore dei dati nel numero immesso	I dati ricadono fuori dell'intervallo specificato oppure sono troppo lunghi.	Correggere gli argomenti del comando e trasferirlo nuovamente.
18	Errore nella lunghezza del frame	La lunghezza massima del frame pari a 131 byte è stata superata.	Verificare i dati e trasferire nuovamente il comando.
19	Non eseguibile	Un batch di memoria di I/O è stato eseguito quando le voci da leggere non erano memorizzate.	Memorizzare le voci da leggere prima di tentare una lettura in batch.
A3	Interrotto per un errore della FCS nella trasmissione dei dati	Si è verificato un errore della FCS nel secondo frame o in uno successivo.	Correggere i dati del comando e trasferirlo nuovamente.

Codice di fine	Descrizione	Causa probabile	Soluzione
A4	Interrotto per un errore di formato nella trasmissione dei dati	Il formato del comando non corrispondeva al numero di byte nel secondo frame o in uno successivo.	Correggere i dati del comando e trasferirlo nuovamente.
A5	Interrotto per un errore dei dati nel numero immesso nella trasmissione dei dati	Si è verificato un errore dei dati nel numero immesso nel secondo frame o in uno successivo oppure si è verificato un errore nella lunghezza dei dati.	Correggere i dati del comando e trasferirlo nuovamente.
A8	Interrotto per un errore nella lunghezza del frame nella trasmissione dei dati	La lunghezza del secondo frame o di uno successivo ha superato il valore massimo pari a 128 byte.	Correggere i dati del comando e trasferirlo nuovamente.

Impostazione

Per impostare la porta seriale di TJ1-MC__ per il protocollo dello slave Host Link è necessario il comando **SETCOM**. Impostare il comando come segue:

SETCOM(velocitàtrasmissione, bit_dati, bit_stop, parità, porta, 5)

Con il comando così impostato, TJ1-MC__ risponde ai comandi Host Link dal master con il numero di nodo specificato. È possibile impostare tale numero di nodo con il parametro **HLS_NODE**.

Esempio

In questo esempio si presume che sia presente il seguente assetto:

- Un Trajexia con un TJ1-MC__.
- Un terminale programmabile NS8
- Un collegamento dalla porta seriale di TJ1-MC__ al terminale programmabile. La porta seriale utilizza la comunicazione RS232C

Esempio

Codice BASIC

Configurazione dello slave Host Link.

```
' Definisce il nodo dello slave Host Link
HLS_NODE = 15
' Definisce il codice modello dello slave Host Link
HLS_MODEL = $FA
' Imposta lo slave Host Link per la porta 1
SETCOM(9600,7,2,2,1,5)
```

Risultato

TJ1-MC__ è in grado di comunicare con il terminale programmabile.

4.3.3 Protocollo definito dall'utente

È possibile implementare un protocollo definito dall'utente utilizzando i seguenti comandi:

Comando BASIC	Descrizione
SETCOM	SETCOM configura la porta di comunicazione seriale e abilita i protocolli Host Link.
GET	GET assegna a una variabile il codice ASCII di un carattere ricevuto.
INPUT	INPUT assegna alle variabili specificate valori numerici di stringa d'ingresso.
KEY	KEY restituisce TRUE o FALSE quando un carattere è stato ricevuto oppure non è stato ricevuto.
LINPUT	LINPUT assegna a una matrice di variabili il codice ASCII dei caratteri ricevuti.
PRINT	PRINT invia una serie di caratteri a un dispositivo di uscita seriale.

Esempio

Si supponga che sia presente la seguente configurazione:

- Un Trajexia con un TJ1-MC__.
- Un sistema F500 di OMRON Vision.
- Una connessione dalla porta seriale di TJ1-MC__ all'F500. La porta seriale utilizza la comunicazione RS232 (porta 1).

Questo programma invia un comando Vision attraverso la porta seriale, legge la risposta dal sistema Vision, la scrive nelle variabili VR e stampa i risultati nella finestra del terminale di Trajexia Tools.

```
' Nel programma STARTUP
' Impostazione della porta RS232 per il sistema di visione
SETCOM(38400,8,1,0,1,0)
' Nel programma applicativo
loop:
  ' Attivazione, fronte di salita nel sistema virtuale
  WAIT UNTIL IN(30)=0
  WAIT UNTIL IN(30)=1
  ' Ripulisce lo schermo
  PRINT CHR(27);"[2J"

  ' Vuota il buffer
  GOSUB clear_buffer

  ' Invia il comando alla porta seriale in base a VR(10)
  IF vision_command=v_measure THEN
    PRINT #1, "M"
    PRINT ">> M"
  ELSEIF vision_command=v_date THEN
    PRINT #1, "DATE"
    PRINT ">> DATE"
  ELSEIF vision_command=v_scene THEN
    PRINT #1,"SCENE ";scene_n
    PRINT ">> SCENE"
  ENDIF

  'Verifica la risposta
  GOSUB read_buffer

  GOTO loop
read buffer:
count=0
resp_status=0
k=-1
TICKS=5000
REPEAT
  IF KEY#1 THEN
    count=count+1
    GET#1, k
    'PRINT k;count
```

```
TABLE(count,k)
'PRINT count
ENDIF
UNTIL TICKS<0 'OR k=13
PRINT "Ricevuti ";count[0];" caratteri"
FOR i=1 TO count
  IF TABLE(i)<>13 THEN
    PRINT CHR(TABLE(i))
  ELSE
    PRINT "'cr'"
  ENDIF
NEXT i
IF TICKS<0 THEN
  PRINT "Timeout nella comunicazione con F500"
  resp_status=3
ELSEIF TABLE(count-2)=79 AND TABLE(count-1)=75 THEN
  PRINT "Risposta OK"
  resp_status=1
ELSE
  PRINT "Risposta non corretta"
  resp_status=2
ENDIF
PRINT "Lo stato della risposta ċ :";resp_status[0]
RETURN
clear_buffer:
PRINT "Svuotamento in corso..."
WHILE KEY#1
  GET#1,k
  PRINT k
WEND
PRINT "Vuoto!"
RETURN
```

4.4 PROFIBUS

4.4.1 Introduzione

PROFIBUS è uno standard internazionale di fieldbus aperto. TJ1-PRT di Trajexia consente Trajexia di comunicare con una rete PROFIBUS. Scambia dati tra il master PROFIBUS e TJ1-MC__. A questo scopo usa le variabili VR di Trajexia.

4.4.2 Configurazione della comunicazione

TJ1-PRT è dotato di due selettori del numero di nodo. È possibile utilizzare i selettori del numero di nodo per assegnare a TJ1-PRT un indirizzo di rete PROFIBUS. Prima di accendere l'alimentazione del sistema Trajexia è necessario assegnare un indirizzo a TJ1-PRT.

Per inizializzare TJ1-PRT, utilizzare il comando BASIC **PROFIBUS**:

PROFIBUS(numero_unità, 2, 1, inizio_uscite, conteggio_uscite, inizio_ingressi, conteggio_ingressi)

dove:

- **numero_unità** è il numero del modulo TJ1-PRT.
- **inizio_uscite** è l'indirizzo iniziale dell'intervallo dei dati di uscita delle variabili VR.
- **conteggio_uscite** è il numero di variabili VR nell'intervallo dei dati di uscita.
- **inizio_entrare** è l'indirizzo iniziale dell'intervallo dei dati di entrata delle variabili VR.
- **conteggio_entrare** è il numero di variabili VR nell'intervallo dei dati di entrata.



Nota

Il numero massimo di variabili VR per lo scambio dei dati è 122.

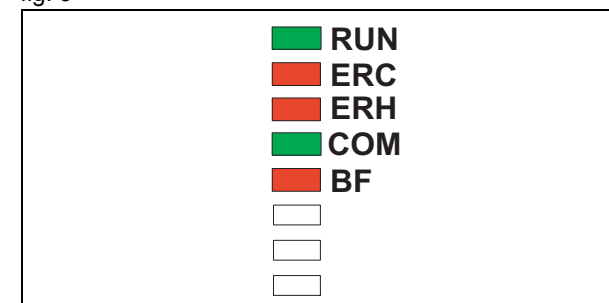
Dopo che è stato eseguito il comando **PROFIBUS(numero_unità, 2, ...)**, le matrici dei dati vengono automaticamente scambiate. I dati scambiati tra TJ1-PRT e il master PROFIBUS sono in formato di numeri interi a 16 bit. Ciascun canale scambiato è compreso tra -32768 e 32767.

Una variabile VR può contenere un numero a 24 bit e anche frammenti. Lo scambio con il master PROFIBUS non supporta i valori che cadono al di fuori dell'intervallo -32768 – +32767 e i frammenti.

Un esempio di sequenza per configurare il modulo TJ1-PRT è il seguente:

1. Impostare il numero di modulo con i due selettori rotativi del modulo TJ1-PRT.
2. Attivare l'alimentazione del sistema. Il LED **RUN** si accende. Il LED **ERH** lampeggia.
3. Creare un programma BASIC contenente il comando **PROFIBUS(2,2,1,10,7,150,3)**. In questo esempio, il sistema inizializza un modulo TJ1-PRT con il numero di modulo 2. Il sistema invia sette canali di uscita ricevuti dal master a VR(10) e VR(16) e tre canali d'ingresso da VR(150) a VR(152) al master.
4. Se la configurazione riesce, il LED **RUN** e il LED **COMM** si accendono. La comunicazione è a questo punto attiva.

fig. 5



Per configurare CJ1-PRM21 con CX-PROFIBUS, procedere come segue:

1. Avviare lo strumento software CX-PROFIBUS.
2. Fare clic con il pulsante destro sulla struttura ad albero MyNetwork.
3. Selezionare **Add Device....**

4. Selezionare la scheda master PROFIBUS.
5. Fare clic su **OK**.

6. Aprire **Device Catalogue** dal menu **View**.

fig. 6



fig. 7

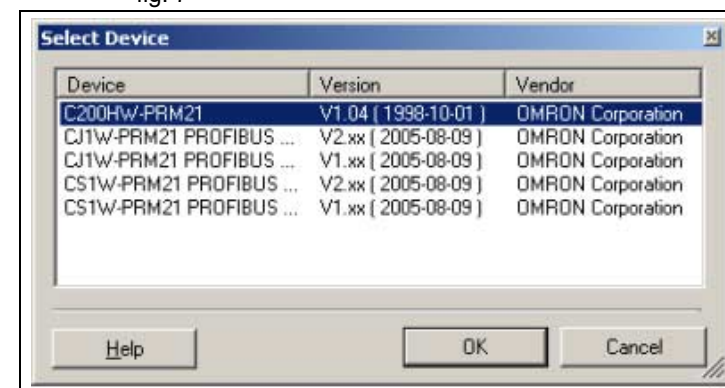
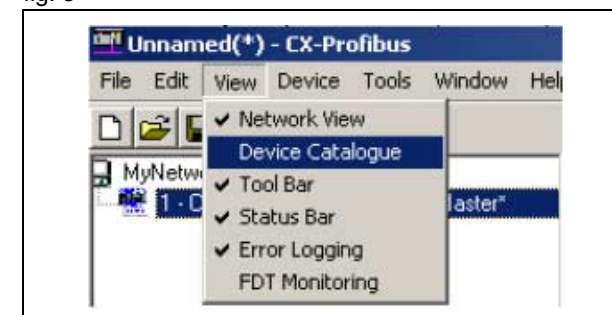


fig. 8



7. Fare clic su **Install GSD Files....** Il file GSD si trova sul CD di Trajexia Tools. È possibile anche scaricarlo dall'Area di download del sito web di OMRON.
8. Fare clic su **Update**. TJ1-PRT compare nell'elenco.
9. Selezionare OMRON TJ1-PRT dall'elenco e fare clic su **Add Device**.

fig. 9

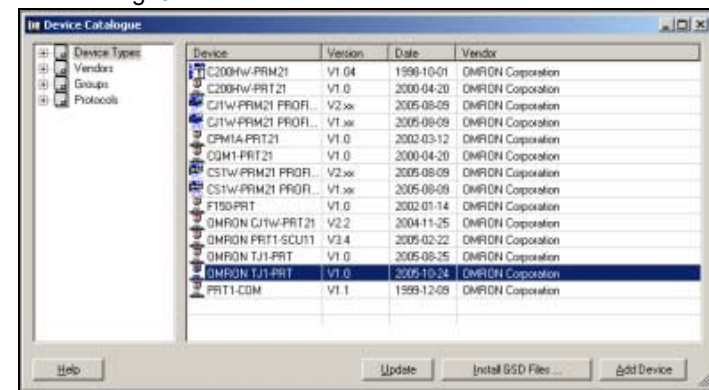
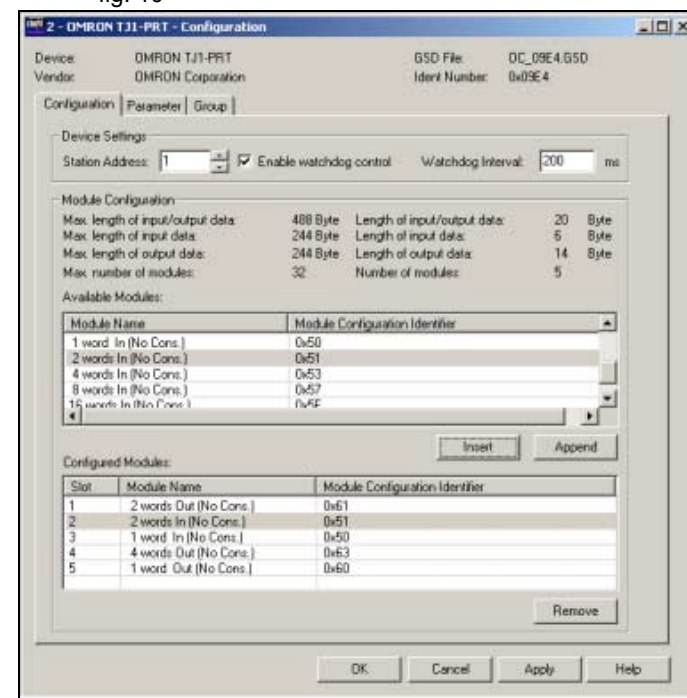


fig. 10

10. Fare doppio clic sul modulo slave TJ1-PRT della struttura ad albero MyNetwork.
11. Impostare il numero di nodo nel campo **Station Address**.
12. Aggiungere (**Insert**) i moduli di ingresso e uscita all'elenco di configurazione di seguito.
13. Assicurarsi che il numero dei canali di ingresso e dei canali di uscita dei moduli selezionati sia uguale al numero selezionato con il comando PROFIBUS.
14. Fare clic su **OK**.



Per configurare CJ1W-PRM21 con CX-PROFIBUS, procedere come segue:

1. Fare doppio clic sul modulo master della struttura ad albero MyNetwork.
2. Impostare lo **Station Address** e lo **Unit Number**.

fig. 11

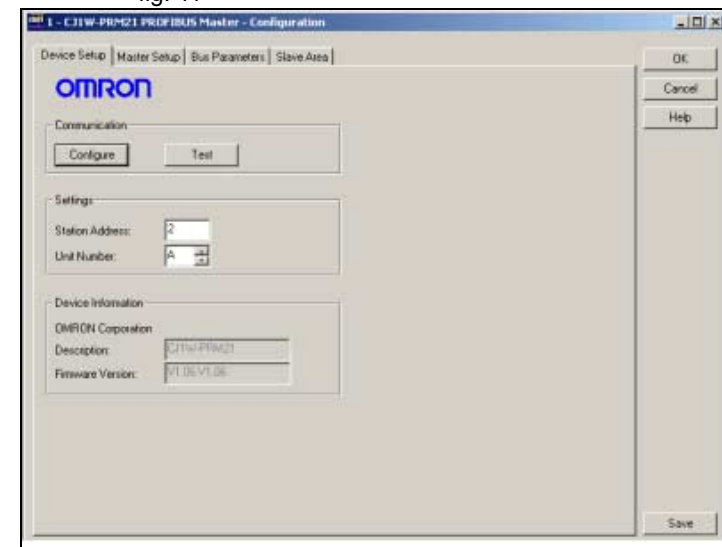
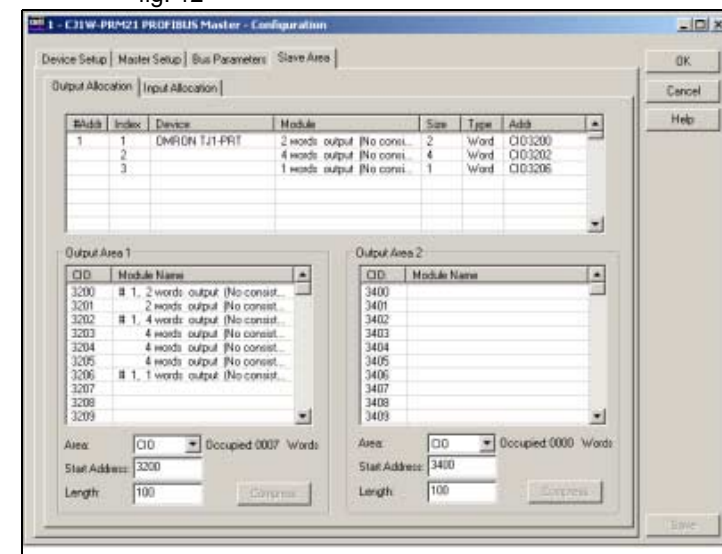


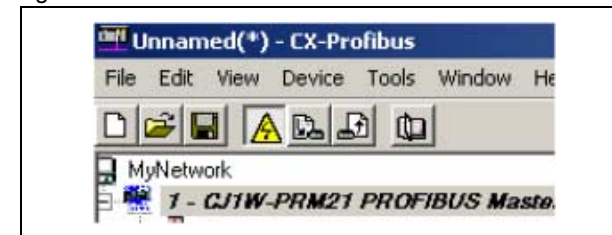
fig. 12

3. Selezionare la scheda **Slave area**.
4. Impostare il campo **Start Address** di **Output Area 1** e **Input Area 1**.
5. Memorizzare il progetto.



6. Fare clic sul pulsante della barra degli strumenti **Device Online/Offline (Toggle)** per collegarsi in rete.
7. Fare clic sul pulsante della barra degli strumenti **Device Download** per scaricare i parametri.

fig. 13



4.4.3 Stato della comunicazione

TJ1-PRT è in grado di fornire informazioni sullo stato a TJ1-MC___. In BASIC è possibile recuperare le informazioni sullo stato con il comando **PROFIBUS** (numero_unità,4,0). Il risultato fornisce le seguenti informazioni:

Bit	Valore	Descrizione
0	0	Impossibile effettuare la configurazione dello scambio di dati di I/O
	1	Configurazione dello scambio di dati di I/O riuscita
1	0	Dati di I/O non disponibili
	1	Dati di I/O disponibili
2	0	Scambio dati attivo in modalità OPERATE
	1	Scambio dati attivo in modalità CLEAR

4.5 DeviceNet

4.5.1 Introduzione

DeviceNet è uno standard internazionale di fieldbus aperto basato sul protocollo CAN. TJ1-DRT consente al sistema Trajexia di comunicare a una rete DeviceNet. Scambia dati tra un master DeviceNet e TJ1-MC__. A questo scopo usa le variabili VR di Trajexia.

4.5.2 Configurazione della comunicazione

TJ1-DRT è dotato di due selettori del numero di nodo. È possibile utilizzare i selettori del numero di nodo per assegnare al modulo TJ1-DRT un numero di nodo

L'intervallo dei numeri di nodo di DeviceNet è compreso tra 0 e 63. Se mediante il selettore si seleziona un numero di nodo non compreso in questo intervallo, ciò corrisponde a selezionare il numero di nodo impostato dal software. I nodi che consentono l'impostazione del software sono quelli compresi tra 64 e 99.

Per inizializzare TJ1-DRT, utilizzare il comando BASIC **DEVICENET**:

DEVICENET(numero_unità, 2, 1, inizio_uscite, conteggio_uscite, inizio_ingressi, conteggio_ingressi)

dove:

- **numero_unità** è il numero del modulo TJ1-DRT.
- **inizio_uscite** è l'indirizzo iniziale dell'intervallo dei dati di uscita delle variabili VR.
- **conteggio_uscite** è il numero di variabili VR nell'intervallo dei dati di uscita.
- **inizio_entrare** è l'indirizzo iniziale dell'intervallo dei dati di entrata delle variabili VR.
- **conteggio_entrare** è il numero di variabili VR nell'intervallo dei dati di entrata.



Nota

Il numero massimo di variabili VR per lo scambio dei dati è 32.



Nota

Se si utilizza un master DeviceNet di OMRON, è consigliabile selezionare conteggio_ingressi o conteggio_uscite con un valore di 4, 8, 16 o 32 per le variabili VR.

Dopo che è stato eseguito il comando

DEVICENET(numero_unità, 2, ...), le matrici dei dati vengono automaticamente scambiate. I dati scambiati tra TJ1-DRT e il master DeviceNet sono in formato di numeri interi a 16 bit. Ciascun canale scambiato è compreso tra -32768 e 32767.

Una variabile VR può contenere un numero a 24 bit e anche frammenti. Lo scambio con il master DeviceNet non supporta i valori che cadono al di fuori dell'intervallo -32768 – +32767 o i frammenti.

Configurazione della rete DeviceNet

Per configurare il master DeviceNet CJ1W/CS1W-DRM21 di OMRON affinché scambi le variabili VR con il sistema Trajexia, procedere come segue:

1. Avviare il CX-Integrator nello strumento software CX-ONE.
2. Selezionare Network dal menu Insert.
3. Selezionare DeviceNet dalla schermata Select Network.
Viene visualizzata la finestra Network.

4. Selezionare CJ1W-DRM21 dall'elenco degli adattatori di comunicazione OMRON.

fig. 14

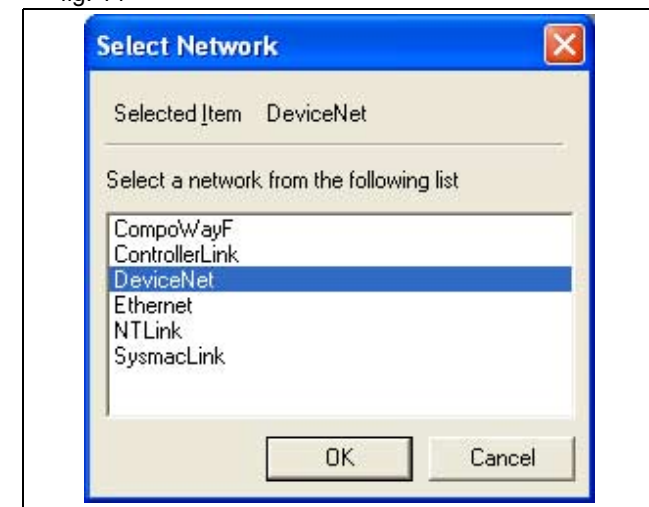
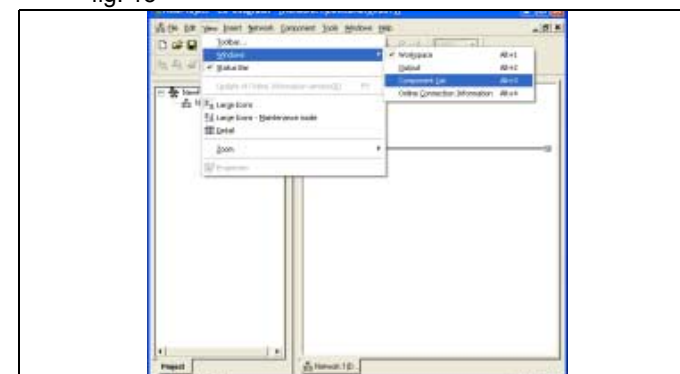


fig. 15



5. Trascinare CJ1W-DRM21 alla finestra Network.

6. Installare il file **EDS** da CX-Integrator.
7. Selezionare No nella finestra di dialogo.
L'icona non è necessaria.

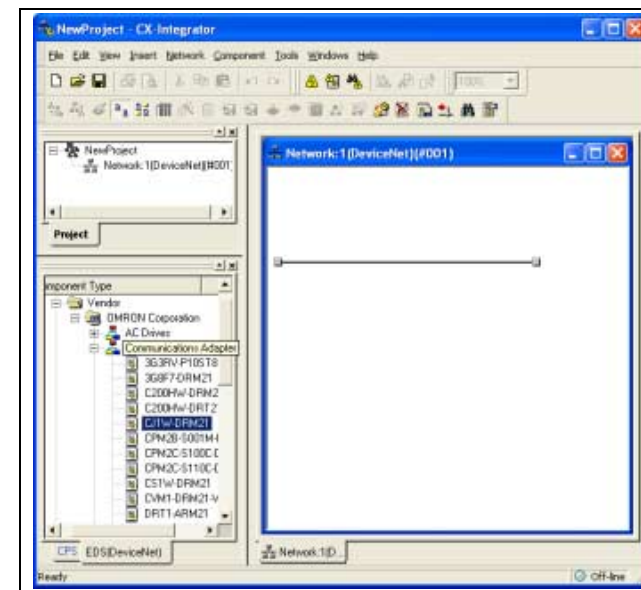
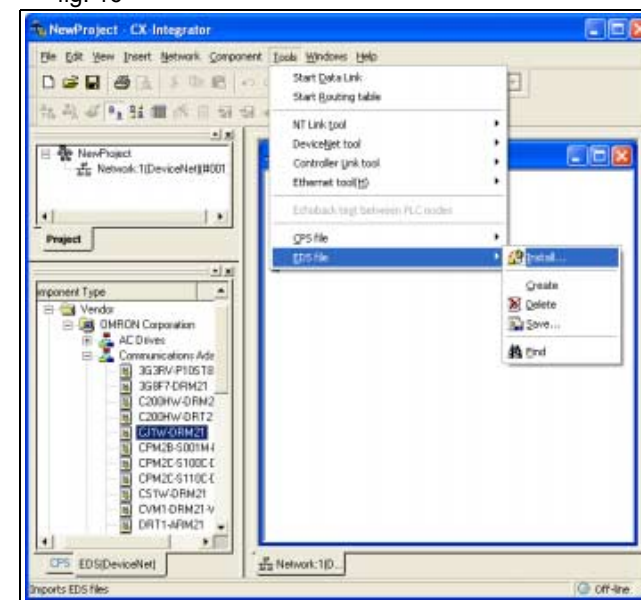


fig. 16



8. Registrare lo slave sul master, fare clic con il pulsante destro sull'icona #01TJ1-DRT.
9. Fare doppio clic sull'icona Master.
10. Selezionare il dispositivo TJ1-DRT.

fig. 17

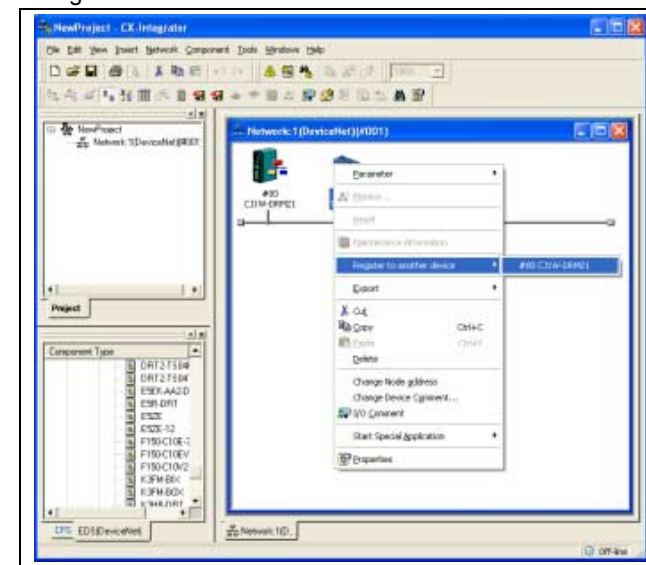
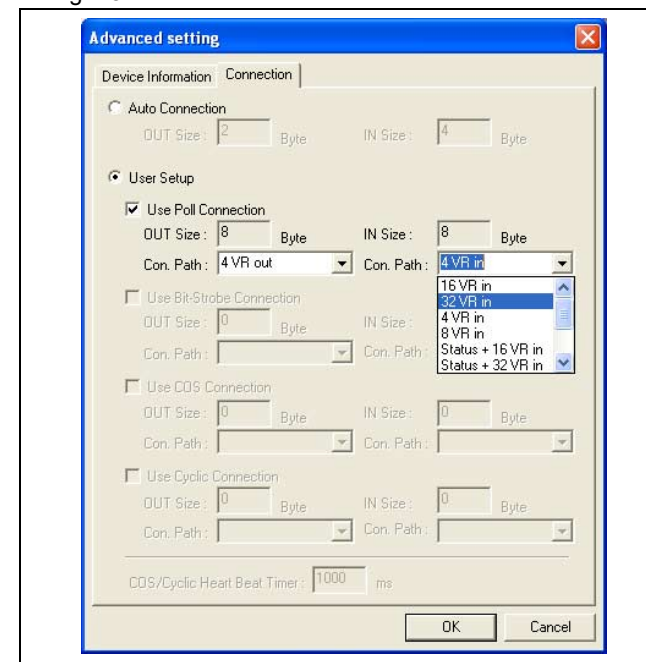


fig. 18

11. Fare clic su Advanced Setup.
12. Fare clic sulla scheda Connection.
13. Selezionare User Setup.
14. Selezionare Poll Connection.
15. Selezionare Con.Path.
16. Scegliere il numero di variabili che è stato selezionato per comunicazione DeviceNet.
17. Fare clic su OK per confermare tutte le finestre di dialogo.
18. Selezionare Work Online dal menu Network.
19. Selezionare Parametro dal menu Component.
20. Fare clic con il pulsante destro sull'icona Master.
21. Selezionare Parameter Download.



4.5.3 Stato della comunicazione

TJ1-DRT può fornire lo stato della comunicazione sia a TJ1-MC__ sia al master DeviceNet. In BASIC è possibile recuperare le informazioni sullo stato con il comando DeviceNet (numero_unità,4,0). Il risultato fornisce le seguenti informazioni:

Bit	Valore	Descrizione
0	0	DeviceNet (numero_unità, 2, ...) non ancora eseguito
	1	DeviceNet (numero_unità, 2, ...) eseguito senza errori
1	0	Nessun collegamento di I/O DeviceNet
	1	Collegamento di I/O DeviceNet in funzione
2	0	Le variabili VR dell'intervallo dati di uscita sono state aggiornate
	1	Le variabili VR dell'intervallo dati di uscita non sono state ancora aggiornate
3	0	Le dimensioni del collegamento di I/O DeviceNet corrispondono al comando DeviceNet (numero_unità, 2,...)
	1	Le dimensioni del collegamento di I/O DeviceNet non corrispondono ancora al comando DeviceNet (numero_unità, 2,...)
4 – 7	0	Sempre zero
8	0	Alimentazione di rete OK
	1	Guasto nell'alimentazione di rete
9	0	Non si è verificato nessun BUSOFF
	1	Si è verificato un BUSOFF
10	0	Nessun errore di duplicazione dell'indirizzo del nodo
	1	Errore di duplicazione dell'indirizzo del nodo
11 –	0	Riservato

È possibile recuperare le informazioni sullo stato presenti nel master DeviceNet selezionando un percorso di connessione che include le informazioni sullo stato. Le informazioni sullo stato includono un bit. Il bit 2 indica che la tensione della rete è scesa sotto al livello impostato nei parametri DeviceNet di TJ1-DRT. È possibile impostare i parametri DeviceNet di TJ1-DRT utilizzando un configuratore DeviceNet. La tensione predefinita è di 11 V.

4.6 MECHATROLINK-II

Il protocollo MECHATROLINK-II è un bus seriale concepito per controllare il movimento in maniera deterministica.

Il numero di dispositivi MECHATROLINK-II determina il tempo di ciclo dello scambio di dati:

- Se sono presenti da 1 a 4 dispositivi, il tempo di ciclo può essere 0,5 ms, 1 ms o 2 ms.
- Se sono presenti da 5 a 8 dispositivi, il tempo di ciclo può essere di 1 ms o 2 ms.
- Se sono presenti da 9 a 16 dispositivi, il tempo di ciclo è di 2 ms.

La trasmissione ciclica attraversa due stadi:

- TJ1-ML__ invia il comando di riferimento agli slave MECHATROLINK.
- Gli slave inviano la retroazione e le informazioni sullo stato a TJ1-ML__.

MECHATROLINK-II utilizza un clock di sincronizzazione e il broadcast dei messaggi per garantire che tutti gli slave eseguano il comando allo stesso tempo.

Inoltre, altre informazioni vengono trasferite a una velocità inferiore, ad esempio i parametri di lettura e scrittura.

Esistono comandi BASIC specifici per comunicare direttamente alle unità slave di MECHATROLINK.

- **DRIVE_CLEAR:** Questo comando reimposta un allarme nel servoazionamento di MECHATROLINK attraverso un messaggio MECHATROLINK.
- **OP(45,ON):** Questo messaggio attiva un'uscita in un modulo di I/O MECHATROLINK remoto.

5 Interfaccia Trajexia Tools

5.1 Introduzione

Trajexia Tools è lo strumento software che consente di programmare il sistema Trajexia. Questo software consente ai tecnici addetti alle applicazioni di gestire i progetti Trajexia e modificare i programmi. Include alcuni utili strumenti descritti più oltre in questo capitolo, come ad esempio i comandi Run/Stop/Step (esecuzione/arresto/azionamento passo-passo) di programmi individuali, l'aggiunta di punti di interruzione, l'esecuzione di comandi diretti, la lettura/scrittura di variabili, le funzioni di oscilloscopio e la programmazione dei servoazionamenti.

La connessione con TJ1-MC__ è tramite Ethernet. Occorre regolare le impostazioni di comunicazione prima del collegamento a un'unità.

Lo strumento software Trajexia Tools è stato creato per funzionare on-line con un TJ1-MC__.

Trajexia Tools include:

- Lo strumento software per TJ1-MC__ (Motion Perfect 2)
- CX-Server
- CX-Drive per programmare e impostare i servoazionamenti e l'inverter.

È possibile usare Trajexia Tools per programmare, attraverso una comunicazione seriale, altre unità di controllo assi:
C200HW-MC402E, R88A-MCW151-E e R88A-MCW151-DRT-E.

5.2 Specifiche e connessioni

5.2.1 Specifiche del PC

Le specifiche del PC da utilizzare con Trajexia Tools sono le seguenti:

Descrizione	Specifiche minime	Specifiche consigliate
CPU	Pentium 300MHz	Pentium, 1GHz
RAM	64 MB	256 MB
Spazio su disco rigido	140 MB	140 MB

Descrizione	Specifiche minime	Specifiche consigliate
Sistema operativo	Windows TM 98	Windows XP
Display	800x600 256 colori	1024x768 Colori a 24 bit
Comunicazioni	Ethernet 10BaseT	Ethernet 100BaseT
Internet	Explorer V5.0	Explorer V6.0

Utilizzare l'ultima versione di Trajexia Tools. Gli aggiornamenti sono disponibili presso il proprio distributore. Il software è anche disponibile sul sito web di Trajexia: www.trajexia.com.

5.2.2 Installazione del software Trajexia Tools

1. Inserire il CD-ROM di Trajexia Tools nell'unità CD-ROM del PC.
2. Il programma di installazione di Trajexia Tools si avvia automaticamente.
3. Se il programma di installazione di Trajexia Tools non si avvia automaticamente, avviarlo manualmente: eseguire **setup.exe** nella directory principale del CD.
4. Nell'elenco a discesa, selezionare la lingua da utilizzare. Fare clic su **OK**.
5. Si apre la finestra di installazione di Trajexia Tools. Fare clic su **Next**.

fig. 1

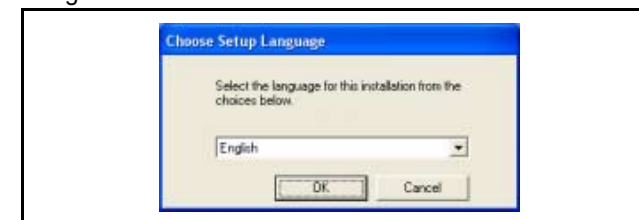


fig. 2



6. Fare clic su **Yes** per accettare l'accordo di licenza e proseguire.

7. Digitare il proprio nome nel campo **Name**.
8. Digitare il nome dell'azienda nel campo **Company**.
9. Digitare il numero di licenza nei campi **Licence**. Il numero di licenza è riportato sull'etichetta attaccata sulla custodia del CD di Trajexia Tools.
10. Fare clic su **Next**.

11. Fare clic su **Yes**.

fig. 3



fig. 4

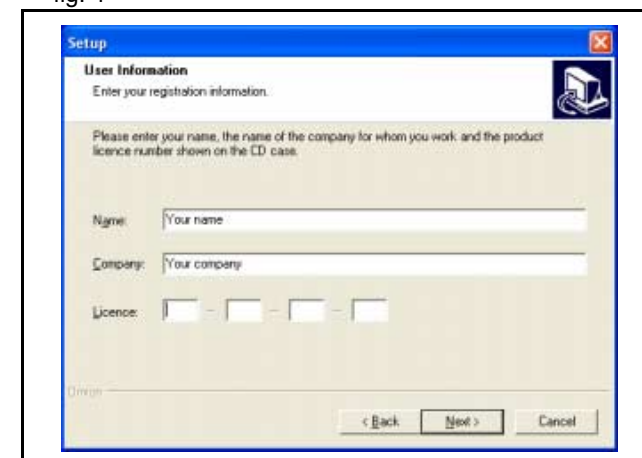
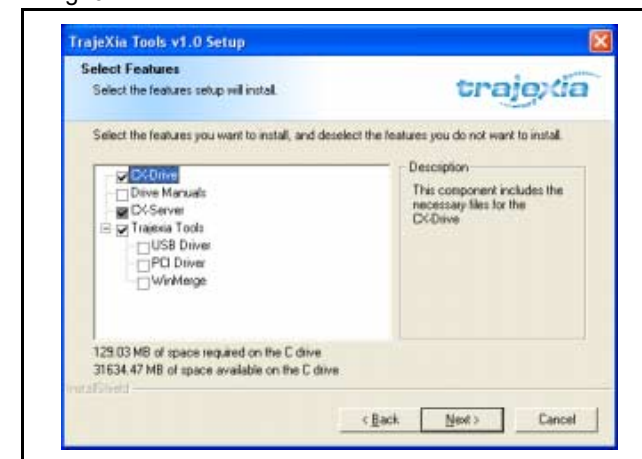


fig. 5



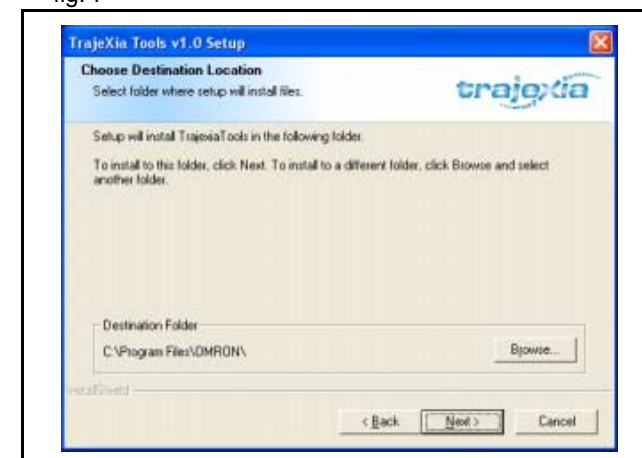
12. Fare clic su **Next**.

fig. 6



13. Fare clic su **Next**.

fig. 7



14. Fare clic su **Next**.

15. Fare clic su **Next**.

16. Il programma di installazione di Trajexia Tools copia i file sul PC. Questa operazione può durare alcuni minuti.

fig. 8

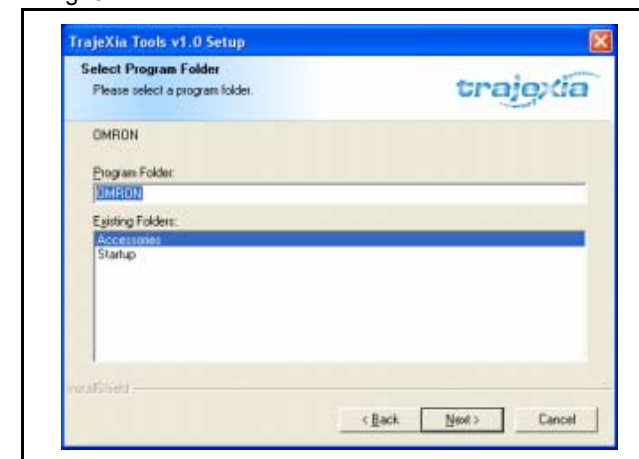
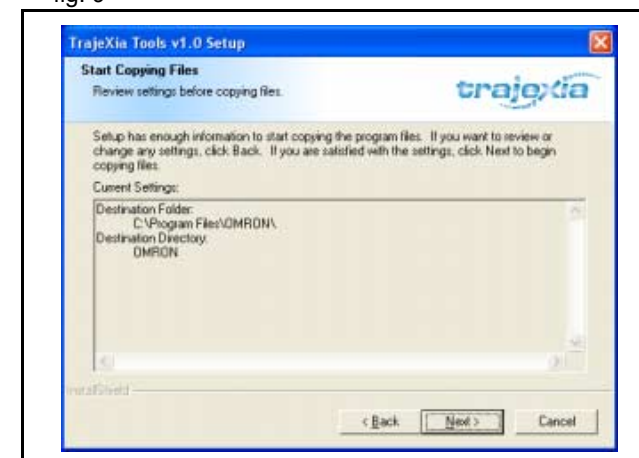


fig. 9



17. Fare clic su **Finish**. Verrà mostrata la finestra del file Readme di CX-Drive. Chiudere la finestra.

fig. 10



5.2.3 Collegamento a TJ1-MC__

È necessario un cavo patch o crossover Ethernet per collegare il PC all'unità TJ1-MC__.

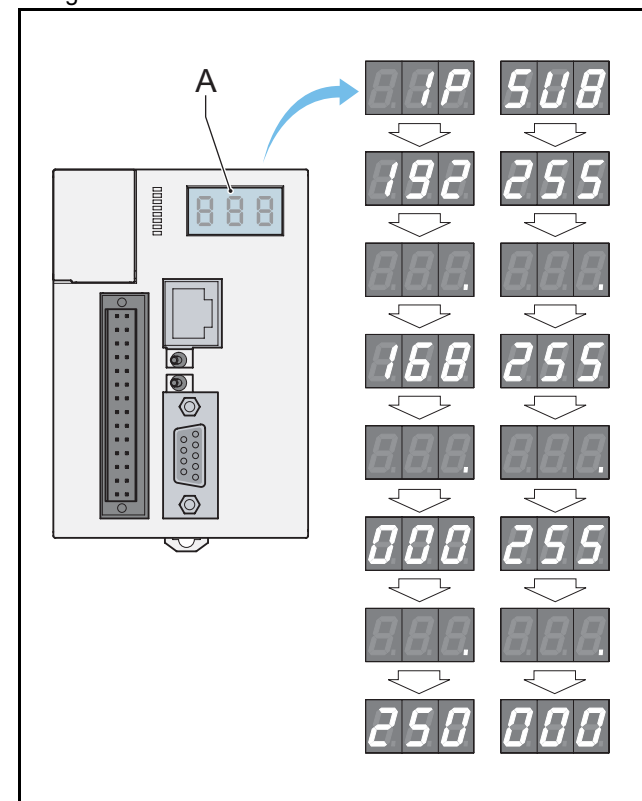


Nota

Se si lavora offline è possibile utilizzare il simulatore. La simulazione consente a Trajexia Tools di collegarsi a un'unità di controllo virtuale. La programmazione offline viene condotta in questo modo. Il "simulatore" non riconosce tutti i comandi specifici di TJ1-MC__.

1. Collegare il sistema Trajexia all'alimentazione principale.
2. Per visualizzare di nuovo l'indirizzo IP e la maschera di sottorete dell'unità TJ1-MC__, spegnere e riaccendere il sistema Trajexia.
3. Collegare il cavo Ethernet alla porta Ethernet del PC.
4. Collegare il cavo Ethernet alla porta Ethernet della TJ1-MC__. L'indirizzo IP dell'unità TJ1-MC__ viene visualizzato quattro volte sul display a LED.
5. All'avvio, il software Trajexia Tools tenta di comunicare con l'unità di controllo. Quando si avvia Trajexia Tools per la prima volta, le impostazioni di comunicazione non sono quelle adatte; pertanto è necessario cancellarle (vedere fig. 12) e immettere le proprie impostazioni.

fig. 11



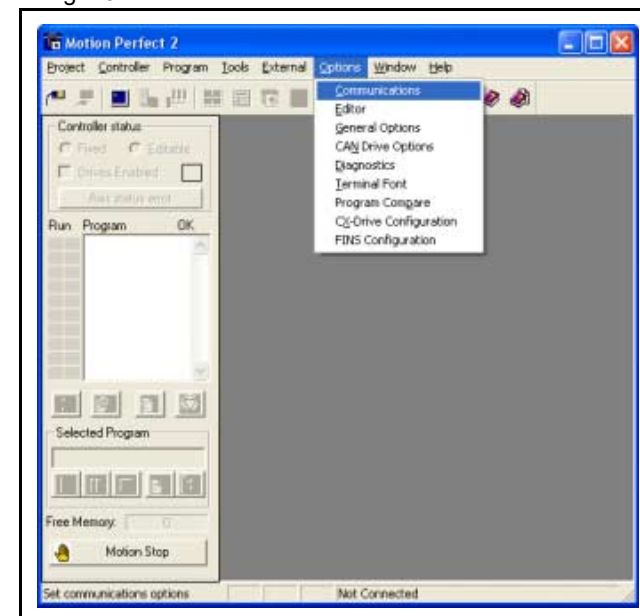
6. Avviare il programma Trajexia Tools sul PC.
Nel menu **Start** di Windows, selezionare:
 - **Programmi**
 - **OMRON**
 - **Trajexia Tools**
 - **Trajexia Tools**
7. Si apre la schermata iniziale di **Trajexia Tools**.
Attendere finché non viene visualizzato il pulsante **Cancel**, quindi fare clic su **Cancel**.

fig. 12



8. Selezionare il menu:
 - **Options**
 - **Communications**

fig. 13



9. Assicurarsi che nell'elenco sia selezionato **ENet0**.
10. Fare clic su **Configure**.

11. Digitare **192.168.0.250** nel campo **Server name/IP address**.
12. Fare clic su **OK**.

13. Fare clic su **OK**.

fig. 14



fig. 15

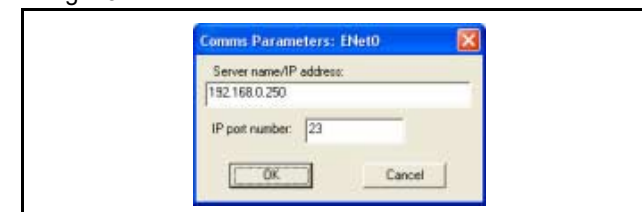


fig. 16



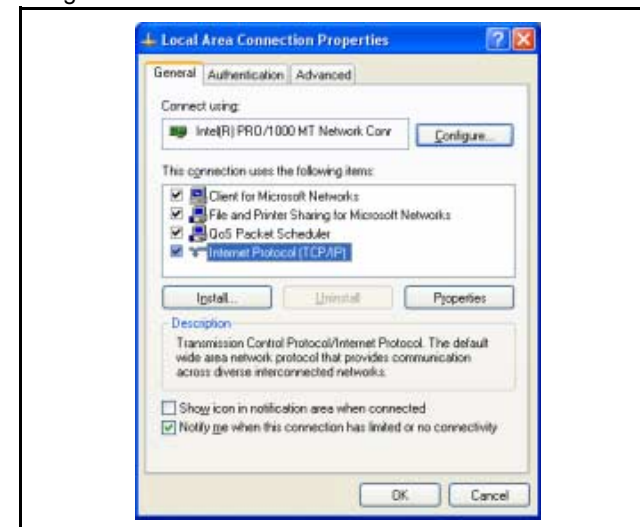
14. Aprire sul PC il Pannello di controllo di Windows.
15. Fare doppio clic sull'icona **Connessioni di rete**.
16. Fare clic con il pulsante destro del mouse sull'icona **Connessione alla rete locale (LAN)**.
Fare clic sul menu **Proprietà**.

17. Fare clic sulla scheda **Generale**.
18. Nell'elenco, selezionare **Protocollo Internet (TCP/IP)**.
19. Fare clic su **Proprietà**.

fig. 17

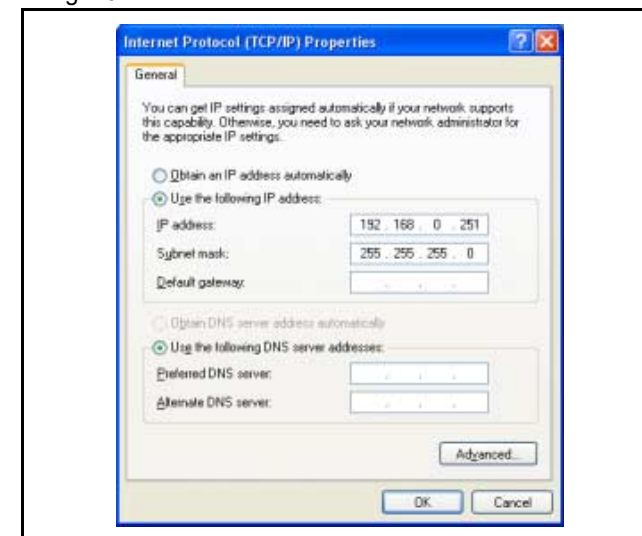


fig. 18



20. Fare clic sulla scheda **Generale**.
21. Selezionare **Utilizza il seguente indirizzo IP**.
22. Digitare l'indirizzo **192.168.0.251** nel campo **Indirizzo IP**.
23. Digitare **255.255.255.0** nel campo **Subnet mask**.
24. Fare clic su **OK**.
25. Fare clic su **OK**.
26. Chiudere la schermata Connessioni di rete.

fig. 19



5.3 Progetti

5.3.1 Progetti di Trajexia Tools

I progetti rendono più facile il processo di creazione e di sviluppo di un'applicazione. Una copia di tutti i programmi, parametri e dati è disponibile sul disco rigido del PC utilizzato per programmare il sistema.

L'utente definisce un progetto e Trajexia Tools mantiene la sincronizzazione tra il progetto presente sul PC e il sistema Trajexia.

I programmi che vengono modificati sono automaticamente duplicati sul PC.

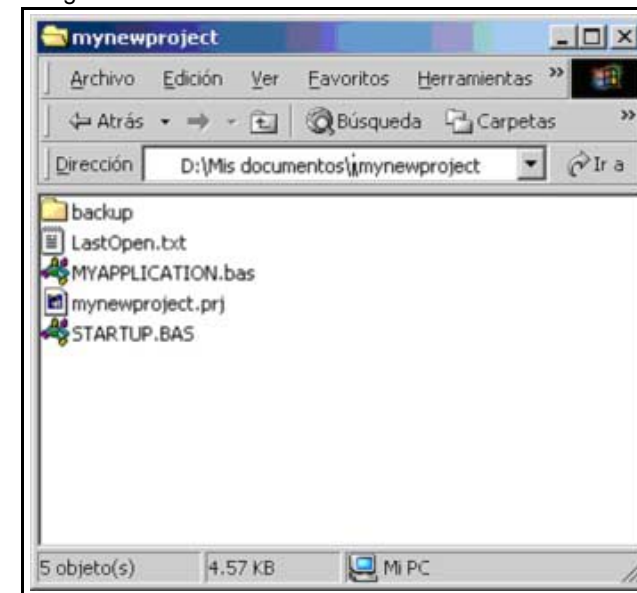
Un progetto Trajexia Tools si compone di un file **project.prj** che contiene la configurazione Trajexia, i parametri del servozionamento e un insieme di programmi **.bas**. Questi file sono conservati in una cartella che ha lo stesso nome del file **.prj**.



Nota:

un programma creato su un computer non può essere correttamente aperto in un altro computer. Per evitare questo problema, copiare l'intera directory del progetto da un computer a un altro.

fig. 20



5.3.2 Finestra Check Project

Trajexia Tools avvia la finestra Check Project quando è collegato al sistema Trajexia. Un raffronto viene eseguito tra i file di programma contenuti nel sistema Trajexia e quelli presenti sul PC.

Se i file di programma sono diversi, la finestra Check Projects mostra i seguenti pulsanti:

- **Save**
- **Load**
- **Change**
- **New**
- **Resolve**
- **Cancel**

Save

Questa opzione carica sul PC il progetto presente nel sistema Trajexia.

Se sul PC è presente un progetto con lo stesso nome, questo viene sovrascritto. Prima di sovrascrivere un programma sul PC, assicurarsi di possederne una copia di backup.

Load

Scarica sul sistema Trajexia il progetto presente nel PC.

Se nel sistema Trajexia è presente un progetto con lo stesso nome, questo viene sovrascritto. Prima di caricare un programma sul sistema Trajexia, assicurarsi di possederne una copia di backup.

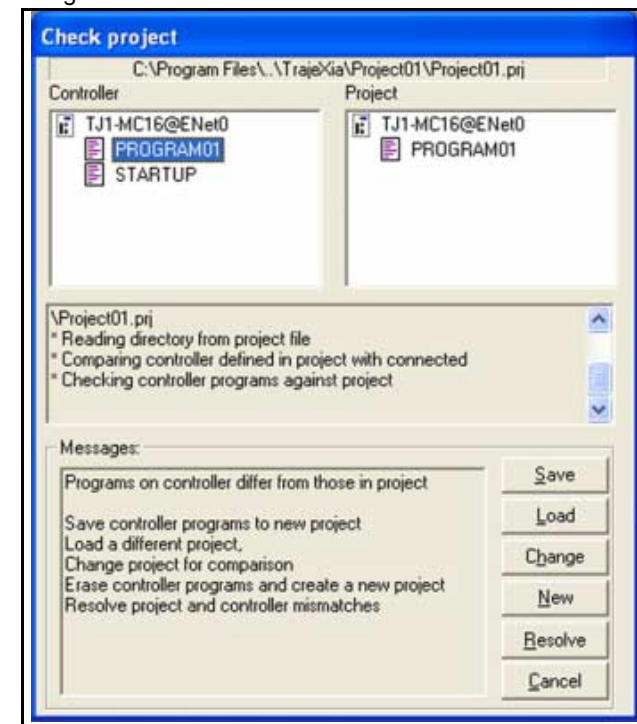
Change

Consente l'apertura di un progetto del PC, diverso dal progetto predefinito.

Se il progetto del sistema Trajexia e quello del PC non coincidono, è possibile utilizzare **Change** per selezionare un progetto diverso sul PC. Trajexia Tools verifica nuovamente entrambi i progetti.

Questa funzionalità è necessaria se si lavora su più applicazioni con progetti diversi.

fig. 21



New

Cancella il progetto presente sul sistema Trajexia e inizia un nuovo progetto sul PC. Trajexia Tools crea una nuova directory che ha lo stesso nome del progetto e contiene il nuovo file di progetto. Il nome della directory deve coincidere con il nome del progetto, in caso contrario il progetto non può aprirsi.

Resolve

Mette a confronto il progetto presente su Trajexia Tools con il progetto presente sul PC. Questa opzione offre la possibilità di salvare (**Save**), caricare (**Load**) o esaminare (**Examine**) a livello individuale le differenze in ciascun programma contenuto dal progetto. Questa opzione consente di modificare un programma offline utilizzando il simulatore, dopo aver scaricato lo stesso programma su TJ1-MC__ e permette anche a più persone contemporaneamente di lavorare sullo stesso progetto.

Cancel

Arresta il processo di connessione. Trajexia Tools si avvia in modalità disconnessa.

5.4 Finestra delle applicazioni di Trajexia Tools

La finestra delle applicazioni di Trajexia Tools è composta dei seguenti elementi:

1. Pannello di controllo
2. Barra dei menu
3. Barra degli strumenti
4. Area di lavoro
5. Barra di stato

5.4.1 Pannello di controllo

Il pannello di controllo offre un accesso rapido e facile alla maggior parte dei comandi usati più di frequente per gestire e mettere a punto un progetto.

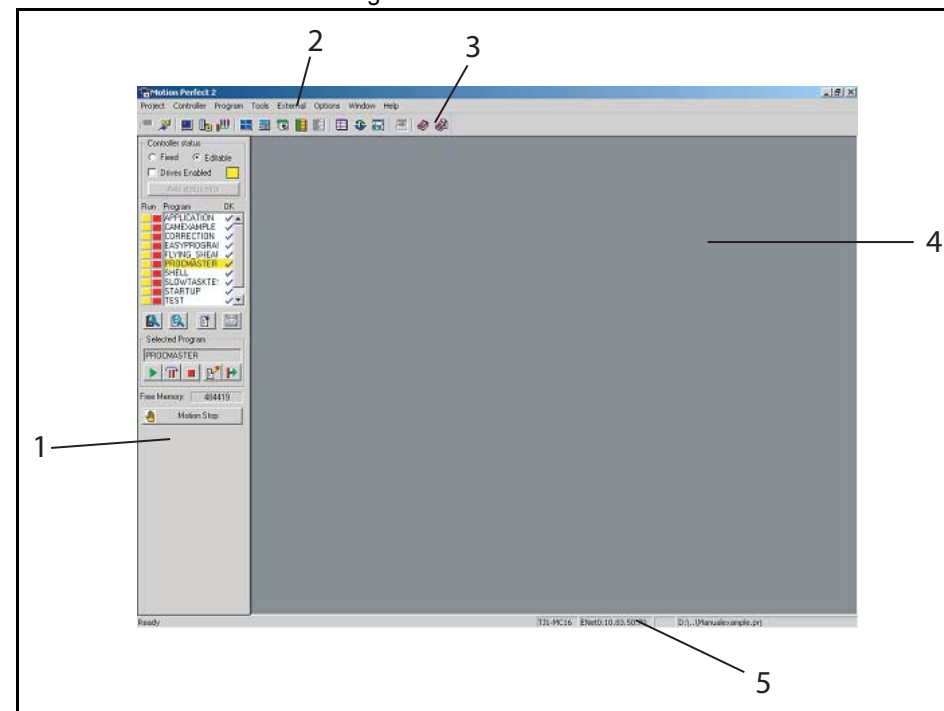
5.4.2 Barra dei menu

La barra dei menu contiene i seguenti elementi:

- **Project**
- **Controller**
- **Program**
- **Tools**
- **External**
- **Options**
- **Window**
- **Help**

Tutti questi menu sono descritti in dettaglio alla sezione "Descrizioni dei menu" (p. 196).

fig. 22



5.4.3 Barra degli strumenti

	Connect	Collega Trajexia Tools al sistema Trajexia. Fare riferimento a Connect nel "Menu Controller" (p. 198).		Digital IO	Fare riferimento a Digital IO Status nel "Menu Tools" (p. 205).
	Disconnect	Scollega Trajexia Tools dal sistema Trajexia. Fare riferimento a Disconnect nel menu "Menu Controller" (p. 198).		Analog input	Fare riferimento ad Analog input nel "Menu Tools" (p. 205).
	Terminal	Editor della riga di comando. Fare riferimento a Terminal nel "Menu Tools" (p. 205).		TABLE values	Fare riferimento a TABLE and VR values nel "Menu Tools" (p. 205).
	Axis Parameters	Fare riferimento a Axis parameters nel "Menu Tools" (p. 205).		VR values.	Fare riferimento a TABLE and VR values nel "Menu Tools" (p. 205).
	Intelligent Drives	Fare riferimento a Intelligent Drives nel "Menu Tools" (p. 205).		Watch variables (Supervisione delle variabili)	Non implementato in Trajexia.
	Oscilloscope	L'oscilloscopio software può essere usato per tracciare i parametri degli assi e dei movimenti. Questa funzionalità è di ausilio nello sviluppo dei programmi e nella messa a punto del sistema. Fare riferimento a Oscilloscope nel "Menu Tools" (p. 205).		Simulator (Simulatore)	Non del tutto implementato in Trajexia.
	Keypad (Tastierino)	Non implementato in Trajexia.		Guida di Trajexia Tools	Apre la Guida di Trajexia Tools.
	Jog Axis	Questa finestra consente all'utente di muovere manualmente gli assi in Trajexia. Fare riferimento a Jog Axis nel "Menu Tools" (p. 205).		Guida di Trio BASIC	Apre la Guida di Trio BASIC.

5.5 Descrizioni dei menu

5.5.1 Menu Project

Il menu **Project** consente di creare, caricare e salvare i progetti Trajexia Tools.

New Project (Nuovo progetto)

Cancella il progetto presente sul sistema Trajexia e inizia un nuovo progetto sul PC. Trajexia Tools crea una nuova directory che ha lo stesso nome del progetto e contiene il nuovo file di progetto. Il nome della directory deve coincidere con il nome del progetto, in caso contrario il progetto non può aprirsi.

Load project (Carica progetto)

Apri un progetto già esistente sul PC. Trajexia Tools scarica sul sistema Trajexia il progetto presente nel PC. Se nel sistema Trajexia è presente un progetto con lo stesso nome, questo viene sovrascritto. Prima di caricare un programma sul sistema Trajexia, assicurarsi di possederne una copia di backup.

Save project as... (Salva progetto con nome)

Carica sul PC il progetto che si trova nel sistema Trajexia e lo salva con nome in una directory del disco fisso del PC.

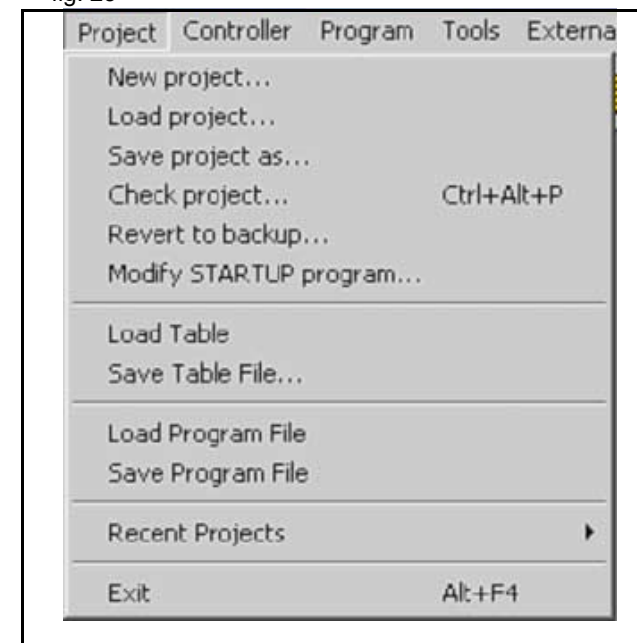
Check project (Controlla progetto)

Un controllo viene eseguito tra il progetto presente nel sistema Trajexia e il PC. Vengono messi a confronto i checksum e il contenuto dei programmi.

Revert to backup... (Ritorna alla versione del backup...)

Ogni volta che si collega a MC16, Trajexia Tools confronta il progetto presente nell'unità di controllo con quello presente nel PC e poi crea una copia di backup. **Revert to backup** può essere usato quando si desidera cancellare tutte le modifiche apportate al progetto e ai programmi BASIC mentre si era collegati all'unità di controllo.

fig. 23



Così facendo, i programmi BASIC vengono riportati alle versioni presenti nella directory di backup.

Modify STARTUP program (Modifica programma STARTUP)

Il programma **Startup** controlla il numero di nodi presenti in un sistema MECHATROLINK-II trasferendoli al progetto. Utilizzare il programma **Modify STARTUP** per modificare un programma di avvio che è stato creato dalla finestra **Intelligent Drives**.

Load table (Carica tabella)

È possibile caricare un elenco di valori presenti in una tabella da un file esterno con estensione ***.lst** o ***.bas**. Il comando importa i valori e li registra nei valori della memoria TABLE.

Save table file... (Salva file di tabella...)

Salva un file ***.lst** o ***.bas** a partire dai valori della memoria TABLE nella directory del progetto.

Load program file (Carica file di programma)

Carica un file che contiene un codice eseguibile in un task.

Save program file (Salva file di programma)

Salva il file di programma in formato **.txt**.

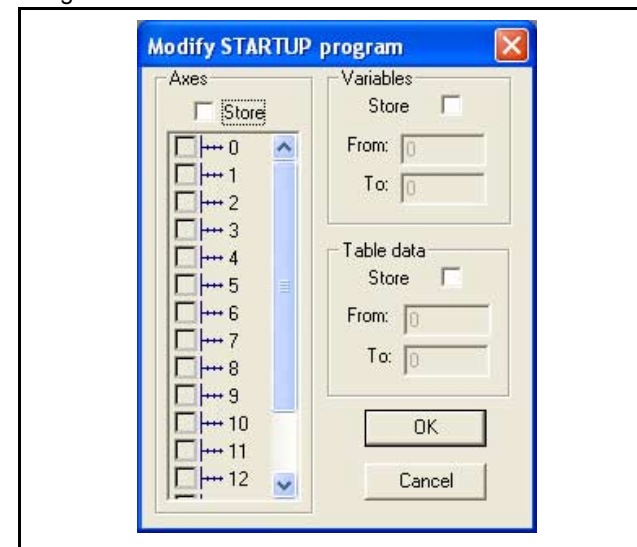
Recent projects (Progetti recenti)

È possibile aprire i progetti recenti che sono stati modificati dal software Trajexia Tools.

Exit (Esci)

Chiude l'applicazione Trajexia Tools.

fig. 24



5.5.2 Menu Controller

Il menu Controller consente di impostare la comunicazione tra il PC e il sistema Trajexia e di controllare quest'ultimo.

Connect

Crea un collegamento con il sistema Trajexia e avvia il gestore del progetto. Disponibile se Trajexia Tools non è collegato al sistema.

Disconnect

Interrompe il collegamento con il sistema Trajexia.
Disponibile quando Trajexia Tools è collegato al sistema.

Connect to simulator (Collega al simulatore)

Non del tutto implementato per Trajexia.

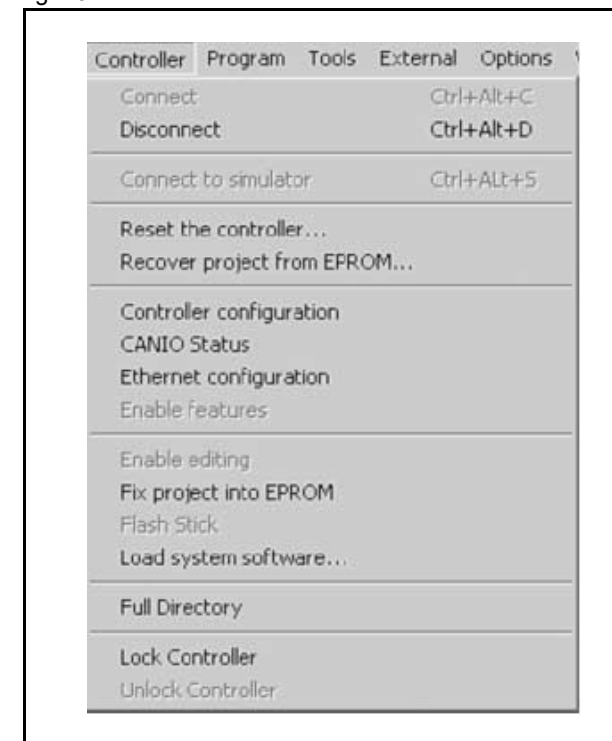
Reset the controller

Esegue un reset software sul sistema Trajexia. L'applicazione Trajexia Tools si scollega dal sistema Trajexia.

Recover project from EPROM

Reimposta il sistema Trajexia e ripristina sul PC i programmi che si trovano sulla memoria EPROM.

fig. 25



Controller configuration

Visualizza lo schermo di configurazione dell'hardware dell'unità di controllo collegata al PC.

Controller: il PC è collegato a Trajexia Motion Controller (TJ1-MC__) dotato di software 1.64 Dev. 94. Il ciclo di servoazionamento è 1000µs.

Axis: mostra quali assi sono disponibili.

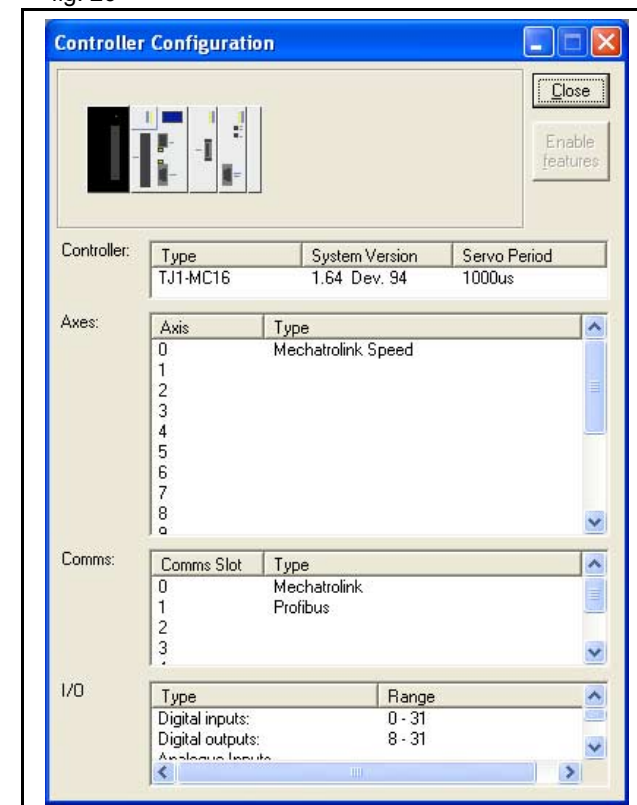
Comms: la capacità di comunicazione di Trajexia Motion Controller.

I/O: il tipo e l'intervallo degli ingressi e delle uscite digitali, analogici e virtuali.

CANIO

Non implementato per Trajexia.

fig. 26



Ethernet configuration

Consente la modifica della configurazione Ethernet e dell'indirizzo IP dell'hardware dell'unità di controllo.

Slot: sempre -1 per Trajexia.

IP address: l'indirizzo IP di Trajexia Motion Controller. Non corrisponde all'indirizzo IP del PC.

Subnet Mask: la maschera di sottorete per Trajexia Motion Controller e il PC deve essere la stessa.

Default gateway: un nodo della rete che funge da ingresso per un'altra rete. È richiesto solo se Trajexia deve comunicare con il dispositivo di un'altra sottorete.

MAC address: indirizzo Media Access Control (Controllo per l'accesso ai supporti), un indirizzo hardware che identifica univocamente ciascun nodo della rete. Questo indirizzo è di sola lettura.

Normal Communications Port Number: la porta TCP utilizzata per comunicare con Trajexia Tools.

Token Communications Port Number: la porta TCP utilizzata per comunicare con il controllo ActiveX PC Motion.

Enable features

Non implementato per Trajexia.

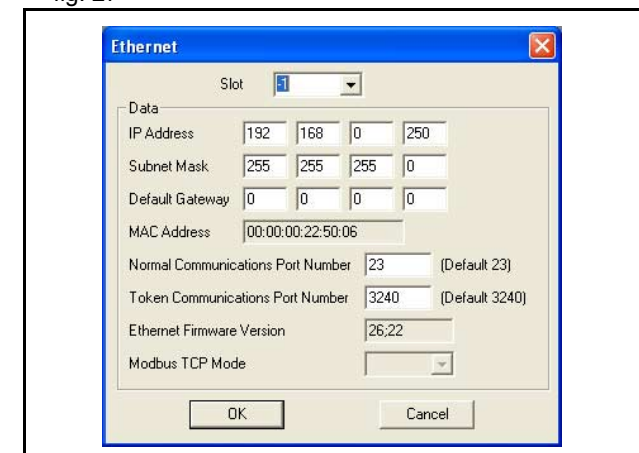
Enable editing

Consente a Trajexia di operare con la versione RAM dei programmi. In questa modalità è possibile modificare i programmi.

Trajexia memorizza i programmi nella RAM (e nelle variabili globali) utilizzando la batteria di backup. Questa opzione cambia internamente il parametro **POWER_UP**.

Se **POWER_UP** è impostato su 0, all'avvio Trajexia utilizza i programmi memorizzati nella RAM dalla batteria di backup, anche se i programmi sono stati salvati sulla memoria flash. Questa funzionalità è disponibile solo quando **POWER_UP=1**.

fig. 27



Fix project into EPROM

Copia i programmi dall'unità di controllo alla memoria flash EPROM. Tutti i programmi contenuti in quel momento nella memoria EPROM vengono sovrascritti. Questa funzionalità cambia **POWER_UP** impostandolo su 1 e dopo l'avvio la RAM viene sovrascritta con i contenuti della memoria EPROM. In questa modalità non è possibile modificare i programmi. Questa funzionalità è disponibile solo quando **POWER_UP=0**.

Load System Software

Trajexia dispone di una memoria flash EPROM per registrare sia i programmi utente che il software di sistema. Utilizzare Load System Software per aggiornare il software di sistema all'ultima versione.

Si apre una finestra di dialogo che richiede l'esecuzione di un backup e la conferma che si desidera continuare con l'operazione.

Si apre un'interfaccia standard per la selezione dei file. Selezionare il file richiesto.



OMRON consiglia di caricare una nuova versione del software di sistema solo se consigliato dal proprio distributore OMRON.



Attenzione

Non caricare software che non è stato specificato per Trajexia Motion Controller. Caricare esclusivamente versioni create appositamente per essere utilizzate con Trajexia.

Nessun'altra versione funziona.

Si apre una finestra di dialogo con la richiesta di confermare che si desidera continuare.

Fare clic su OK per iniziare. Il caricamento nella memoria flash EPROM dura circa 7 minuti.



Attenzione

Non interrompere il processo di aggiornamento del software. Un'interruzione del processo di comunicazione danneggia l'unità Trajexia. Se dopo il caricamento nella memoria flash EPROM il recupero dell'unità Trajexia risulta impossibile, contattare il proprio rappresentante.

Una volta terminato il download, un checksum conferma che il caricamento nella memoria flash EPROM è riuscito. Per completare l'operazione, selezionare **Yes** nella finestra di dialogo di conferma.

Aprire la finestra **Controller Configuration** e confermare la presenza della nuova versione del sistema.

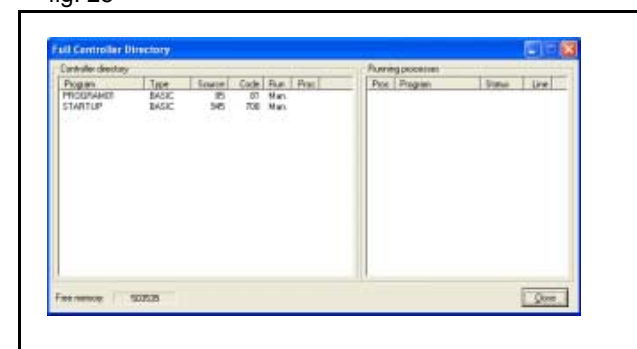
Full directory

Da utilizzare per controllare il contenuto e la struttura dei file della directory di TJ1-MC__.

Lock/Unlock

Blocca il sistema Trajexia per impedire l'accesso non autorizzato. Quando il sistema Trajexia è bloccato, non è possibile elencare, modificare o salvare alcun programma Trajexia. Il sistema Trajexia non è disponibile attraverso il software Trajexia Tools; sono disponibili solo il terminale e la finestra di dialogo di sblocco. Per bloccare il sistema, inserire un numero a 7 cifre. Per sbloccare il sistema, utilizzare lo stesso numero.

fig. 28



5.5.3 Menu Program

Il menu Program contiene le voci che consentono di programmare un progetto Trajexia.

New

Crea un nuovo programma.

Edit

Apri un progetto per consentirne la modifica.

È possibile avviare l'editor anche dal pannello di controllo.

Usando il menu Program viene dapprima richiesto, tramite una finestra di selezione, di confermare il programma che si desidera modificare.

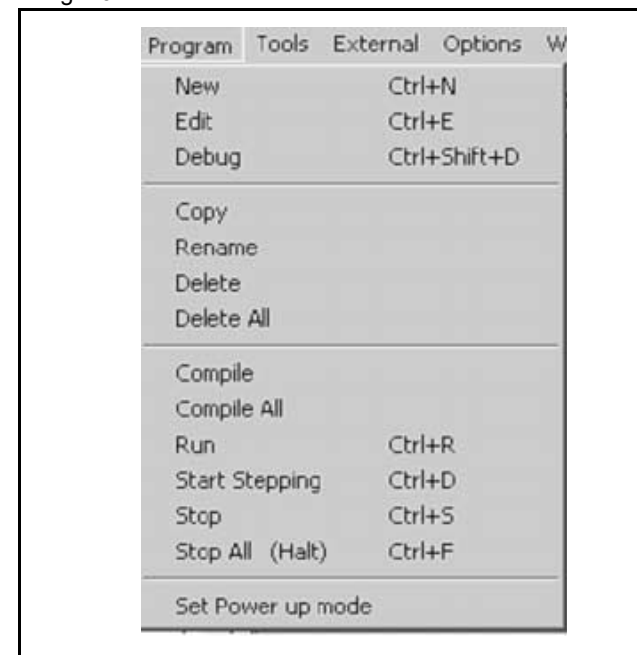
L'editor di Trajexia Tools è stato concepito per funzionare in maniera analoga a quella dell'editor di testo di un qualsiasi PC. Le operazioni standard, come ad esempio la funzione di modifica dei blocchi di testo, la funzione di ricerca e sostituzione di un testo e quella di stampa, sono tutte supportate e conformi alle combinazioni di scelta rapida presenti in Windows. Oltre a ciò, dispone di una funzione di evidenziazione della sintassi BASIC, di una funzione di formattazione dei programmi e di una funzione di debug dei programmi.

Debug

Controlla la sintassi dei programmi e propone delle possibili soluzioni. Il programma viene aperto in una speciale modalità di registrazione che effettua l'esecuzione riga per riga. È possibile impostare dei punti di interruzione all'interno del programma che ne sospendono l'esecuzione quando vengono raggiunti. Nella finestra di debug, la riga di codice corrente è evidenziata.

Quando un programma viene eseguito in modalità debug, qualunque editor è impostato sulla modalità debug e diventa di sola lettura.

fig. 29



Copy

Copia i contenuti di un programma in un altro programma.

Rename

Cambia il nome di un programma.

Delete

Elimina il programma dalla struttura dei file.

Delete all

Elimina tutti i programmi dalla struttura dei file.

Compile

Compila il programma correntemente nel progetto.

Compile all

Compila tutti i programmi correntemente nel progetto.

Run

Esegue il programma corrente all'interno del processo specificato.

Start stepping

Esegue il programma corrente all'interno del processo specificato in modalità passo-passo (riga per riga).

Stop

Il comando Stop arresta il programma dell'unità di controllo TJ1-MC__. Non equivale al comando Motion Stop. Il programma si arresta al termine del ciclo della CPU. I servomotori mantengono la propria posizione.

Stop all

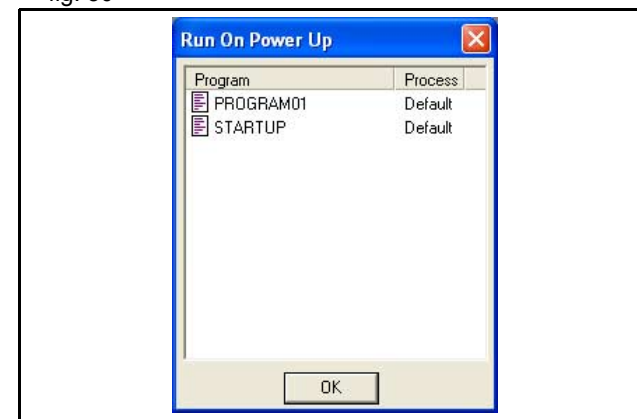
Il comando Stop all arresta tutti i programmi dell'unità di controllo TJ1-MC__. Non equivale al comando Motion Stop. I programmi si arrestano al termine del ciclo della CPU. I servomotori mantengono la propria posizione.

Set power up mode

È possibile ottenere l'esecuzione automatica dei programmi di TJ1-MC__ all'avvio del sistema. Selezionare **Set Powerup Mode** per aprire la finestra di dialogo **Run On Power Up** dialog window.

Selezionare il programma che si desidera fare eseguire automaticamente. Sulla parte destra della finestra appare un piccolo menu a comparsa. Se si desidera che Trajexia esegua il processo all'avvio, scegliere default come numero di processo. È anche possibile selezionare specificamente il processo.

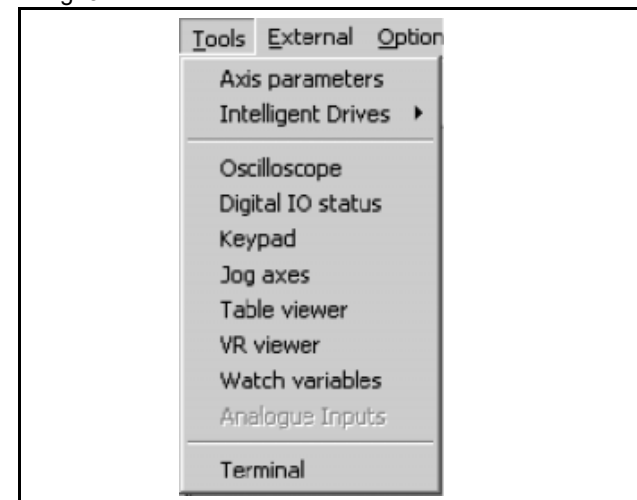
fig. 30



5.5.4 Menu Tools

Per accedere agli strumenti di Trajexia Tools, utilizzare il menu Tools o il pulsante equivalente sulla barra degli strumenti.

fig. 31



Axis parameters

La finestra Axis Parameters consente di monitorare e cambiare i parametri di movimento per qualsiasi asse del sistema Trajexia.

La finestra raggruppa i parametri in due insiemi:

- Insieme 1 (la metà superiore della finestra): contiene i parametri che possono essere cambiati dall'utente.
- Insieme 2 (la metà inferiore della finestra): contiene i parametri impostati dal software di sistema del sistema Trajexia quando questo elabora i comandi e controlla lo stato degli ingressi esterni.

L'elemento di separazione che divide i due insiemi di dati può essere spostato con il mouse.

Quando l'utente modifica un parametro delle unità, tutti i parametri che utilizzano quel valore di parametro vengono nuovamente letti e regolati in base a quel fattore.

Alcuni dei parametri influenzati da questo parametro sono, ad esempio:

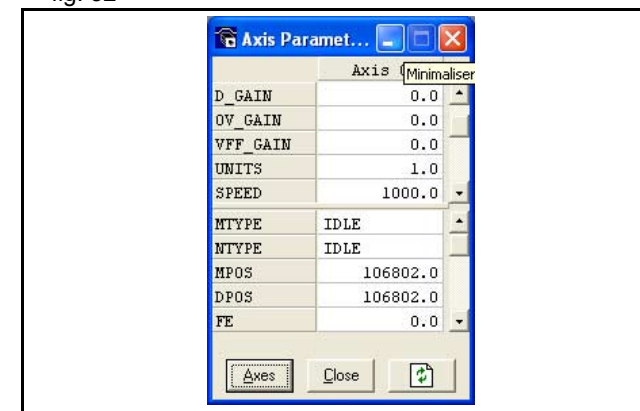
- **SPEED**
- **ACCEL**
- **MPOS**

Il parametro **axistatus** mostra lo stato dell'asse. Il colore dei caratteri con cui è scritto il valore del parametro ne indica lo stato:

- Verde: nessun errore
- Rosso: errore

car	Descrizione
w	Avviso intervallo FE
a	Errore comunicazione driver.
m	Errore driver remoto
f	Limite di avanzamento
r	Limite di marcia indietro
d	Ingresso dati

fig. 32



car	Descrizione
h	Ingresso sospensione alimentazione
e	Errore seguente
x	Limite software di avanzamento
y	Limite software di marcia indietro
c	Annullamento movimento in corso
o	Errore encoder

Le opzioni della finestra Axis Parameter sono:

Axes: seleziona gli assi di cui sono visualizzati i dati.

Refresh: per ridurre il carico sul sistema Trajexia, i parametri dell'insieme 1 sono letti solo quando lo schermo viene visualizzato per la prima volta o quando viene impostato il valore del parametro. Se il valore di un parametro viene modificato, il valore visualizzato potrebbe non essere quello corretto. Il pulsante forza Trajexia Tools a leggere nuovamente l'intera selezione.



Nota:

Se si modifica il valore di un parametro, è necessario aggiornare il display prima di effettuare una nuova modifica.

Intelligent Drives

Intelligent Drives consente di accedere ai programmi di configurazione e avvio per ognuno dei servoazionamenti configurabili.

Lo strumento Intelligent Drives mostra la configurazione di Trajexia rilevata all'accensione. Selezionando l'unità, si apre la scheda successiva.

Nella parte superiore della finestra è visualizzato

TJ1-MC__ con i suoi diversi moduli. In questo esempio:

- TJ1-MC__ con numero di modulo –1
- TJ1-ML__ con numero di modulo 0
- TJ1-PRT__ con numero di modulo 1
- TJ1-FL02__ con numero di modulo 2

Se nel sistema esistono più TJ1-ML__, vengono visualizzate più schede.

Il pulsante **Modify STARTUP program** crea un programma STARTUP per la configurazione individuata.

Nella scheda che corrisponde a TJ1-ML__ è possibile vedere le informazioni corrispondenti agli slave MECHATROLINK-II rilevati (inclusi l'inverter e i moduli di I/O).

Facendo clic sul pulsante **Config** (disponibile solo per il servoazionamento e l'inverter), si apre la finestra successiva:

Scheda **Status**:

- **Drive ID/Motor ID/Firmware Version:** mostrano informazioni relative al servoazionamento e al servomotore.
- **Drive Status** mostra il contenuto del canale **DRIVE_STATUS** per quell'asse.
- **Drive I/O** mostra il contenuto del canale **DRIVE_INPUTS** per quell'asse.
- **Drive Clear** esegue il comando **DRIVE_CLEAR** (cancellazione dell'allarme del servoazionamento) per quell'asse.
- **Drive Reset** esegue il comando **DRIVE_RESET** (accensione software) per quell'asse.
- **Drive Monitor** seleziona il monitor da aggiornare in **DRIVE_MONITOR**.

fig. 33

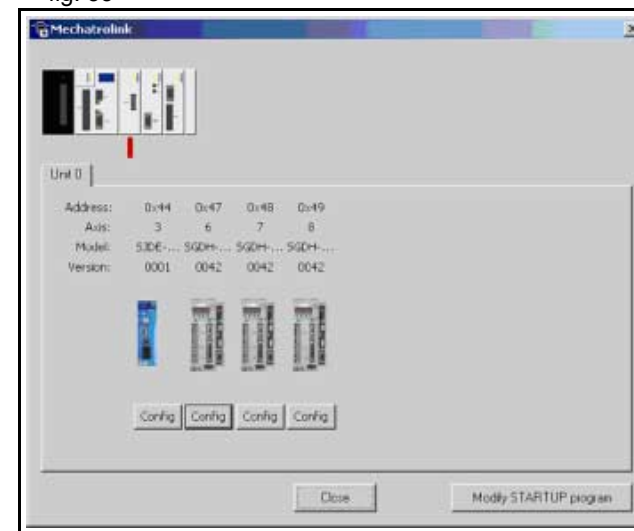
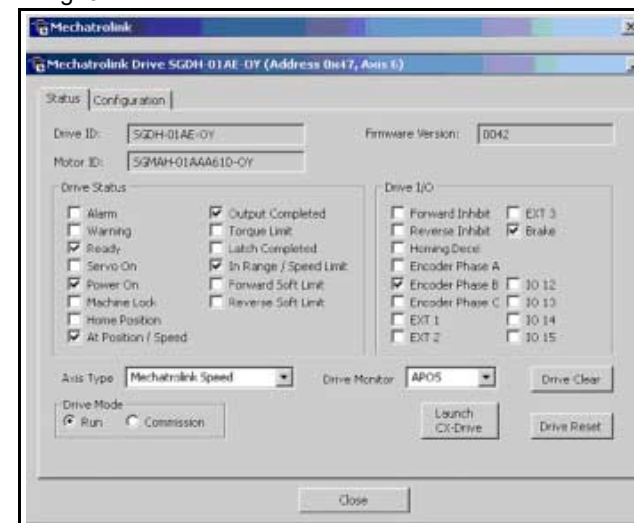


fig. 34

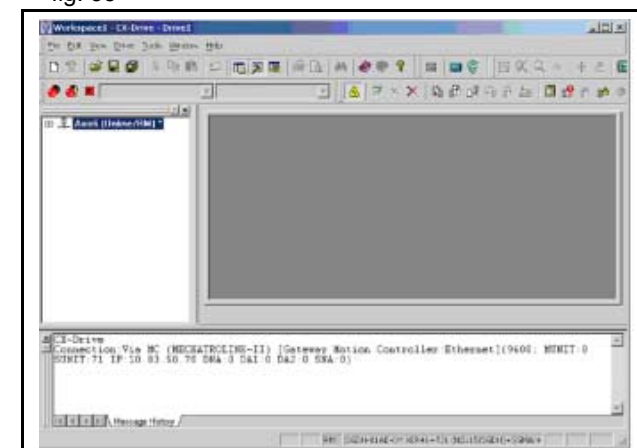


- **Axis Type** seleziona l'**ATYPE** per l'asse. Questo valore viene incluso nel programma STARTUP.
- **Drive Mode**: Run o Commission.
 - Quando l'asse è impostato su Run, il suo stato RUN e i suoi movimenti sono interamente controllati dai programmi in esecuzione su TJ1-MC__.
 - Quando l'asse è impostato su Commission, il funzionamento e i movimenti sono controllati esternamente tramite FINS; questa modalità è implementata per essere usata con la funzionalità Jog e impostata tramite CX-Drive. Lo scopo è di evitare che i programmi entrino in conflitto. Durante la messa a punto, l'asse viene considerato un asse virtuale dai programmi.
È possibile leggere e scrivere contemporaneamente i parametri del servozionamento attraverso Programs o FINS, a prescindere dalla modalità.
- **Launch CX-Drive**: attraverso Trajexia Tools è possibile solamente leggere e scrivere i parametri di un servozionamento. Se è richiesta una più elevata funzionalità di servozionamento, ad esempio Read alarm code (Lettura codice allarme), Jog, Set rigidity (Impostazione rigidità), Autotuning, occorre avviare CX-Drive. Facendo clic su questo pulsante, si avvia CX-Drive collegandolo all'asse corrente tramite TJ1-ML__.

L'unica funzionalità di servozionamento non supportata da CX-Drive attraverso MECHATROLINK-II è la funzionalità Trace (Traccia), ma in sostituzione è possibile utilizzare l'oscilloscopio di Trajexia Tools.

Se si modifica un parametro del servozionamento attraverso CX-Drive, Trajexia Tools non lo rileva immediatamente. Evitare con attenzione che lo stesso parametro abbia valori diversi sul servozionamento e nel progetto.

fig. 35



Scheda **Configuration**.

La scheda **Configuration** mostra una finestra per la modifica dei parametri identica a quella presente in CX-Drive. Per maggiori dettagli, controllare le informazioni in CX-Drive.

Le nuove funzionalità sono:

Il pulsante **Save**: salva i parametri di servozionamento correnti nel progetto Trajexia (nel file *.prj).

Cancel Registration Mode (Annulla modalità di registrazione): quando nel servozionamento la registrazione è attiva, se si desidera ottenere una risposta rapida e affidabile non è possibile scrivere i parametri. Questo equivale ad eseguire **REGIST(-1)**.

Launch CX-Drive: lo stesso del pulsante descritto per la scheda Status.

Oscilloscope

L'oscilloscopio del software può essere usato per tracciare i parametri degli assi e dei movimenti. Questa funzionalità aiuta lo sviluppo dei programmi e la messa a punto del sistema. Esistono quattro canali, ognuno dei quali capace di registrare un (1) campionamento per **SERVO_PERIOD**, tramite ciclo manuale o attivazione attraverso programma.

L'unità di controllo registra i dati alla frequenza selezionata e carica le informazioni nell'oscilloscopio per la visualizzazione. Se viene utilizzata una base temporale più ampia, i dati vengono recuperati in sezioni e anche il grafico viene tracciato in sezioni sul display. La modalità in cui si trova l'unità di controllo, cioè Program oppure Manual, determina il momento in cui l'unità inizia la registrazione.

- **Modalità Program**: l'oscilloscopio inizia a registrare i dati quando riceve un'istruzione in questo senso dal programma presente sull'unità di controllo.
- **Modalità Manual**: l'oscilloscopio inizia subito a registrare i dati.

Canali dell'oscilloscopio

Ciascun canale dell'oscilloscopio dispone di controlli per tutti e quattro i blocchi di controllo dei canali. Ciascun blocco di controllo dei canali è contraddistinto da un bordino colorato che corrisponde al colore del display di quel canale. I diversi controlli sono descritti di seguito:

fig. 36

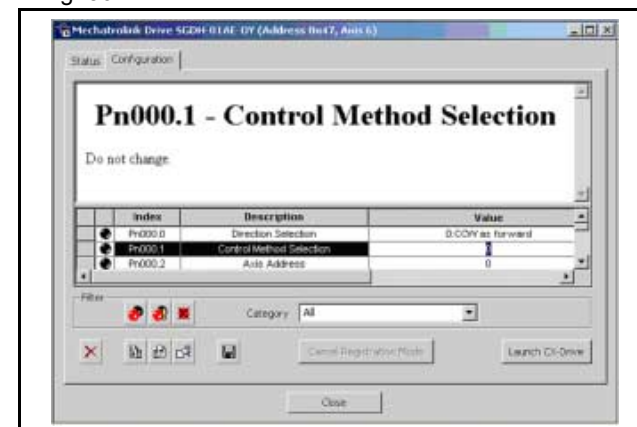
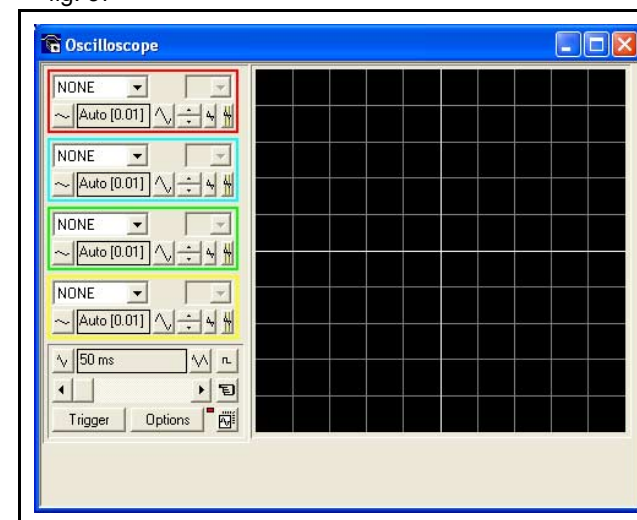


fig. 37



Parameter to display: i parametri che l'oscilloscopio è in grado di registrare e visualizzare vengono selezionati dal menu a discesa nell'angolo in alto a sinistra di ciascun blocco di controllo dei canali. Il tipo di parametro selezionato imposta l'etichetta successiva tra asse (Ax) e canale (Ch).

Per tracciare i punti registrati nella memoria TABLE dell'unità di controllo, selezionare il parametro TABLE e il canale che ha i primi e gli ultimi punti configurati tramite la finestra di dialogo delle opzioni avanzate.

Se il canale non è necessario, selezionare NONE nella casella di riepilogo dei parametri.

Axis/Channel number: una casella di riepilogo con menu a discesa che consente la selezione dell'asse o del canale per un parametro o canale di movimento relativo a un parametro di ingresso/uscita digitale o analogico.

Y range DOWN/Y range UP: la scala verticale viene selezionata per ciascun canale ed è possibile configurarla in modalità automatica o manuale. In modalità automatica, l'oscilloscopio calcola la scala appropriata una volta completata la registrazione e prima di visualizzare la traccia. Se durante il funzionamento l'oscilloscopio continua a riavviarsi, non sarà in grado di selezionare una scala verticale idonea. Occorre perciò intervenire arrestandolo e poi riavviandolo di nuovo.

Nella modalità manuale, l'utente seleziona la scala più appropriata.

Y Shift: il valore di offset verticale utilizzato per spostare verticalmente una traccia sul display. Questo controllo è utile quando due o più tracce sono identiche.

Reset Y: questo pulsante azzerà il valore di Y shift.

Enable/Disable cursor bars: quando l'oscilloscopio smette di funzionare e una traccia viene visualizzata, è possibile abilitare le barre di scorrimento. Le barre di scorrimento sono due barre verticali, dello stesso colore della traccia del canale. Indicano la posizione dei valori minimi e massimi della traccia. I valori rappresentati dalle barre sono mostrati sotto al display dell'oscilloscopio.

Le barre di scorrimento vengono abilitate e disabilitate premendo il pulsante del cursore. Le barre di scorrimento possono essere selezionate e spostate usando il cursore del mouse.

General controls

I controlli generali si trovano in basso a sinistra, sullo schermo dell'oscilloscopio. I diversi controlli sono descritti di seguito:

Time base: il valore della base temporale è il valore temporale di ciascuna divisione orizzontale dell'oscilloscopio. La base temporale viene selezionata attraverso i pulsanti su/giù della scala, ai due lati della casella dei valori della base temporale corrente. Se la base temporale è maggiore di un valore predefinito, i dati vengono presi dall'unità di controllo in sezioni e non come traccia continua di dati.

Le sezioni di dati vengono tracciate sul display a mano a mano che vengono ricevute. L'ultimo punto è di colore bianco.

X shift: se la traccia viene completata mentre il valore della base temporale viene modificato per produrre un rilevamento più veloce, solo una parte della traccia sarà visualizzata. Per visualizzare la traccia completa, utilizzare la barra di scorrimento X shift.

Se l'oscilloscopio è configurato sia per registrare i parametri di movimento che per tracciare i dati di tabella, il numero dei punti tracciati sul display può essere determinato dal parametro di movimento. Gli altri punti della tabella che non risultano visibili, possono essere resi visibili utilizzando la barra di scorrimento.

La traccia del parametro di movimento non può essere spostata.

Single/continuous trigger: in modalità singola, l'oscilloscopio rimane in funzione solo finché non viene premuto il pulsante Trigger e un insieme di dati registrati dall'unità di controllo viene recuperato e visualizzato.

In modalità continua, l'oscilloscopio rimane sempre in funzione e recupera i dati dall'unità di controllo ogni volta che viene attivato e i dati vengono registrati. L'oscilloscopio rimane in funzione fino a quando non si fa clic una seconda volta sul pulsante Trigger.

Trigger/Halt data capture: facendo clic sul pulsante Trigger, l'oscilloscopio viene abilitato. Se si trova in modalità manuale, l'unità di controllo inizia immediatamente a registrare i dati.

Se si trova in modalità Program, aspetta fino a quando non incontra un comando di attivazione in un programma in esecuzione.

Quando il pulsante di Trigger è stato premuto, il testo del pulsante cambia in "Halt". Se l'oscilloscopio è in modalità singola, una volta che i dati sono stati registrati e tracciati sul display, il pulsante Trigger ritorna a chiamarsi "Trigger", indicando che l'operazione è stata completata. L'oscilloscopio può essere arrestato in qualsiasi momento quando è in funzione e il pulsante "Trigger" mostra il testo "Halt".

Clear configuration: la configurazione corrente dell'oscilloscopio (lo stato di tutti i controlli) è salvata quando la finestra dell'oscilloscopio viene chiusa e recuperata quando viene riaperta.

Il pulsante di ripristino della configurazione (posto in fondo a destra del pannello di controllo dell'oscilloscopio) reimposta tutti i controlli ai rispettivi valori predefiniti.

L'indicatore di stato: l'indicatore di stato di trova fra i le opzioni e i pulsanti di ripristino della configurazione. Questa spia ha un colore diverso in base allo stato corrente dell'oscilloscopio, come segue:

- Rosso: oscilloscopio arrestato.
- Nero: l'unità di controllo sta aspettando che l'oscilloscopio completi la registrazione dei dati acquisiti.
- Giallo: il recupero dei dati dall'unità di controllo è in corso.

Set capture options: facendo clic su questo pulsante d'opzione, viene visualizzata la finestra di dialogo per la configurazione avanzata dell'oscilloscopio.

Opzioni avanzate dell'oscilloscopio

Informazioni generali:



Visualizzazione dei punti registrati nella memoria TABLE dell'unità di controllo

Se l'oscilloscopio è configurato sia per i parametri di movimento che per i parametri di tabella, il numero dei punti tracciati sul display è determinato dalla base temporale (e dai campionamenti per divisione). Se il numero di punti da tracciare per il parametro di tabella è maggiore del numero di punti del parametro di movimento, i punti di tabella addizionali non vengono visualizzati, ma possono essere visti facendo scorrere la traccia della tabella con la barra di scorrimento orizzontale. La traccia del parametro di movimento non si sposta.



Caricamento dei dati dall'unità di controllo all'oscilloscopio
Se la base temporale complessiva è maggiore di un valore predefinito, i dati vengono presi dall'unità di controllo in blocchi, pertanto l'aggiornamento del display si svolgerà in sezioni. L'ultimo punto tracciato della sezione corrente è visualizzato in colore bianco. Se l'oscilloscopio è configurato sia per registrare i parametri di movimento che per tracciare i dati di tabella, i dati di tabella vengono letti in un blocco unito mentre i parametri di movimento vengono letti su base continua oppure in blocchi, a seconda della base temporale. Anche se l'oscilloscopio si trova in modalità continua, i dati non vengono rilette: solo i parametri di movimento vengono continuamente letti dall'unità di controllo.



Abilitazione/disabilitazione dei controlli dell'oscilloscopio
Quando l'oscilloscopio è in funzione, tutti i controlli dell'oscilloscopio sono disabilitati, ad eccezione del pulsante Trigger. Pertanto, se fosse necessario modificare la base temporale o la scala verticale, occorre arrestare l'oscilloscopio e poi riavviarlo.



Precisione della visualizzazione
L'unità di controllo registra i valori di parametro alla velocità di campionamento richiesta dalla tabella, dopodiché trasmette le informazioni all'oscilloscopio. Quindi la traccia visualizzata è accurata in rapporto alla base temporale selezionata. Tuttavia, trascorre un intervallo tra il momento in cui i dati sono registrati dall'unità di controllo e quello in cui vengono visualizzati sull'oscilloscopio, a causa del tempo richiesto per caricare i dati tramite il collegamento di comunicazione.

Samples per division (Campionamento per suddivisione):

l'impostazione predefinita dell'oscilloscopio prevede la registrazione di cinque punti per ogni suddivisione della griglia (base temporale). È possibile regolare questo valore utilizzando la barra di scorrimento adiacente.

Per ottenere la massima velocità di campionamento possibile, ridurre a 1 il numero dei campionamenti per ogni suddivisione della griglia e aumentare la scala della base temporale al valore più veloce (1 ciclo di servoazionamento per ogni suddivisione della griglia).

Table range used for data capture (Intervallo di tabella utilizzato per l'acquisizione dei dati): prima di caricare nella finestra i valori richiesti per i dati di parametro, l'unità di controllo registra questi valori come dati di tabella. Come impostazione predefinita, il valore di tabella più basso utilizzato nell'oscilloscopio è zero. Tuttavia, se questo contrasta con i programmi in esecuzione sull'unità di controllo, che potrebbero anch'essi necessitare di questa sezione della tabella, il valore di tabella più basso può essere modificato.

Il valore di tabella più alto utilizzato nell'oscilloscopio è, in seguito, automaticamente aggiornato in base al numero di canali in uso e il numero di campionamenti per ogni suddivisione della griglia.

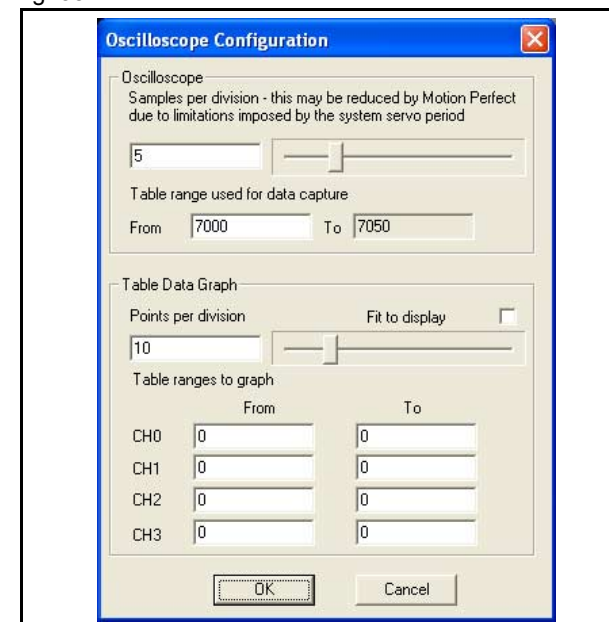
Se si inserisce un valore di tabella più basso che fa superare al valore di tabella più alto il massimo consentito sull'unità di controllo, l'oscilloscopio utilizzerà il valore originale.

Table Data Graph (Grafico dei dati della tabella): i valori della tabella dell'unità di controllo possono essere tracciati direttamente: i riquadri di testo per specificare i limiti della tabella consentono all'utente di inserire fino a quattro gruppi di indici (primo/ultimo).

Controlli dei parametri

Se vengono registrati ingressi analogici, la risoluzione più rapida dell'oscilloscopio (velocità di campionamento) corrisponde al numero di canali analogici in msec (in altre parole, per 2 ingressi analogici la velocità di campionamento più elevata è di 2 msec). La risoluzione viene calcolata dividendo il valore della scala della base temporale per il numero di campionamenti per ciascuna suddivisione della griglia. La quantità dei valori di tabella che possono essere inseriti per i canali non può superare le dimensioni massime della memoria TABLE del controllore; analogamente per l'oscilloscopio, non è possibile inserire

fig. 38



un valore di tabella più basso. Non è neppure ammesso aumentare il campionamento per ogni suddivisione della griglia a un valore che causa il superamento del valore di tabella massimo dell'unità di controllo da parte del valore di tabella superiore dell'oscilloscopio. Se quando si incrementa il numero dei campionamenti per ogni suddivisione della griglia la maggiore velocità della scala della base richiede a una risoluzione impossibile, l'oscilloscopio automaticamente reimposta il numero di campionamenti per ogni suddivisione della griglia.

Digital IO status (Stato dell'I/O digitale)

Questa finestra consente all'utente di visualizzare tutti i canali di I/O e commutare lo stato dei canali di uscita. In opzione, consente all'utente di inserire anche una descrizione per ciascuna riga di I/O.

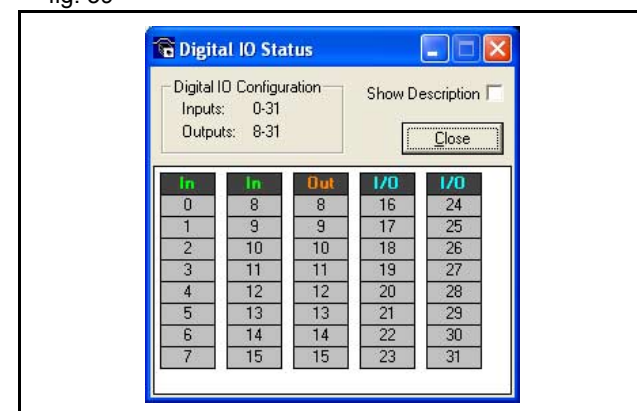
Digital inputs: mostra il numero totale dei canali di ingresso di Trajexia.

Digital outputs: mostra il numero totale dei canali di uscita di Trajexia.

Display: il display è suddiviso in insiemi di 8 indicatori, che rappresentano blocchi di 8 ingressi o uscite:

- **Insiemi di ingresso (In)**
Rappresentano lo stato degli ingressi digitali. Da In(0) a In(15) sono gli ingressi digitali incorporati di TJ1-MC__. Gli ingressi digitali aggiuntivi del sistema sono mappati automaticamente a partire da In(32).
- **Insiemi di uscita (Out)**
Rappresentano lo stato delle uscite digitali. Da OP(8) a OP(15) sono le uscite digitali incorporate di TJ1-MC__. Le uscite digitali aggiuntive del sistema sono mappate automaticamente a partire da OP(32).
- **Insiemi di Ingresso/Uscita (I/O)**
Rappresentano I/O virtuali che è possibile utilizzare all'interno del programma come flag utente. Impostando una di queste uscite virtuali, verrà impostata anche l'entrata virtuale corrispondente. Se un indicatore è grigio, l'ingresso o uscita corrispondenti non sono attivi. Se è colorato (giallo, verde, arancione, rosso, azzurro o magenta), ciò significa che l'ingresso o uscita corrispondenti sono attivi. I diversi colori sono utilizzati per rappresentare i diversi tipi di ingresso e uscita.
Facendo clic su un indicatore che rappresenta un'uscita (o su un ingresso e un'uscita collegati), quell'uscita cambierà stato. Facendo clic su un indicatore che rappresenta un'uscita, non si produce alcun effetto.

fig. 39



- Alcuni circuiti d'uscita richiedono una fonte di alimentazione esterna. In questo caso, lo stato d'ingresso degli I/O collegati internamente non viene indicato correttamente se la fonte di alimentazione esterna è assente, poiché persino nel caso in cui l'uscita fosse attiva, lo stato dell'ingresso non cambierebbe. La stessa situazione si verifica quando un'uscita entra in una condizione di limite di corrente a causa di un guasto o di un sovraccarico.
- Show description: selezionando o deselectando la casella di controllo Show Description si possono visualizzare e nascondere le descrizioni. Le descrizioni sono memorizzate nel file di progetto.

Keypad (Tastierino)

Non applicabile per Trajexia.

Jog Axes

Questa finestra consente all'utente di muovere gli assi su Trajexia. Questa finestra sfrutta le funzionalità offerte dai canali di I/O virtuali bidirezionali (da 16 a 27) di Trajexia per impostare gli ingressi di jog. Gli ingressi di avanzamento, marcia indietro e jog rapido vengono identificati scrivendo nei parametri degli assi corrispondenti e si prevede che vengano collegati agli interruttori NC. Ciò significa che quando l'ingresso è attivo (in tensione a +24 V) la funzione di jog corrispondente è DISABILITATA mentre quando l'ingresso non è attivo (0 V) la funzione di jog è ABILITATA.

Le funzioni di jog qui implementate disabilitano la funzione di jog rapido; ciò significa che la velocità alla quale sarà eseguito il jog è stabilita dal parametro degli assi JOGSPEED. Per di più, questa finestra limita la velocità di jog all'intervallo compreso tra 0 e la velocità richiesta, laddove la velocità richiesta viene fornita dal parametro degli assi SPEED.

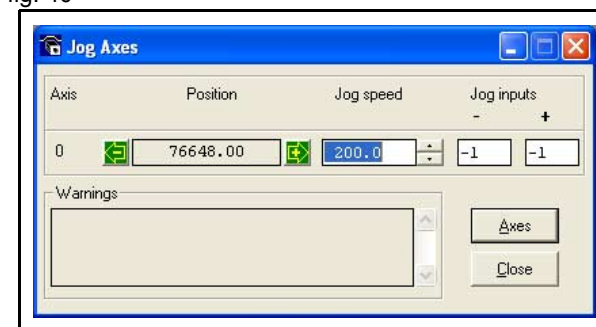
Prima di consentire l'avvio di un jog, la finestra del jog verifica che tutti i dati in essa impostati e quelli presenti su Trajexia siano validi e tali da permettere l'esecuzione di un jog.

Jog reverse: questo pulsante avvia un jog all'indietro.

A questo scopo, verificare in sequenza quando segue:

- Se si tratta di un asse SERVO e il servoazionamento non è attivo, il messaggio di avvertimento è impostato.
- Se il WatchDog non è attivo, il messaggio di avvertimento è impostato.

fig. 40



- Se la velocità di jog è 0, il messaggio di avvertimento è impostato.
- Se la velocità di accelerazione di quest'asse è 0, il messaggio di avvertimento è impostato.
- Se la velocità di decelerazione di quest'asse è 0, il messaggio di avvertimento è impostato.
- Se l'ingresso del jog all'indietro è fuori intervallo, il messaggio di avvertimento è impostato.
- Se su quest'asse è già in corso di esecuzione un movimento che non è un movimento jog, il messaggio di avvertimento è impostato.

Jog forward: questo pulsante avvia un jog in avanti. Viene eseguita una verifica identica a quella per il jog all'indietro. Se non sono impostati messaggi di avvertimento, nella finestra degli avvertimenti viene impostato il messaggio **"Forward jog set on axis?"** (Jog in avanti impostato sull'asse?), la validità dell'ingresso **FAST_JOG** viene revocata, lo **slittamento** è impostato sul valore dato nel controllo della velocità di jog e, da ultimo, l'uscita **JOG_FWD** viene disattivata, abilitando così la funzione di jog in avanti.

Jog speed: è la velocità alla quale viene eseguito il jog. Questa finestra limita questo valore all'intervallo compreso tra 0 e la velocità richiesta per quest'asse, laddove la velocità richiesta viene fornita dal parametro degli assi **SPEED**. Il valore può essere modificato tramite scrittura diretta nel controllo oppure utilizzando il controllo della velocità di jog. La barra di scorrimento modifica la velocità di jog in frazioni crescenti o decrescenti pari a 1 unità al secondo.

Jog inputs: sono gli ingressi che saranno associati alle funzioni di jog in avanti/all'indietro. Devono essere compresi nell'intervallo compreso tra 8 e il numero totale degli ingressi del sistema, poiché i canali di ingresso da 0 a 7 non sono bidirezionali e lo stato dell'ingresso non può essere impostato dall'uscita corrispondente. Se si desidera che la funzione di jog sia disabilitata, l'ingresso deve essere attivo, mentre non deve essere attivo se si vuole che funzione di jog all'indietro sia abilitata. Per rispettare questa condizione, quando l'ingresso è impostato su un numero valido prima si attiva l'uscita corrispondente e poi si imposta il parametro degli assi **REV_JOG**.

Axes: questo pulsante visualizza un riquadro per la selezione degli assi che consente all'utente di selezionare gli assi da visualizzare nella finestra Jog Axes. Come impostazione predefinita, gli assi fisici collegati all'unità di controllo vengono visualizzati.

fig. 41



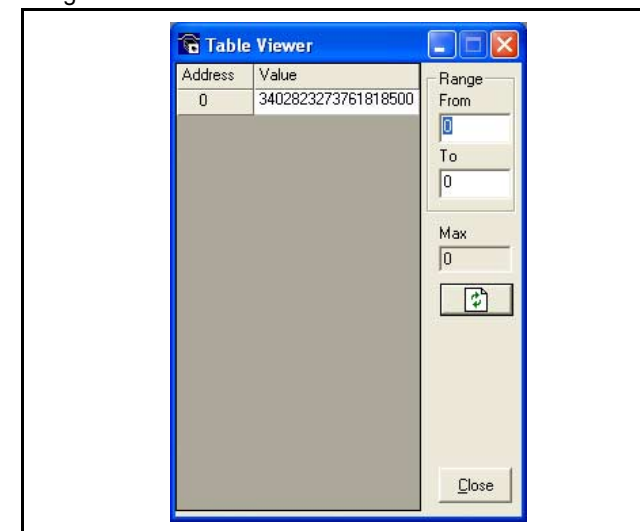
TABLE viewer

Gli strumenti TABLE e VR Editor sono molto simili. Un intervallo dei valori presenti in memoria viene visualizzato su un'interfaccia in stile foglio elettronico dalla quale possono essere modificati. Per modificare un valore, selezionarlo con il mouse, digitare il nuovo valore e premere Invio. La modifica può essere apportata mentre i programmi sono in esecuzione e sarà subito attiva.

Opzioni:

- **Range**
Entrambi gli strumenti dispongono dell'opzione che consente di impostare l'inizio e la fine dell'intervallo da visualizzare. Nello strumento TABLE Viewer il valore massimo visualizza il valore più alto che è possibile leggere (cioè il parametro di sistema **TSIZE**). Se l'intervallo dei valori supera la capacità di visualizzazione della finestra di dialogo, una barra di scorrimento consentirà di vedere tutti i valori.
- Pulsante di aggiornamento
Lo schermo non si aggiorna automaticamente; pertanto, se un valore della memoria TABLE o VR viene modificato dal programma, non sarà possibile visualizzare il nuovo valore fino a quando il display non viene aggiornato.

fig. 42



Watch variables (Supervisione delle variabili)

Non implementato per Trajexia.

Ingressi analogici

Controlla il valore presente nel modulo remoto degli ingressi analogici. Quando uno o più moduli AN2900 vengono individuati, gli ingressi sono automaticamente aggiunti al sistema a partire da AIN0.

Terminal

La finestra Terminal è un editor di testo che offre un collegamento diretto con il sistema Trajexia. La maggior parte delle funzioni che devono essere svolte durante l'installazione, la programmazione e la messa a punto di un sistema con Trajexia sono state automatizzate dalle opzioni disponibili nel menu di Trajexia Tools. Tuttavia, qualora fosse necessaria una comunicazione diretta, è possibile utilizzare la finestra Terminal.

Select channel: quando Trajexia Tools è collegato all'unità di controllo, lo strumento Terminal visualizza una finestra di dialogo per la selezione del canale di comunicazione.

Il canale 0 viene utilizzato per la riga di comando Trajexia mentre i canali 5, 6 e 7 vengono impiegati per la comunicazione con i programmi in esecuzione su Trajexia.

Per collegare uno strumento Terminal a un canale, selezionare il canale desiderato e premere **OK**. Per ogni canale è consentito il collegamento di un solo strumento Terminal (o strumento del tastierino) per volta.

fig. 43

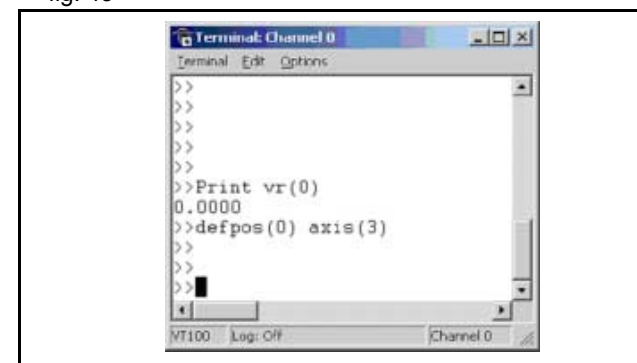
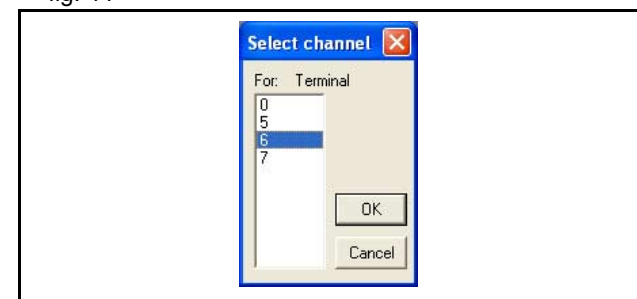


fig. 44



5.5.5 Menu Options

Attraverso il menu **Options** vengono impostate le opzioni di sistema per il sistema:

Communications

Consente la visualizzazione e la selezione delle impostazioni di comunicazione. Le impostazioni possono essere modificate solamente offline. Le varie opzioni sono.

- Serial: per altre unità di controllo assi OMRON (C200HW-MC402-E e R88-MCW151-E).
- USB: non utilizzato.
- Simulazione: utilizzato per lavorare offline; simula il sistema di un'unità di controllo assi virtuale.
- Ethernet: l'opzione utilizzata per Trajexia.
- PCI: non utilizzato.

Editor

Modifica le diverse opzioni dell'editor di testo.

fig. 45

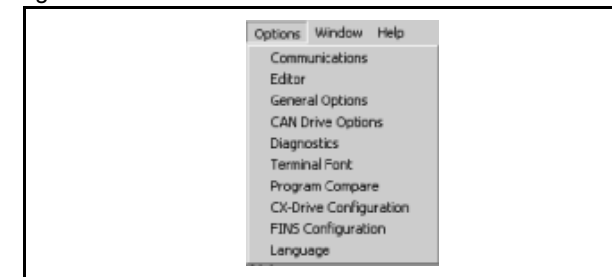
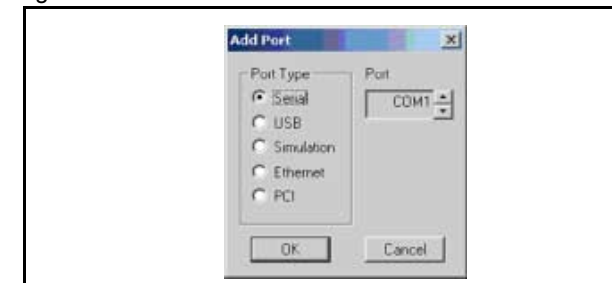


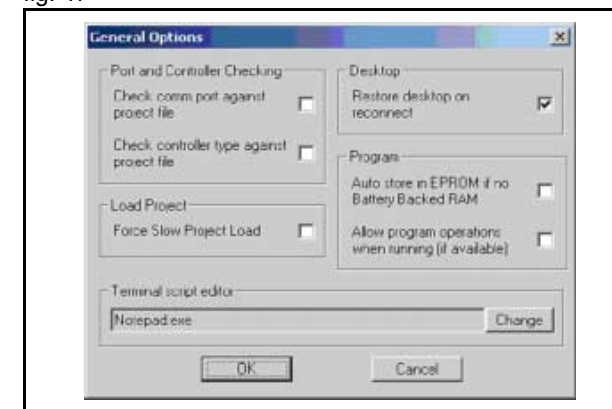
fig. 46



General Options

Consente l'impostazione di varie opzioni del sistema.

fig. 47



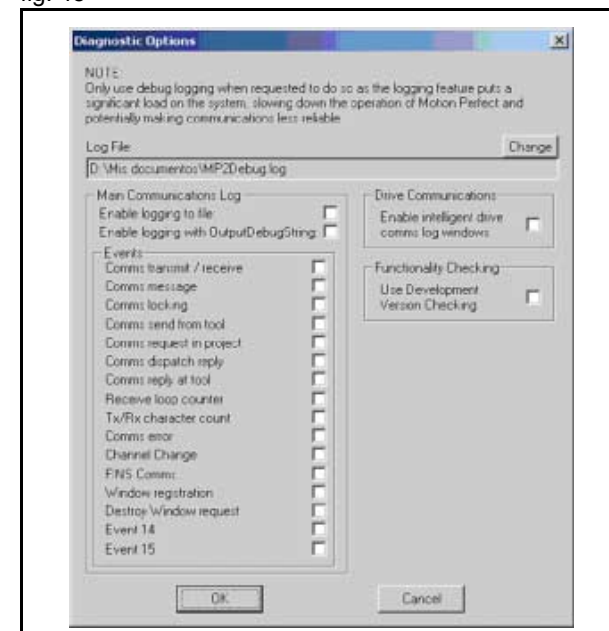
CAN Drive

Non implementato in Options.

Diagnostics

Consente di selezionare gli eventi da memorizzare in un file **.log** a fini diagnostici.

fig. 48



Terminal Font

Seleziona il font da utilizzare nella finestra del terminale. Molto utile per la messa a punto.

Program Compare

Consente di eseguire un confronto tra i programmi

CX-Drive Configuration

Consente di selezionare la directory del database di CX-Drive.

FINS Configuration

Seleziona la porta e il timeout per la comunicazione FINS.

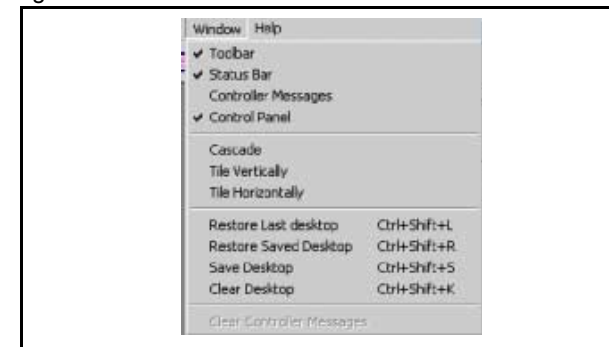
Language (Lingua)

Attualmente il programma è disponibile solo in inglese.

5.5.6 Menu Window

- Restore Last desktop/Restore Saved Desktop/Save Desktop/ Clear Desktop (Ripristina ultimo desktop/Ripristina desktop salvato/Salva desktop/Cancella desktop): strumenti a disposizione dell'utente per gestire e configurare rapidamente il desktop in base alle proprie necessità.
- Clear Controller Messages: cancella i contenuti della finestra **Controller Messages**.

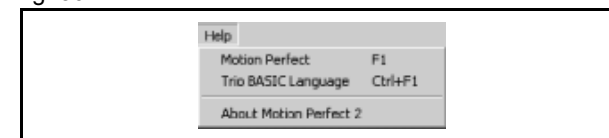
fig. 49



5.5.7 Menu Help

- Motion Perfect: apre la Guida di Trajexia Tools.
- Trio BASIC Language: la Guida dei comandi e dei parametri BASIC.
- About Motion Perfect 2: mostra la versione di Trajexia Tools.

fig. 50



6 Esempi e suggerimenti

In questo capitolo vengono forniti esempi e suggerimenti di due tipi:

- Esempi procedurali
- Esempi pratici.

6.1 Esempi procedurali

6.1.1 Programma Startup

Lo scopo di questo programma è di mettere a confronto la configurazione MECHATROLINK-II rilevata con quella prevista (la configurazione prevista è la configurazione presente al momento della creazione del programma).

Il programma STARTUP esegue queste azioni:

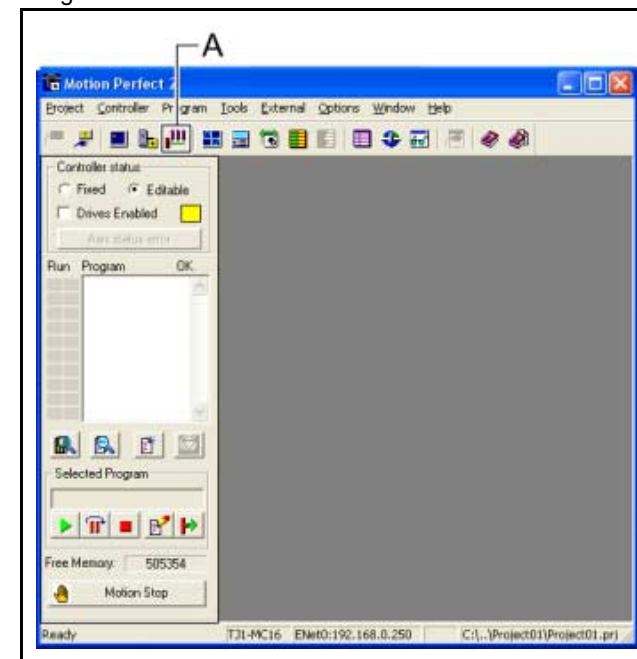
- Controlla il numero di nodi presenti nel sistema.
- Controlla che il numero di nodi coincida.
- Controlla se tutti i dispositivi sono collegati e alimentati.
- In presenza di qualsiasi discrepanza, il programma si arresta.
- Imposta il corretto **ATYPE** selezionato nella finestra Intelligent Axis.
- Imposta la modalità: **Run** (Esecuzione) o **Commissioning** (Messa a punto).

Come usare il programma Startup

L'uso consigliato del programma **STARTUP** è il seguente:

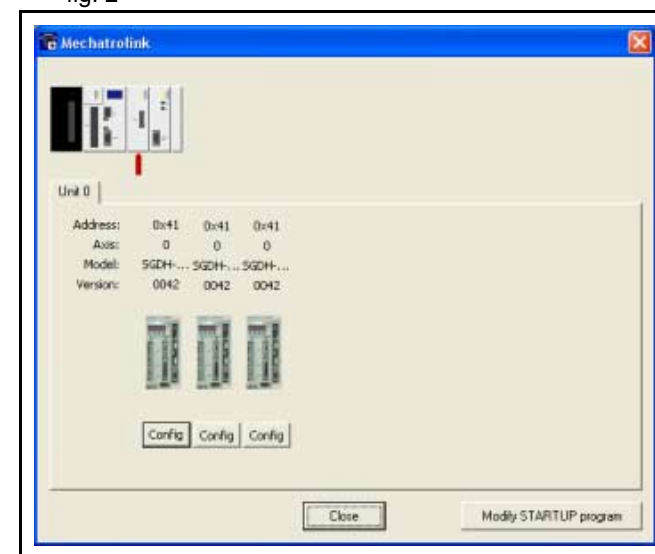
1. Fare clic sul pulsante **Intelligent drives** (Unità intelligenti) (A).

fig. 1



2. Fare clic sul pulsante **Modify STARTUP program** (Modifica programma STARTUP).
3. Al termine della sezione creata automaticamente, mettere il proprio codice di applicazione. Normalmente si tratta dell'inizializzazione di variabili e di parametri degli assi.
4. Alla fine del programma STARTUP, eseguire i programmi applicativi. Si consiglia di eseguire il programma "SHELL", spiegato di seguito in questa sezione.
5. Impostare l'esecuzione del programma STARTUP all'accensione.

fig. 2



**Nota**

Per avviare il proprio programma applicativo, OMRON consiglia di usare l'istruzione **RUN "tuo_programma"** alla fine del programma Startup. Il programma applicativo viene avviato al termine di un'esecuzione riuscita e senza errori del programma Startup.

Quando si imposta un programma applicativo per “essere eseguito all'avvio”, sussiste il rischio che la macchina si avvii se è presente un errore nel bus MECHATROLINK-II.

Esempio

```
'=====
'LA PRIMA PARTE DEL PROGRAMMA VIENE GENERATA
'AUTOMATICAMENTE DALLA FINESTRA INTELLIGENT AXIS DI
'TRAJEXIA TOOLS. CONSISTE IN UNA SEQUENZA DI VERIFICA CHE
'CONTROLLA CHE LA CONFIGURAZIONE DEGLI ASSI RILEVATA
'SIA QUELLA PREVISTA.
'IN CASO AFFERMATIVO, IL PROGRAMMA SI FERMA E AVVIA
"SHELL".
'IN CASO NEGATIVO, IL PROGRAMMA SI ARRESTA E NESSUN
ALTRO PROGRAMMA SI AVVIA.
'QUESTO PROGRAMMA VA IMPOSTATO PER ESSERE ESEGUITO
ALL'ACCENSIONE IN 'UN
'TASK A BASSA PRIORITÁ (1, IN QUESTO ESEMPIO)
'=====
'Avvia la sezione MECHATROLINK
' Controlla i dispositivi rilevati
' Unitá 0
IF NOT MECHATROLINK(0,3,0) THEN
    PRINT "Errore nel conteggio dei dispositivi per
    l'unitá 0"
    STOP
ELSE
    IF VR(0) <> 3 THEN
        PRINT "Conteggio dei dispositivi per l'unitá 0
        non corretto"
        STOP
    ENDIF
ENDIF
ENDIF
```

```
IF NOT MECHATROLINK(0,4,0,0) THEN
    PRINT "Errore nell'acquisizione dell'indirizzo
    dell'unit  0, stazione 0"
    STOP
ELSE
    IF VR(0) <> 65 THEN
        PRINT "Indirizzo non corretto per l'unit  0,
        stazione 0"
        STOP
    ENDIF
ENDIF
IF NOT MECHATROLINK(0,4,1,0) THEN
    PRINT "Errore nell'acquisizione dell'indirizzo
    dell'unit  0, stazione 1"
    STOP
ELSE
    IF VR(0) <> 66 THEN
        PRINT "Indirizzo non corretto per l'unit  0,
        stazione 1"
        STOP
    ENDIF
ENDIF
IF NOT MECHATROLINK(0,4,2,0) THEN
    PRINT "Errore nell'acquisizione dell'indirizzo
    dell'unit  0, stazione 2"
    STOP
ELSE
    IF VR(0) <> 67 THEN
        PRINT "Indirizzo non corretto per l'unit  0,
        stazione 2"
        STOP
    ENDIF
ENDIF
' Imposta i tipi di asse
' Unit  0
ATYPE AXIS(0)=40
ATYPE AXIS(1)=40
ATYPE AXIS(2)=40
' Imposta i servozionamenti in modalit  di esecuzione
' Unit  0
MECHATROLINK(0,20,65)
MECHATROLINK(0,20,66)
MECHATROLINK(0,20,67)
```

```
'Arresta la sezione MECHATROLINK
'=====
'QUESTA SEZIONE DEVE ESSERE IMPOSTATA MANUALMENTE
DALL'UTENTE
'IN BASE ALL'APPLICAZIONE. LE AZIONI CONSUETE SONO
'INIZIALIZZAZIONE DELLE VARIABILI, IMPOSTAZIONE DEGLI
ASSI/SERVOAZIONAMENTI, DENOMINAZIONE
'DELLE VARIABILI GLOBALI E AVVIO DEL PROGRAMMA "SHELL".
'=====
'Definisce i nomi delle variabili globali
GLOBAL "stato_progetto",100
GLOBAL "stato_allarme",101
GLOBAL "azione",102
'Inizializza le variabili
VR(0)=0
stato_progetto=0
stato_allarme=0
azione=0
'Avvia il programma SHELL
RUN "SHELL",2
STOP
```

6.1.2 Impostazioni del guadagno

L'impostazione del guadagno è correlata al sistema meccanico al quale il motore è collegato. Vi sono tre concetti principali:

- Coefficiente di inerzia
- Rigidità
- Frequenza di risonanza.

Questi concetti sono descritti nel Manuale di riferimento hardware al capitolo “Filosofia del sistema”.

Questa sezione mostra valori di parametro dimostrativi per:

- Guadagno anello di velocità
- Guadagno posizione proporzionale
- Guadagno velocità feed-forward.

I valori dimostrativi dei parametri di programma e movimento del sistema Trajexia sono forniti di seguito. Notare che sono appropriati per encoder a 13 bit.

Valore del parametro di servoazionamento	Descrizione
Pn103 = 716	Coefficiente di inerzia
Pn110 = 0012	Nessun autotuning
Pn202=1	Numeratore del rapporto di riduzione
Pn203=1	Denominatore del rapporto di riduzione

Valori dei parametri di movimento	Descrizione
UNITS =1	Funzionamento in conteggi encoder
SPEED=200000	Impostazione velocità
ACCEL=1000000	Impostazione accelerazione
DECEL=1000000	Impostazione decelerazione
MOVEMENT=81920	10 giri

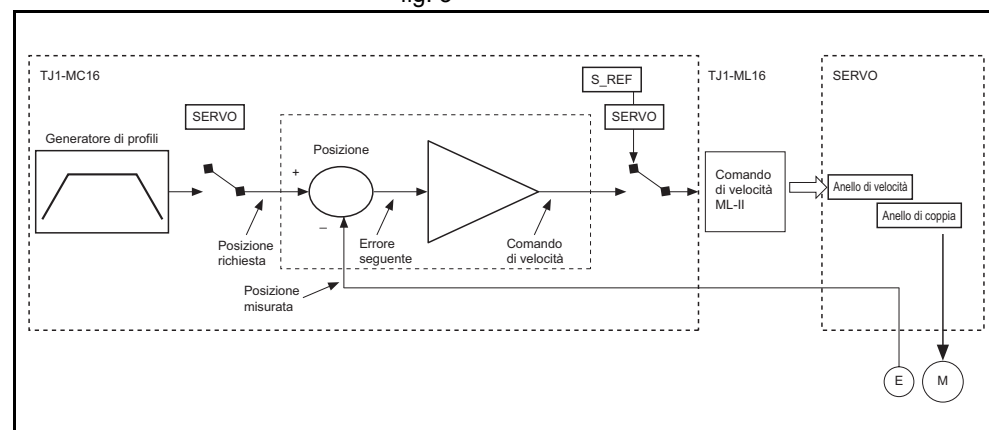
Esempi di modalità velocità

In questa modalità, l'anello di posizione viene chiuso in Trajexia mentre l'anello di velocità viene chiuso nel servoazionamento. Il parametro degli assi **Speed** viene inviato al servoazionamento attraverso la rete MECHATROLINK-II e legge la retroazione sulla posizione.

```

BASE (0)
ATYPE=41 'Modalità velocità di MECHATROLINK
SERVO=1
WDOG=1
DEFPOS (0)
anello:
    MOVE (81920)
    WAIT IDLE
    WA (100)
    DEFPOS (0)
    GOTO anello
  
```

fig. 3



Esempio 1

Solo il guadagno proporzionale possiede un valore impostato:
l'errore di inseguimento è proporzionale alla velocità.

I valori di parametro per l'esempio sono:

Valori dei parametri di movimento
P_Gain=131072
VFF_GAIN=0
Fn001=4



Nota:

I colori e la scala dell'oscilloscopio per la modalità velocità sono i seguenti:

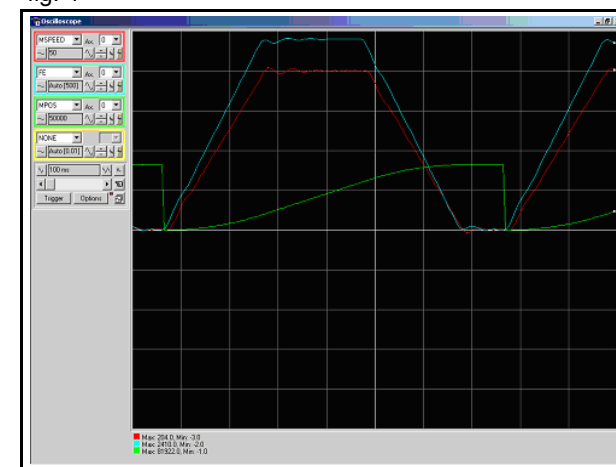
Rosso: MSPEED (velocità degli assi misurata).

L'unità è 50 unità/ms/divisione

Blu: FE (errore di inseguimento). L'unità dipende dal grafico

Verde: MPOS (posizione degli assi misurata).
50000 unità/ms/divisione

fig. 4



Esempio 2

Il valore della rigidità viene aumentato. La grandezza dell'errore rimanente invariata, ma l'ondulazione, la stabilità della velocità e la sovraelongazione migliorano.

I valori di parametro per l'esempio sono:

Valori dei parametri di movimento
P_Gain=131072
VFF_GAIN=0
Fn001=6

Esempio 3

Il parametro **P_GAIN** viene ulteriormente incrementato.

L'errore di inseguimento diminuisce proporzionalmente.

I valori di parametro per l'esempio sono:

Valori dei parametri di movimento
P_Gain=200000
VFF_GAIN=0
Fn001=6

fig. 5

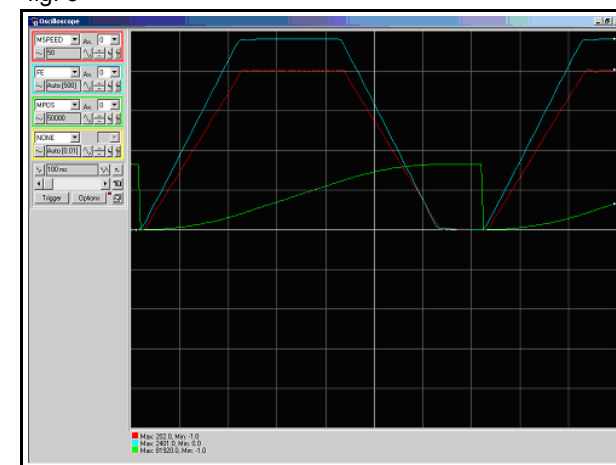
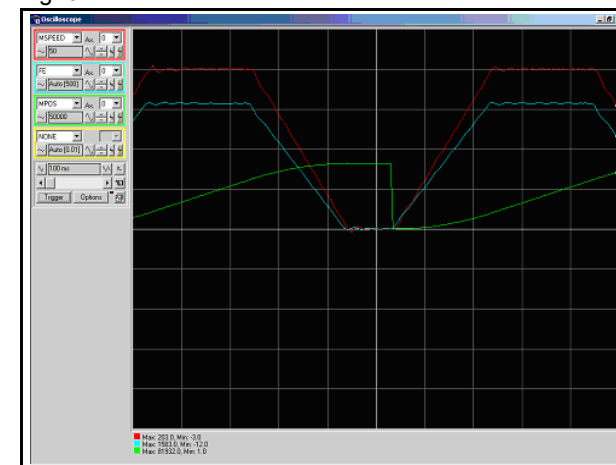


fig. 6



Esempio 4

Il valore del parametro **P_GAIN** è due volte il valore dell'esempio 1.
L'errore di inseguimento è dimezzato, ma è riscontrabile una vibrazione dovuta a guadagni eccessivi.
I valori di parametro per l'esempio sono:

Valori dei parametri di movimento
P_Gain=262144
VFF_GAIN=0
Fn001=6

Esempio 5

Il valore del parametro **P_GAIN** è impostato su quello dell'esempio 1. Il valore di **VFF_GAIN** viene aumentato. L'errore di inseguimento viene ridotto senza diminuire la stabilità. L'errore di inseguimento non è proporzionale alla velocità.
I valori di parametro per l'esempio sono:

Valori dei parametri di movimento
P_Gain=131072
VFF_GAIN=1400000
Fn001=6

fig. 7

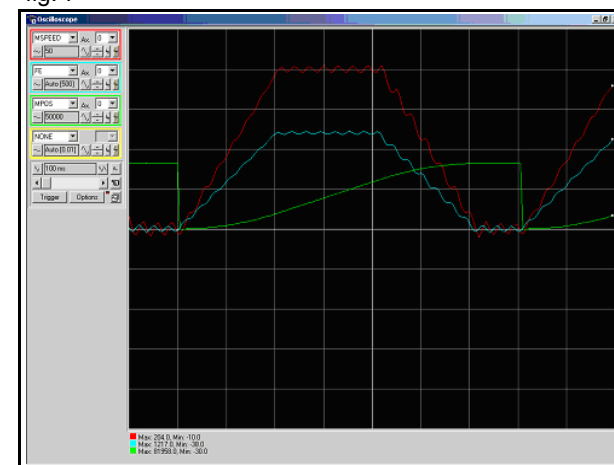
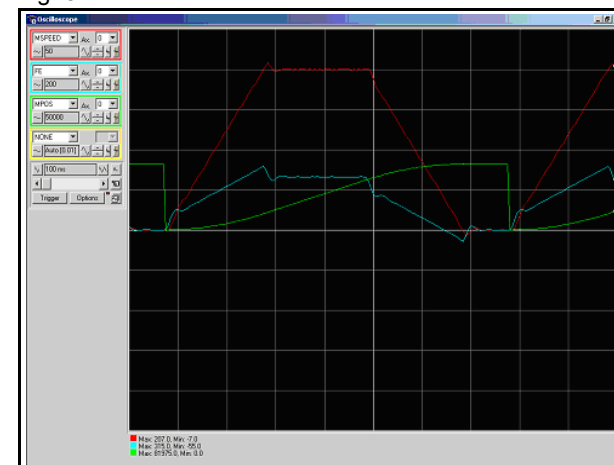


fig. 8



Esempio 6

Se VFF_GAIN ha questo valore, l'errore di inseguimento è proporzionale all'accelerazione e più basso che non nel caso del guadagno proporzionale (la scala è di 20 unità/divisione). L'errore di inseguimento si avvicina allo zero in presenza di una velocità costante.

L'effetto negativo di questo insieme di valori è rappresentato dalla sovraelongazione e sottoelongazione che si verificano quando l'accelerazione viene modificata; è possibile ridurre, ma non eliminare, questo inconveniente aumentando il guadagno dell'anello di velocità, se il sistema è in grado di gestire un guadagno elevato.

I valori di parametro per l'esempio sono:

Valori dei parametri di movimento
P_Gain=131072
VFF_GAIN=1573500
Fn001=6

Esempio 7

Il valore della rigidità viene incrementato da 6 a 8. La sovraelongazione/sottoelongazione è inferiore, ma la vibrazione del motore è aumentata.

I valori di parametro per l'esempio sono:

Valori dei parametri di movimento
P_Gain=131072
VFF_GAIN=1573500
Fn001=8

fig. 9

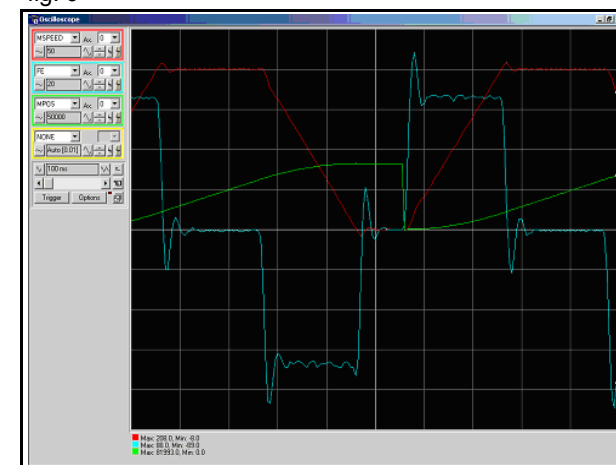
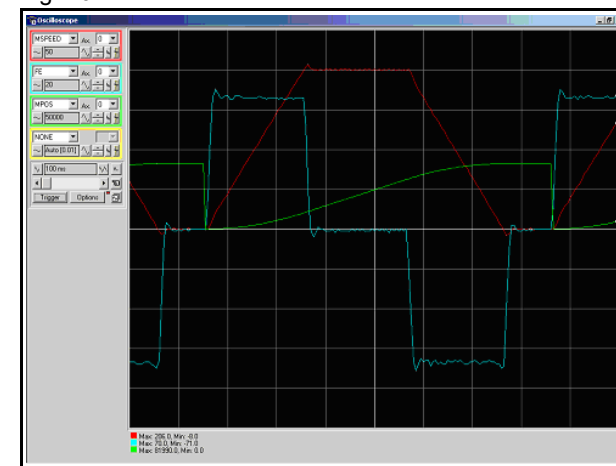


fig. 10



Esempio 8

Contrariamente a P_GAIN, dove più alto è il valore, meglio è (il limite è fissato dal momento in cui il sistema meccanico inizia a vibrare), per VFF_GAIN esiste un valore ottimale (quello dell'esempio 6); quando si supera tale valore ottimale si ottengono errori proporzionali alla velocità/accelerazione, ma di segno diverso. La correzione richiesta risulta troppo grande.

I valori di parametro per l'esempio sono:

Valori dei parametri di movimento
P_Gain=131072
VFF_GAIN=1650000
Fn001=6

Esempi di modalità posizione

In questa modalità, la posizione e l'anello di velocità vengono chiusi nel servozionamento. TJ1-ML__ invia al servozionamento il comando di posizione attraverso la rete MECHATROLINK-II e legge la retroazione sulla posizione.

Notare che questo sistema non ha un ritardo di campionamento rispetto all'anello di posizione del servozionamento, la Posizione_richiesta del ciclo "n" rispetto alla Posizione_misurata del ciclo "n".

Trajexia, per la gestione interna, continua ad utilizzare il proprio anello di posizionamento; pertanto l'errore di inseguimento letto nel parametro Axis di Trajexia non corrisponde a quello effettivo presente nel servozionamento. Per leggere l'errore di inseguimento corretto utilizzare DRIVE_MONITOR.

Regolare contemporaneamente la rigidità del servozionamento, il guadagno dell'anello di velocità e il guadagno dell'anello di posizionamento, utilizzando solamente il guadagno di posizione proporzionale. I risultati ottenuti sono analoghi a quelli che si hanno con la modalità velocità di MECHATROLINK-II, con i seguenti vantaggi:

- La regolazione è più semplice: occorre impostare solo la rigidità (Fn001) e, se necessario, il guadagno feed-forward (Pn109).

fig. 11

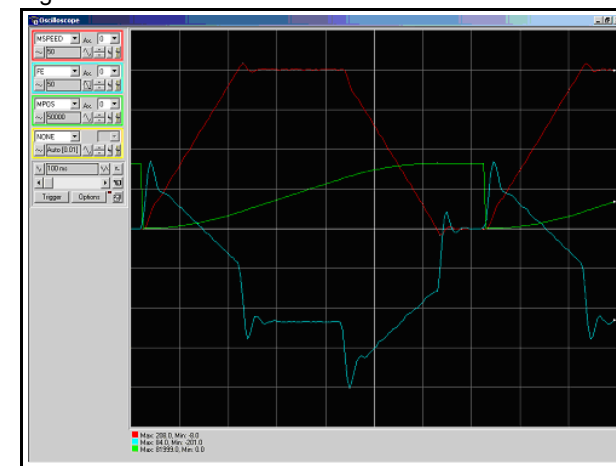
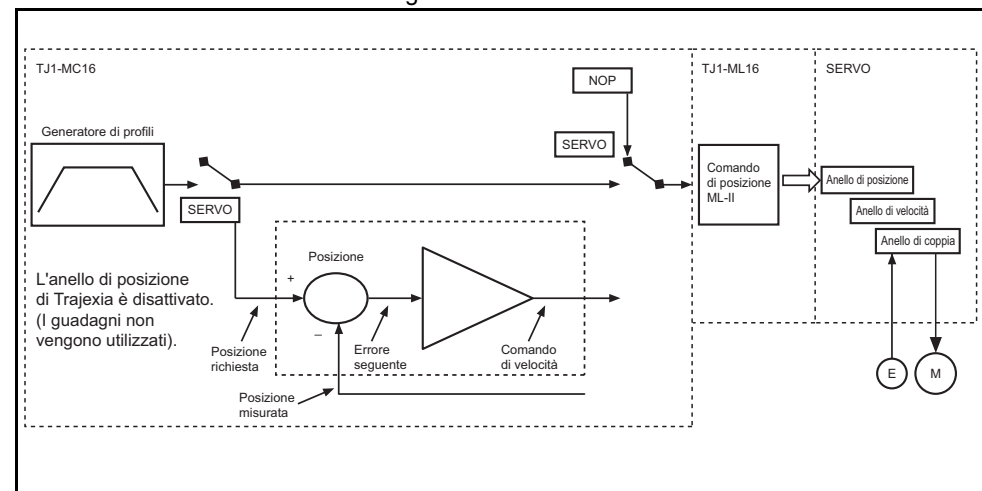


fig. 12



- L'anello di posizionamento nel servozionamento è più veloce (250µs) che non in Trajexia e gira congiuntamente all'anello di velocità.
- Non vi è alcun ritardo di campionamento tra la “Posizione di riferimento” e la “Posizione misurata”.

Se si desidera effettuare una regolazione fine, è possibile modificare individualmente i diversi parametri di guadagno.

```
BASE(0)
ATYPE=41 'Modalità posizione di MECHATROLINK
SERVO=1
DRIVE_CONTROL=2 'Per controllare l'errore
di inseguimento in
'DRIVE_MONITOR
```

```
WDOG=1
DEFPOS(0)
anello:
    MOVE(81920)
    WAIT IDLE
    WA(100)
    DEFPOS(0)
```

```
GOTO anello
```

Esempio 1

L'errore di inseguimento è proporzionale alla velocità.
Esiste un “profilo graduale” dovuto alla bassa impostazione della rigidità (guadagno basso).



Nota:

i colori e la scala dell'oscilloscopio per la modalità posizione sono i seguenti:

Rosso: MSPEED (velocità degli assi misurata).

L'unità è 50 unità/ms/divisione

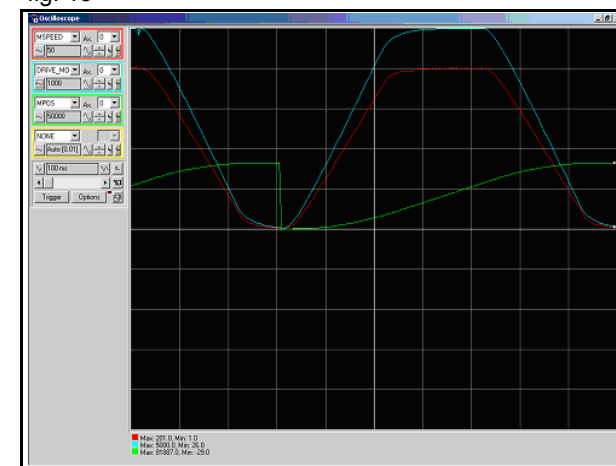
Blu: DRIVE_ MONITOR (impostato come errore di inseguimento nel servozionamento).

L'unità dipende dal grafico

Verde: MPOS (posizione degli assi misurata).

50000 unità/ms/divisione

fig. 13



I valori di parametro per l'esempio sono:

Valori dei parametri di movimento
Fn001=4
Pn109=0

Esempio 2

L'errore di inseguimento diminuisce all'aumentare della rigidità.

I valori di parametro per l'esempio sono:

Valori dei parametri di movimento
Fn001=6
Pn109=0

Esempio 3

In presenza di un guadagno elevato il motore inizia a vibrare, ma il profilo risulta più stabile che non in modalità velocità di MECHATROLINK-II.

I valori di parametro per l'esempio sono:

Valori dei parametri di movimento
Fn001=8
Pn109=0

fig. 14

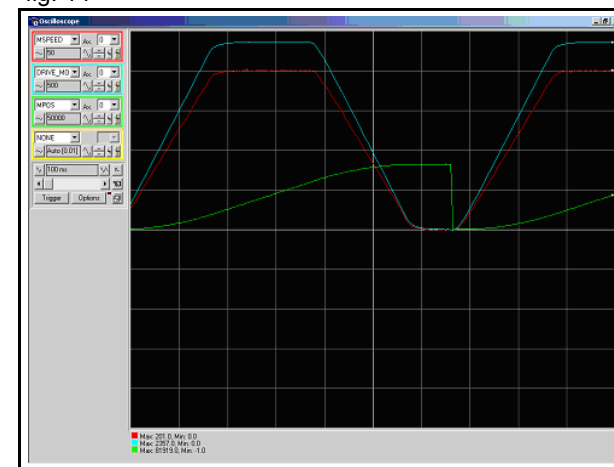
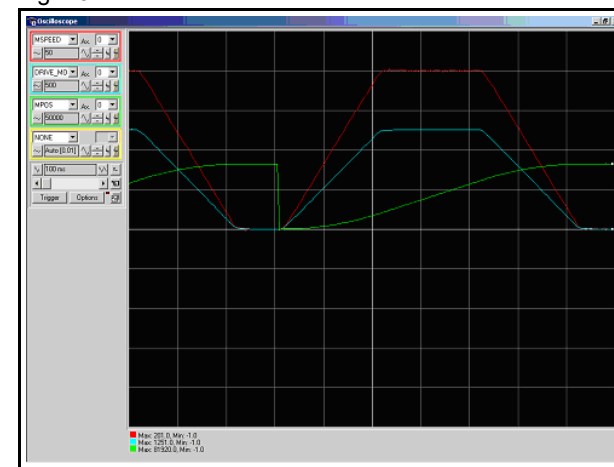


fig. 15



Esempio 4

L'effetto del guadagno feed-forward è di ridurre l'errore di inseguimento in maniera proporzionale all'accelerazione. I valori di parametro per l'esempio sono:

Valori dei parametri di movimento	
Fn001=6	
Pn109=95	

Esempio 5

Quando il feed-forward è impostato sul 100%, l'errore di inseguimento è molto contenuto e proporzionale all'accelerazione. Il valore ottimale di una correzione al 100% è il valore massimo impostabile. Il valore di parametro di Pn109 è più facile da impostare rispetto al valore di parametro di VFF_GAIN.

I valori di parametro per l'esempio sono:

Valori dei parametri di movimento	
Fn001=6	
Pn109=100	

fig. 16

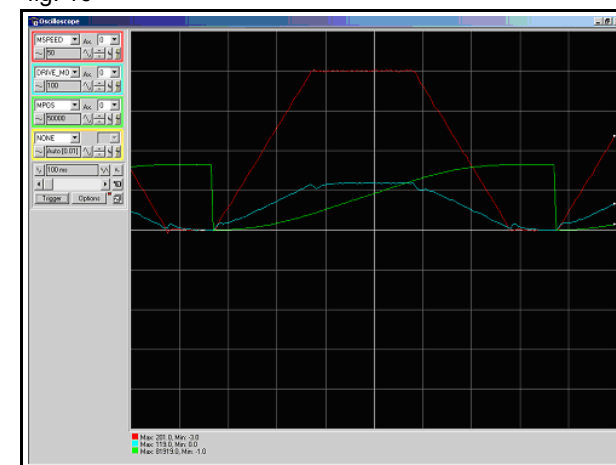
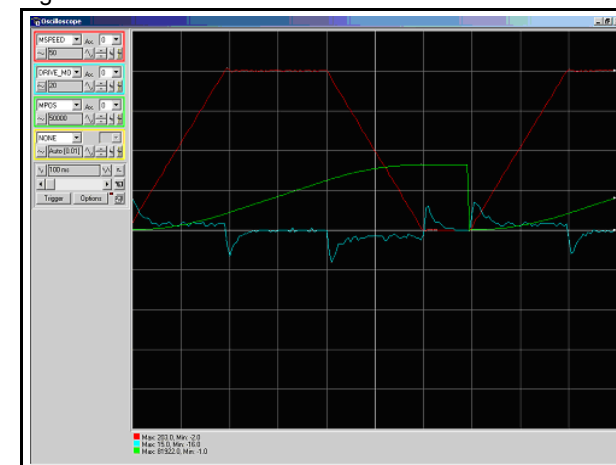


fig. 17



6.1.3 Impostazione del parametro degli assi **UNITS** e del rapporto di riduzione

Nel controllo degli assi meccanici con Trajexia TJ1-MC___, un servoazionamento e un servomotore, le uniche unità di misura che l'hardware comprende sono i conteggi encoder. Tutti i comandi inviati al driver per muovere un asse sono espressi in conteggi encoder. e tutte le informazioni di ritorno sulle posizioni dell'asse sono anch'esse espresse in conteggi encoder. Durante la scrittura di programmi BASIC per produrre dei movimenti o una sequenza di movimenti, un utente potrebbe voler definire egli stesso le unità di misura che preferisce, ad esempio millimetri, centimetri, metri, gradi, "prodotti", "rotazioni", "stazioni". Il parametro degli assi **UNITS** contiene il fattore di conversione tra conteggi encoder e unità definite dall'utente. Tutti i parametri degli assi relativi al movimento e tutti gli argomenti dei comandi degli assi che determinano la quantità di movimento sono espressi in unità utente. Questo parametro consente all'utente di definire le unità più comode per lavorare. Ad esempio, per una parte mobile che esegue un movimento lineare, potrebbe essere preferibile usare i millimetri o le frazioni di millimetro. Per una parte mobile che esegue un movimento rotatorio, invece, potrebbe essere preferibile usare i gradi o le frazioni di grado. Per maggiori informazioni sul parametro degli assi **UNITS**, vedere la sezione 3.2.270.

L'utente deve essere tuttavia consapevole del fatto che il parametro degli assi **UNITS** non è il solo elemento importante nella conversione dei conteggi encoder in unità utente. Alcuni parametri di servoazionamento e alcune caratteristiche del sistema meccanico sono anch'essi rilevanti. Le sezioni di seguito descrivono quali parametri di servoazionamento sono importanti per questa conversione. Vengono inoltre forniti alcuni esempi di impostazione di questi parametri e del parametro degli assi **UNITS**, tenendo in considerazione le caratteristiche meccaniche del sistema.

Conversione tra conteggi encoder e unità definite dall'utente

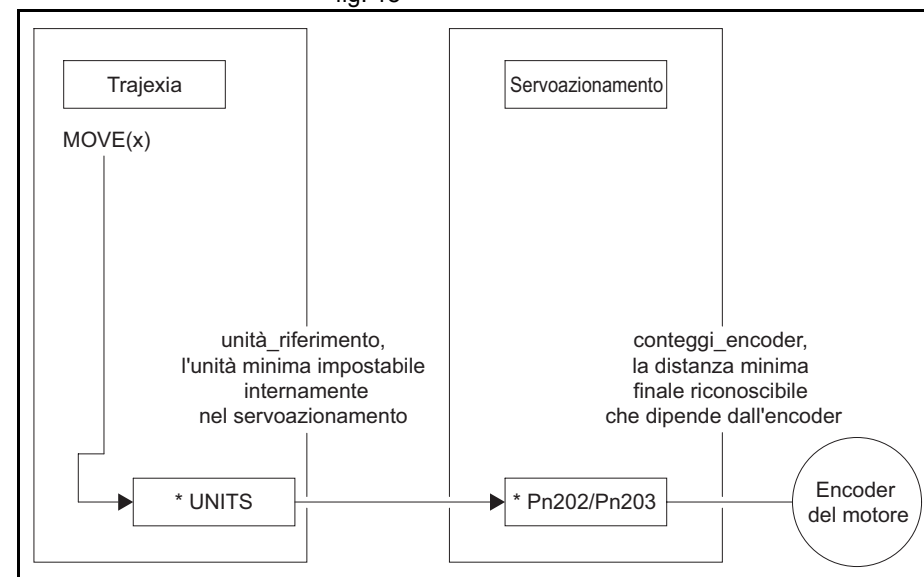
Nella conversione dei conteggi encoder in unità utente, due parametri molto importanti del servozionamento Sigma-II sono Pn202 e Pn203. Se si utilizza un servomotore con un encoder assoluto, è necessario anche impostare il parametro Pn205. Il parametro Pn202 è il denominatore del rapporto di riduzione elettronico (G1). Il parametro Pn203 è il numeratore del rapporto di riduzione elettronico (G2). Il servomotore ruota utilizzando il valore del segnale del comando di posizione inviato da TJ1-MC__, moltiplicato per il rapporto di riduzione elettronico (Pn202, Pn203). Dal lato uscita (servomotore), il segnale è espresso in numero di impulsi dell'encoder. Per maggiori informazioni sui parametri Pn202 e Pn203 del servozionamento, consultare il Manuale del servozionamento Sigma-II.

Il parametro degli assi **UNITS** esprime effettivamente il rapporto esistente tra le unità che l'utente desidera utilizzare nel programma e la posizione inviata al servozionamento tramite il bus MECHATROLINK-II. Prendendo in considerazione il rapporto di riduzione elettronico, l'equazione che esprime la relazione tra le unità utente, il parametro **UNITS**, i parametri Pn202 e Pn203, gli impulsi dell'encoder e le unità di misurazione meccanica è:

$$\frac{Pn202}{Pn203} \cdot UNITS = \frac{y \cdot conteggi\_encoder}{x \cdot unità\_utente}$$

dove y è il numero dei conteggi encoder e x è la quantità espressa in unità utente.

fig. 18



Esempio 1

Il sistema meccanico consiste in un semplice tavolo girevole. Viene utilizzato un servomotore con un encoder incrementale a 13 bit e il rapporto di riduzione della trasmissione è di 1:10. Le unità utente desiderate sono i gradi. Questo sistema può essere descritto attraverso le seguenti equazioni:

$$1 \cdot \text{rivoluzione\_motore} = 2^{13} \cdot \text{conteggi\_encoder}$$

$$10 \cdot \text{rivoluzione\_motore} = 1 \cdot \text{ciclo\_macchina}$$

$$1 \cdot \text{ciclo\_macchina} = 360^\circ$$

Combinando queste equazioni, si ottiene il seguente risultato:

$$\frac{Pn202}{Pn203} \cdot \text{UNITS} = \frac{2^{13} \cdot \text{conteggi\_encoder}}{1 \cdot \text{rivoluzione\_motore}} \frac{10 \cdot \text{rivoluzione\_motore}}{1 \cdot \text{rivoluzione\_macchina}} \frac{1 \cdot \text{rivoluzione\_macchina}}{360^\circ} =$$

$$\frac{2^{13} \cdot 10}{360} \frac{\text{conteggi\_encoder}}{\text{grado}}$$

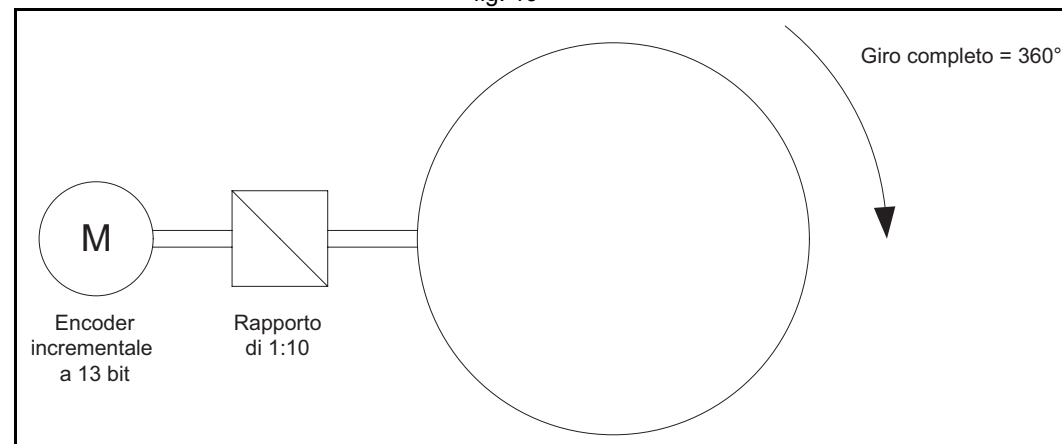
E quindi:

$$\frac{Pn202}{Pn203} \cdot \text{UNITS} = \frac{2^{13} \cdot 10}{360}$$

Da questa equazione è possibile derivare i valori per Pn202, Pn203 e **UNITS**, con le seguenti restrizioni e raccomandazioni:

1. Pn202 e Pn203 sono numeri interi.
2. UNITS non deve avere una quantità infinita di decimali. Questi infatti possono causare errori di arrotondamento che, producendo piccoli errori di posizione, provocano poi considerevoli errori di posizione cumulativi.
3. Per motivi di stabilità, è necessario evitare situazioni dove Pn202/Pn203 è minore di 0,01 o maggiore di 100.

fig. 19



È quindi possibile riscrivere l'ultima equazione come segue:

$$\text{UNITS} \cdot \frac{\text{Pn202}}{\text{Pn203}} = 2^{13} \frac{10}{360}$$

Una soluzione di questa equazione è:

$$\text{UNITS} = 2^{13} = 8.192$$

$$\text{Pn202} = 10$$

$$\text{Pn203} = 360$$

Tenendo in considerazione il terzo requisito dell'elenco precedente (evitare situazioni dove $\text{Pn202}/\text{Pn203}$ è minore di 0,01 o maggiore di 100), è possibile riscrivere l'ultima equazione come segue:

$$\text{UNITS} \cdot \frac{\text{Pn202}}{\text{Pn203}} = 2^{13} \frac{10}{360} = 2^8 \frac{2^5}{36} = 2^8 \frac{32}{36}$$

Che fornisce la soluzione:

$$\text{UNITS} = 2^8 = 256$$

$$\text{Pn202} = 32$$

$$\text{Pn203} = 36$$

Con questi valori, il comando **MOVE(28)** ruota il tavolo di 28 gradi in direzione positiva.

Impostazione dell'encoder assoluto

L'encoder assoluto conserva la posizione corrente del motore, anche in assenza di alimentazione. L'encoder assoluto fornisce la posizione nell'ambito di un giro (cioè, una frazione da 0 a 1 escluso) ed è provvisto di un contatore multigiro. Il comportamento multigiro dell'encoder assoluto può essere impostato attraverso il parametro Pn205 del servozionamento Sigma-II. Questo parametro regola il numero massimo di giri che il contatore può conteggiare prima che si verifichi un overflow. Per maggiori

informazioni sul parametro Pn205 del servozionamento, consultare il Manuale del servozionamento Sigma-II. Prendendo in considerazione il valore di questo parametro, il valore di posizione massimo che l'encoder può segnalare è:

$$\text{valore\_max\_conteggio\_encoder} = (\text{Pn205} + 1) \cdot \text{conteggi\_encoder} - 1$$

che produce Pn205 giri completi più la posizione compresa all'interno di un giro (la frazione da 0 a 1 escluso). Quando si stabilisce il collegamento tra MECHATROLINK e il servozionamento, la posizione dell'encoder assoluto viene letta sul servozionamento e il relativo valore scritto in **MPOS** (dopo la conversione: **UNITS** × Pn202/Pn203). Quando il sistema meccanico deve coprire una distanza limitata, come nel caso di una vite a circolazione di sfere, il parametro Pn205 dovrebbe essere impostato su un valore sufficientemente alto da produrre un overflow del contatore al di fuori della posizione effettiva. Ciò viene denominato asse limitato o asse finito. Un esempio tipico di asse limitato è una vite a circolazione di sfere, come viene mostrato in fig. 24. Quando il sistema meccanico si muove sempre nella stessa direzione, raggiunge l'overflow del contatore multigiro. In tal caso, il valore di Pn205 deve garantire che l'overflow si verifichi sempre nella stessa posizione in relazione alla macchina. Questo viene denominato asse illimitato e un esempio tipico di ciò è una tavola rotante come quella mostrata in fig. 20. Può essere ottenuto attraverso la seguente equazione: il valore più piccolo di m tale per cui:

$$n \cdot \text{cicli\_macchina} = m \cdot \text{rivoluzione\_motore}$$

Dal momento che n e m sono numeri interi: $\text{Pn205} = m - 1$. Questa impostazione è spiegata nell'esempio di seguito.

Esempio 2

Il sistema meccanico consiste in un semplice tavolo girevole mostrato in figura. Viene utilizzato un servomotore con un encoder assoluto a 16 bit. Il rapporto di riduzione della trasmissione è di 1:10. Le unità utente desiderate sono i gradi. Il tavolo girevole è suddiviso in sei sezioni di 60 gradi. Pertanto il ciclo_macchina è pari a 60 gradi.

Applicando l'ultima equazione a quanto sopra si ottiene:

$$10 \cdot \text{rivoluzione\_motore} = 1 \cdot \text{rivoluzione\_macchina} = 6 \cdot \text{ciclo\_macchina}$$

La semplificazione di questa equazione dà:

$$5 \cdot \text{rivoluzione\_motore} = 3 \cdot \text{ciclo\_macchina}$$

Il risultato di questo è:

$$\text{Pn205} = 5 - 1 = 4$$

Calcolando i parametri come si è fatto nell'esempio 1 si ottiene:

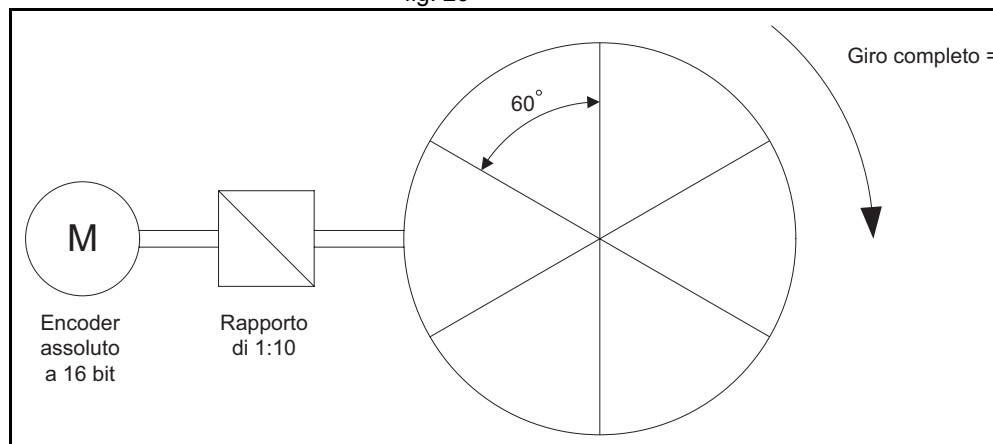
$$\text{UNITS} = 2^{11} = 2.048$$

$$\text{Pn202} = 32$$

$$\text{Pn203} = 36$$

Per garantire il corretto overflow sia in Trajexia che nel servozionamento, è necessario impostare due parametri degli assi aggiuntivi: **REP_DIST** = 60 e **REP_OPTION** = 1. Con queste impostazioni, il comando **MOVE(35)** ruota il tavolo di 35 gradi in direzione positiva. L'intervallo dei possibili valori **MPOS** e **DPOS** va da 0 a 60 gradi.

fig. 20





È necessario inizializzare l'encoder assoluto prima di usarlo per la prima volta, quando la batteria si è scaricata in assenza di alimentazione e quando viene modificata l'impostazione del limite di multigirotto nel parametro Pn205. È possibile eseguire l'inizializzazione sul display del servozionamento ovvero con lo strumento software. Per maggiori dettagli sull'inizializzazione dell'encoder assoluto, consultare il Manuale del servozionamento Sigma-II.



È possibile reimpostare il contatore multigirotto, ma non è possibile ridefinire la posizione all'interno di un giro (la frazione compresa tra 0 e 1 escluso). Per regolare l'offset zero, utilizzare il parametro Pn808. Per maggiori dettagli, consultare il manuale del Modulo di interfaccia MECHATROLINK-II NS115.



All'avvio, la posizione dell'encoder assoluto viene letta sul motore e scritta in **MPOS** utilizzando la seguente conversione:

- Per **MPOS**:

$$\text{MPOS\_assoluto} = \text{posizione\_ass\_encoder} \cdot \frac{1}{\text{UNITS}} \cdot \frac{\text{Pn203}}{\text{Pn202}}$$

- Tale conversione è corretta se:

$$(\text{Pn205} + 1) \cdot \frac{\text{Pn203}}{\text{Pn202}} \cdot \text{conteggi\_encoder} < 5.000.000$$

- Se questo valore è maggiore di 5.000.000, è possibile che **MPOS** contenga valori errati all'avvio. Per evitare questo problema, aggiungere il codice di programma **DEFPOS = ENCODER/UNITS** dopo tutte le inizializzazioni **UNITS**.

Esempio 3

Il sistema meccanico utilizza un servomotore con un encoder assoluto a 17 bit. Il rapporto di riduzione meccanico della trasmissione è di 1:6,31. Una rotazione della puleggia sposta la parte mobile di 320 mm lungo il nastro. La lunghezza totale del nastro, e quindi la capacità di spostamento totale della parte mobile, è di 4160 mm.

Le unità di misura meccanica devono essere i mm. Ciò significa che tutti i parametri e i comandi degli assi inviati a Trajexia sono espressi in mm. Utilizzando la stessa procedura dell'esempio 1, l'equazione che esprime la relazione tra le unità utente e il conteggio encoder è:

$$\frac{Pn202}{Pn203} \text{ UNITS} = \frac{2^{17} \cdot \text{conteggi\_encoder}}{1 \cdot \text{rivoluzione\_motore}} \frac{6,31 \cdot \text{rivoluzione\_motore}}{1 \cdot \text{rivoluzione\_puleggia}} \frac{1 \cdot \text{rivoluzione\_puleggia}}{320 \text{ mm}} =$$

$$\frac{2^{17} \cdot 6,31}{320} \frac{\text{conteggi\_encoder}}{\text{mm}}$$

Quindi:

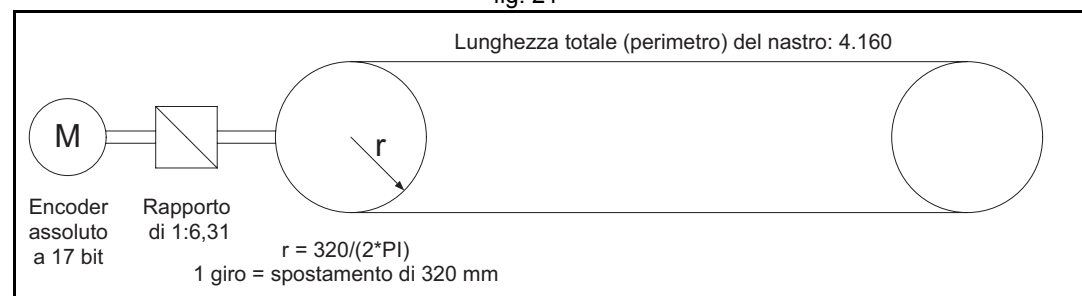
$$\frac{Pn202}{Pn203} \text{ UNITS} = \frac{2^{17} \cdot 6,31}{320} = \frac{2^{17}}{2^5} \frac{631}{1.000} = 2^{12} \frac{631}{8.125} = 2^{12} \frac{631}{2^3 \cdot 125} = 2^9 \frac{631}{125}$$

Una soluzione è:

$$\begin{aligned} \text{UNITS} &= 2^9 = 512 \\ Pn202 &= 631 \\ Pn203 &= 125 \end{aligned}$$

Notare che nel calcolo non è stato utilizzato il raggio della puleggia. Ciò è stato fatto per evitare l'uso di π , che non può essere espresso come frazione. Nelle pulegge dentate, normalmente vengono usati il numero di denti e i millimetri per dente.

fig. 21



Il calcolo dell'impostazione limite multigiro è:

$$m \cdot \text{rivoluzione\_motore} = n \cdot \text{ciclo\_macchina}$$

$$m \cdot \text{rivoluzione\_motore} = n \cdot \text{ciclo\_macchina} \frac{4.160 \cdot \text{rivoluzione\_puleggia}}{320 \cdot \text{ciclo\_macchina}} = n \cdot 13 \cdot \text{rivoluzione\_puleggia}$$

$$= n \cdot 13 \frac{6 \cdot 31 \text{ rivoluzione\_motore}}{1 \text{ rivoluzione\_puleggia}} = n \cdot 82,03 \cdot \text{rivoluzione\_puleggia}$$

$$m = n \cdot 82,03$$

Il numero intero m più piccolo per il quale questa equazione è valida è 8203. Ciò dà come risultato $Pn205 = 8202$. Inoltre, per limitare l'intervallo delle unità di movimento all'intervallo di movimento della parte mobile, occorre impostare i seguenti parametri degli assi: **REP_DIST = 4260** e **REP_OPTION = 1**. Con queste impostazioni, l'esecuzione di **MOVE(38)** sposta in avanti la parte mobile di 38 mm. L'intervallo dei possibili valori **MPOS** e **DPOS** va da 0 a 4160 mm.

Esempio 4

Il sistema meccanico utilizza un servomotore con un encoder assoluto a 17 bit. Il rapporto di riduzione meccanico della trasmissione è di 1:12,24. Le unità di misura meccanica devono essere i decimi di grado. Pertanto, la distanza di ripetizione totale per il giro completo della parte mobile è pari a 3600 decimi di grado.

Seguendo la stessa procedura dell'esempio 1, si ha:

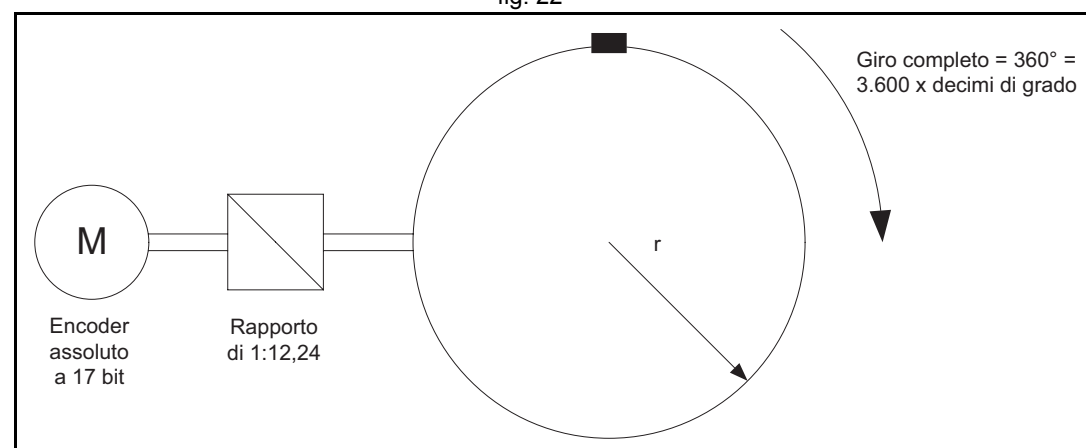
$$\frac{Pn202}{Pn203} \text{ UNITS} = \frac{2^{17} \cdot \text{conteggi\_encoder}}{1 \cdot \text{rivoluzione\_motore}} \frac{12,24 \cdot \text{rivoluzione\_motore}}{1 \cdot \text{rivoluzione\_macchina}} \frac{1 \cdot \text{rivoluzione\_puleggia}}{3.600 \text{ decimi di grado}} =$$

$$= \frac{2^{17} \cdot 12,24}{3.600} \frac{\text{conteggi\_encoder}}{\text{decimo di grado}}$$

Quindi:

$$\text{UNITS} = \frac{Pn202}{Pn203} = 2^{17} \frac{1.224}{360.000}$$

fig. 22



Una soluzione è:

```
UNITS = 217 = 131.072  
Pn202 = 1.224  
Pn203 = 360.000
```

Dal momento che il massimo comun divisore di Pn202 e Pn203 deve essere 1, si ottiene: Pn202 = 17 e Pn203 = 500. Pertanto, i parametri sono:

```
UNITS = 131.072  
Pn202 = 17  
Pn203 = 500  
Pn205 = 16  
REP_DIST = 3.600  
REP_OPTION = 1
```

Per calcolare l'impostazione limite multigiro Pn205, si ha:

$$m \cdot \text{rivoluzione\_motore} = n \cdot \text{ciclo\_macchina} = n \cdot 12,24 \cdot \text{rivoluzione\_motore}$$

La soluzione evidente è: $n = 100$ e $m = 1224$.

Ovvero, se si semplificano i fattori: $n = 25$ e $m = 306$.

Quindi: $Pn205 = m - 1 = 305$. Con queste impostazioni, l'esecuzione di **MOVE(180)** sposta la parte mobile in avanti di 180 decimi di grado ovvero 18 gradi.

Esempio 5

Il sistema meccanico utilizza un servomotore con un encoder assoluto a 17 bit. Il rapporto di riduzione della trasmissione è di 1:10. La puleggia è dotata di 12 denti, ciascuno dei quali dista dall'altro 50 mm. Un giro completo della puleggia equivale a 144 stazioni sulla ruota principale. La distanza tra due stazioni è di 50 mm. Le unità di misura meccanica devono essere i mm. La distanza di ripetizione totale deve corrispondere alla distanza tra due stazioni, cioè 50 mm.

Seguendo la stessa procedura dell'esempio 1, si ha:

$$\begin{aligned} \frac{Pn202}{Pn203} \text{ UNITS} &= \\ \frac{2^{17} \cdot \text{conteggi\_encoder}}{1 \cdot \text{rivoluzione\_motore}} \cdot \frac{10 \cdot \text{rivoluzione\_motore}}{1 \cdot \text{rivoluzione\_puleggia}} \cdot \frac{1 \cdot \text{rivoluzione\_puleggia}}{12 \cdot \text{stazione}} \cdot \frac{1 \cdot \text{stazione}}{50 \text{ mm}} &= \\ = \frac{2^{17} \cdot 10}{12 \cdot 50} \frac{\text{conteggi\_encoder}}{\text{mm}} \end{aligned}$$

Pertanto, se si utilizza il sistema meccanico per impostare il rapporto di riduzione elettronico, si ha:

$$\text{UNITS} \frac{Pn202}{Pn203} = \frac{2^{17}}{50} \frac{10}{12}$$

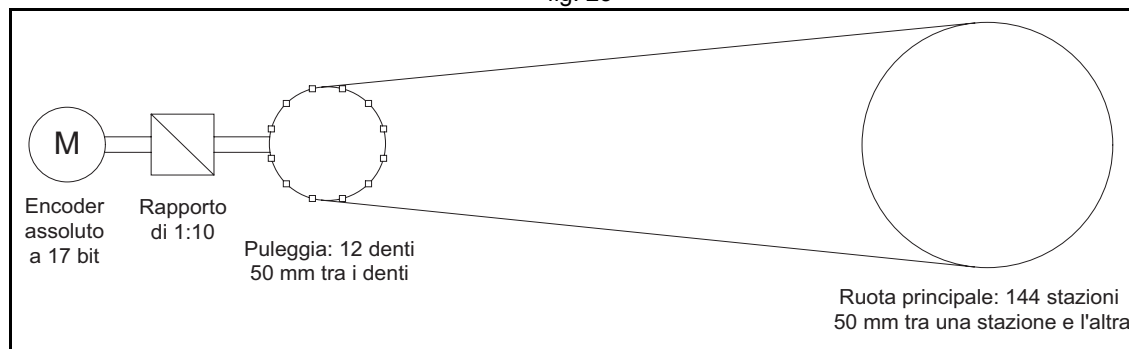
Una possibile soluzione è:

$$\begin{aligned} \text{UNITS} &= \frac{2^{17}}{50} \\ Pn202 &= 5 \\ Pn203 &= 6 \\ Pn205 &= 4 \end{aligned}$$

Dal momento che $2^{17}/50$ è un numero con una quantità infinita di decimali, si può procedere come segue:

$$\text{UNITS} \frac{Pn202}{Pn203} = 2^{17} \frac{10}{50 \cdot 12} = 2^{17} \frac{10}{600} = 2^{17} \frac{1}{60} = 2^{17} \frac{1}{2^2 \cdot 15} = 2^{15} \frac{1}{15}$$

fig. 23



Pertanto, i parametri sono:

$UNITS = 2^{15} = 32.768$
 $Pn202 = 1$
 $Pn203 = 15$
 $Pn205 = 4$
 $REP\_DIST = 50$
 $REP\_OPTION = 1$

Con queste impostazioni, l'esecuzione di **MOVE(50)** sposta in avanti la parte mobile di 50 mm ovvero di una stazione.

Esempio 6

Il sistema meccanico consiste in una vite a circolazione di sfere. Utilizza un servomotore con un encoder assoluto a 17 bit. Il rapporto di riduzione della trasmissione è di 1:3. Il passo della vite a circolazione di sfere è di 10 mm per rivoluzione. La distanza di spostamento totale della vite a circolazione di sfere è di 540 mm. Le unità di misura meccanica devono essere i mm.

Seguendo la stessa procedura dell'esempio 1, si ha:

$$\frac{Pn202}{Pn203} UNITS = \frac{2^{17} \cdot \text{conteggi\_encoder}}{1 \cdot \text{rivoluzione\_motore}} \cdot \frac{3 \cdot \text{rivoluzione\_motore}}{1 \cdot \text{rivoluzione\_vite\_a\_sfere}} \cdot \frac{1 \cdot \text{rivoluzione\_vite\_a\_sfere}}{10 \text{ mm}} =$$

$$= \frac{2^{17} \cdot 3}{10} \frac{\text{conteggi\_encoder}}{\text{mm}}$$

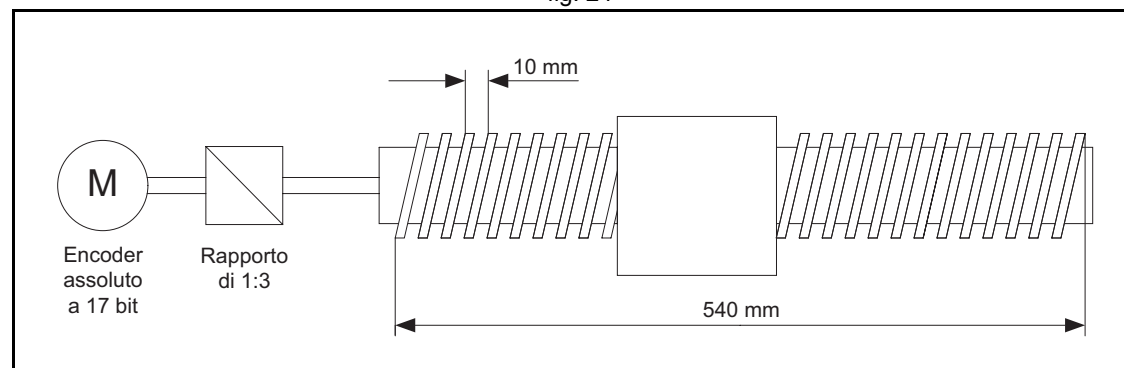
Quindi:

$$\frac{Pn202}{Pn203} UNITS = 2^{17} \frac{3}{10} = 2^{17} \frac{3}{2 \cdot 5} = 2^{16} \frac{3}{5}$$

Una soluzione è:

$UNITS = 2^{16} = 65.536$
 $Pn202 = 3$
 $Pn203 = 5$

fig. 24



Il calcolo del parametro di impostazione del limite di multigiro Pn205 non è necessario in questo caso perché la vite a circolazione di sfere è un sistema ad asse fisso (limitato). È sufficiente impostare questo valore a un livello abbastanza elevato per collocare l'overflow del contatore al di fuori della posizione effettiva. Inoltre, dal momento che l'asse è limitato, non è importante impostare il parametro **REP_OPTION**, poiché l'impostazione di **REP_DIST** deve essere sufficientemente ampia da cadere fuori della posizione effettiva massima (540 mm). Una soluzione è: **REP_DIST** = 1000 e **REP_OPTION** = 0. Con queste impostazioni, l'esecuzione di **MOVE(17)** sposta in avanti la vite a circolazione di sfere di 17 mm.

6.1.4 Mappatura degli ingressi e delle uscite del servozionamento

L'unità di controllo di Trajexia è dotata di uno spazio digitale di I/O composto di 256 ingressi digitali e 256 uscite digitali.

L'intervallo delle uscite digitale consiste di quattro parti:

- Uscite digitali 0 – 7.
Queste uscite non esistono fisicamente su TJ1-MC__. Scrivendo su queste uscite, non succede nulla. Leggendo queste uscite, viene restituito uno 0.
- Uscite digitali 8 – 15.
Queste uscite esistono fisicamente su TJ1-MC__. È possibile accedervi fisicamente sul connettore a molla a 28 pin sul lato anteriore di TJ1-MC__ (per maggiori dettagli, vedere il Manuale di riferimento hardware). Scrivendo su queste uscite, esse si attivano ed emettono un segnale a 24 V c.c. Leggendo queste uscite, esse restituiscono il proprio stato corrente. Per scrivere e leggere queste uscite, utilizzare il comando **OP**.
- Uscite digitali 16 – 31.
Queste uscite sono esclusivamente uscite software. Non esistono fisicamente su TJ1-MC__, è possibile scrivervi e leggerne lo stato corretto. Queste uscite sono utilizzate principalmente nei programmi BASIC per eseguire alcune sequenze di controllo che richiedono uscite non necessariamente fisiche. Per scrivere e leggere queste uscite, utilizzare il comando **OP**.

- Uscite digitali 32 – 255.
Queste uscite sono fisicamente presenti solo se a TJ1-MC__ sono collegati moduli di I/O digitali aggiuntivi tramite il bus MECHATROLINK-II. La lettura e la scrittura di queste uscite, se non esistono fisicamente (cioè se i moduli di I/O non sono collegati) non produce alcun effetto. Per scrivere e leggere queste uscite, utilizzare il comando **OP**.

Tutte le uscite sono caratteristica esclusiva dell'unità di controllo. Non vi si accede in base ai singoli assi.

L'intervallo degli ingressi digitali consiste di quattro parti:

- Ingressi digitali 0 – 15.
Questi ingressi esistono fisicamente su TJ1-MC__. È possibile accedervi fisicamente sul connettore a molla a 28 pin sul lato anteriore di TJ1-MC__ (per maggiori dettagli, vedere il Manuale di riferimento hardware). Questi ingressi sono attivi (ON) quando viene loro applicato un segnale a 24 V c.c. Leggendoli, essi restituiscono il proprio stato corrente. Per leggere questi ingressi, utilizzare il comando **IN**.
- Ingressi digitali 16 – 31.
Questi ingressi sono esclusivamente ingressi software. Non esistono fisicamente su TJ1-MC__, ma è possibile leggerli. Questi ingressi sono utilizzati principalmente nei programmi BASIC per eseguire alcune sequenze di controllo che richiedono ingressi non necessariamente fisici. Per leggere questi ingressi, utilizzare il comando **IN**.
- Ingressi digitali 32 – 255.
Questi ingressi sono fisicamente presenti solo se a TJ1-MC__ sono collegati moduli di I/O digitali aggiuntivi tramite il bus MECHATROLINK-II. La lettura di queste uscite, se non esistono fisicamente (cioè se i moduli di I/O non sono collegati), restituisce 0. Per leggere questi ingressi, utilizzare il comando **IN**.

Tutti gli ingressi sono caratteristica esclusiva dell'unità di controllo. Non vi si accede in base ai singoli assi.

Ingressi dei servozionamenti di MECHATROLINK-II nello spazio di I/O di Trajexia

Il comando BASIC **IN** consente di accedere agli ingressi fisicamente presenti in un programma BASIC. Questi ingressi possono trovarsi incorporati nell'unità di controllo o essere collegati attraverso il bus MECHATROLINK-II bus.

La mappatura di alcuni ingressi del servozionamento entra nello spazio di I/O di Trajexia. Pertanto è possibile accedere a questi ingressi attraverso il programma BASIC. Trajexia supporta tale accesso solo per i servozionamenti collegati al sistema Trajexia attraverso il bus MECHATROLINK-II.

La selezione degli ingressi del servozionamento con mappatura dentro allo spazio I/O di Trajexia dipende dal valore del parametro Pn81E.

Il valore consigliato di Pn81E è 0x4321.

Se questo valore viene impostato come valore predefinito, si ha la seguente mappatura per il servozionamento Sigma-II:

Bit DRIVE_MONITOR	Segnale del connettore	Ingresso Trajexia
6	CN1-44	N/D
7	CN1-45	N/D
8	CN1-46	N/D
12	CN1-40	28
13	CN1-41	29
14	CN1-42	30
15	CN1-43	31

Se questo valore viene impostato come valore predefinito, si ha la seguente mappatura per il servozionamento Junma:

Bit DRIVE_MONITOR	Segnale del connettore	Ingresso Trajexia
2	CN1-2	26
6	CN1-1	27

Quando, come in questo caso, la mappatura degli ingressi del servozionamento entra nello spazio di I/O di Trajexia, l'accesso agli ingressi è consentito solo all'interno del programma e in base all'asse; l'accesso consueto attraverso il comando **IN** non è ammesso. L'unico modo di usare questi ingressi all'interno del programma consiste nell'assegnarli ai parametri degli assi **DATUM_IN**, **FHOLD_IN**, **FWD_IN** e **REV_IN**.

Vengono utilizzati gli ingressi del servozionamento degli assi, in base agli assi su cui vengono impostati questi parametri.

Ad esempio, vi sono due servozionamenti Sigma-II assegnati agli assi 0 e 3 dell'unità di controllo. Per entrambi gli assi si desidera utilizzare il segnale d'ingresso CN1-41 come ingresso del limite di marcia indietro. Ciò è possibile utilizzando i seguenti comandi:

REV_IN AXIS(0) = 29

REV_IN AXIS(3) = 29

Notare che sebbene i parametri **REV_IN** per entrambi gli assi abbiano lo stesso valore, cioè 29, gli ingressi reali utilizzati non sono gli stessi. Per l'asse 0 viene utilizzato l'ingresso CN1-41 del primo driver (assegnato a quell'asse), mentre per l'asse 3 viene sì usato lo stesso ingresso CN1-41, ma dell'altro driver (quello assegnato all'asse 3). Ecco perché si dice che l'accesso a questi ingressi avviene in base all'asse: non si tratta di ingressi esclusivi per l'intera unità di controllo. In generale, questi due ingressi possiedono ognuno uno stato diverso contemporaneamente. Notare altresì che non è possibile accedere né all'uno né all'altro di questi due ingressi utilizzando il comando **IN**. Il comando **IN(29)**, ad esempio, restituisce lo stato dell'ingresso software 29 dell'unità di controllo (unico per tutti gli assi), che ha uno stato diverso rispetto agli ingressi del servozionamento con mappatura allo stesso numero.

Il Manuale di programmazione di Trajexia contiene maggiori informazioni sulla mappatura degli ingressi dei servozionamenti di MECHATROLINK-II nello spazio di I/O di Trajexia, sotto la descrizione dei comandi BASIC **DATUM_IN**, **FHOLD_IN**, **FWD_IN** e **REV_IN**.

6.1.5 Ricerca dell'origine

La funzionalità di ricerca dell'origine o di ricerca dello zero viene spesso considerata una particolare sequenza di movimenti di un asse nella fase di avvio della macchina. Nella maggior parte dei casi, tale sequenza si svolge automaticamente, senza richiedere l'intervento da parte dell'operatore della macchina. In generale, una procedura di ricerca dell'origine abbina una posizione a un asse specifico. La procedura dipende dagli encoder utilizzati (assoluti o relativi), dal sistema impiegato (lineare o circolare) e dalla struttura meccanica della macchina. Gli encoder assoluti non richiedono alcun movimento durante la procedura di ricerca dell'origine, poiché le posizioni esatte vengono trasferite direttamente al sistema. Per gli altri tipi di encoder, un movimento è necessario, poiché la posizione esatta all'interno del sistema risulta sconosciuta. Sostanzialmente si tratta di un movimento a bassa velocità in una certa direzione fino a quando non viene raggiunto un determinato punto di misurazione. Tale punto di misurazione può essere analizzato su due direzioni per aumentarne la precisione.

All'avvio, le posizioni correnti degli assi che utilizzano encoder incrementali hanno valore 0. Dal momento che tali posizioni non coincidono con lo 0 meccanico della macchina, è necessario eseguire la sequenza di ricerca dello zero. Se si utilizza un encoder assoluto, la posizione assoluta viene letta all'avvio sull'encoder e la ricerca dello zero non è necessaria. In questo caso, è necessario eseguire una sequenza di avvio una sola volta durante la messa a punto della macchina.

All'atto pratico, esistono diverse sequenze di ricerca dell'origine.

Si distinguono sotto i seguenti aspetti:

- I mezzi utilizzati per individuare le posizioni limite della parte mobile (sensori, interruttori, ecc.)
- La posizione o riferimento di origine (punto zero).
- Le possibili posizioni della parte mobile in rapporto alle posizioni limite e alla posizione di origine.

Trajexia include alcune sequenze fondamentali di ricerca dello zero predefinite:

- **DATUM(0)**
Strettamente parlando, questa non è una ricerca dell'origine. Questo comando imposta **DPOS=MPOS** e cancella gli errori degli assi.

- **DATUM(1)**
Questo comando esegue una ricerca dell'origine muovendosi in avanti e utilizzando la fase "Z" di un encoder come interruttore per la ricerca dello zero.
- **DATUM(2)**
Esegue una ricerca dell'origine muovendosi indietro e utilizzando la fase "Z" di un encoder come interruttore per la ricerca dello zero.
- **DATUM(3)**
Esegue una ricerca dell'origine muovendosi in avanti e utilizzando l'ingresso selezionato in **DATUM_IN** come interruttore per la ricerca dello zero.
- **DATUM(4)**
Esegue una ricerca dell'origine muovendosi indietro e utilizzando l'ingresso selezionato in **DATUM_IN** come interruttore per la ricerca dello zero.
- **DATUM(5)**
Esegue una ricerca dell'origine muovendosi in avanti e utilizzando l'ingresso selezionato in **DATUM_IN** come interruttore per la ricerca dello zero, dopodiché cerca la fase "Z" successivo di un encoder.
- **DATUM(6)**
Esegue una ricerca dell'origine muovendosi indietro e utilizzando l'ingresso selezionato in **DATUM_IN** come interruttore per la ricerca dello zero, dopodiché cerca la fase "Z" successivo di un encoder.

Per maggiori dettagli su queste sequenze di ricerca dello zero predefinite, vedere la sezione 3.2.68.

In alcune situazioni sono richieste sequenze di ricerca dello zero più complesse:

- Ricerca dell'origine con interruttore assoluto e finecorsa.
- Ricerca dell'origine rispetto ai finecorsa.
- Ricerca dell'origine rispetto a componenti hardware che arrestano il movimento.
- Ricerca dell'origine tramite l'impulso di riferimento dell'encoder "fase Zero".
- Ricerca dell'origine statica, tramite imposizione di una posizione ottenuta da un riferimento utente.
- Ricerca dell'origine statica, tramite imposizione di una posizione derivata da un encoder assoluto.

La figura mostra lo scenario di una ricerca dell'origine di carattere generale. Questa semplice sequenza di ricerca dell'origine si sviluppa in 3 fasi:

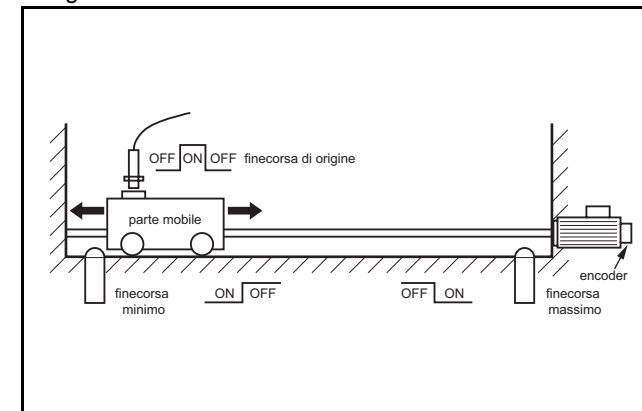
1. Ricerca di un segnale.
2. Ricerca di un altro segnale.
3. Spostamento dell'asse fino a una posizione predefinita.



Per motivi di sicurezza, i finecorsa sono solitamente chiusi. Per questo motivo, nella figura a lato e nelle altre figure proposte in questa sezione, il livello di segnale basso è indicato da ON mentre il livello di segnale alto è indicato da OFF.

È importante notare che prima di eseguire una qualsiasi operazione di ricerca dell'origine è necessario impostare i parametri degli assi **UNITS**, **REP_DIST** e **REP_OPTION**, nonché i parametri del servozionamento Pn202, Pn203 e Pn205, correttamente e conformemente al sistema meccanico e alle unità di misura desiderate utilizzate nella programmazione. Questi parametri influenzano la ricerca dell'origine, specie se viene utilizzato un encoder assoluto. Per maggiori informazioni sull'impostazione di questi parametri, vedere la sezione 6.1.2.

fig. 25



Ricerca dell'origine con interruttore assoluto e finecorsa

La funzione di ricerca dell'origine si svolge attraverso la ricerca di un finecorsa esterno la cui posizione è assoluta e tale da definire la posizione di origine. L'esempio di questa procedura di ricerca dello zero è mostrata in figura.

La figura mostra gli scenari possibili per una ricerca dell'origine con interruttore assoluto e finecorsa. Questi scenari dipendono dalla posizione in cui si trova la parte mobile al momento dell'accensione.

Un esempio di programma che esegue una simile sequenza di ricerca dell'origine viene fornita di seguito.

```
'Selettore d'origine assoluto: IN0
'Finecorsa sinistro: IN1
'Finecorsa destro: IN2
BASE(0)
DATUM_IN=0
FW_IN=2
RV_IN=1
SERVO=ON
WDOG=ON
DATUM(4)
WA(1)
WAIT UNTIL MTYPE=0 OR IN(1)=OFF
IF IN(1)=ON
    FORWARD
    WAIT UNTIL IN(0)=ON
    WAIT UNTIL IN(0)=OFF
    CANCEL
```

fig. 26

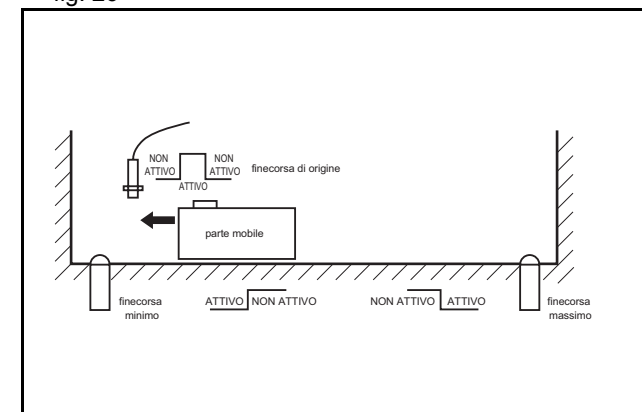
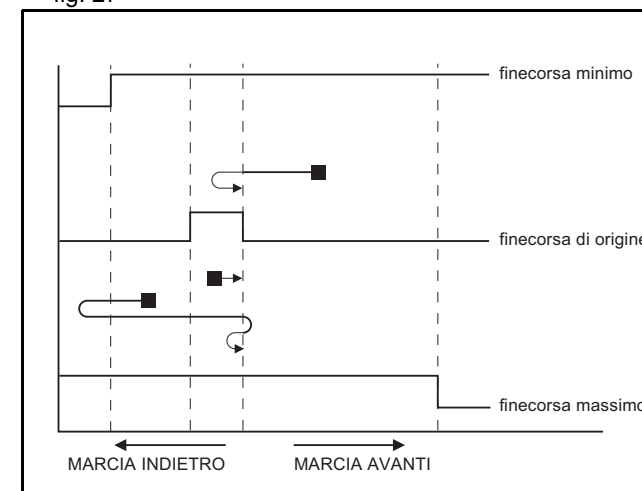


fig. 27



```
DATUM (4)
WA (1)
WAIT IDLE
ENDIF
```

Ricerca dell'origine rispetto ai finecorsa

Questa funzione di ricerca dell'origine viene svolta cercando un sensore esterno che utilizza solo i finecorsa. L'esempio di questa procedura di ricerca dello zero è mostrata in figura.

Gli scenari possibili per una ricerca dell'origine rispetto ai finecorsa, dipende dalla posizione della parte mobile all'accensione, come mostrato in figura.

Un esempio di programma che esegue una simile sequenza di ricerca dell'origine viene fornita di seguito.

```
'Selettore d'origine e finecorsa sinistro: IN0
'Finecorsa destro: IN1
BASE (0)
DATUM_IN=0
SERVO=ON
WDOG=ON
DATUM (4)
WA (1)
WAIT IDLE
```

fig. 28

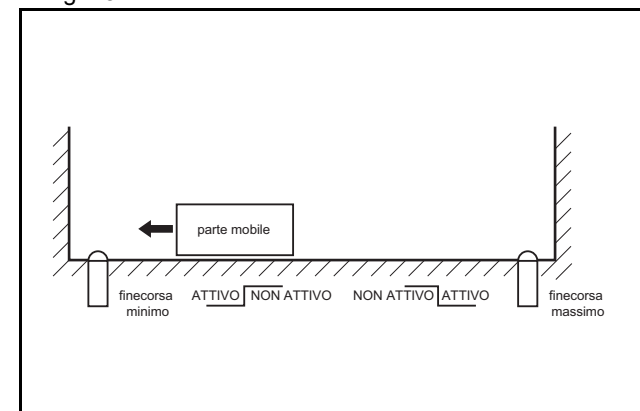
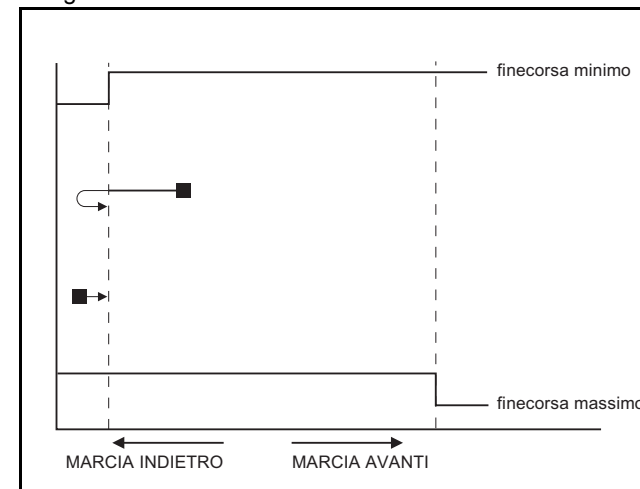


fig. 29



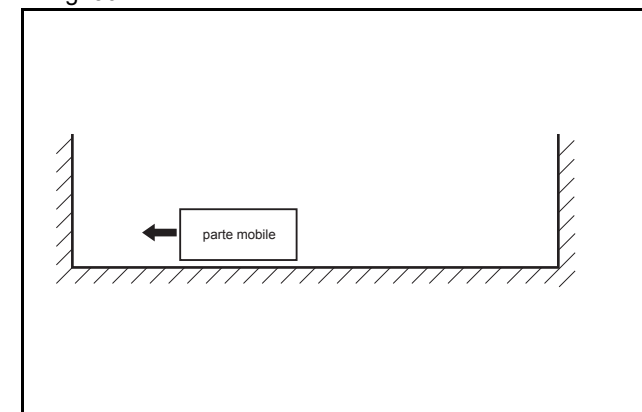
Ricerca dell'origine rispetto a componenti hardware che arrestano il movimento

Questa procedura di ricerca dell'origine esegue la ricerca rispetto a un oggetto fisico che blocca meccanicamente il movimento. Non vi sono finecorsa, selettori di posizione assoluti o impulsi di riferimento. La posizione dell'origine viene identificata individuando un valore di coppia particolare rispetto agli elementi bloccanti. È necessario un limite di coppia adeguato che non danneggi i meccanismi durante il processo di ricerca dell'origine. L'esempio di questa procedura di ricerca dello zero è mostrata in figura.

Un esempio di programma che esegue una simile sequenza di ricerca dell'origine viene fornita di seguito.

```
BASE(0)
DRIVE_CONTROL=11 Tiene la coppia sotto controllo con
DRIVE_MONITOR
SERVO=ON
WDOG=ON
SPEED=CREEP
REVERSE
WA(1)
WAIT UNTIL DRIVE_MONITOR < -100
    'Aspetta fino a quando non viene applicata
    un particolare valore di coppia
CANCEL
DEFPOS(0)
MOVEABS(10) 'Questo è necessario, diversamente la
posizione
    'viene mantenuta, con relativa pressione sui
    limiti meccanici della
    'macchina e conseguente scatto del motore
    per sovraccarico
```

fig. 30



Ricerca dell'origine tramite l'impulso di riferimento dell'encoder “fase Zero”

Questa procedura di ricerca dell'origine esegue la ricerca seguendo il segnale “fase Zero” dell'encoder. Tale segnale è noto altresì come “fase” o “impulso di riferimento”. Compare una volta per ogni rivoluzione completa dell'encoder. L'esempio di questa procedura di ricerca dello zero è mostrata in figura.

Gli scenari possibili per una ricerca effettuata utilizzando l'impulso di riferimento “fase Zero”, dipende dalla posizione della parte mobile all'accensione, come mostrato in figura.

Un esempio di programma che esegue una simile sequenza di ricerca dell'origine viene fornita di seguito.

```
'Selettore d'origine e finecorsa sinistro: IN0
'Finecorsa destro: IN1
REV_IN=-1
BASE(0)
DATUM_IN=0
SERVO=ON
WDOG=ON
DATUM(6)
WA(1)
WAIT IDLE
```

Ricerca dell'origine statica, tramite imposizione di una posizione ottenuta da un riferimento utente

Questa procedura di ricerca dell'origine esegue una ricerca dell'origine statica, imponendo direttamente una posizione effettiva. Non esegue alcun movimento fisico.

```
DATUM(0)
```

fig. 31

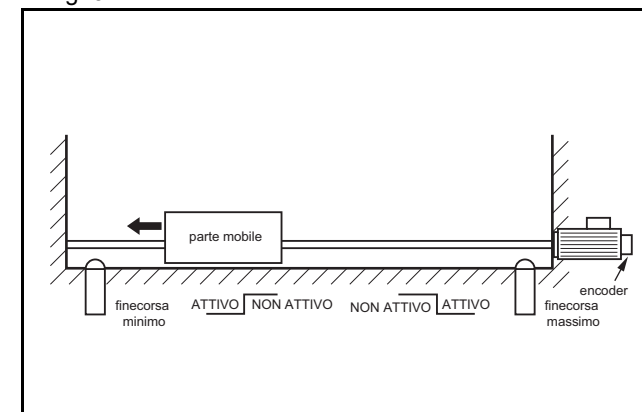
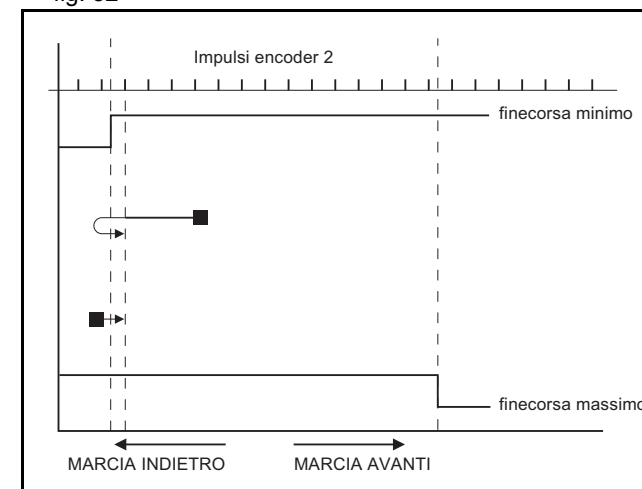


fig. 32



Ricerca dell'origine statica, tramite imposizione di una posizione derivata da un encoder assoluto

Questa procedura di ricerca dell'origine imposta la posizione effettiva in base alla posizione di un encoder assoluto.

Non esegue alcun movimento fisico. È possibile solo su di un asse con un encoder assoluto in un anello di controllo.

6.1.6 Registrazione

La registrazione, detta anche “ritenzione” o “registrazione dell'evento”, è la memorizzazione in tempo reale della posizione di un asse quando viene attivato un ingresso esterno.

L'informazione registrata, vale a dire memorizzata, viene elaborata più tardi, non in tempo reale, dal programma applicativo.

La registrazione differisce dall'elaborazione di un ingresso o segnale interrupt. Con la registrazione non viene generato nessun evento quando l'ingresso di registrazione viene attivato. Inoltre, la normale esecuzione del programma applicativo non viene disturbata né interrotta. Viene semplicemente memorizzata la posizione di un asse. Tale informazione può essere utilizzata, come gli altri parametri o valori, all'interno di un programma.

Il programma può disporre delle informazioni registrate subito dopo la registrazione.

Il vantaggio della registrazione sta nella rapidità con la quale viene eseguita. Pertanto, la posizione dell'asse che viene memorizzata è molto precisa. Per ottenere una simile velocità e precisione, la registrazione viene effettuata attraverso l'hardware e l'ingresso della registrazione deve trovarsi sulla stessa scheda dove è presente l'ingresso dell'encoder, che fornisce le informazioni sulla posizione degli assi.

L'acquisizione e memorizzazione della posizione dell'asse sono eseguite in tempo reale dall'hardware. L'elaborazione delle informazioni non viene eseguita in tempo reale dal programma applicativo.

Il comando degli assi REGIST

In Trajexia, per eseguire una registrazione si usa il comando degli assi **REGIST**. Questo comando è accompagnato da un argomento. L'argomento determina quale ingresso esterno è registrato, indipendentemente dal fatto che la registrazione sia eseguita sul fronte di salita o sul fronte di discesa e a prescindere dall'uso della funzione di windowing o di altre opzioni. Per maggiori informazioni sul comando **REGIST**, fare riferimento alla sezione 3.2.219.

La registrazione varia da un asse all'altro, in base al collegamento degli assi stessi al sistema. Se un asse è collegato attraverso il bus MECHATROLINK-II, la registrazione viene effettuata nell'hardware del servozionamento. Se un asse è collegato attraverso l'interfaccia analogica del servozionamento e di TJ1-FL02, la registrazione viene effettuata nell'hardware di TJ1-FL02. I diversi tipi di registrazione sono descritti di seguito.

Registrazione nel servozionamento Sigma-II

La registrazione nel servozionamento Sigma-II si verifica quando un asse assegnato al servozionamento Sigma-II è collegato al sistema Trajexia attraverso il bus MECHATROLINK-II.

Il servozionamento Sigma-II è dotato di tre ingressi di registrazione, ma un solo blocco hardware; pertanto è possibile usare un solo ingresso per volta. Gli ingressi fisici si trovano ai pin CN1-44, CN1-45 e CN1-46 del connettore CN1 a 50 pin, ma Trajexia utilizza gli ingressi logici EXT1, EXT2 ed EXT3 per associare gli ingressi fisici a quelli logici. Tale associazione viene eseguita tramite impostazione del parametro Pn511 del servozionamento. Per maggiori informazioni sull'impostazione di questa associazione e del parametro Pn511, fare riferimento alla sezione 3.2.219, tabella 1. L'ingresso utilizzato per la registrazione è determinato dall'argomento del comando **REGIST**.

Per l'acquisizione, il servozionamento Sigma-II ci impiega circa $3\text{ }\mu\text{s}$. Dal momento che le informazioni dell'encoder vengono aggiornate ogni $62,5\text{ }\mu\text{s}$, è necessario eseguire un'interpolazione per ottenere il valore corretto della posizione acquisita (vedere la figura). Considerato che la velocità del motore non può cambiare molto in $62,5\text{ }\mu\text{s}$, la precisione risultante è molto elevata.

I tempi di trasmissione delle informazioni sono:

- Tempo d'avvio della registrazione: da $0,625\text{ ms}$ a 4 ms
- Tempo di ricezione della registrazione: $3,5\text{ ms}$:
- Tempo di acquisizione della registrazione: $3\text{ }\mu\text{s}$.

È inoltre possibile utilizzare la fase "Z" dell'encoder per registrare la posizione di un asse. Questa azione è eseguibile altresì mediante il comando **REGIST**.

Registrazione nel servozionamento Junma

Nel servozionamento Junma la registrazione si svolge come nel servozionamento Sigma-II, con una sola differenza: Vi sono un solo ingresso fisico e un solo blocco logico, pertanto non è necessaria alcuna impostazione del servozionamento. L'ingresso fisico è associato al blocco logico EXT1 e solo il fronte di salita può essere utilizzato per la registrazione.

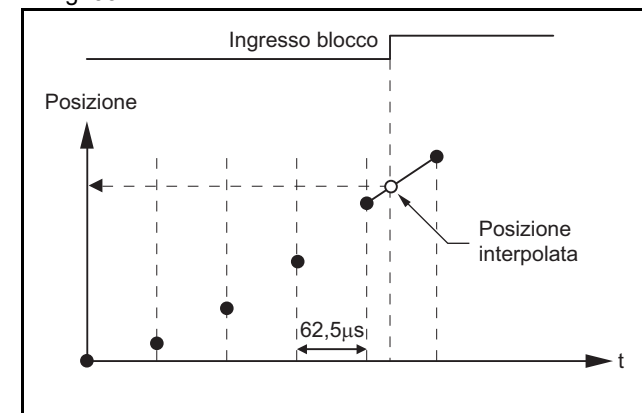
La registrazione su TJ1-FL02

TJ1-FL02 è dotato di due ingressi fisici di registrazione e due blocchi per ogni ingresso encoder, che possono essere utilizzati indipendentemente. È quindi possibile utilizzare due ingressi di registrazione indipendenti allo stesso tempo. Per maggiori informazioni su come utilizzare entrambi gli ingressi di registrazione di TJ1-FL02 contemporaneamente, fare riferimento alle sezioni 3.2.170, 3.2.171, 3.2.217, 3.2.218 e 3.2.219.

Il tempo di acquisizione è $0,5\text{ }\mu\text{s}$. Poiché la posizione dell'encoder viene letta costantemente sull'ingresso encoder line driver, non è richiesta alcuna interpolazione.

Il tempo di trasmissione delle informazioni acquisite è pari a solo un ciclo **SERVO_PERIOD**.

fig. 33



Uso della registrazione in programmi applicativi

Esistono un comando dell'asse (**REGIST**) e due parametri dell'asse (**MARK** e **REG_POS**). Questi comandi e parametri consentono di controllare e utilizzare la funzionalità di registrazione nei programmi BASIC.

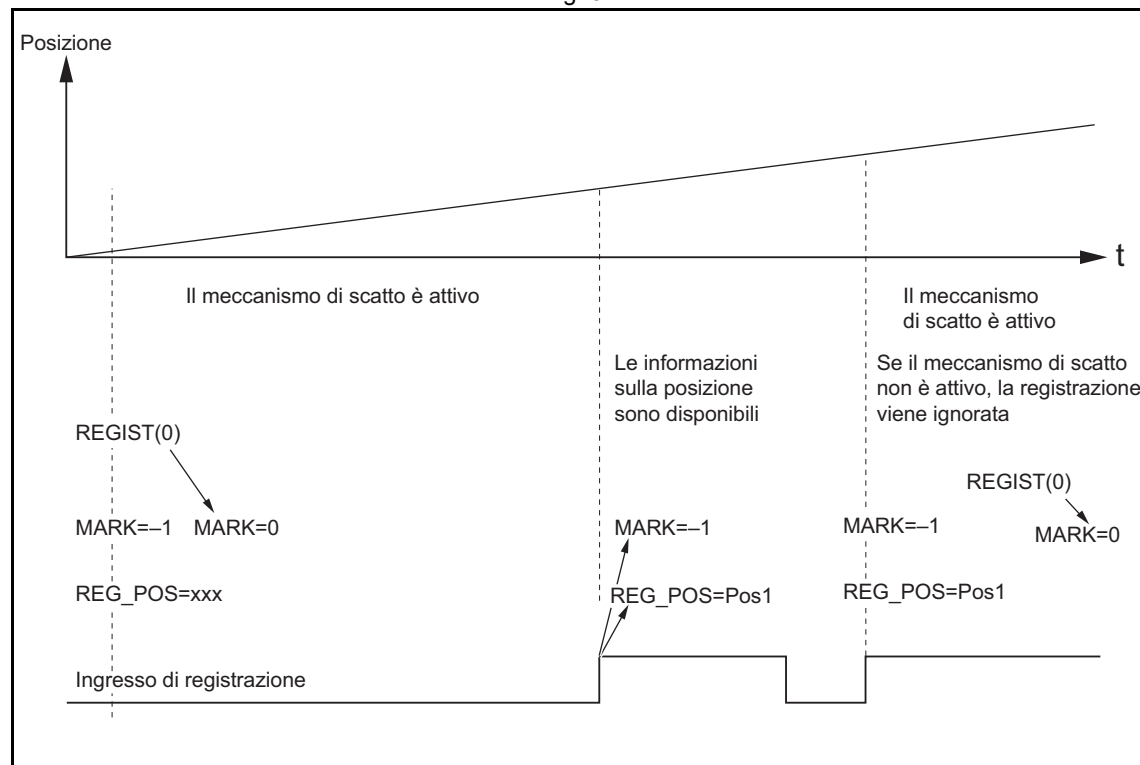
- **REGIST** acquisisce la posizione di un asse quando viene individuato un segnale di registrazione. Le impostazioni variabili dipendono dal tipo di asse. Fare riferimento alla sezione 3.2.219.
- **MARK** è un flag che indica se la posizione è stata acquisita oppure no. Per il secondo ingresso di registrazione di TJ1-FL02, è disponibile anche il parametro **MARKB**. Per maggiori informazioni, fare riferimento alle sezioni 3.2.170 e 3.2.171.
- **REG_POS** contiene la posizione dell'asse acquisita. Solo se il flag **MARK** indica che la posizione è stata acquisita con successo, è possibile considerare affidabile il valore **REG_POS**. Per il secondo ingresso di registrazione di TJ1-FL02, è disponibile anche il parametro **REG_POSB**. Per maggiori informazioni, fare riferimento alle sezioni 3.2.217 e 3.2.218.

La figura mostra la sequenza d'esecuzione dei comandi e delle registrazioni del programma di esempio fornito di seguito.

```

BASE (N)
REGIST(0)
WAIT UNTIL MARK=0
loop:
  WAIT UNTIL MARK=-1
  PRINT "Posizione acquisita in: "; REG_POS
  REGIST(0)
  WAIT UNTIL MARK=0
GOTO loop
  
```

fig. 34



Funzione di registrazione e windowing

La funzione di windowing limita la registrazione a un intervallo specifico di posizioni dell'asse. La funzione viene selezionata assegnandole il valore corretto come argomento del comando **REGIST**. Le funzioni di windowing sono controllate da due parametri degli assi, **OPEN_WIN** e **CLOSE_WIN**. Per maggiori informazioni su **REGIST**, **OPEN_WIN** e **CLOSE_WIN**, fare riferimento alle sezioni 3.2.49, 3.2.198 e 3.2.219

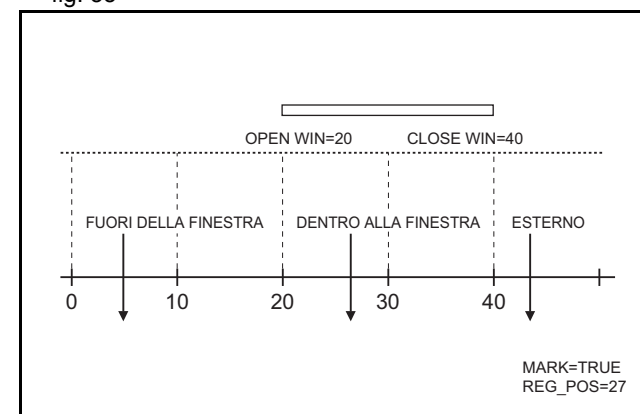
Esistono due tipi di windowing:

- Il windowing inclusivo che consente la registrazione solo all'interno della finestra specificata di posizioni degli assi. Con questa funzione di windowing, gli eventi di registrazione sono ignorati se la posizione misurata dell'asse è minore del parametro degli assi **OPEN_WIN** o maggiore del parametro **CLOSE_WIN**.
- Il windowing esclusivo che consente la registrazione solo al di fuori di una finestra specifica di posizioni degli assi. Con questa funzione di windowing, gli eventi di registrazione sono ignorati se la posizione misurata dell'asse è maggiore al parametro degli assi **OPEN_WIN** o minore del parametro **CLOSE_WIN**.

Quando viene utilizzata la funzione di windowing, il processo interno si svolge come segue:

1. **REGIST** + fa eseguire il windowing nel programma.
2. **MARK** = 0 fa scattare il blocco.
3. La posizione viene acquisita e trasmessa al processore Trajexia.
4. La posizione acquisita si trova dentro alla finestra inclusiva o fuori della finestra esclusiva?
 - In caso di risposta affermativa, **MARK** = -1 e **REG_POS** viene aggiornato.
 - In caso di risposta negativa, si torna al punto 2 (il blocco scatta di nuovo sotto gli occhi dell'utente).

fig. 35



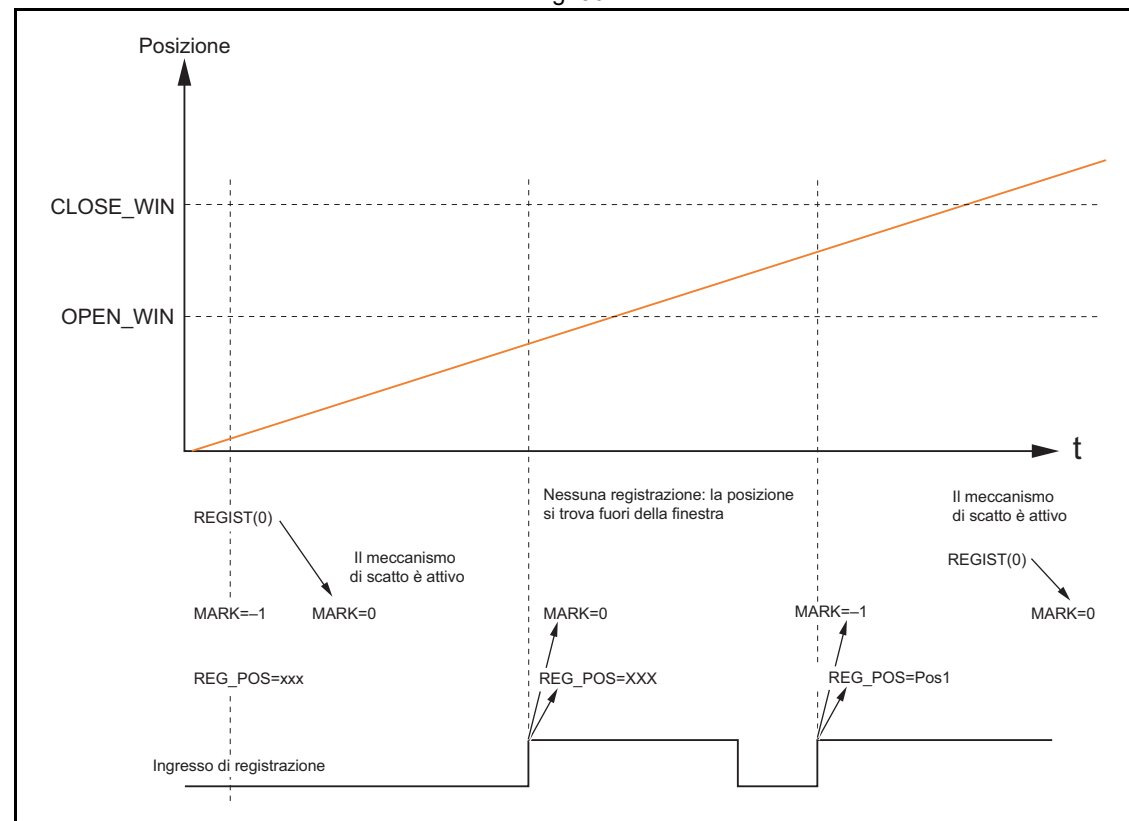
L'illustrazione mostra la sequenza di esecuzione dei comandi sopraelencati e quando si verificano gli eventi di registrazione, qualora si utilizzi il windowing inclusivo.

Tra questi eventi, intercorrono i seguenti tempi:

- Trajexia riceve il blocco.
- Trajexia decide di far scattare nuovamente il blocco.
- Il blocco scatta.

L'intercorrere di questi tempi produce indeterminatezza nei margini della finestra nel caso in cui le fasi possano essere individuati nelle vicinanze dei fronti. Il fenomeno è più marcato per gli assi collegati al sistema tramite il bus MECHATROLINK-II per via dei ritardi provocati dal bus. Per compensare questi ritardi, l'utente deve impostare la larghezza dei margini su valori sufficientemente ampi.

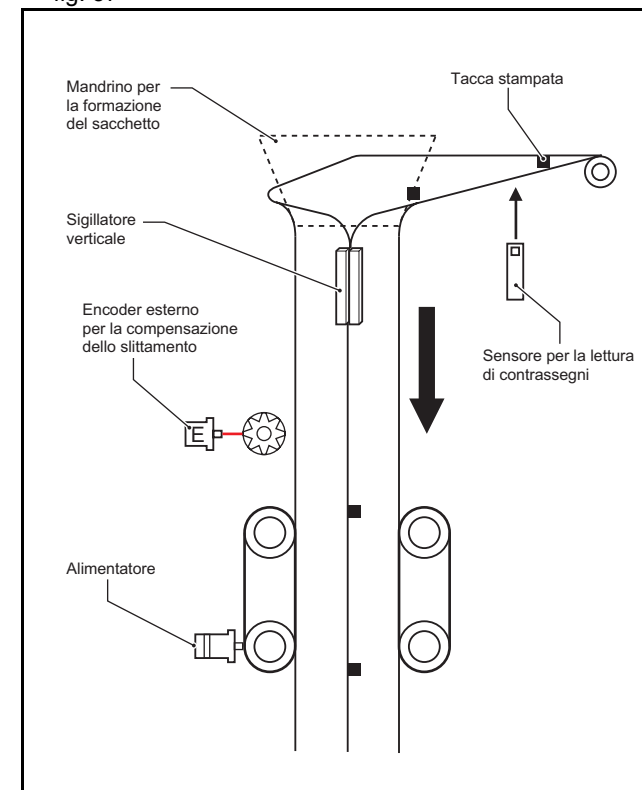
fig. 36



Esempio: correzione della posizione di un asse

L'illustrazione mostra una macchina per l'imballaggio e la sigillatura di prodotti in sacchetti. Il materiale che costituisce il sacchetto proviene da un rullo di pellicola di plastica che viene dapprima svolto, quindi modellato a forma di tubo da un mandrino meccanico e, contemporaneamente, sigillato verticalmente. Il movimento dell'alimentatore è intermittente e la lunghezza di avanzamento coincide con la lunghezza del sacchetto. Una volta che il sacchetto ha raggiunto la lunghezza prevista, il sigillatore lo chiude per consentirne il riempimento con il prodotto. Dopodiché, viene fatto avanzare il tratto di pellicola per il sacchetto successivo e il processo ha nuovamente inizio.

fig. 37



L'alimentatore può funzionare in due modalità: senza contrassegno di registrazione e con contrassegno di registrazione. Il funzionamento senza tacche di registrazione consiste in un semplice movimento incrementale da un punto all'altro. In questo caso, non è garantito che l'alimentatore copra l'esatta distanza prevista dallo schema progettuale. Se ad esempio la lunghezza prevista per ogni sacchetto è di 200 mm, ma la lunghezza reale è di 200,1 mm, il semplice movimento incrementale da punto a punto senza correzioni comporta l'accumulo di 0,1 mm per ogni sacchetto. Se il numero di sacchetti non è elevato, la differenza non risulta visibile, ma dopo 500 sacchetti l'errore raggiunge i 50 mm, vale a dire il 25% della lunghezza del sacchetto.

Lavorando con le tacche di registrazione, l'unità di controllo assi esegue un movimento incrementale fino a raggiungere una certa posizione. Se durante il processo di posizionamento viene rilevato la tacca di registrazione, la posizione di riferimento viene modificata in fase di funzionamento di modo che il movimento abbia termine in un punto ben preciso dopo la tacca di registrazione. In tal modo, viene sempre garantito il mantenimento della medesima distanza in relazione al contrassegno di registrazione.

fig. 38

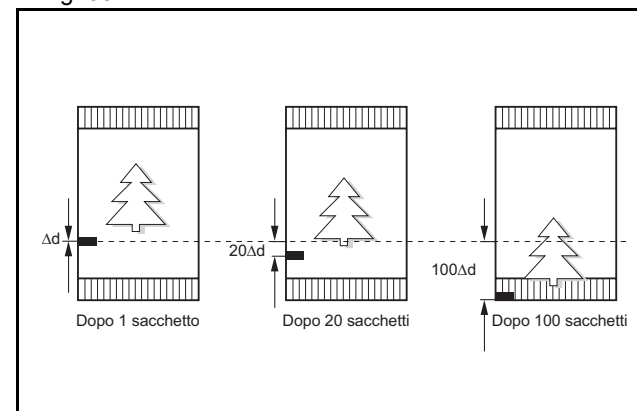
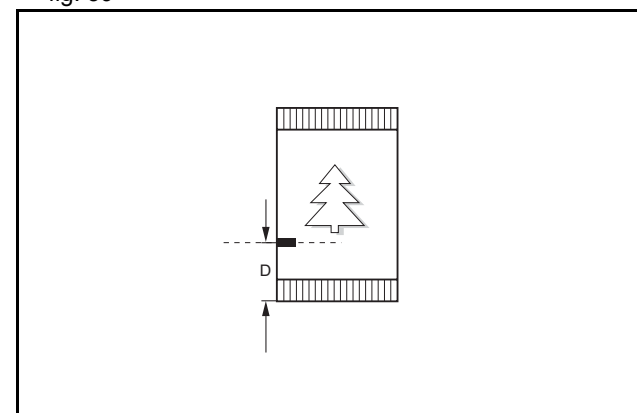


fig. 39



Il profilo del movimento e la relativa modifica generata dalla tacca di registrazione sono mostrati alla fig. 39.

Il programma BASIC per questo esempio è il seguente:

```
DEFPOS(0)
REGIST(3) 'Fa scattare la registrazione della tacca
MOVE(bag_length) 'Effettua lo spostamento fino alla
distanza teorica
WA(1)
WAIT UNTIL MARK OR MTYPE=0
IF MARK THEN
    end_position=REG_POS+distance_after_mark
    MOVEMODIFY(end_position)
    'Corregge la distanza in base alla tacca
ENDIF
```

Esempio: avvio di un asse slave da un punto preciso di un asse principale

La figura mostra una lama semovente che taglia l'estremità superiore di alcune assi di legno. Non appena la fotocellula individua il bordo dell'asse in arrivo, inizia la sincronizzazione del movimento della lama semovente con la posizione corretta sull'asse.

Se il movimento viene avviato dal programma in seguito all'individuazione di un segnale da parte della fotocellula, interviene sempre almeno un **SERVO_PERIOD** di indeterminatezza.

Al contrario, il movimento viene avviato utilizzando il comando **MOVELINK** con **opzione_collegamento=1**, il che vuol dire che il collegamento all'asse principale inizia quando l'evento di registrazione si verifica sull'asse di collegamento (principale).

La sequenza di programma corrispondente è:

REGIST(2) AXIS(master)

MOVELINK(dst,lnk_dst,lnk_acc,lnk_dec,master,1) AXIS(slave)

Per maggiori informazioni sul comando **MOVELINK** e l'argomento **opzione_collegamento**, fare riferimento alla sezione 3.2.180.

fig. 40

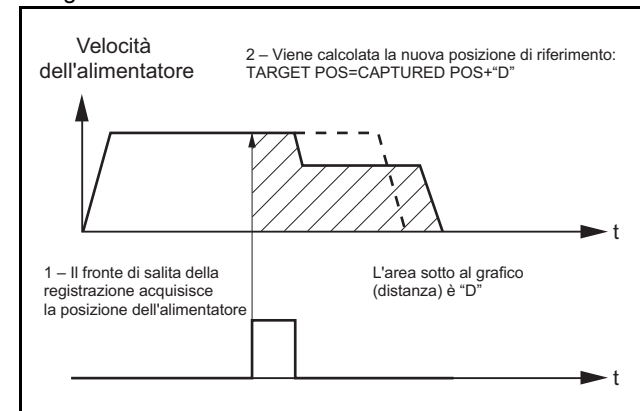
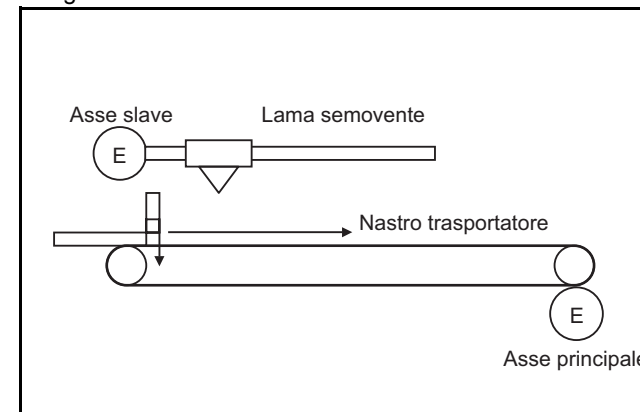
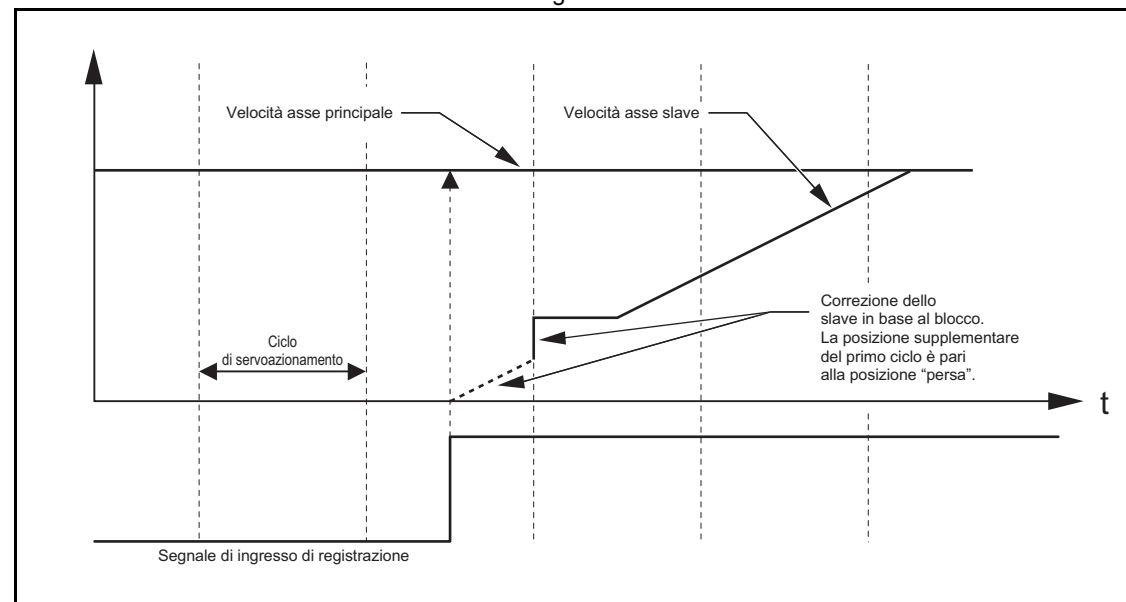


fig. 41



L'immagine mostra come la posizione dell'asse slave viene corretta, utilizzando l'evento di registrazione che si verifica sull'asse principale per avviare il movimento dell'asse slave. Ciò elimina completamente l'influenza del **SERVO_PERIOD** e il fatto che l'evento di registrazione possa verificarsi in qualsiasi momento del **SERVO_PERIOD**.

fig. 42



6.1.7 Registrazione e monitoraggio

Funzione oscilloscopio in Trajexia Tools

L'oscilloscopio software è un elemento standard di Trajexia Tools. L'oscilloscopio può essere utilizzato per registrare e rappresentare graficamente i parametri di sistema e degli assi. Questo può essere d'aiuto nello sviluppo del sistema, nella sua messa a punto e nella risoluzione dei problemi. Per maggiori informazioni sull'oscilloscopio software e sulle relative funzionalità e capacità, fare riferimento alla sezione 5.5.4.

La registrazione di un determinato asse e dei parametri di sistema da parte dell'oscilloscopio può essere avviata in due modi: manualmente o tramite programma. L'avvio manuale viene effettuato utilizzando lo strumento oscilloscopio. I parametri vengono registrati nella memoria TABLE dell'unità di controllo. L'intervallo della memoria TABLE dove vengono registrati i parametri può essere impostato attraverso la finestra

Oscilloscope Configuration (vedere sezione 5.5.4).

L'avvio manuale consente all'utente di visualizzare in tempo reale i cambiamenti dei parametri degli assi e di sistema, mentre il sistema è in funzione. I cambiamenti dei valori di parametro vengono rappresentati graficamente non appena si verificano. Il limite dell'avvio manuale è rappresentato dal fatto che richiede l'interazione dell'utente; ciò significa che l'avvio della registrazione non è sincronizzato con il movimento che viene analizzato. Inoltre, con l'avvio manuale la registrazione è limitata a 200 campionamenti per canale.

Utilizzo dell'oscilloscopio

L'avvio della registrazione attraverso un programma costituisce un'alternativa non sottoposta alle limitazioni dell'avvio manuale. L'avvio tramite programma memorizza i parametri degli assi e di sistema nella memoria di TJ1-MC__. In seguito, i parametri vengono trasmessi all'oscilloscopio per una rappresentazione grafica. I parametri degli assi e di sistema vengono registrati nella memoria TABLE. L'intervallo di memoria utilizzato è definito dai parametri del comando **SCOPE**. Quando i parametri si trovano nella memoria TABLE, è possibile configurare l'oscilloscopio per visualizzare un intervallo della memoria TABLE invece dei parametri degli assi e di sistema. Il momento esatto in cui ha inizio la registrazione può essere stabilito con precisione poiché è controllato dal comando **TRIGGER**. Ciò significa che l'inizio della registrazione è sincronizzato con il movimento. Non vi è alcun limite di 200 campionamenti per canale e l'oscilloscopio mostra tutti i campionamenti (dati della memoria TABLE) configurati attraverso la finestra **Oscilloscope Configuration**.

Esempio

Questa sezione fornisce un esempio pratico dell'uso dei comandi **SCOPE** e **TRIGGER** unitamente all'oscilloscopio per il controllo dei parametri degli assi e per la risoluzione degli errori del sistema. Per maggiori informazioni sui comandi **SCOPE** e **TRIGGER**, fare riferimento alle sezioni 3.2.237 e 3.2.265

Supponiamo che il sistema consista di due assi, **AXIS(0)** e **AXIS(1)**. **AXIS(0)** è l'asse principale. Esegue un semplice movimento in avanti. **AXIS(1)** è l'asse slave. Deve seguire l'asse principale in base alla regola del coseno:

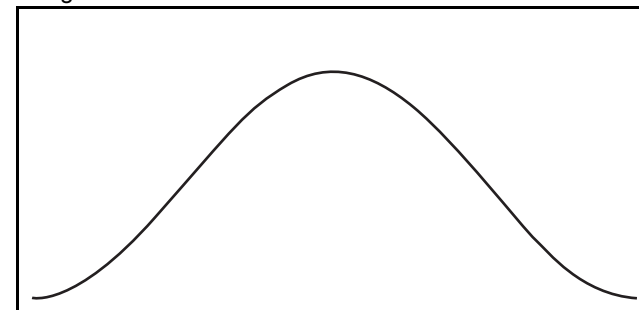
$$x_1 = end\_pos \cdot \frac{1}{2} \left(1 - \cos \left(\frac{2\pi \cdot x_0}{999} \right) \right)$$

dove x_0 è la posizione dell'asse principale **AXIS(0)** e x_1 è la posizione dello slave **AXIS(1)**. È possibile collegare i due assi con il comando **CAMBOX**. Per maggiori dettagli, fare riferimento alla sezione 3.2.42. Supponiamo inoltre che il parametro **end_pos** non sia costante, ma possa cambiare in base alle diverse condizioni del sistema. La parte del programma che crea la tabella CAM è:

```
'Valori CAM iniziali
VR(end_pos)=15
current_end_pos=VR(end_pos)
FOR i=0 TO 999
    TABLE(i, VR(end_pos)*(1-COS(2*PI*i/999))/2)
NEXT i
...
loop:
IF VR(end_pos)<>current_end_pos THEN
'Ricalcola la tabella CAM
    FOR i=0 TO 999
        TABLE(i, VR(end_pos)*(1-COS(2*PI*i/999))/2)
    NEXT i
    current_end_pos=VR(end_pos)
ENDIF
...
GOTO loop
```

Il valore **VR(end_pos)** può essere cambiato attraverso un altro programma o, esternamente, attraverso un'altra unità di controllo che utilizza il sistema di messaggistica FINS. In tal caso occorre ricalcolare la tabella **CAM**.

fig. 43



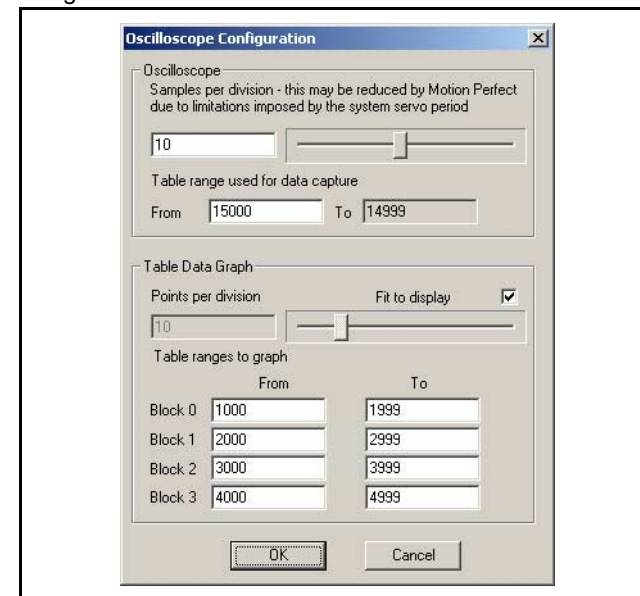
La creazione della tabella CAM è completa. L'inizializzazione dei parametri degli assi e di sistema desiderati per la registrazione è:

```
'Inizializzazioni
FOR i=0 TO 1
  BASE(i)
  ATYPE=40
  UNITS=8192
  REP_DIST=20
  REP_OPTION=1
  FE_LIMIT=1
  DRIVE_CONTROL=11
  SPEED=8
  ACCEL=50
  DECEL=50
  DEFPOS(0)
  SERVO=ON
  CANCEL
NEXT i
WDOG=ON
BASE(1)
'Impostazioni dell'oscilloscopio:
'1 campionamento ogni 2 cicli di servoazionamento
'Informazioni memorizzate da TABLE(1000) a TABLE(4999)
'I canali di acquisizione sono 4; pertanto, vi sono 1000 campionamenti per canale.
'MPOS AXIS(0) ċ memorizzato da TABLE(1000) a TABLE(1999)
'DPOS AXIS(1) ċ memorizzato da TABLE(2000) a TABLE(2999)
'La coppia di riferimento per AXIS(1) ċ memorizzata da
'TABLE(3000) a TABLE(3999)
'MSPEED AXIS(1) ċ memorizzato da TABLE(4000) a TABLE(4999)
'L'acquisizione interessa 1000 campionamenti * 2ms / campionamento = 2 secondi
SCOPE(ON,2,1000,4999,MPOS AXIS(0),DPOS,DRIVE_MONITOR,MSPEED)
FORWARD AXIS(0) 'Sposta in avanti l'asse principale
TRIGGER 'Inizia la registrazione e la memorizzazione dei parametri
WHILE NOT MOTION_ERROR
  'CAMBOX che inizia alla posizione 1 di AXIS(0)
  CAMBOX(0,999,UNITS,10,0,2,1)
  WAIT UNTIL MPOS AXIS(0)<1
  'L'acquisizione ha inizio quando l'asse principale si trova in
  'una posizione compresa tra 0 e 1. Ulteriori condizioni
```

```
'sono:
'- L'acquisizione precedente   terminata
'(SCOPE_POS=1000)
'- Si dispone dei permessi
(VR(activate_trigger)=ON)
IF SCOPE_POS=1000 AND VR(activate_trigger)=ON THEN
    TRIGGER
    PRINT "Attivato"
ENDIF
WAIT IDLE
WEND
HALT
```

Per visualizzare il risultato dell'acquisizione nell'oscilloscopio di Trajexia Tools,   necessario che la finestra **Oscilloscope Configuration** sia impostata come mostrato in figura.   inoltre necessario disabilitare ogni ulteriore acquisizione per evitare la mescolanza di due acquisizioni differenti nelle stesse voci della memoria TABLE.

fig. 44



Il risultato dell'acquisizione è mostrato in figura.

Nell'esempio fornito qui sopra, il valore del parametro **UNITS** è impostato in conteggi encoder. La posizione dell'asse principale **MPOS AXIS(0)** è fornita in rosso (Blocco tabella 0, da Table(1000) a Table(1999), vedere le impostazioni della finestra **Oscilloscope Configuration**). La posizione aumenta in maniera lineare poiché la velocità dell'asse principale è costante.

La posizione richiesta dell'asse slave **DPOS AXIS(1)** è fornita in blu (Blocco tabella 1, da Table(2000) a Table(2999), vedere le impostazioni della finestra **Oscilloscope Configuration**).

La curva di questo grafico è quella di un coseno. Corrisponde alla tabella CAM che è stata creata.

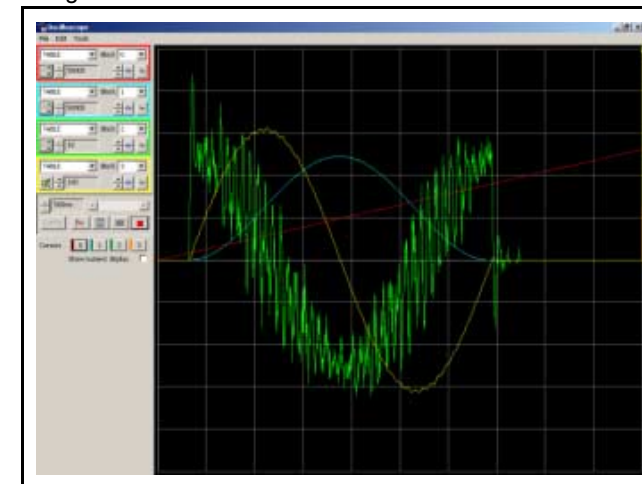
La posizione misurata dell'asse slave **MSPEED AXIS(1)** è fornita in giallo (Blocco tabella 3, da Table(4000) a Table(4999), vedere le impostazioni della finestra **Oscilloscope Configuration**).

Questo grafico consiste in una curva sinusoidale perché la velocità è una derivata della posizione e la derivata del coseno è il seno. Alle alte velocità si riscontrano alcune ondulazioni.

Il grafico verde corrisponde alla coppia del motore per l'asse slave impostato con **DRIVE_COMMAND=11** quale percentuale della coppia nominale. La coppia è proporzionale all'accelerazione.

Dal momento che l'accelerazione è una derivata della velocità e la velocità è una curva sinusoidale, la curva dell'accelerazione (e così pure della coppia) è un seno. Vi è un valore massimo all'avvio e un altro valore massimo all'arresto, poiché l'accelerazione è discontinua. La curva della coppia presenta anche un'oscillazione ad alta frequenza che suggerisce la presenza di una frequenza di risonanza eliminabile utilizzando le impostazioni del filtro notch nel servozionamento Sigma-II. L'alta frequenza viene rafforzata perché si riflette anche nella curva della velocità. Per maggiori informazioni sulle impostazioni del filtro notch, fare riferimento al manuale del servozionamento Sigma-II.

fig. 45



Risoluzione dei problemi tramite l'oscilloscopio

Quando i dati richiesti sono acquisiti e registrati nelle voci della memoria TABLE, è possibile utilizzare l'oscilloscopio per visualizzarli. Questo può essere d'aiuto nella messa a punto del sistema e nella risoluzione dei problemi. L'esempio fornito in questa sezione mostra in che modo un bug difficile da analizzare può essere spiegato e risolto chiaramente utilizzando i dati acquisiti e l'oscilloscopio.

Il parametro **end_pos**, che definisce i valori della tabella CAM, dipende da condizioni esterne al sistema. Pertanto, un programma che viene eseguito all'interno di un altro task o, addirittura, che controlla un dispositivo utilizzando la comunicazione FINS, può modificare la tabella durante l'esecuzione del programma principale che collega due assi. Supponiamo che queste variazioni di condizione, che producono un cambiamento del parametro **end_pos**, si verifichino perlopiù quando gli assi non sono collegati, cioè quando il comando **CAMBOX** non è in corso di esecuzione. Supponiamo inoltre che, quando gli assi sono collegati, la condizione cambi molto raramente. La variazione del parametro **end_pos** fa partire un nuovo calcolo della tabella CAM durante l'esecuzione del comando **CAMBOX**. Di conseguenza, una parte della posizione richiesta dell'asse slave segue il profilo così com'era prima della variazione, mentre l'altra parte segue il profilo modificato. Il risultato finale è una discontinuità del profilo che produce nell'asse una velocità indefinita e provoca questo errore: WDOG si disattiva e tutti gli assi si arrestano. Un simile scenario è difficile da analizzare quando non si sa che cosa sta succedendo. L'unica cosa che l'utente vede è che nell'asse slave si verifica un errore a intervalli di alcune ore o anche meno spesso. Ma l'oscilloscopio può chiaramente mostrare dove si trova il problema. Per usare l'oscilloscopio, è necessario che tutti i parametri desiderati siano acquisiti nel momento in cui si verifica un errore. Ciò è possibile se si strutturano i programmi applicativi in un certo modo. Le procedure ottimali di programmazione consigliano di disporre di un programma d'avvio separato che viene eseguito all'accensione del sistema e ne controlla l'integrità, verificando se tutti i dispositivi previsti sono collegati e inizializzati. Per un esempio di programma

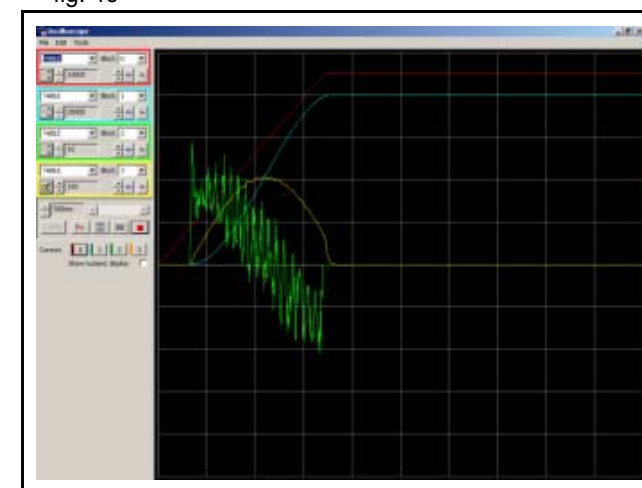
d'avvio, vedere la sezione 6.1.1. Si consiglia di impostare il programma d'avvio in maniera tale per cui, una volta terminata la sua esecuzione, faccia partire solo un programma che tuteli la sicurezza e l'integrità dell'applicazione e l'esecuzione di tutti gli altri programmi applicativi. Tale programma viene solitamente denominato programma SHELL. Per maggiori informazioni sulla stesura di un programma SHELL, vedere la sezione 6.2.1.

Supponiamo che il programma sia scritto in modo da contenere la seguente frazione di codice:

```
'Quando si verifica un errore, tutti i programmi
vengono arrestati. Non viene
'effettuata nessuna nuova acquisizione
dall'oscilloscopio e alle voci selezionate della
memoria TABLE
'sono state memorizzate le ultime registrazioni di dati
in cui
'si è verificato un errore. Pertanto, è possibile
recuperare questa
'registrazione e analizzarla.
loop:
    IF MOTION_ERROR<>0 THEN HALT
GOTO loop
```

Quando si verifica un errore su un asse, questo codice di programma provoca l'arresto di tutti i programmi e tutte le registrazioni. I dati sono già stati acquisiti nella memoria TABLE ed è possibile iniziare ad usare l'oscilloscopio per visualizzare lo stato dei parametri desiderati nel momento in cui si è verificato l'errore. Utilizzando lo scenario sopra descritto e le impostazioni dell'oscilloscopio come quelle mostrate in fig. 44, si ottiene il risultato illustrato in fig. 46. La posizione misurata dell'asse principale, in rosso, non pare essere la causa poiché non denota alcuna discontinuità. Si esclude altresì l'esistenza di un problema meccanico poiché la coppia, in verde, presenta valori bassi. Nel momento in cui il problema si è verificato, la velocità dell'asse slave, in giallo, era bassa e uniforme: pertanto neanche questa è la causa.

fig. 46



L'azione successiva è l'analisi della tabella CAM, per vedere quali valori sono stati usati per richiedere la posizione dell'asse slave. A questo scopo, si modifica la configurazione dell'oscilloscopio per mostrare un blocco di valori compreso tra Table(0) e Table(999) in rosso, poiché è in queste voci della memoria TABLE che la tabella CAM viene creata (vedere sopra per la parte del programma che crea la tabella CAM). La configurazione modificata è mostrata in figura.

fig. 47

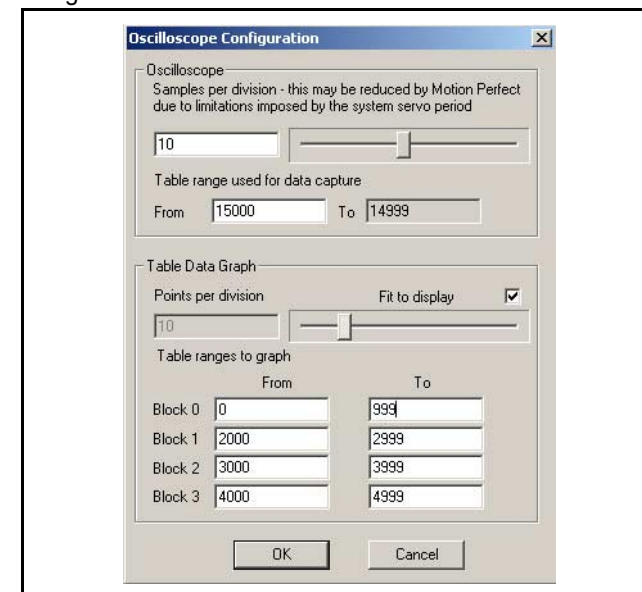
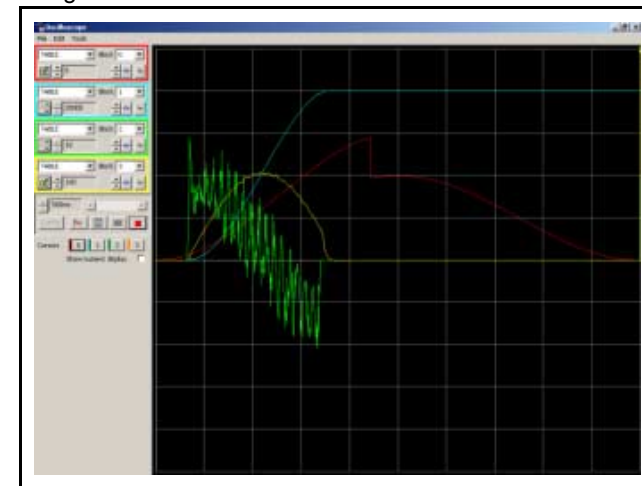


fig. 48

Il risultato è mostrato in figura. Il grafico rosso fa chiaramente vedere una discontinuità dei valori di posizione che l'asse slave deve seguire. Dal momento che la velocità è una derivata della posizione, nel punto di discontinuità della curva di posizione la velocità assume un valore elevato. (In teoria tale valore è pari a infinito; all'atto pratico si tratta solo di un valore molto elevato). Questa è la causa dell'errore. Il grafico rosso mostra dove si trova la radice del problema. L'ampiezza della curva del coseno e, di conseguenza, il parametro **end_pos** sono stati cambiati durante l'esecuzione del comando **CAMBOX**. La soluzione è semplice: impedire che il parametro **end_pos** venga modificato durante l'esecuzione di **CAMBOX**. A questo scopo, occorre modificare i programmi presenti in Trajexia oppure in un'altra unità di controllo (se il parametro è modificato fuori del raggio d'azione dei programmi applicativi, ad esempio da un messaggio FINS).





La base temporale dei punti della tabella CAM non coincide con quella dell'acquisizione degli altri segnali. Il momento in cui si verifica la discontinuità della tabella CAM (grafico rosso) coincide con quello dell'interruzione del movimento. Per analizzare ciò, controllare su base individuale i valori della posizione utilizzando un foglio elettronico. Per analizzare dettagliatamente i punti della tabella è possibile esportarli su un foglio elettronico e sottoporli a un esame più complesso.

6.2 Esempi pratici.

6.2.1 Programma shell

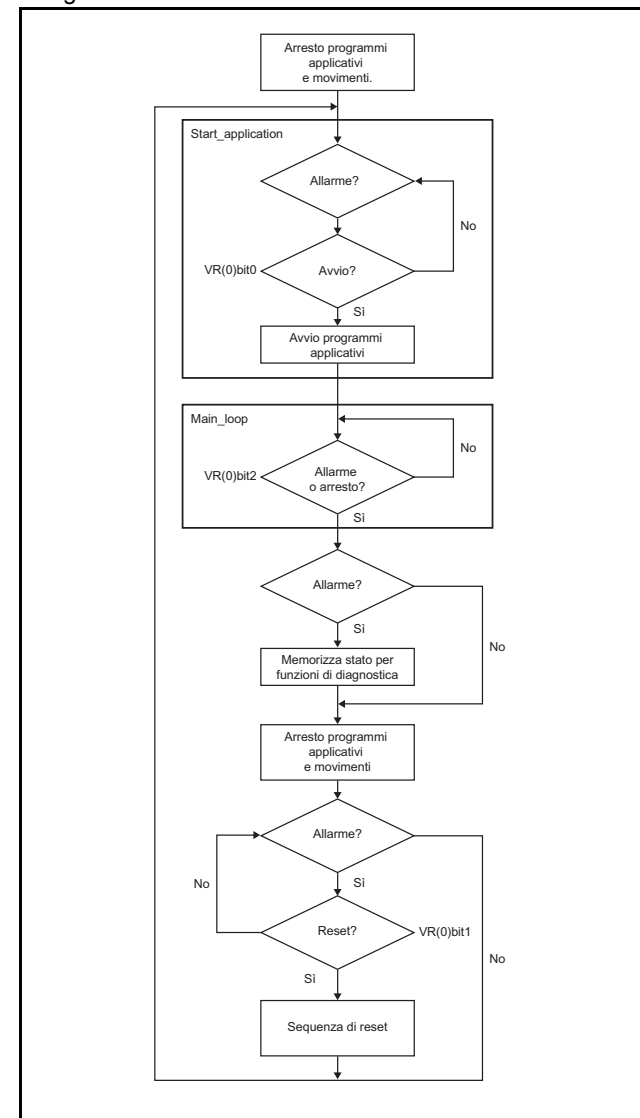
Le procedure ottimali di programmazione richiedono la stesura di un buon programma shell. Un programma shell avvia, arresta e ripristina i programmi applicativi. Il programma shell non è una necessità, ma imparte una struttura all'applicazione e aumenta l'efficacia del metodo impiegato per programmare l'unità di controllo assi.

Di seguito viene presentato un esempio di programma shell. Si consiglia vivamente di modificare il programma per adattarlo alle necessità specifiche dell'applicazione. Il corretto funzionamento del programma va controllato prima di poterlo considerare affidabile. Il programma è solitamente impostato per avviarsi, con un basso livello di priorità, al momento dell'accensione.

Esempio

```
'=====
'ESEMPIO DI PROGRAMMA SHELL
'QUESTA VERSIONE È STATA SVILUPPATA PER ESSERE USATA
'CON SERVOAZIONAMENTI MECHATROLINK
'ADATTARE QUESTO PROGRAMMA IN BASE ALLA PROPRIA
'APPLICAZIONE
'=====
'IN QUESTO ESEMPIO QUALSIASI ERRORE CAUSA L'ARRESTO
'DI TUTTI I PROGRAMMI E TUTTI I MOVIMENTI
'VIENE UTILIZZATA UNA VARIABILE GLOBALE (VR(0)) PER LA
'GESTIONE DEL PROGRAMMA.
' VR(0)bit0 per avviare l'applicazione (fronte
'di salita)
' VR(0)bit1 per il RESET di tutti gli allarmi (fronte
'di salita)
' VR(0)bit2 per ARRESTARE l'applicazione (fronte
'di salita)
'=====
'Questo esempio riguarda un'applicazione con tre
'servoazionamenti
'AXIS 1, 2 e 3
'=====
```

fig. 49



```
'Inizializzazione delle variabili
max_axis=2
'project_status
'=0 valore iniziale
'=1 programmi arrestati in assenza di errori
'=2 programmi arrestati in presenza di errori
'=3 programmi in esecuzione
project_status=0

'alarm_status
'=0 Nullo
'=1 Allarme sull'asse 0
'=2 Allarme sull'asse 1
'=3 Allarme sull'asse 3
'=4 Allarme su MECHATROLINK
'=5 In condizioni ottimali
alarm_status=5

'Azione
'=0 Nullo
'=1 Premere reset per riavviare
'=2 Reset in corso
'=3 OK
action=3

GOSUB stop_all
GOSUB start_application

loop:
  'In presenza di comando di errore o arresto
  IF MOTION_ERROR<>0 OR READ_BIT(1,0) THEN GOSUB alarm_sequence

  'Cancellare l'allarme del servomotorio, se presente
  IF (DRIVE_STATUS AXIS(0) AND 2)>0 THEN DRIVE_CLEAR AXIS(0)
  IF (DRIVE_STATUS AXIS(1) AND 2)>0 THEN DRIVE_CLEAR AXIS(1)
  IF (DRIVE_STATUS AXIS(1) AND 2)>0 THEN DRIVE_CLEAR AXIS(1)
GOTO loop

alarm_sequence:
  IF MOTION_ERROR<>0 THEN
    project_status=2
    action=1
```

```
'ERRORE DI DIAGNOSTICA
'Ricerca errori MECHATROLINK in corso
IF (AXISSTATUS AXIS(0) AND 4)<>0 THEN
    alarm_status=4
ELSEIF (AXISSTATUS AXIS(1) AND 4)<>0 THEN
    alarm_status=4
ELSEIF (AXISSTATUS AXIS(1) AND 4)<>0 THEN
    alarm_status=4
ELSE
    'Ricerca errore dell'asse in corso
    alarm_status=ERROR_AXIS+1
ENDIF
ELSE
    project_status=1
ENDIF
GOSUB stop_all
GOSUB reset_all
GOSUB start_application
RETURN

stop_all:
'ARRESTA I PROGRAMMI
STOP "APPLICATION"

'ARRESTA GLI ASSI
FOR i= 0 TO max_axis
    BASE(i)
    CANCEL(1) 'Cancella NTYPE
    WA(1)
    CANCEL(1) 'Cancella eventuale buffer di programma
    CANCEL 'Cancella MTYPE
NEXT i
'Rilascia il comando RUN nei servozionamenti
WDOG=0
'Apri l'anello di posizione
FOR i= 0 TO max_axis
    BASE(i)
    WAIT IDLE
    SERVO=0
NEXT i
RETURN
```



```
reset_all:
  WHILE MOTION_ERROR<>0
    'Attende il fronte di salita nell'ingresso RESET
    WAIT UNTIL READ_BIT(2,0)=0
    WAIT UNTIL READ_BIT(2,0)=1
    action=2
    'Ripristino servomotori contenenti errori
    FOR i=0 TO max_axis
      BASE(i)
      'Nel caso di errori di ML-II il ripristino è:
      IF (AXISSTATUS AND 4)<>0 THEN
        MECHATROLINK(0,0)
        WA(3000)
        DATUM(0)
        RUN "startup",1
        STOP
      ENDIF
      'In caso di errore del servomotore
      IF (AXISSTATUS AND 8)<>0 THEN DRIVE_CLEAR
    NEXT i
    WA(100)
    'In caso di errore dell'asse
    DATUM(0)
  WEND
  project_status=1 'Arrestato in assenza di errori
  alarm_status=5
  action=3
RETURN

start_application:
  'Attende il fronte di salita nel bit 0 di VR(0)
  WHILE READ_BIT(0,0)=0
    IF MOTION_ERROR<>0 THEN RETURN
  WEND
  WHILE READ_BIT(0,0)=1
    IF MOTION_ERROR<>0 THEN RETURN
  WEND
  RUN "APPLICATION"
  project_status=3 'Applicazione in corso di esecuzione
RETURN
```

6.2.2 Programma di inizializzazione

Il programma di inizializzazione imposta i parametri degli assi. Questi parametri dipendono dalla risoluzione dell'encoder per motori e dalla velocità massima del motore.



Nota:

Per queste informazioni, fare riferimento al catalogo del motore e del servozionamento.

```
'=====
'ESEMPIO DI PROGRAMMA DI INIZIALIZZAZIONE
'QUESTA VERSIONE È STATA SVILUPPATA PER ESSERE USATA
CON SERVOAZIONAMENTI MECHATROLINK
'ADATTARE QUESTO PROGRAMMA IN BASE ALLA PROPRIA
APPLICAZIONE
'=====

BASE(x)
restart=0
inertia_ratio=set_load_inertia_ratio

'-----
'ESEMPIO 1
'Dati motore SGMAH-01AAA61D-OY
'-----
enc_resolution=2^13 'encoder a 13 bit
max_speed=5000 'velocità max. 5000 giri/min

'-----
'ESEMPIO 2
'Dati motore SGMAH-01A1A61D-OY
'-----
enc_resolution=2^16 'encoder a 16 bit
max_speed=5000 'velocità max. 5000 giri/min

'-----
'SCRITTURA DEI PARAMETRI SUL SERVOAZIONAMENTO
'-----
DRIVE_WRITE($103,2,inertia_ratio) 'Scrivo il rapporto
di inerzia
```

```

DRIVE_READ($110,2,10)
IF VR(10)<>$0012 THEN
    DRIVE_WRITE($110,2,$0012,1)
    'Pn110=0012h (autotuning disabilitato)
    restart=1
ENDIF
DRIVE_READ($202,2,10)
IF VR(10)<>1 THEN
    DRIVE_WRITE($202,2,1,1)
    'Pn202=1 (numeratore del rapporto di riduzione nel servozionamento. Il valore predefinito ĉ 4)
    restart=1
ENDIF
DRIVE_READ($511,2,10)
IF VR(10)<>$6548 THEN
    DRIVE_WRITE($511,2,$6548,1)
    'Pn511 imposta gli ingressi di registrazione nel servozionamento
    restart=1
ENDIF
DRIVE_READ($81E,2,10)
IF VR(10)<>$4321 THEN
    DRIVE_WRITE($81E,2,$4321,1)
    'Pn81E=$4321 Per rendere gli ingressi digitali del servozionamento
    'disponibili alla lettura attraverso il canale DRIVE_INPUTS
    restart=1
ENDIF
IF restart=1 THEN DRIVE_RESET

'-----
'Guadagni iniziali per MECHATROLINK_SPEED
'-----
'L'esperienza dimostra che questa impostazione ĉ un buon punto iniziale
P_GAIN=INT(214748.3648*max_speed/enc_resolution)
'Questo ĉ il valore ottimale. Se necessario, effettuare l'impostazione
VFF_GAIN=INT(60000*1073741824/enc_resolution/max_speed)

'-----
'Guadagni iniziali per la modalitř MECHATROLINK_POSITION
'-----
'Modificare la rigiditř (Fn001) in base al 'sistema meccanico
'Se necessario, modificare il guadagno feedforward Pn109

```

```
'-----  
'Parametro iniziale di AXIS  
'-----  
'Se impostato su 1 (e Pn202=Pn203=1) le UNITS sono 'conteggi encoder  
UNITS=1  
'Il FE teorico fa funzionare il motore alla "max_speed"  
'senza VFF_GAIN in MECHATROLINK SPEED  
FE_LIMIT=1073741824/P_GAIN/UNITS  
'SPEED è impostata a 1/3 di "max_speed"  
SPEED=(max_speed73)*enc_resolution/60/UNITS  
'ACCEL in 200 ms da 0 a "max_speed"  
ACCEL=SPEED/0.2  
'ACCEL in 200 ms da "max_speed" a 0  
ACCEL=SPEED/0.2
```

6.2.3 Programma per asse singolo

Questo è un semplice programma per far funzionare un solo asse.

Esempio

```
'GOSUB ricerca dell'origine
```

```
BASE (0)
```

```
DEFPOS (0)
```

```
WA (100)
```

```
loop:
```

```
    MOVE (1440)
```

```
    WAIT IDLE
```

```
    WA (100)
```

```
GOTO loop
```

In questo esempio le unità di misura sono i gradi, pertanto:

- Encoder a 13 bit
- Pn202=32
- Pn203=45
- **UNITS=32**

Il grafico in figura è tipico di questo movimento da un punto all'altro con accelerazione lineare. Osservare quanto segue:

- In fase di accelerazione lineare, il grafico della posizione è una parabola (poiché la velocità è una derivata della posizione).
- In fase di velocità costante, il grafico della posizione è una linea retta.
- In fase di decelerazione lineare, il grafico della posizione è una parabola inversa.
- In fase di arresto, il grafico della posizione è costante.
- Quando si verifica un overflow (**MPOS** ≥ **REP_DIST**), la posizione passa a 0 se **REP_OPTION=1** o a **-REP_DIST** se **REP_OPTION=0**.

fig. 50



- L'errore di inseguimento è proporzionale alla velocità se si utilizza solo il guadagno proporzionale nell'anello di posizione.
- La coppia (che è fornita da **DRIVE_MONITOR** come percentuale della coppia nominale del motore quando si imposta **DRIVE_CONTROL=11**) è proporzionale all'accelerazione in base alla formula:

$$Coppia_{totale} = J_{totale} \times \alpha + Coppia_{frizione}$$

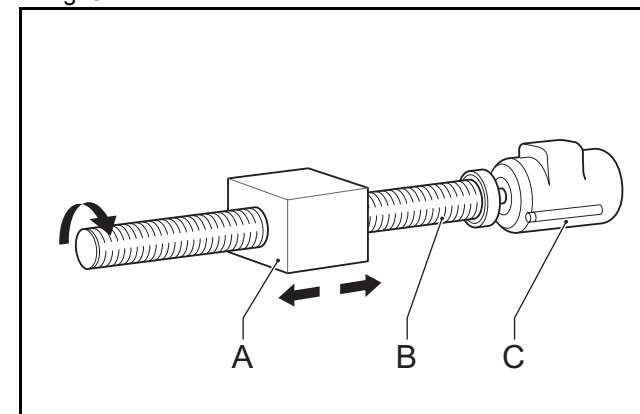
dove $Coppia_{frizione}$ ha solitamente un valore contenuto, α è l'accelerazione angolare e J l'inerzia del sistema.

6.2.4 Posizione tramite individuazione del prodotto

Una vite a sfere si sposta in avanti alla velocità di slittamento fino a quando non raggiunge un prodotto e un microselettore (IN(2)) non si accende.

La vite a sfera si arresta immediatamente, la posizione in cui il prodotto è stato individuato viene indicata e la vite a sfera ritorna ad alta velocità nella posizione iniziale.

fig. 51

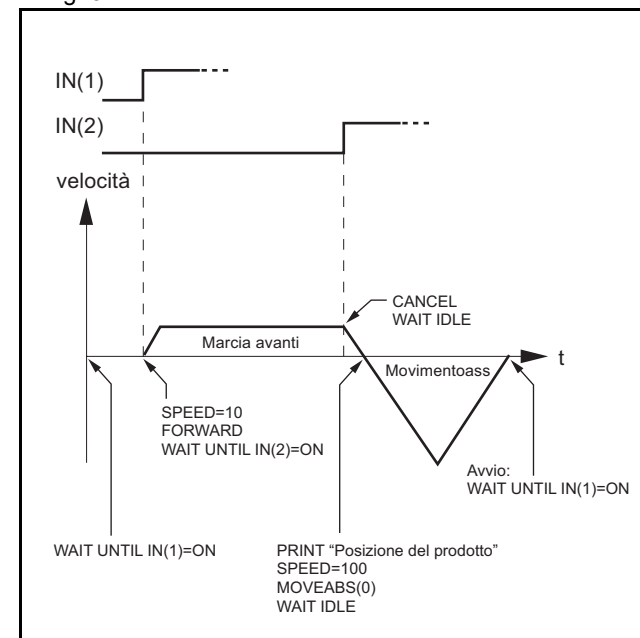


Esempio

```

start:
    WAIT UNTIL IN(1)=ON
    SPEED=10
    FORWARD
    WAIT UNTIL IN(2)=ON
    prod_pos=MPOS
    CANCEL
    WAIT IDLE
    PRINT "Posizione del prodotto: "; prod_pos
    SPEED=100
    MOVEABS(0)
    WAIT IDLE
GOTO start
  
```

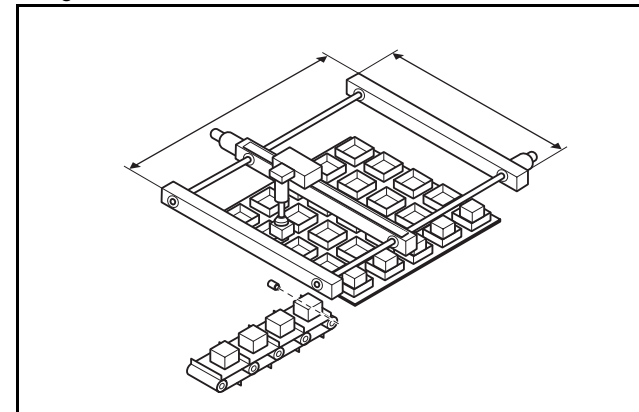
fig. 52



6.2.5 Posizione su una griglia

Una tavola quadrata con lato di 1 m è suddivisa in una griglia 5x5. Ognuna delle posizioni della griglia contiene un riquadro da riempire usando lo stesso schema a quadrati da 100x100 mm. Un ugello di riempimento controllato dall'uscita digitale 8 deve essere aperto durante l'operazione di riempimento del riquadro e rimanere chiuso altrimenti.

fig. 53



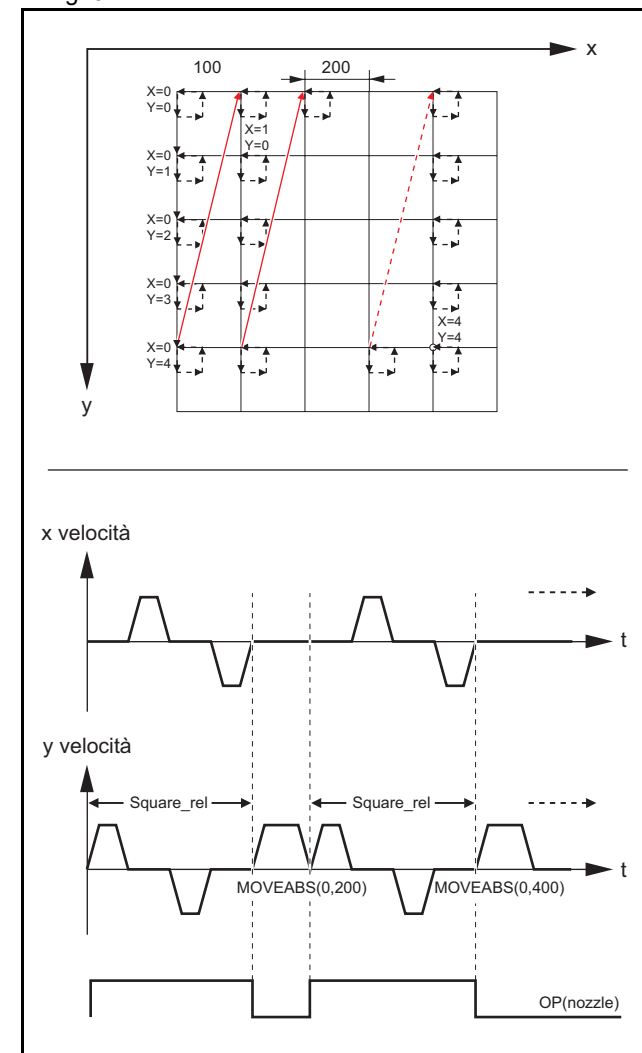
Esempio

```

nozzle = 8
start:
  FOR x = 0 TO 4
    FOR y = 0 TO 4
      MOVEABS(x*200, y*200)
      WAIT IDLE
      OP(nozzle, ON)
      GOSUB square_rel
      OP(nozzle, OFF)
    NEXT y
  NEXT x
GOTO start
square_rel:
  MOVE(0, 100)
  MOVE(100, 0)
  MOVE(0, -100)
  MOVE(-100, 0)
  WAIT IDLE
  WA(1000)
RETURN

```

fig. 54



6.2.6 Programma per un alimentatore di sacchetti

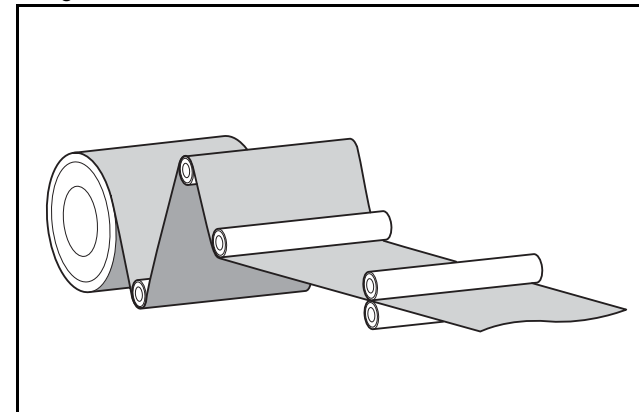
Un alimentatore di sacchetti fa avanzare una pellicola per una distanza fissa stabilita dall'operatore. La figura mostra un tipico alimentatore di sacchetti all'interno di una macchina.

La modalità di funzionamento degli alimentatori di sacchetti è duplice.

- Senza tacca: fa avanzare la pellicola per una distanza preimpostata, nel caso di pellicole dal colore uniforme.
- Con tacca: fa avanzare la pellicola fino a una tacca stampato.

Il programma contenuto in questa sezione mostra il codice tipico per un alimentatore di sacchetti.

fig. 55



Esempio

```

=====
'Programma per un ALIMENTATORE DI SACCHETTI
=====
'In caso di funzionamento con tacca, qualora mancasse
una tacca
'sposta la pellicola per la distanza teorica. Qualora
tuttavia la tacca mancasse per
'diversi sacchetti consecutivi, arresta l'operazione.
'Un'uscita digitale si attiva in un momento fissato per
tagliare
'il sacchetto.
=====

```

```

'Inizializzazione delle variabili
start_signal=7
max_fail=3
program_alarm=0
failed=0
feeder_axis=2
BASE(feeder_axis)
'Il contatore di posizione (MPOS,DPOS) va da 0 a 999999
'e quindi torna a 0
UNITS=27
SPEED=100
ACCEL=1000
DECEL=1000
REP_DIST=1000000
REP_OPTION=1
SERVO=ON
WDOG=ON

```

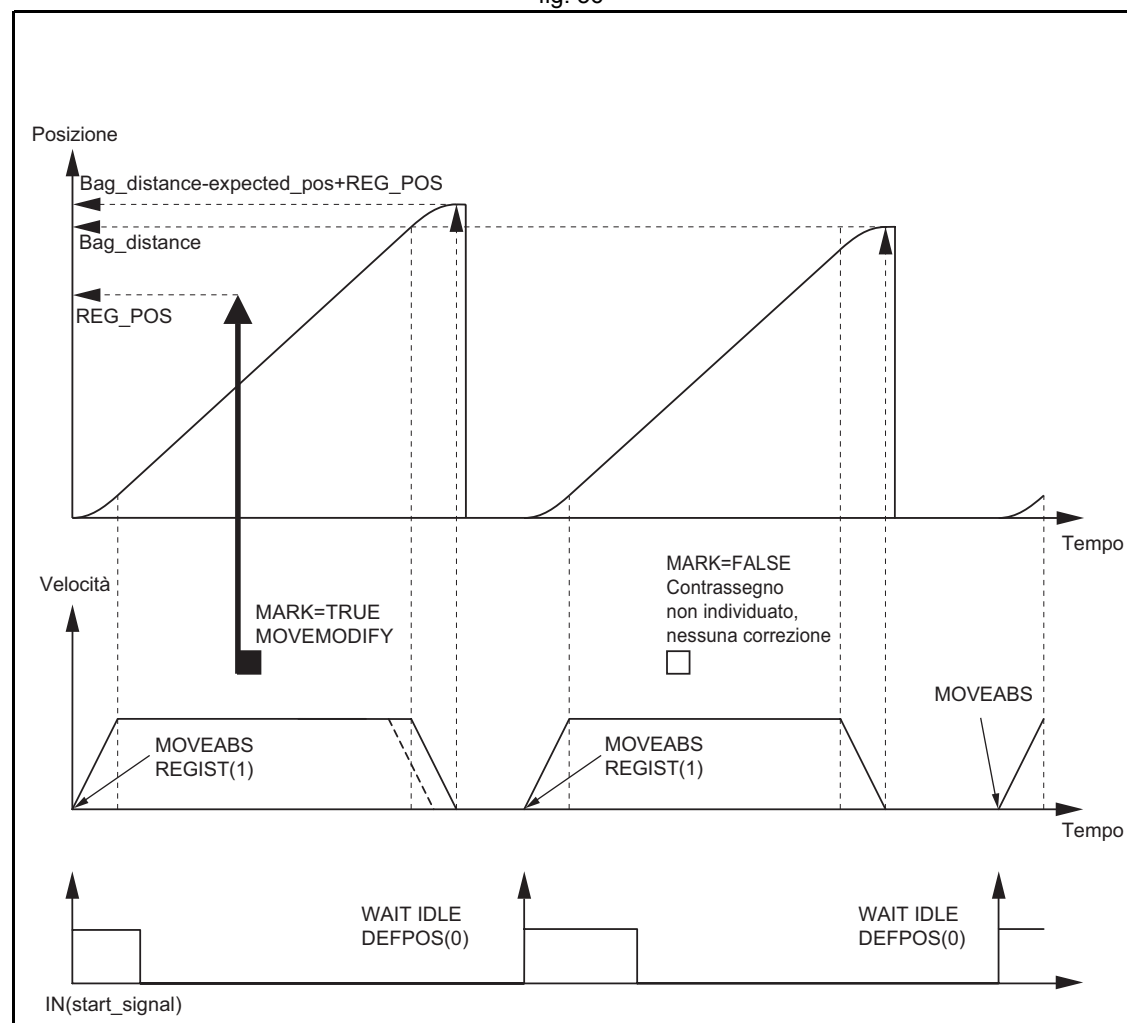
```

'Programma principale
loop:
'Definisce la posizione corrente come zero
DEFPOS(0)

'Attende il fronte di salita nell'ingresso digitale
'"start_signal"

```

fig. 56



```
WAIT UNTIL IN(start_signal)=0
WAIT UNTIL IN(start_signal)=1
```

```
'Sposta il sacchetto per la distanza prevista
MOVEABS (bag_distance)
WAIT UNTIL MTYPE=2 'Per verificare che MOVEABS sia eseguito
```

```
'Se si lavora con MARK, attiva il meccanismo di scatto
'MARK=FALSE quando il meccanismo di scatto è attivato e TRUE quando non è attivato
IF work_with_mark AND MARK THEN
    REGIST(1)
    WAIT UNTIL MARK=0
ENDIF
```

```
'Attende fino al completamento del movimento o all'individuazione del contrassegno
WAIT UNTIL MTYPE=0 OR (MARK AND work_with_mark)
```

```
'Funzionamento con contrassegno
IF work_with_mark THEN
    IF MARK THEN 'Se è stata individuata la tacca, la posizione viene corretta
        MOVEMODIFY (bag_distance-expected_pos+REG_POS)
        failed=0

    ELSE 'Se la tacca non è stato individuata
        PRINT "Tacca non individuata"
        failed=failed+1
        IF failed>max_fail THEN 'Dopo diverse individuazioni errate consecutive, arresta l'applicazione
            PRINT "Tacca persa in via definitiva"
            program_alarm=3
            STOP
        ENDIF
    ENDIF
ENDIF
```

```
'Attende fino al completamento del movimento di alimentazione
WAIT IDLE
GOTO loop
```

6.2.7 Tabella CAM all'interno di un programma

Mostra come creare una tabella CAM all'interno di un programma e utilizzare il comando di movimento **CAMBOX**.

Il profilo utilizzato è il punto di partenza COS. Si tratta di un profilo piuttosto comune per applicazioni come gli alimentatori, poiché:

- Il movimento fornisce un'accelerazione graduale senza cambiamenti improvvisi, riducendo al minimo lo scorrimento del materiale
- Offre una decelerazione rapida, riducendo il tempo di ciclo. Durante la decelerazione non si verifica alcuno scorrimento di materiale e l'attrito aiuta a portare l'arresto a zero.

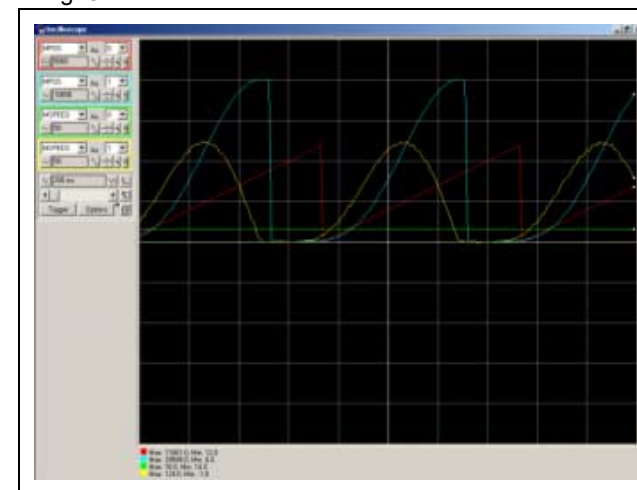
Esempio

```
start:
  GOSUB filltable
  WDOG=1 'Imposta i servoazionamenti su RUN
  BASE(1)
  SERVO=1 'Abilita l'anello di posizione sull'asse 1
  BASE(0)
  SERVO=1 'Abilita l'anello di posizione sull'asse 0
  'Il contatore di posizione conteggia da 0 a 11999
  'e quindi di nuovo a 0
  REP_OPTION=1
  REP_DIST=12000
  SPEED=200
  FORWARD

BASE(1)
loop:
  CAMBOX(in_tbl,end_tbl,1,lnk_dst,master,opt,start)
  WAIT IDLE
GOTO loop

filltable:
  'La struttura della CAM è memorizzata nelle voci da
  TABLE(0) a
  'TABLE(360)
  npoints=360
```

fig. 57



```

in_tbl=0
end_tbl=in_tbl+npoints
'La distanza dell'asse principale per la creazione
della CAM
lnk_dst=10000
'Asse principale
master=0
'La CAM inizia esattamente quando l'asse principale
raggiunge
'la posizione "start"
opt=2
start=1000

k=100
'Riempire la memoria TABLE con la forma d'onda
appropriata
FOR i= in_tbl TO end_tbl
    TABLE(i, (k*(COS(PI*i/npoints)-1))^2)
NEXT i
RETURN

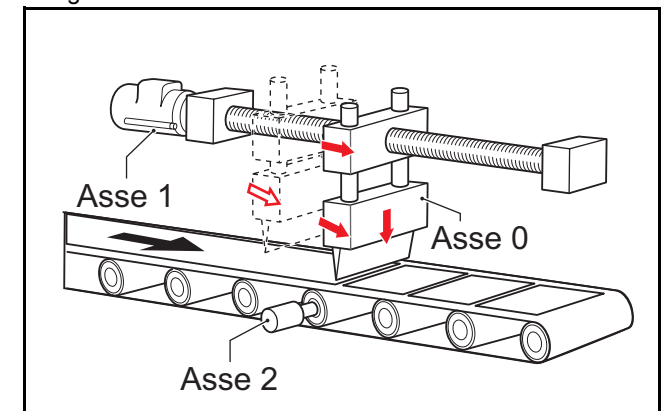
```

6.2.8 Programma per lama semovente

Questo è un esempio di programma per lama semovente. In questa applicazione vi sono tre assi:

- L'asse 0, shear_axis (asse della lama), che effettua l'avanzamento della lama.
- L'asse 1, flying_axis (asse semovente), che è la lama semovente.
- L'asse 2, line_axis (asse di linea), che trasporta il materiale.

fig. 58



Esempio

```

=====
'Programma per LAMA SEMOVENTE
=====
'Esempio tipico di applicazione per lama semovente.
'Un asse (line_axis) trasporta il materiale.
'Un secondo asse (flying_axis) è la lama semovente
stessa.
'Un terzo asse (shear_axis) effettua l'avanzamento
della lama.
'La distanza di sincronizzazione deve essere
sufficientemente lunga
'per consentire il taglio alla velocità massima.
'Il ritorno della lama semovente viene eseguito a una
velocità tale che il tempo di attesa è pari a zero
(ottimizzazione
'del movimento).
'Ancora una volta si presume che tutto sia stato
calcolato in maniera tale da non superare la velocità
massima del motore
'alla velocità massima della linea.
=====

```

```

cut_counter=0
line_axis=2
shear_axis=0
flying_axis=1

```

```

SERVO AXIS(line_axis)=ON
SERVO AXIS(flying_axis)=ON
SERVO AXIS(shear_axis)=ON
WDOG=ON

```

'PRIMO CICLO

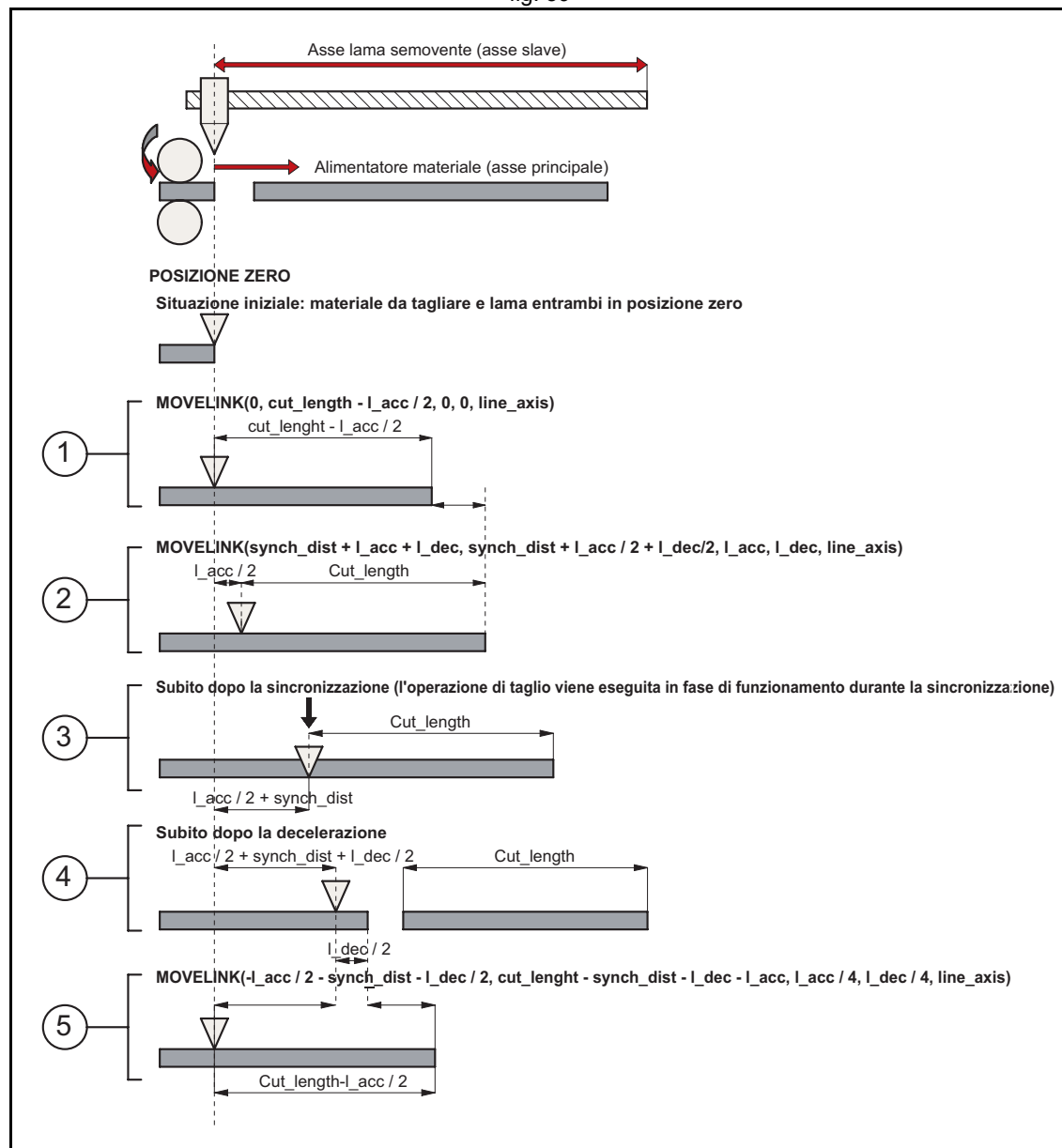
```

'Effettua un primo taglio del materiale
MOVEABS(end_pos) AXIS(shear_axis)
WAIT UNTIL MTYPE AXIS(shear_axis)=2
WAIT IDLE AXIS(shear_axis)

```

'La prima volta si sperimenta una certa attesa perché

fig. 59



```
'il materiale   appena stato tagliato
wait_distance=cut_lenght-l_acc/2
MOVELINK(0,wait_distance,0,0,line_axis) AXIS(flying_axis)
WAIT UNTIL MTYPE AXIS(flying_axis)=22

'La linea viene avviata
FORWARD AXIS(line_axis)

loop:

'Aggiorna la velocit  della linea a ogni ciclo
SPEED AXIS(line_axis)=line_speed

'Movimento di taglio a velocit  sincronizzata
line_cut=synch_dist+l_acc+l_dec
shear_cut=synch_dist+l_acc/2+l_dec/2
MOVELINK(shear_cut,line_cut,l_acc,l_dec,line_axis) AXIS(flying_axis)
WAIT UNTIL MPOS AXIS(flying_axis)>l_acc/2

'Attiva la lama quando   sincronizzata con la linea
'Bassa velocit  per il taglio
SPEED AXIS(shear_axis)=cut_speed
MOVEABS(end_pos) AXIS(shear_axis)
MOVEABS(0) AXIS(shear_axis)
WAIT UNTIL NTYPE AXIS(shear_axis)=2
'Alta velocit  per il ritorno
WAIT LOADED AXIS(shear_axis)
SPEED AXIS(shear_axis)=return_speed

cut_counter=cut_counter+linch

'Il ritorno in posizione   sincronizzato con il master in maniera tale
'che non vi sono tempi di attesa
line_back=cut_length-synch_dist-l_dec-l_acc
shear_cut=l_acc/2+synch_dist+l_dec/2
MOVELINK(-shear_cut,line_back,l_acc/4,l_dec/4,line_axis) AXIS(flying_axis)

GOTO loop
```


Il grafico velocità-tempo mostra le fasi dell'esempio appena fornito. Le diverse fasi sono:

1. Il ciclo iniziale: l'asse slave aspetta che il prodotto raggiunga la lunghezza giusta per tagliarlo ($\text{cut\_length} - \text{distance\_to\_accelerate} / 2$). Quando si usa il comando **MOVELINK** è necessario dividere $\text{distance\_to\_accelerate}$, poiché quando si effettua la sincronizzazione, il movimento dell'asse principale risulta doppio rispetto a quello dell'asse slave.
2. L'asse slave accelera per sincronizzarsi con il master. Al termine dell'accelerazione, la distanza relativa tra il margine del prodotto e la lama è cut_length .
3. Questa è la parte della sincronizzazione: la distanza relativa tra il margine del prodotto e la lama rimane uguale. Viene eseguito il taglio nel materiale. Questa azione produce un nuovo margine nel materiale.
4. La parte della decelerazione: il materiale prosegue e la lama si arresta.
5. Ritorno ad alta velocità: le distanze sono calcolate in maniera tale per cui quando lo slave raggiunge la sua posizione originale, il margine del prodotto si trova nella posizione corretta per iniziare un nuovo taglio.

Un nuovo movimento ha inizio (fase 2).

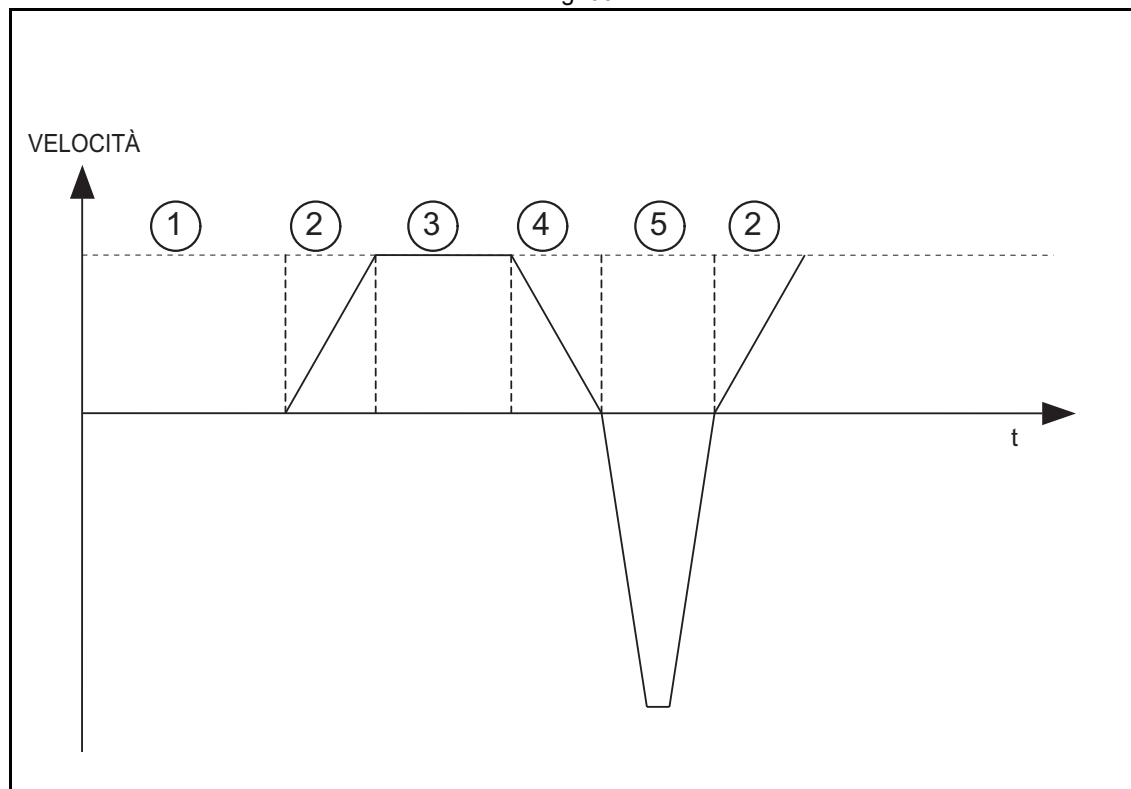
6.2.9 Programma di correzione

Questa applicazione è destinata a un'etichettatrice rotante.

Le costanti sono:

- Il prodotto arriva sopra un nastro trasportatore (asse principale) che funziona a velocità costante.
- Un'etichettatrice rotante sincronizzata 1:1 con il nastro trasportatore attacca le etichette.
- La distanza tra i prodotti è fissa e meccanicamente garantita.

fig. 60



La distanza tra le etichette non è mai esattamente costante, pertanto una correzione è necessaria. Ciò si ottiene sovrapponendo un asse virtuale al movimento dell'etichettatrice. La differenza tra la posizione prevista e la posizione effettiva viene misurata da una fotocellula. Questo è il fattore di correzione. Ogni volta che viene effettuata una correzione, la posizione di origine viene aggiornata di conseguenza.

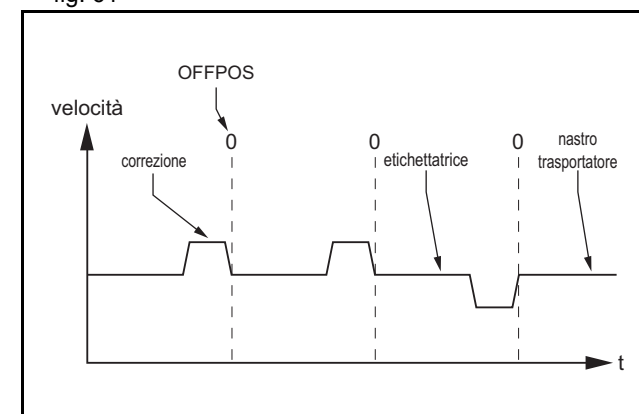
Esempio

```
conveyor=0
labeller=1
virtual=15
SERVO AXIS(conveyor)=1
SERVO AXIS(labeller)=1
WDOG=1
```

```
BASE(labeller)
CONNECT(1,conveyor)
ADDAX(virtual)
FORWARD AXIS(conveyor)
REGIST(1)
WAIT UNTIL MARK=0
```

```
loop:
  WAIT UNTIL MARK
  correction=REG_POS+expected_pos
  MOVE(correction) AXIS(virtual)
  WAIT IDLE AXIS(virtual)
  OFFPOS=-label_length+correction
  REGIST(1)
  WAIT UNTIL MARK=0
GOTO loop
```

fig. 61



7 Soluzione dei problemi

7.1 Tensione e strumenti di analisi

Controllare la tensione sui terminali di ingresso dell'alimentatore. Assicurarsi che la tensione rientri nell'intervallo specificato. Se la tensione ricade fuori dell'intervallo specificato, il sistema è soggetto a errori di funzionamento.

Per diagnosticare gli errori di TJ1-MC__ e TJ1-ML__ e risolvere i problemi di queste unità, utilizzare lo strumento software Trajexia Tools.
Per diagnosticare gli errori di TJ1-PRT e risolvere i problemi di questa unità, utilizzare un'utilità di configurazione e monitoraggio PROFIBUS (ad esempio **OMRON CX-PROFIBUS**).



Attenzione

Scollegare tutti i cavi prima di controllare se sono bruciati. Anche dopo aver controllato la conduzione nei cavi, permane un rischio di conduzione a causa del circuito di ritorno.



Attenzione

Se il segnale dell'encoder viene perso, è possibile che il servomotore si metta a funzionare in modo incontrollato o che si generi un errore. Accertarsi che il motore sia scollegato dal sistema meccanico prima di controllare il segnale dell'encoder.



Attenzione

Prima di iniziare la risoluzione dei problemi assicurarsi che nessuno si trovi all'interno degli impianti in cui si trovano le macchine e che queste non possano essere danneggiate neanche nel caso in cui il servomotore iniziasse a funzionare in modo incontrollato. Accertarsi che sia possibile arrestare immediatamente le macchine con un freno di emergenza se il motore si mettesse a funzionare in modo incontrollato.

7.2 TJ1-MC__

7.2.1 Errori di sistema

Gli errori di sistema vengono visualizzati sul display a LED di TJ1-MC__ come **Enn**, dove **nn** rappresenta il codice di errore.

Codice di errore	Descrizione	Causa	Soluzione
E00	Errore SRAM BASIC	L'hardware di TJ1-MC__ è guasto.	Sostituire TJ1-MC__.
E01	Errore canale basso SRAM di sistema	L'hardware di TJ1-MC__ è guasto.	Sostituire TJ1-MC__.
E02	Errore canale alto SRAM di sistema	L'hardware di TJ1-MC__ è guasto.	Sostituire TJ1-MC__.
E03	Errore di batteria scarica	La tensione della batteria è troppo bassa.	Sostituire la batteria.
...	L'hardware è guasto	L'hardware di TJ1-MC__ è guasto.	Sostituire TJ1-MC__.



Nota:

Per maggiori informazioni, fare riferimento alla sezione 3.2.254.

7.2.2 Errori degli assi

Gli errori degli assi vengono visualizzati sul display a LED di TJ1-MC__ come **Ann**, dove **nn** rappresenta il numero dell'asse che ha provocato l'errore.

Le cause possibili sono due:

- Valore errato o fuori intervallo del parametro degli assi impostato.
- Errore o allarme del servozionamento assegnato all'asse.

Le due cause e le rispettive soluzioni sono le seguenti:

- Valore errato o fuori intervallo del parametro degli assi.
- Errore o allarme del servozionamento assegnato all'asse.

Valore errato o fuori intervallo del parametro degli assi.

Se il valore di un parametro degli assi è errato o fuori intervallo, si verifica un errore dell'asse. Nessun allarme o errore viene visualizzato sul display del servozionamento assegnato all'asse.

Il comando **AXISSTATUS** consente di visualizzare la causa dell'errore.

Nella finestra del terminale di Trajexia Tools digitare **PRINT AXISSTATUS AXIS(nn)**, dove **nn** corrisponde al numero dell'asse. Il valore restituito dal comando **AXISSTATUS** contiene il codice di errore dell'asse.

Vedere il comando **AXISSTATUS**.

È anche possibile aprire la finestra **Axis Parameter** di Trajexia Tools e verificare il campo **AXISSTATUS** dell'asse che ha provocato l'errore.

I bit che indicano la causa dell'errore vengono visualizzati in grosse lettere rosse. Per eliminare l'errore, procedere come segue:

1. Correggere il valore.
2. Reimpostare l'unità di controllo oppure fare clic sul pulsante **Axis status error**.

Errore o allarme del servozionamento assegnato all'asse

Se un errore o un allarme sul servozionamento assegnato all'asse causa un errore dell'asse, l'allarme del servozionamento viene visualizzato sul relativo display a LED. È anche possibile aprire la finestra **Axis Parameter** di Trajexia Tools e verificare il campo **AXISSTATUS** dell'asse che ha provocato l'errore. Nel valore restituito dal comando **AXISSTATUS**, il secondo bit (bit **a**: errore di comunicazione del servozionamento) e/o il terzo bit (bit **m**: allarme del servozionamento) sono visualizzati a grandi lettere rosse.

Per eliminare l'errore, procedere come segue:

1. Fare riferimento al Manuale del servozionamento per stabilire la causa dell'errore e risolverlo.
2. Reimpostare l'unità di controllo oppure fare clic sul pulsante **Axis status error**.

7.2.3 Errori delle unità

Gli errori delle unità vengono visualizzati sul display a LED di TJ1-MC__ come **Unn**, dove **nn** rappresenta il numero dell'unità che ha provocato l'errore.

Le cause possibili sono quattro:

- Unità difettosa.
- Unità non collegata al bus Trajexia.
- Un modulo di I/O o l'inverter di un modulo MECHATROLINK-II è perso o scollegato.
- Nessun terminatore.

Unità difettosa

Sul display viene visualizzato il codice di errore **U0n**, dove **n** è compreso tra 0 e 6 e rappresenta il numero dell'unità che ha provocato l'errore.

Per il risolvere il problema, sostituire l'unità difettosa.

Unità non collegata al bus Trajexia

Sul display viene visualizzato il codice di errore **U0n**, dove **n** è compreso tra 0 e 6 e rappresenta il numero dell'unità che ha provocato l'errore.

Per risolvere il problema, controllare il connettore per bus dell'unità.

Un modulo di I/O o l'inverter di un modulo MECHATROLINK-II è perso o scollegato

Sul display viene visualizzato il codice di errore **U0n**, dove **n** è il numero di TJ1-ML__ al quale è collegato il modulo MECHATROLINK-II che ha provocato l'errore.

È possibile impostare flag di sistema per abilitare e disabilitare questi errori. Come impostazione predefinita, gli errori sono abilitati.

Per disabilitare gli errori, digitare **COORDINATOR_DATA(7,1)** nella finestra del terminale di Trajexia Tools.

Per abilitare gli errori, digitare **COORDINATOR_DATA(7,0)** nella finestra del terminale di Trajexia Tools.

Per visualizzare l'impostazione corrente, digitare **PRINT COORDINATOR_DATA(7)** nella finestra del terminale di Trajexia Tools.

Per cancellare lo stato di errore dopo la correzione, procedere come segue:

- Ricollegare il modulo di I/O o l'inverter di MECHATROLINK-II che era stato perso.
- Digitare MECHATROLINK(n, 5, station, -1) nella finestra del terminale di Trajexia Tools:
dove **n** è il numero di TJ1-ML__ al quale è collegato il modulo MECHATROLINK-II interessato e **station** è il numero del dispositivo MECHATROLINK-II che è stato perso.

Se si desidera utilizzare il sistema senza il dispositivo perso, è possibile ricollegare a TJ1-ML__ tutti i dispositivi disponibili. Per eseguire questa operazione, digitare **MECHATROLINK(n, 0)** nella finestra del terminale di Trajexia Tools, dove **n** è il numero di TJ1-ML__ che riferisce l'errore.

Nessun terminatore

Sul display viene visualizzato il codice di errore **U07**.

Per risolvere il problema, controllare il collegamento del terminatore o sostituire il terminatore se è difettoso.

7.2.4 Errori di configurazione

Gli errori di configurazione vengono visualizzati sul display a LED di TJ1-MC__ come **Cnn**, dove **nn** rappresenta il numero dell'unità che ha provocato l'errore.

Le cause che provocano un errore di configurazione sono:

- Il sistema dispone di troppe unità dello stesso tipo e non segue le regole per l'aggiunta di unità al sistema stesso.
- Le stazioni MECHATROLINK-II collegate a TJ1-ML__ sono troppe.
- Gli assi del sistema sono troppi.
- Le stazioni MECHATROLINK-II non relative agli assi collegate al sistema sono troppe.

Per risolvere questo problema, modificare il sistema di modo che rispetti le regole per l'aggiunta di unità al sistema stesso. Vedere il Manuale di riferimento hardware.

7.2.5 Sostituire la batteria

Per sostituire la batteria di backup, procedere come segue:

1. Assicurarsi che il modulo di alimentazione rimanga acceso per almeno cinque minuti. In caso contrario, il condensatore che fornisce il backup alla memoria di TJ1-MC__ quando la batteria non è collegata non si carica completamente e si rischia una perdita dei dati in memoria.
2. Aprire l'alloggiamento della batteria sollevando il coperchio.
3. Tirare i cavi bianco e rosso così da poter estrarre la batteria vecchia.
4. Assicurarsi di eseguire le due azioni successive nel giro di 30 secondi onde evitare la perdita di dati nella memoria RAM.
5. Scollegare i cavi dalla vecchia batteria.
6. Collegare i cavi alla nuova batteria.
7. Inserire la nuova batteria nel suo alloggiamento.
8. Chiudere il coperchio dell'alloggiamento della batteria.

7.3 TJ1-PRT

7.3.1 Errori di sistema

Indicazione	Problema	Soluzione
Nessun LED è acceso né lampeggia	L'alimentazione è assente.	Attivare l'alimentazione.
	TJ1-PRT è difettoso.	Sostituire TJ1-PRT.
Il LED ERH è acceso	Errore di comunicazione tra TJ1-MC__ e TJ1-PRT.	Reimpostare TJ1-MC__. Se il problema non si risolve, sostituire TJ1-MC__.
Il LED ERC è acceso	Errore del modulo. TJ1-PRT è difettoso.	Sostituire TJ1-PRT.

7.3.2 Errore di comunicazione nei dati di I/O

Indicazione	Problema	Soluzione
Il LED COMM è spento e il LED BF è acceso	La configurazione PROFIBUS è errata, la comunicazione con l'asse principale è assente.	- Controllare che TJ1-PRT abbia lo stesso indirizzo di stazione presente nella configurazione dell'asse principale. - Controllare che nessun indirizzo di stazione sia usato due volte.
	Il cablaggio di PROFIBUS è errato.	- Controllare che siano collegati i pin giusti del connettore CN1. - Controllare che non vi siano cortocircuiti o interruzioni della linea. - Controllare che sia in uso il tipo di cavo corretto. - Controllare che le linee tronche non siano troppo lunghe.
	Le terminazioni della rete PROFIBUS non sono corrette.	Collocare le terminazioni della rete PROFIBUS nei punti corretti.
	Il modulo master PROFIBUS è difettoso.	Sostituire il modulo master
	TJ1-PRT è difettoso.	Sostituire TJ1-PRT.

Indicazione	Problema	Soluzione
Il LED COMM è spento e il LED BF lampeggia	La configurazione PROFIBUS è errata, la comunicazione con l'asse principale è assente.	- Controllare che nel modulo master sia in uso il file GDS corretto. - Controllare la configurazione e i dati contenuti nei parametri dello slave. - Controllare che la rete sia stata configurata per comunicare alla velocità di trasmissione supportata da TJ1-PRT.
	Non sono stati selezionati i dati di configurazione per lo slave.	Controllare la configurazione del modulo master.
	TJ1-PRT è difettoso.	Sostituire TJ1-PRT.

7.4 TJ1-DRT

7.4.1 Errori di sistema

Indicazione	Problema	Soluzione
Nessun LED è acceso o lampeggia	L'alimentazione è assente.	Attivare l'alimentazione.
	TJ1-DRT è difettoso.	Sostituire TJ1-DRT.
Il LED ERH è acceso	Errore di comunicazione tra TJ1-MC__ e TJ1-DRT.	Reimpostare TJ1-MC__. Se il problema non si risolve, sostituire TJ1-MC__.
Il LED ERC è acceso	Errore del modulo. TJ1-DRT è difettoso.	Sostituire TJ1-DRT.

7.4.2 Errore di comunicazione nei dati di I/O

Indicazione	Problema	Soluzione
Il NOK lampeggia e il LED NF è spento	Il master DeviceNet non comunica con TJ1-DRT.	Configurare e avviare il master DeviceNet.
Il NOK è spento e il LED NF è acceso	Errore di duplicazione dell'indirizzo del nodo.	Controllare l'indirizzo del nodo.
	Errore del cavo di rete.	Controllare i cavi di rete.

7.5 TJ1-ML__

7.5.1 Errori di sistema

Indicazione	Problema	Soluzione
Tutti i LED sono spenti	L'alimentazione è assente.	Attivare l'alimentazione.
	TJ1-ML__ è difettoso.	Sostituire TJ1-ML__.

7.5.2 Errori del bus

Indicazione	Problema	Soluzione
Il LED BF è acceso	Errore nel cavo del bus MECHATROLINK-II.	Controllare che nei cavi del MECHATROLINK-II che collegano le stazioni all'unità non vi siano interruzioni e irregolarità (cortocircuiti tra le linee di comunicazione A e B, cortocircuiti in qualsiasi linea di comunicazione con schermatura).
	Il terminatore del bus MECHATROLINK-II manca o è danneggiato.	Montare un terminatore del bus MECHATROLINK-II sull'ultima stazione della catena oppure sostituirlo.
	La stazione MECHATROLINK-II collegata al modulo viene persa a causa di un'interruzione dell'alimentazione o di un errore dell'interfaccia MECHATROLINK-II alla stazione.	Controllare l'alimentazione e l'interfaccia MECHATROLINK-II della stazione che ha provocato il problema. Sostituire la stazione se necessario.
	TJ1-ML__ è difettoso.	Sostituire TJ1-ML__.



Nota:

Dopo aver risolto la causa di un errore, assicurarsi di inizializzare nuovamente il bus MECHATROLINK-II sull'unità in cui è comparso l'errore. Digitare nella finestra del terminale di Trajexia Tools:

MECHATROLINK(n, 0)

dove **n** rappresenta il numero dell'unità alla quale è collegata l'unità che ha provocato l'errore.

7.6 TJ1-FL02

7.6.1 Errori di sistema

Indicazione	Problema	Soluzione
Tutti i LED sono spenti	L'alimentazione è assente.	Attivare l'alimentazione.
	TJ1-FL02 è difettoso.	Sostituire TJ1-FL02.
Il LED RUN è acceso Il LED EN A o EN B è spento	L'asse per il quale il LED EN è spento non è abilitato.	Abilitare l'asse: eseguire sull'asse WDOG=ON e/o AXIS_ENABLE .
Il LED RUN è acceso Il LED EN A o EN B lampeggia	Nell'asse per cui il LED EN lampeggia è presente un errore.	TJ1-MC__ indica il numero dell'asse in cui è presente l'errore. Eliminare la causa dell'errore dell'asse e cancellare l'errore dell'asse oppure riavviare il sistema.

A

Asse flessibile

Errori 305

Attenzione, sicurezza 16

B

BASIC

Specifiche matematiche 27

Strutture dei dati 25

Variabili 25

Batteria 302

C

Collegamento

del PC 186

Comandi BASIC 33

per Host Link 159

Comando

Asse 33

Comunicazione 36

Controllo del programma 38

I/O 36

Programma 37

Sistema 38

Task 40

Confronto di file di programma 192

Confronto di Trajexia 192

Confronto tra progetti 192

Connessione di rete 187

Costanti 36

Creazione di applicazioni 191

D

Descrizioni dei menu 196

DEVICENET

Configurazione della comunicazione 173

Errori	304
Interfaccia	173
Stato della comunicazione	178
E	
Errori	
Assi	300
Configurazione	302
TJ1-MC__	300
Unità	301
Esecuzione del movimento	28
Esempio	
Caratteristiche del servozionamento	251
Impostazione delle unità	239
Impostazioni del guadagno	229
Modalità posizione	235
Modalità velocità	230
Posizione su una griglia	289
Posizione tramite individuazione del prodotto	287
Programma di correzione	298
Programma di inizializzazione	283
Programma per asse singolo	286
Programma per lama semovente	295
Programma per un alimentatore di sacchetti	291
Programma shell	279
Programma Startup	225
Registrazione	259
Registrazione e monitoraggio	269
Ricerca dell'origine	253
Ricerca dello zero	253
Tabella CAM	294
Esempio delle caratteristiche del servozionamento	251
Esempio di correzione	298
Esempio di guadagno	229
Esempio di inizializzazione	283

Esempio di lama semovente	295
Esempio di modalità posizione	235
Esempio di modalità velocità	230
Esempio di posizione su una griglia	289
Esempio di posizione tramite individuazione del prodotto	287
Esempio di programma per un alimentatore di sacchetti	291
Esempio di registrazione	259
Esempio di registrazione e monitoraggio	269
Esempio di ricerca dell'origine	253
Esempio di ricerca dello zero	253
Esempio di shell	279
Esempio di Startup	225
Esempio di tabella CAM	294
Esempio di unità	239
Esempio per asse singolo	286
F	
Finestra delle applicazioni	194
Funzione	
I/O	36
Matematica	37
Sistema	38
I	
Icone	195
Informazioni generali sul sistema	22
Installazione del software	181
Intelligent Drives	208
Interfaccia	
DEVICENET	173
Ethernet	153
MECHATROLINK	179
PROFIBUS	167
Seriale	158
Interfaccia della riga di comando	30

Interfaccia Profibus	167
Interfaccia seriale	158
IO status	216
J	
Jog	217
M	
MECHATROLINK	
Errori	304
Protocollo	179
Menu	
Help	224
Options	221
Program	203
Project	196
Tools	205
Unità di controllo	198
Window	224
Modificatore	
Slot	38
Multitasking	23
O	
Operando	37
Matematico	37
Oscilloscope	210
P	
Panoramica dei protocolli	153
Panoramica delle interfacce	153
Panoramica sull'hardware	23
Parametro	
Asse	34
Comunicazione	36
I/O	36

Sistema	39
Slot	38
Task	40
PC	
Avvio di Trajexia Tools	187
Connessione diretta	154
Connessione remota	155
PROFIBUS	
Configurazione della comunicazione	167
Errori	302
Stato della comunicazione	172
Programmazione in BASIC	24
Programmi BASIC	30
Protocollo	
Client FINS	158
definito dall'utente	165
DEVICENET	173
master Host Link	159
MECHATROLINK	179
PROFIBUS	167
Slave FINS	156
slave Host Link	163
Trajexia Tools	156
Protocollo definito dall'utente	165
Protocollo dello slave FINS	156
Protocollo Ethernet	153
Protocollo slave	
Host Link	163
Protocollo Trajexia Tools.	156
S	
Sicurezza, ambiente operativo	17
Sicurezza, assemblaggio dell'unità	21
Specifiche	
del PC	180

STARTUP program	
Modify	208
Strumento di programmazione	180
T	
TABLE viewer	219
V	
VR Editor	219

Storico delle revisioni

Il suffisso al numero di catalogo stampato sulla copertina del manuale indica il codice di revisione del documento.

Codice di revisione	Data	Contenuto modificato
01	Agosto 2006	Originale
02	Ottobre 2006	Aggiornamento per DeviceNet
03	Maggio 2007	Aggiornamento con TJ1-MC04 e TJ1-ML04. Comandi, esempi di programmazione e suggerimenti BASIC migliorati.