

## RD1A RD12A



**EtherCAT**®



Versione RS-232

- Attuatore rotativo con interfaccia EtherCAT conforme a ETG.1000
- Include il protocollo CoE e l'EtherCAT State Machine
- Motore brushless, coppia nominale 5 Nm
- Encoder assoluto multigiro reale a 18 bit
- RD12A con freno integrato, coppia di stazionamento 17 Nm
- Per operazioni di cambio formato e posizionamento automatizzato

Descrive i seguenti modelli:

- RD1A-P8-Txx-EC-...
- RD12A-P8-Txx-EC-...

Indice generale

Norma di sicurezza	26
Identificazione	28
Installazione meccanica	29
Connessioni elettriche	33
Interfaccia EtherCAT®	54
Interfaccia MODBUS®	115
Lista parametri di default	186

Questa pubblicazione è edita da Lika Electronic s.r.l. 2021. All rights reserved. Tutti i diritti riservati. Alle Rechte vorbehalten. Todos los derechos reservados. Tous droits réservés.

Il presente manuale e le informazioni in esso contenute sono proprietà di Lika Electronic s.r.l. e non possono essere riprodotte né interamente né parzialmente senza una preventiva autorizzazione scritta di Lika Electronic s.r.l. La traduzione, la riproduzione e la modifica totale o parziale (incluse le copie fotostatiche, i film, i microfilm e ogni altro mezzo di riproduzione) sono vietate senza l'autorizzazione scritta di Lika Electronic s.r.l.

Le informazioni contenute nel presente manuale sono soggette a modifica senza preavviso e non devono essere in alcun modo ritenute vincolanti per Lika Electronic s.r.l. Lika Electronic s.r.l. si riserva il diritto di apportare delle modifiche al presente testo in qualunque momento e senza nessun obbligo di informazione a terzi.

Questo manuale è periodicamente rivisto e aggiornato. All'occorrenza si consiglia di verificare l'esistenza di aggiornamenti o nuove edizioni di questo manuale sul sito istituzionale di Lika Electronic s.r.l. Lika Electronic s.r.l. non si assume alcuna responsabilità per eventuali errori o omissioni riscontrabili in questo documento. Valutazioni critiche di questo manuale da parte degli utilizzatori sono gradite. Ogni eventuale osservazione ci è utile nella stesura della futura documentazione, al fine di redigere un prodotto che sia quanto più chiaro, utile e completo possibile. Per inviarci i Vostri commenti, suggerimenti e critiche mandate una e-mail all'indirizzo [info@lika.it](mailto:info@lika.it).

EtherCAT® è un marchio registrato e una tecnologia brevettata, concessi in licenza da Beckhoff Automation GmbH, Germania

The logo for Lika, consisting of the word "lika" in a bold, lowercase, sans-serif font. The letter "i" has a dot above it.

# Indice generale

Manuale d'uso.....	1
Indice generale.....	3
Indice analitico.....	9
Convenzioni grafiche e iconografiche.....	11
Informazioni preliminari.....	12
Glossario dei termini EtherCAT.....	14
Glossario dei termini MODBUS.....	22
<b>1 Norma di sicurezza.....</b>	<b>26</b>
1.1 Sicurezza.....	26
1.2 Avvertenze elettriche.....	26
1.3 Avvertenze meccaniche.....	27
<b>2 Identificazione.....</b>	<b>28</b>
<b>3 Installazione meccanica.....</b>	<b>29</b>
<b>4 Connessioni elettriche.....</b>	<b>33</b>
4.1 Collegamento messa a terra (Figura 1 e Figura 2).....	34
4.2 Connettori (Figura 4 e Figura 5).....	34
4.2.1 Connettore alimentazione.....	35
4.2.2 Connettori interfaccia EtherCAT (PORT IN e PORT OUT).....	35
4.2.3 Configurazione di rete: topologie, cavi, hub, switch - Raccomandazioni.....	36
4.2.4 Impostazione indirizzo.....	37
4.2.5 Terminazione di linea.....	37
4.2.6 Ingressi / uscita + porta di servizio RS-232 MODBUS.....	37
4.3 Led di diagnostica (Figura 4 e Figura 6).....	38
4.4 Tappo a vite per accesso interno (Figura 4 e Figura 7).....	41
4.5 Pulsanti Preset / Jog (Figura 8).....	42
4.5.1 Pulsanti JOG + e JOG - (Figura 8).....	43
4.5.2 Pulsante PRESET (Figura 8).....	43
<b>5 Quick reference.....</b>	<b>45</b>
<b>6 Funzioni.....</b>	<b>47</b>
6.1 Principio di funzionamento.....	47
6.2 Tipi di movimento: jog e posizione.....	48
Jog: controllo di velocità.....	48
Posizione: controllo di posizione e velocità.....	49
6.3 Ingressi e uscita digitali.....	50
6.4 Distance per revolution, Preset, Max delta pos / Positive delta e Max delta neg / Negative delta.....	51
<b>7 Interfaccia EtherCAT®.....</b>	<b>54</b>
7.1 Configurazione con ambiente di sviluppo TwinCAT di Beckhoff.....	54
7.1.1 Impostazione della scheda di rete.....	54
7.1.2 Aggiunta di nuovi moduli I/O (Box).....	57
7.2 Impostazione della modalità di comunicazione.....	61
7.2.1 Modalità di comunicazione FreeRun.....	61
7.3 Process Data Object.....	64
7.4 Dizionario oggetti COE.....	65
7.5 Dati Online.....	67
7.6 Nozioni di base sul protocollo EtherCAT®.....	68

7.6.1	Trasmissione dati.....	69
7.6.2	Modello a livelli ISO/OSI.....	70
7.6.3	Topologia.....	70
7.6.4	Terminazione di linea.....	71
7.6.5	Indirizzo dispositivo.....	72
7.6.6	Modalità di comunicazione.....	73
	FreeRun.....	73
	Sincrono con SM3.....	74
	Sincrono con DC SYNC0.....	74
7.6.7	EtherCAT State Machine (ESM).....	75
7.6.8	Configurazione dello Slave.....	76
7.6.9	Temporizzazione e sincronizzazione.....	77
	Sync Manager.....	77
	Buffered Mode (3-Buffer Mode).....	77
	Mailbox Mode (1-Buffer Mode).....	78
7.6.10	Watchdog.....	78
7.7	CANopen Over EtherCAT (CoE).....	80
	7.7.1 File XML.....	80
	7.7.2 Tipi di messaggi.....	81
	7.7.3 Process Data Object (PDO).....	81
	7.7.4 Service Data Object (SDO).....	81
7.8	Dizionario oggetti.....	82
	7.8.1 Oggetti della Communication Profile Area (DS 301).....	84
	<b>1000-00 Device type</b> .....	84
	<b>1008-00 Manufacturer Device Name</b> .....	84
	<b>1009-00 Manufacturer Hardware version</b> .....	84
	<b>100A-00 Manufacturer Software version</b> .....	84
	<b>1018 Identity Object</b> .....	84
	01 Vendor ID.....	84
	02 Product code.....	84
	03 Revision number.....	85
	04 Serial number.....	85
	<b>1600 Receive PDO Mapping</b> .....	85
	01 Mapped Object 001.....	85
	02 Mapped Object 002.....	86
	<b>1A00 Transmit PDO Mapping</b> .....	86
	01 Mapped Object 001.....	86
	02 Mapped Object 002.....	86
	<b>1C00 Sync Manager Communication Type</b> .....	87
	01 SM MailBox Receive (SM0).....	87
	02 SM MailBox Send (SM1).....	87
	03 SM PDO output (SM2).....	87
	04 SM PDO input (SM3).....	87
	<b>1C12-00 Sync Manager RxPDO assign</b> .....	87
	01 SubIndex 001.....	87
	<b>1C13-01 Sync Manager TxPDO assign</b> .....	87
	01 SubIndex 001.....	88
	<b>1C32 Output Sync Manager Parameter</b> .....	88
	01 Synchronization Type.....	88
	02 Cycle time.....	88
	03 Shift Time.....	88

04 Synchronization Types supported.....	88
05 Minimum Cycle Time.....	88
06 Calc and Copy Time.....	88
09 Delay Time.....	89
0C Cycle Time Too Small.....	89
<b>1C33 Input Sync Manager Parameter.....</b>	<b>89</b>
01 Synchronization Type.....	89
02 Cycle time.....	89
03 Shift Time.....	89
04 Synchronization Types supported.....	89
05 Minimum Cycle Time.....	90
06 Calc and Copy Time.....	90
0C Cycle Time Too Small.....	90
7.8.2 Oggetti della Manufacturer Specific Profile Area.....	91
<b>2101-00 HMS Serial Number.....</b>	<b>91</b>
<b>2102-00 HMS_FW_Major.....</b>	<b>91</b>
<b>2103-00 HMS_FW_Minor.....</b>	<b>91</b>
<b>2104-00 HMS_FW_Build.....</b>	<b>91</b>
<b>2105-00 Position Offset.....</b>	<b>91</b>
<b>2106-00 Real Speed [rpm].....</b>	<b>91</b>
<b>2107-00 Electronics Temperature [°C].....</b>	<b>92</b>
<b>2108-00 Motor Temperature [°C].....</b>	<b>92</b>
<b>2109-00 Real Current.....</b>	<b>92</b>
<b>210A-00 Following error [pulse].....</b>	<b>92</b>
<b>210B-00 Pos. Limit Switch [pulse].....</b>	<b>92</b>
<b>210C-00 Neg. Limit Switch [pulse].....</b>	<b>93</b>
<b>210D Parameter Error List.....</b>	<b>93</b>
<b>210E Alarms List.....</b>	<b>94</b>
Machine data not valid.....	94
Flash memory error.....	94
Counting error.....	94
Following error.....	94
Axis not synchronized.....	94
Target not valid.....	94
Emergency.....	94
Overcurrent.....	94
Electronics Overtemperature.....	95
Motor Overtemperature.....	95
Undervoltage.....	95
Watch dog.....	95
Hall sequence.....	95
Overvoltage.....	96
<b>2200 Control Word.....</b>	<b>96</b>
Jog +.....	96
Jog -.....	97
Stop.....	97
Alarm reset.....	97
Incremental jog.....	98
Start.....	98
Emergency.....	98
Save parameters.....	99

Load default parameters.....	99
Setting the preset.....	100
Axis torque.....	100
OUT 1.....	100
Brake disabled.....	100
<b>2201-00 Target Position.....</b>	<b>101</b>
<b>2202 Status Word.....</b>	<b>102</b>
Axis in position.....	102
Drive enabled.....	102
SW limit switch +.....	103
SW limit switch -.....	103
Alarm.....	103
Axis running.....	103
Executing a command.....	103
Target position reached.....	103
Button 1 Jog +.....	103
Button 2 Jog -.....	104
Button 3 Preset.....	104
PWM saturation.....	104
IN 1.....	104
IN 2.....	105
IN 3.....	105
<b>2203-00 Real Position.....</b>	<b>105</b>
<b>2204-00 Distance per revolution.....</b>	<b>105</b>
<b>2205-00 Position tolerance.....</b>	<b>106</b>
<b>2206-00 Settling time.....</b>	<b>106</b>
<b>2207-00 Max following error.....</b>	<b>106</b>
<b>2208-00 Proportional gain.....</b>	<b>107</b>
<b>2209-00 Integral gain.....</b>	<b>107</b>
<b>220A-00 Acceleration.....</b>	<b>107</b>
<b>220B-00 Deceleration.....</b>	<b>107</b>
<b>220C-00 Max delta pos.....</b>	<b>107</b>
<b>220D-00 Max delta neg.....</b>	<b>108</b>
<b>220E-00 Jog speed.....</b>	<b>110</b>
<b>220F-00 Work speed.....</b>	<b>110</b>
<b>2210-00 Count direction.....</b>	<b>111</b>
<b>2211-00 Preset.....</b>	<b>111</b>
<b>2212-00 Step jog.....</b>	<b>112</b>
7.9 SDO Abort code.....	114
7.10 Emergency Error Code.....	114
7.11 AL Status Error Code.....	114
<b>8 Interfaccia MODBUS®.....</b>	<b>115</b>
8.1 Configurazione del dispositivo mediante software Lika.....	115
8.2 Pagina "Configurazione seriale".....	117
8.3 Pagina "Principale".....	119
8.4 Comandi MODBUS.....	121
8.5 Box "Stato".....	123
8.6 Pagina "Allarmi e stati".....	124
8.7 Pagina "Programming firmware".....	126
8.8 Pagina "Parametri".....	129

8.9 Pagina "Programma".....	132
8.10 Principi guida del protocollo MODBUS Master / Slave.....	135
8.11 Frame MODBUS.....	136
8.12 Modalità di trasmissione.....	137
8.12.1 Modalità di trasmissione RTU.....	137
8.13 Codici funzione.....	140
8.13.1 Codici funzione implementati.....	140
<b>03 Read Holding Register</b> .....	140
<b>04 Read Input Register</b> .....	142
<b>06 Write Single Register</b> .....	144
<b>16 Write Multiple Register</b> .....	146
8.14 Parametri di programmazione.....	150
8.14.1 Parametri Holding Register.....	150
<b>Distance per revolution [0x00]</b> .....	151
<b>Position window [0x01]</b> .....	152
<b>Position window time [0x02]</b> .....	152
<b>Max following error [0x03-0x04]</b> .....	152
<b>Kp position loop [0x05]</b> .....	152
<b>Ki position loop [0x06]</b> .....	153
<b>Acceleration [0x07]</b> .....	153
<b>Deceleration [0x08]</b> .....	153
<b>Positive delta [0x09-0x0A]</b> .....	153
<b>Negative delta [0x0B-0x0C]</b> .....	154
<b>Jog speed [0x0D]</b> .....	155
<b>Work speed [0x0E]</b> .....	156
<b>Code sequence [0x0F]</b> .....	156
<b>Offset [0x10-0x11]</b> .....	156
<b>Preset [0x12-0x13]</b> .....	157
<b>Jog step length [0x14]</b> .....	157
<b>Extra commands register [0x29]</b> .....	158
Control by PC.....	158
<b>Control Word [0x2A]</b> .....	158
Jog +.....	158
Jog -.....	159
Stop.....	159
Alarm reset.....	159
Incremental jog.....	160
Start.....	160
Emergency.....	160
Watch dog enable.....	161
Save parameters.....	161
Load default parameters.....	161
Setting the preset.....	162
Axis torque.....	162
OUT 1.....	162
Brake disabled.....	162
<b>Target position [0x2B-0x2C]</b> .....	163
8.14.2 Parametri Input Register.....	165
<b>Alarms register [0x00]</b> .....	165
Machine data not valid.....	165

Flash memory error.....	165
Counting error.....	165
Following error.....	165
Axis not synchronized.....	165
Target not valid.....	166
Emergency.....	166
Overcurrent.....	166
Electronics Overtemperature.....	166
Motor Overtemperature.....	166
Undervoltage.....	166
Watch dog.....	166
Hall sequence.....	167
Overvoltage.....	167
<b>Status word [0x01]</b> .....	167
Axis in position.....	167
Drive enabled.....	168
SW limit switch +.....	168
SW limit switch -.....	168
Alarm.....	168
Axis running.....	168
Executing a command.....	168
Target position reached.....	169
Button 1 Jog +.....	169
Button 2 Jog -.....	169
Button 3 Preset.....	169
PWM saturation.....	170
IN 1.....	170
IN 2.....	170
IN 3.....	170
<b>Current position [0x02-0x03]</b> .....	170
<b>Current velocity [0x04]</b> .....	170
<b>Position following error [0x05-0x06]</b> .....	171
<b>Temperature value [0x07]</b> .....	171
<b>Wrong parameters list [0x08-0x09]</b> .....	171
<b>Motor voltage [0x0A]</b> .....	172
<b>Current value [0x0B]</b> .....	172
<b>Hall [0x0C]</b> .....	172
<b>Duty cycle [0x0D]</b> .....	172
<b>DIP switch baud rate [0x0E]</b> .....	173
<b>DIP switch node ID [0x0F]</b> .....	173
<b>SW Version [0x10]</b> .....	173
<b>HW Version [0x11]</b> .....	173
8.15 Codici eccezione.....	176
8.16 Esempi di programmazione.....	180
8.16.1 Utilizzo del codice funzione 03 Read Holding Register.....	180
8.16.2 Utilizzo del codice funzione 04 Read Input Register.....	181
8.16.3 Utilizzo del codice funzione 06 Write Single Register.....	183
8.16.4 Utilizzo del codice funzione 16 Write Multiple Register.....	185
<b>9 Lista parametri di default.....</b>	<b>186</b>

# Indice analitico

<b>1</b>	
1000-00 Device type.....	84
1008-00 Manufacturer Device Name.....	84
1009-00 Manufacturer Hardware version.....	84
100A-00 Manufacturer Software version.....	84
1018 Identity Object.....	84
1600 Receive PDO Mapping.....	85
1A00 Transmit PDO Mapping.....	86
1C00 Sync Manager Communication Type.....	87
1C12-00 Sync Manager RxPDO assign.....	87
1C13-01 Sync Manager TxPDO assign.....	87
1C32 Output Sync Manager Parameter.....	88
1C33 Input Sync Manager Parameter.....	89
<b>2</b>	
2101-00 HMS Serial Number.....	91
2102-00 HMS_FW_Major.....	91
2103-00 HMS_FW_Minor.....	91
2104-00 HMS_FW_Build.....	91
2105-00 Position Offset.....	91
2106-00 Real Speed [rpm].....	91
2107-00 Electronics Temperature [°C].....	92
2108-00 Motor Temperature [°C].....	92
2109-00 Real Current.....	92
210A-00 Following error [pulse].....	92
210B-00 Pos. Limit Switch [pulse].....	92
210C-00 Neg. Limit Switch [pulse].....	93
210D Parameter Error List.....	93
210E Alarms List.....	94
2200 Control Word.....	96
2201-00 Target Position.....	101
2202 Status Word.....	102
2203-00 Real Position.....	105
2204-00 Distance per revolution.....	105
2205-00 Position tolerance.....	106
2206-00 Settling time.....	106
2207-00 Max following error.....	106
2208-00 Proportional gain.....	107
2209-00 Integral gain.....	107
220A-00 Acceleration.....	107
220B-00 Deceleration.....	107
220C-00 Max delta pos.....	107
220D-00 Max delta neg.....	108
220E-00 Jog speed.....	110
220F-00 Work speed.....	110
2210-00 Count direction.....	111
2211-00 Preset.....	111
2212-00 Step jog.....	112
<b>A</b>	
Acceleration [0x07].....	153
Alarm.....	103, 168
Alarm reset.....	97, 159
Alarms register [0x00].....	165
Axis in position.....	102, 167
Axis not synchronized.....	94, 165
Axis running.....	103, 168
Axis torque.....	100, 162
<b>B</b>	
BOOTSTRAP.....	76
Brake disabled.....	100, 162
Button 1 Jog +.....	103, 169
Button 2 Jog -.....	104, 169
Button 3 Preset.....	104, 169
<b>C</b>	
Calc and Copy Time.....	88, 90
Code sequence [0x0F].....	156
Control by PC.....	158
Control Word [0x2A].....	158
Counting error.....	94, 165
Current position [0x02-0x03].....	170
Current value [0x0B].....	172
Current velocity [0x04].....	170
Cycle time.....	88 e seg.
Cycle Time Too Small.....	89 e seg.
<b>D</b>	
Deceleration [0x08].....	153
Delay Time.....	89
DIP switch baud rate [0x0E].....	173
DIP switch node ID [0x0F].....	173
Distance per revolution [0x00].....	151
Drive enabled.....	102, 168
Duty cycle [0x0D].....	172
<b>E</b>	
Electronics Overtemperature.....	95, 166
Emergency.....	94, 98, 160, 166
Executing a command.....	103, 168
Extra commands register [0x29].....	158
<b>F</b>	
Flash memory error.....	94, 165
Following error.....	94, 165
<b>H</b>	
Hall [0x0C].....	172
Hall sequence.....	95, 167

HW Version [0x11].....	173	PRE-OPERATIONAL.....	76
<b>I</b>		Preset [0x12-0x13].....	157
IN 1.....	104, 170	Product code.....	84
IN 2.....	105, 170	PWM saturation.....	104, 170
IN 3.....	105, 170	<b>R</b>	
Incremental jog.....	98, 160	Revision number.....	85
INIT.....	76	<b>S</b>	
<b>J</b>		SAFE-OPERATIONAL.....	76
Jog -.....	97, 159	Save parameters.....	99, 161
Jog +.....	96, 158	Serial number.....	85
Jog speed [0x0D].....	155	Setting the preset.....	100, 162
Jog step length [0x14].....	157	Shift Time.....	88 e seg.
<b>K</b>		SM MailBox Receive (SM0).....	87
Ki position loop [0x06].....	153	SM MailBox Send (SM1).....	87
Kp position loop [0x05].....	152	SM PDO input (SM3).....	87
<b>L</b>		SM PDO output (SM2).....	87
Load default parameters.....	99, 161	Start.....	98, 160
<b>M</b>		Status word [0x01].....	167
Machine data not valid.....	94, 165	Stop.....	97, 159
Mapped Object 001.....	85 e seg.	SubIndex 001.....	87 e seg.
Mapped Object 002.....	86	SW limit switch -.....	103, 168
Max following error [0x03-0x04].....	152	SW limit switch +.....	103, 168
Minimum Cycle Time.....	88, 90	SW Version [0x10].....	173
Motor Overtemperature.....	95, 166	Synchronization Type.....	88 e seg.
Motor voltage [0x0A].....	172	Synchronization Types supported.....	88 e seg.
<b>N</b>		<b>T</b>	
Negative delta [0x0B-0x0C].....	154	Target not valid.....	94, 166
<b>O</b>		Target position [0x2B-0x2C].....	163
Offset [0x10-0x11].....	156	Target position reached.....	103, 169
OPERATIONAL.....	76	Temperature value [0x07].....	171
OUT 1.....	100, 162	<b>U</b>	
Overcurrent.....	94, 166	Undervoltage.....	95, 166
Overvoltage.....	96, 167	<b>V</b>	
<b>P</b>		Vendor ID.....	84
Position following error [0x05-0x06].....	171	<b>W</b>	
Position window [0x01].....	152	Watch dog.....	95, 166
Position window time [0x02].....	152	Watch dog enable.....	161
Positive delta [0x09-0x0A].....	153	Work speed [0x0E].....	156
		Wrong parameters list [0x08-0x09].....	171

# Convenzioni grafiche e iconografiche

Per rendere più agevole la lettura di questo testo sono state adottate alcune convenzioni grafiche e iconografiche. In particolare:

- i parametri e gli oggetti sia propri dell'interfaccia che del dispositivo Lika sono evidenziati in **VERDE**;
- gli allarmi sono evidenziati in **ROSSO**;
- gli stati sono evidenziati in **FUCSIA**.

Scorrendo il testo sarà inoltre possibile imbattersi in alcune icone che evidenziano porzioni di testo di particolare interesse o rilevanza. Talora esse possono contenere prescrizioni di sicurezza atte a richiamare l'attenzione sui rischi potenziali legati all'utilizzo del dispositivo. Si raccomanda di seguire attentamente le prescrizioni elencate nel presente manuale al fine di salvaguardare la sicurezza dell'utilizzatore oltre che le performance del dispositivo. I simboli utilizzati nel presente manuale sono i seguenti:

	Questa icona, accompagnata dal termine <b>ATTENZIONE</b> , evidenzia le porzioni di testo che contengono informazioni della massima importanza per l'operatore concernenti l'uso corretto e sicuro del dispositivo. Le istruzioni accompagnate da questo simbolo devono essere seguite scrupolosamente dall'operatore. La loro mancata osservanza può generare malfunzionamenti e danni sia al dispositivo che alla macchina sulla quale il dispositivo è installato e procurare lesioni anche gravi agli operatori al lavoro in prossimità.
	Questa icona, accompagnata dal termine <b>NOTA</b> , evidenzia le porzioni di testo che contengono notazioni importanti ai fini di un uso corretto e performante del dispositivo. Le istruzioni accompagnate da questo simbolo devono essere tenute bene in considerazione da parte dell'operatore. La loro mancata osservanza può procurare l'esecuzione di procedure errate di settaggio da parte dell'utilizzatore e conseguentemente un funzionamento errato o inadeguato del dispositivo.
	Questa icona evidenzia le porzioni di testo che contengono suggerimenti utili per agevolare l'operatore nel settaggio e l'ottimizzazione del dispositivo. Talora il simbolo è accompagnato dal termine <b>ESEMPIO</b> quando le istruzioni di impostazione dei parametri siano seguite da esemplificazioni che ne chiarifichino l'utilizzo.

# Informazioni preliminari

Questo manuale ha lo scopo di descrivere le caratteristiche tecniche, l'installazione e l'utilizzo corretto e sicuro degli **attuatori rotativi DRIVECOD modelli RD1A e RD12A con interfaccia EtherCAT**.

Le unità RD1A e RD12A sono dispositivi di posizionamento che integrano in uno un motore brushless con riduttore, un azionamento, un encoder assoluto multigiro e la logica per il posizionamento. Gli attuatori rotativi RD1A e RD12A trovano impiego in una varietà di settori industriali per svolgere funzioni di posizionamento e cambio formati. Tipiche applicazioni includono, tra le tante, linee di confezionamento, industria alimentare e farmaceutica, macchinari per la lavorazione del legno e del metallo, macchine per la carta, linee di trasporto e imballaggio, piegatrici, impianti di riempimento e imbottigliamento, macchine da stampa, cambio formati, cambio utensili, battute mobili, posizionamenti mandrino, movimentazione ventose.

La versione RD12A si differenzia dalla versione RD1A per la presenza di un freno di stazionamento integrato. Il freno ha lo scopo di inibire la possibilità di qualsiasi movimento dell'asse all'arresto del dispositivo.

Gli attuatori rotativi RD1A e RD12A sono disponibili con le seguenti interfacce:

- RD1xA-x-xxx-**CB**-... = interfaccia CANopen DS301;
- RD1xA-x-xxx-**EC**-... = interfaccia EtherCAT;
- RD1xA-x-xxx-**MB**-... = interfaccia MODBUS RTU (RS-485);
- RD1xA-x-xxx-**PB**-... = interfaccia Profibus-DP;
- RD1xA-x-xxx-**PL**-... = interfaccia POWERLINK;
- RD1xA-x-xxx-**PT**-... = interfaccia Profinet.

Il presente manuale descrive il modello con interfaccia EtherCAT. Per informazioni sugli attuatori che integrano altri bus di campo / interfacce Ethernet, riferirsi alla specifica documentazione.

Nella versione MODBUS la parametrizzazione e la messa in funzione possono essere effettuate tramite un software progettato da Lika Electronic e compreso nella fornitura che consente l'impostazione dei parametri di lavoro dell'unità DRIVECOD, l'attivazione e il controllo dei movimenti programmati e il monitoraggio del funzionamento da qualsiasi PC con sistema operativo Windows (Windows XP o successivo). Nella versione EtherCAT la parametrizzazione e la messa in funzione possono essere effettuate utilizzando lo stesso software, tramite un'**interfaccia seriale di servizio RS-232 in protocollo MODBUS**.

Per una più agevole consultazione questo manuale è diviso in tre parti.

Nella prima parte sono fornite le informazioni generali riguardanti l'unità di posizionamento comprendenti le norme di sicurezza, le istruzioni di montaggio meccanico e le prescrizioni relative alle connessioni elettriche, nonché ulteriori informazioni sul funzionamento e la corretta messa a punto del dispositivo.

Nella seconda parte, intitolata **Interfaccia EtherCAT**, sono fornite tutte le informazioni sia generali che specifiche relative all'interfaccia EtherCAT. In questa sezione sono descritte le caratteristiche dell'interfaccia e gli oggetti EtherCAT che l'unità implementa.

Nella terza parte infine, intitolata **Interfaccia MODBUS**, sono fornite tutte le informazioni sia generali che specifiche relative all'interfaccia MODBUS. Come detto in precedenza, la versione EtherCAT è dotata di un'interfaccia seriale di servizio RS-232 in protocollo MODBUS. Tramite un software specifico progettato da Lika Electronic e compreso nella fornitura essa permette la parametrizzazione e la messa a

punto del dispositivo prima dell'installazione in una rete EtherCAT. Nella sezione **Interfaccia MODBUS** sono descritte le caratteristiche dell'interfaccia e i registri MODBUS che l'unità implementa.

# Glossario dei termini EtherCAT

EtherCAT, come molte altre interfacce di collegamento in rete, si avvale di una terminologia specifica. La tabella qui sotto contiene alcuni dei termini tecnici che sono utilizzati in questa guida per descrivere l'interfaccia EtherCAT. Sono elencati in ordine alfabetico.

<b>Acknowledge telegram (AT)</b>	Telegramma in cui ogni Slave inserisce i propri dati.
<b>Algoritmo</b>	Sequenza finita e completamente determinata di operazioni grazie alle quali i valori dei dati in uscita possono essere calcolati a partire dai dati in ingresso.
<b>Application class</b>	Configurazione di un Oggetto Drive tramite un set di oggetti funzionali supportati da telegrammi standard.
<b>Application mode</b>	Tipo di applicazione che può essere richiesto da un PDS.
<b>Application object</b>	Classi di oggetti multipli che gestiscono e forniscono uno scambio di messaggi run time attraverso la rete nel dispositivo.
<b>Application process</b>	Parte di una applicazione distribuita in rete, che si trova in un dispositivo e indirizzata in maniera non ambigua.
<b>Application relationship</b>	Associazione cooperativa tra due o più application-entity-invocation al fine di scambiare informazioni e coordinare operazioni congiunte. La relazione è attivata o da uno scambio di application-protocol-data-unit o come risultato di attività di preconfigurazione.
<b>Applicazione</b>	Funzione o struttura dati per la quale si utilizzano o producono dati. Elemento software funzionale specifico per la soluzione di un problema di misura e controllo in ambito industriale e di processo.
<b>Asse</b>	Elemento logico all'interno di un sistema di automazione (per esempio un sistema di motion control) che rappresenta una qualche forma di movimento.
<b>Attributo</b>	Descrizione di una caratteristica o peculiarità visibile esternamente di un oggetto, proprietà o caratteristica di un'entità. Gli attributi di un oggetto contengono informazioni sulle porzioni variabili di un oggetto. Tipicamente, forniscono informazioni di stato e gestiscono il funzionamento di un oggetto. Gli attributi possono anche determinare il comportamento di un oggetto. Gli attributi si dividono in class attribute e instance attribute.
<b>Behaviour</b>	Indicazione della risposta di un oggetto a un particolare evento
<b>Bit</b>	Unità di informazione che consiste di un 1 o di uno 0. E' l'unità di dato minima che può essere trasmessa.

<b>Canale</b>	Rappresentazione di un singolo oggetto di gestione fisica o logica di uno Slave per il controllo della trasmissione dei dati.
<b>CANopen</b>	Protocollo del livello di applicazione come definito in EN 50325-4.
<b>Ciclico</b>	Eventi che si ripetono in maniera regolare e ripetitiva.
<b>Ciclo di comunicazione</b>	Accumulazione di tutti i telegrammi tra due telegrammi di sincronizzazione Master.
<b>Ciclo operativo</b>	Periodo del control loop all'interno di un drive o di un'unità di controllo.
<b>CIP™</b>	Common Industrial Protocol (si veda IEC 61158 Tipo 2, IEC 61784-1 e IEC 61784-2 CPF2).
<b>Classe</b>	Descrizione di un set di oggetti che condividono gli stessi attributi, operazioni, metodi, relazioni e semantica.
<b>Classe di applicazione (Application class)</b>	Configurazione di un Oggetto Drive tramite un set di oggetti funzionali supportati da telegrammi standard.
<b>Client</b>	Oggetto che utilizza i servizi di un altro oggetto (Server) per eseguire un task. Iniziatore di un messaggio al quale il Server risponde.
<b>Comandi</b>	Set di comandi da un programma di controllo applicazione a un PDS per controllare il comportamento di un PDS o di un elemento funzionale del PDS.
<b>Communication cycle</b>	Accumulazione di tutti i telegrammi tra due telegrammi di sincronizzazione Master.
<b>Communication object</b>	Componente che gestisce e fornisce uno scambio run time di messaggi attraverso la rete.
<b>Comportamento (behaviour)</b>	Indicazione della risposta di un oggetto a un particolare evento
<b>Connessione</b>	Collegamento logico tra due oggetti applicazione all'interno dello stesso o di differenti dispositivi.
<b>Consume</b>	Azione di ricevimento di dati da un fornitore (provider).
<b>Consumer</b>	Nodo o sink che riceve dati da un fornitore (provider).
<b>Control device</b>	Unità fisica che contiene – in un modulo/sottogruppo o un dispositivo – un programma applicativo per controllare il PDS.
<b>Control unit</b>	Control device, dispositivo di controllo.
<b>Control word</b>	Due byte adiacenti all'interno del Master data telegram che contengono comandi per il drive destinatario.
<b>Controller</b>	Dispositivo di controllo che è associato a uno o più drive (assi), l'host del sistema di automazione complessivo.
<b>Controllo</b>	Azione mirata su o in un processo per soddisfare obiettivi indicati.
<b>Conveyance path</b>	Flusso unidirezionale delle APDU attraverso una relazione applicativa.

<b>Cycle time</b>	Periodo temporale tra due eventi consecutivi ricorrenti ciclicamente.
<b>Data consistency</b>	Mezzi per la trasmissione e l'accesso coerenti di input -o output- data object tra Client e Server e all'interno di essi.
<b>Data exchange</b>	Su richiesta; trasmissione non ciclica (service channel).
<b>Data type</b>	Relazione tra valori e codifica per dati di quel tipo secondo le definizioni di IEC 61131-3. Set di valori abbinato a un set di operazioni permesse.
<b>Data type object</b>	Impostazione nel dizionario oggetti indicante il tipo di dato.
<b>Dati</b>	Termine generico usato per riferirsi a qualsiasi informazione trasmessa in un bus.
<b>Dati ciclici</b>	Parte di un telegramma che non muta il proprio significato durante l'operazione ciclica dell'interfaccia. I dati real time ad alta priorità trasmessi da una connessione CIP Motion su base periodica.
<b>Dati di processo</b>	Insieme di oggetti applicativi che vengono trasmessi in maniera ciclica e non ciclica con funzioni di misura e controllo.
<b>Default gateway</b>	Dispositivo con almeno due interfacce in due diverse sottoreti IP che agisce come router per la sottorete.
<b>Device profile</b>	Insieme di informazioni e funzionalità correlate a un dispositivo che permettono coerenza tra dispositivi simili della stessa tipologia. Descrizione di un dispositivo sulla scorta dei suoi parametri e del suo funzionamento in conformità a un modello che descrive i dati del dispositivo e il comportamento visto attraverso la rete, indipendente da ogni tecnologia di rete.
<b>Diagnosis information</b>	Tutti i dati disponibili a un Server per scopi di manutenzione.
<b>Dispositivo</b>	Dispositivo di campo (field device). Entità fisica indipendente collegata in rete di un sistema di automazione industriale capace di eseguire funzioni stabilite in un particolare contesto e delimitato dalle sue interfacce. Entità che controlla mediante l'esecuzione e/o la lettura di funzioni e si interfaccia con altre entità simili all'interno di un sistema di automazione. Entità fisica connessa a un bus di campo composta da almeno un elemento di comunicazione (elemento di rete) e che può avere un elemento di controllo e/o un elemento finale (trasduttore, attuatore, ecc.).
<b>Dispositivo di controllo</b>	Unità fisica che contiene - in un modulo/sottogruppo o un dispositivo - un programma applicativo per controllare il PDS.
<b>Distributed clocks</b>	Metodo per sincronizzare gli Slave e mantenere una base temporale globale.
<b>Dizionario oggetti</b>	Struttura dati con indirizzamento mediante indice e sub-indice che contiene la descrizione dei data type object, dei communication object e degli application object.

	Lista di oggetti con indice unico a 16 bit e sub-indice a 8 bit come definito in EN 50325-4.
<b>DL</b>	Data-link-layer.
<b>DLPDU</b>	Data-link-protocol-data-unit.
<b>Drive Object</b>	Elemento funzionale di una Drive Unit.
<b>Drive Unit</b>	Dispositivo logico che comprende tutti gli elementi funzionali a una unità di processamento centrale.
<b>Elemento funzionale</b>	Entità di software o software combinato con hardware, capace di eseguire una funzione specificata di un dispositivo.
<b>Error class</b>	Raggruppamento generale per definizioni di errore affini e corrispondenti codici di errore.
<b>Error code</b>	Identificazione di un tipo specifico di errore all'interno di una classe di errore (error class).
<b>Errore</b>	Discrepanza tra un valore o una condizione calcolata, osservata o misurata e il valore o la condizione stabilita o teoricamente corretta.
<b>EtherCAT State Machine</b>	Lo Slave EtherCAT è una macchina a stati; la comunicazione e le caratteristiche di funzionamento dipendono dallo stato corrente del dispositivo.
<b>Event data</b>	Dato real time a media priorità che è trasferito da una connessione CIP Motion solo dopo l'occorrenza di evento specificato.
<b>Evento</b>	Occorrenza di un cambio di condizioni.
<b>Feed forward</b>	Valore di comando usato per compensare il ritardo nel control loop.
<b>Feedback variable</b>	Variabile che rappresenta una variabile controllata e che viene ritornata a un elemento di comparazione.
<b>Fieldbus memory management unit</b>	Funzione che stabilisce una o più corrispondenze tra gli indirizzi logici e la memoria fisica.
<b>Fieldbus memory management unit entity</b>	Elemento singolo della fieldbus memory management unit: una corrispondenza tra uno spazio indirizzo logico coerente e una locazione di memoria fisica coerente.
<b>Frame</b>	Sinonimo di DLPDU.
<b>FreeRun</b>	Modo di comunicazione asincrono.
<b>Full Slave</b>	Dispositivo Slave che supporta l'indirizzamento di dati sia fisico che logico.
<b>HMI</b>	Human Machine Interface.
<b>Host</b>	Dispositivo che copre la funzionalità automation di un dispositivo di automazione.
<b>I/O data</b>	Dati d'ingresso (input) e d'uscita (output) che necessitano tipicamente di un aggiornamento su base regolare (per esempio un cambio di stato periodico), come comandi, set-

	point, stati e valori correnti.
<b>Identification number (IDN)</b>	Designazione di dati operativi sotto il quale è protetto un blocco dati con il proprio attributo, nome, unità, valori di input minimo e massimo e i dati.
<b>Index</b>	Indirizzo di un oggetto in un processo applicativo.
<b>Input data</b>	Dati trasmessi da una sorgente esterna a un dispositivo, risorsa o elemento funzionale.
<b>Interfaccia</b>	Limite condiviso tra due entità definite come appropriate da caratteristiche funzionali, caratteristiche di segnale o altre caratteristiche.
<b>Little endian</b>	Rappresentazione di dati di campi a più byte dove il byte meno significativo è trasmesso per primo.
<b>Logical power drive system</b>	Modello che include PDS e rete di comunicazione accessibile attraverso un'interfaccia PDS generica.
<b>Mapping</b>	Corrispondenza tra due oggetti in modo che un oggetto sia parte dell'altro oggetto.
<b>Mapping parameter</b>	Set di valori che definiscono la corrispondenza tra gli application object e i process data object.
<b>Marca temporale</b>	Valore temporale del tempo di sistema associato ai dati di connessione CIP Motion che trasmette il tempo assoluto in cui il dato associato è stato acquisito o che può essere utilizzato per determinare quando il dato associato sarà applicato.
<b>Master</b>	Dispositivo che controlla la trasmissione dati nella rete e stabilisce l'accesso al mezzo fisico degli Slave mediante messaggi e che costituisce l'interfaccia al sistema di controllo. Nodo che attribuisce agli altri nodi il diritto di trasmettere.
<b>Master data telegram (MDT)</b>	Telegramma nel quale il Master inserisce i propri dati.
<b>Medium</b>	Cavo, fibra ottica o altro mezzo mediante i quali vengono trasmessi i segnali di comunicazione tra due o più punti.
<b>Messaggio</b>	Serie ordinata di ottetti previsti per l'invio di informazioni. Normalmente utilizzato per inviare informazioni tra peer a livello applicativo.
<b>Modello</b>	Rappresentazione matematica o fisica di un sistema o di un processo, basato con sufficiente precisione su regole note, identificazione e supposizioni stabilite.
<b>Modo operativo</b>	Caratterizzazione del modo e della misura in cui l'operatore umano interviene nell'apparecchiatura di controllo.
<b>Motion</b>	Ciascun aspetto delle dinamiche di un asse.
<b>Motion Axis Object</b>	Oggetto che definisce gli attributi, i servizi e il comportamento di un asse basato su dispositivo motion (o PDS) conformemente alla specifica CIP Motion, inclusi comunicazioni, Device Control, ed elementi FE Basic Drive come definiti in IEC 61800-7.

<b>Nodo</b>	Entità DL singola come essa appare in un collegamento locale. Endpoint di un collegamento in una rete o punto in cui due o più collegamenti si incontrano [derivato da IEC 61158-2].
<b>Oggetto</b>	Rappresentazione astratta di un particolare componente all'interno di un dispositivo. Un oggetto può essere: <ol style="list-style-type: none"> <li>1. una rappresentazione astratta delle capacità di un dispositivo. Gli oggetti si possono comporre di alcuni o tutti i componenti seguenti: <ul style="list-style-type: none"> <li>○ dati (informazioni che cambiano nel tempo);</li> <li>○ configurazione (parametri per il funzionamento);</li> <li>○ metodi (procedure che possono essere eseguite usando dati e configurazione);</li> </ul> </li> <li>2. un insieme di dati (nella forma di variabili) e metodi (procedure) correlati per agire su quei dati che sono stati chiaramente definiti interfaccia e comportamento.</li> </ol>
<b>Oggetto di comunicazione</b>	Componente che gestisce e fornisce uno scambio run time di messaggi attraverso la rete.
<b>Output data</b>	Dati originati in un dispositivo, risorsa o elemento funzionale e trasferiti poi da esso ai sistemi esterni.
<b>P-Device</b>	Dispositivo di campo e l'host per i Drive Object.
<b>Parametro</b>	Dato che rappresenta un'informazione sul dispositivo che può essere letta da o scritta in un dispositivo, per esempio attraverso la rete o un'interfaccia HMI locale.
<b>PDO</b>	Process Data Object.
<b>PDS</b>	Power Drive System.
<b>Process Data Object (PDO)</b>	Oggetto di comunicazione con funzionalità real time. Struttura descritta mediante la mappatura di parametri che contengono una o più entità dati di processo.
<b>Producer</b>	Nodo o sorgente che invia dati a uno o più consumer.
<b>Profilo</b>	Rappresentazione di una interfaccia PDS in base ai suoi parametri, blocchi parametro e funzionamento in conformità con un profilo di comunicazione e un profilo dispositivo.
<b>Profilo dispositivo (device profile)</b>	Insieme di informazioni e funzionalità correlate a un dispositivo che permettono coerenza tra dispositivi simili della stessa tipologia. Descrizione di un dispositivo sulla scorta dei suoi parametri e del suo funzionamento in conformità a un modello che descrive i dati del dispositivo e il comportamento visto attraverso la rete, indipendente da ogni tecnologia di rete.
<b>Protocollo</b>	Convenzione su formato dati, sequenze temporali e correzione di errori nello scambio dati di un sistema di comunicazione.
<b>Reference variable</b>	Variabile di input di un elemento di comparazione in un sistema di controllo che determina il valore desiderato della

	variabile di controllo ed è dedotto dalla variabile di comando.
<b>Rete</b>	Gruppo di nodi collegati mediante un qualche tipo di mezzo di comunicazione, incluso qualsiasi ripetitore, bridge, router e gateway a livello lower layer che si frapponga.
<b>Risorsa</b>	Entità con capacità di processo o informazione.
<b>Scambio dati (Data exchange)</b>	Su richiesta; trasmissione non ciclica (service channel).
<b>Segmento</b>	Insieme di un Master reale con uno o più Slave.
<b>Server</b>	Oggetto che fornisce servizi a un altro oggetto (Client).
<b>Service data</b>	Dati real time a bassa priorità associati a un service message fornito da un controller che vengono trasmessi da una connessione CIP Motion su base periodica.
<b>Servizio</b>	Operazione o funzione che un oggetto o una classe di oggetti esegue su richiesta di un altro oggetto e/o di una classe di oggetti.
<b>Set-point</b>	Valore o variabile usati come dato di output del programma di controllo applicativo per controllare il PDS.
<b>Sincronizzato</b>	Condizione per cui il valore di clock locale nel drive è agganciato al clock Master del tempo di sistema (System Time) distribuito.
<b>Sincronizzazione clock</b>	Rappresentazione di una sequenza di interazioni per sincronizzare i clock di tutti i time receiver a un time Master.
<b>Sincrono con DC SYNC0</b>	In questa modalità i dati sono campionati e successivamente copiati nel buffer Sync Manager in corrispondenza del segnale SYNC0 generato dall'unità di capture/compare dell'ESC.
<b>Sincrono con SM3</b>	In questa modalità i dati sono campionati e successivamente copiati nel buffer Sync Manager non appena i dati precedenti sono stati letti dal Master (evento SM); in questo modo i nuovi valori campionati risultano sincroni con le letture da parte del Master.
<b>Slave</b>	Entità DL con accesso al mezzo fisico solo a seguito di autorizzazioni dello Slave precedente o del Master. Nodo cui il diritto di trasmettere viene assegnato dal Master.
<b>Slave base</b>	Dispositivo Slave che supporta unicamente un indirizzamento fisico dei dati.
<b>Stato</b>	Insieme di informazioni dal PDS al programma di controllo applicativo che riflettono lo stato o la modalità del PDS o di un elemento funzionale del PDS.
<b>Status word</b>	Due byte adiacenti all'interno del drive telegram che contengono l'informazione sullo stato.
<b>Sub-indice</b>	Sotto-indirizzo di un oggetto all'interno del dizionario oggetti.
<b>Supervisore</b>	Dispositivo che gestisce la fornitura di dati di configurazione (gruppi di parametri) e la raccolta di dati di diagnostica da P-Device e/o controller.

<b>Switch</b>	Bridge MAC come definito in IEEE 802.1D.
<b>Sync Manager</b>	Il Sync Manager ha il compito di sincronizzare la trasmissione dati tra Master e Slave e impedisce che la stessa area di memoria venga scritta da eventi differenti. Insieme di elementi di controllo per coordinare l'accesso agli oggetti utilizzati simultaneamente.
<b>Sync manager channel</b>	Elementi di controllo singolo per coordinare l'accesso agli oggetti utilizzati simultaneamente.
<b>System Time</b>	Valore di tempo assoluto come definito nella specifica CIP Sync nel contesto di un sistema temporale distribuito in cui tutti i dispositivi hanno un clock locale che è sincronizzato con un clock Master comune.
<b>Telegramma</b>	Messaggio.
<b>Telegramma standard</b>	Insieme di dati d'ingresso e d'uscita per una modalità applicativa.
<b>Tempo di ciclo (Cycle time)</b>	Periodo temporale tra due eventi consecutivi ricorrenti ciclicamente.
<b>Tempo di sistema (System Time)</b>	Valore di tempo assoluto come definito nella specifica CIP Sync nel contesto di un sistema temporale distribuito in cui tutti i dispositivi hanno un clock locale che è sincronizzato con un clock Master comune.
<b>Time stamp</b>	Valore temporale del tempo di sistema associato ai dati di connessione CIP Motion che trasmette il tempo assoluto in cui il dato associato è stato acquisito o che può essere utilizzato per determinare quando il dato associato sarà applicato.
<b>Tipo</b>	Elemento hardware o software che specifica gli attributi comuni condivisi da tutte le istanze del tipo.
<b>Topologia</b>	Architettura della rete fisica in riferimento alla connessione tra le stazioni del sistema di comunicazione.
<b>Unità di controllo</b>	Control device, dispositivo di controllo.
<b>Use case</b>	Specificazione di classe di una sequenza di azioni, incluse le varianti, che un sistema (o altra entità) può eseguire, interagendo con gli attori del sistema.
<b>Valore attuale</b>	Valore di una variabile in un determinato istante.
<b>Variabile</b>	Entità software che può assumere valori diversi, uno alla volta.

# Glossario dei termini MODBUS

MODBUS, come molte altre interfacce di collegamento in rete, si avvale di una terminologia specifica. La tabella qui sotto contiene alcuni dei termini tecnici che sono utilizzati in questa guida per descrivere l'interfaccia MODBUS. Sono elencati in ordine alfabetico.

<b>Address field</b>	Contiene l'indirizzo dello Slave.
<b>Application Process</b>	L'Application Process è la funzione a livello di Application Layer.
<b>Bus</b>	Un bus è un mezzo di comunicazione che connette diversi nodi. I dati possono essere trasmessi mediante circuiti seriali o paralleli, ossia via conduttori elettrici o fibra ottica.
<b>Client</b>	Un Client è qualsiasi dispositivo di rete che invia richieste dati ai Server. MODBUS segue il modello Client/Server. I Master MODBUS sono intesi come Client, mentre gli Slave MODBUS sono intesi come Server.
<b>Codice funzione</b>	MODBUS è un protocollo richiesta/risposta e offre servizi specificati da codici funzione. Il codice funzione è inviato da un Client a un Server e indica quale tipo di azione il Server deve eseguire. I codici funzione MODBUS sono elementi delle PDU richiesta/risposta di MODBUS. Il campo codice funzione di un'unità dati MODBUS è codificato in un byte. I codici validi sono compresi in un range da 1 a 255 decimale (il range 128 – 255 è riservato e utilizzato per le exception response). Il codice funzione "0" non è valido. Gli attuatori Lika implementano solamente codici funzione pubblici.
<b>Cyclic Redundancy Check (CRC)</b>	Metodo di verifica degli errori nel quale il destinatario del frame calcola un resto dividendo il contenuto del frame per un divisore che sia un numero primo binario e confronta il resto calcolato con il valore memorizzato nel frame e fornito dal nodo mittente.
<b>Data encoding</b>	MODBUS usa una architettura 'big-Endian' per dati e indirizzi. Questo significa che quando viene trasmessa una quantità numerica maggiore di un singolo byte, il byte più significativo viene trasmesso per primo.
<b>Exception code</b>	Codice che gli Slave devono restituire nel caso di problemi. Tutte le eccezioni sono segnalate mediante l'aggiunta di 0x80 al codice funzione della richiesta.
<b>Exception response</b>	MODBUS opera sulla base del comune modello client/server (Master/Slave): il Client (Master) invia un telegramma di richiesta (service request) al Server (Slave), e il Server risponde con un telegramma di risposta. Se il Server non è in grado di processare una richiesta, allora ritornerà un codice funzione di

	errore (exception response) che è il codice funzione originale più 80H (cioè con il bit più significativo impostato a 1).
<b>Function code</b>	<p>MODBUS è un protocollo richiesta/risposta e offre servizi specificati da codici funzione. Il codice funzione è inviato da un Client a un Server e indica quale tipo di azione il Server deve eseguire. I codici funzione MODBUS sono elementi delle PDU richiesta/risposta di MODBUS.</p> <p>Il campo codice funzione di un'unità dati MODBUS è codificato in un byte. I codici validi sono compresi in un range da 1 a 255 decimale (il range 128 – 255 è riservato e utilizzato per le exception response). Il codice funzione "0" non è valido. Gli attuatori Lika implementano solamente codici funzione pubblici.</p>
<b>Holding register</b>	Nel modello di dati MODBUS, un Holding register è il dato in uscita. Un Holding register ha una dimensione di 16 bit, può essere modificato da un programma applicativo e permette un accesso in lettura-scrittura o sola lettura.
<b>IEEE 1588</b>	Questo standard definisce un protocollo che abilita la sincronizzazione di clock in dispositivi collegati in rete distribuita (per esempio collegati via Ethernet).
<b>Input register</b>	Nel modello di dati MODBUS, un Input register è il dato in ingresso. Un Input register ha una dimensione di 16 bit, è fornito da un sistema I/O e permette un accesso in sola lettura.
<b>LRC Checking</b>	Nella modalità ASCII, i messaggi includono un campo di verifica degli errori che si basa su un calcolo Longitudinal Redundancy Checking (LRC) eseguito sul contenuto del messaggio, esclusi gli iniziali 'due punti' la coppia di caratteri finali CRLF. Si applica indipendentemente dal metodo di parity checking usato per i singoli caratteri del messaggio.
<b>Master</b>	Un Master è qualsiasi dispositivo di rete che invia richieste dati agli Slave.
<b>Messaggio</b>	<p>Il servizio di messaggistica MODBUS offre una comunicazione Client/Server tra dispositivi collegati in rete. Il modello Client / Server si basa su quattro tipi di messaggi:</p> <ul style="list-style-type: none"> <li>• MODBUS Request</li> <li>• MODBUS Confirmation</li> <li>• MODBUS Indication</li> <li>• MODBUS Response</li> </ul> <p>I servizi di messaggistica MODBUS sono utilizzati per lo scambio di informazioni.</p>
<b>Modalità di trasmissione ASCII</b>	Quando i dispositivi sono impostati per comunicare su una linea seriale MODBUS in modalità ASCII (American Standard Code for Information Interchange), ciascun byte (8 bit) in un messaggio viene trasmesso sotto forma di due caratteri ASCII. Questa modalità è utilizzata quando il link di comunicazione fisica o le capacità del dispositivo non permettono la conformità con i requisiti del modo RTU per quanto riguarda

	la gestione dei timer.
<b>Modalità di trasmissione RTU</b>	RTU sta per Remote Terminal Unit. Quando i dispositivi comunicano su una linea seriale MODBUS utilizzando la modalità RTU, ciascun byte (8 bit) nel messaggio contiene due caratteri esadecimali a 4 bit. Il vantaggio più significativo di questa modalità risiede nel fatto che la maggiore densità di caratteri permette una migliore gestione dei dati rispetto alla modalità ASCII a parità di baud rate. Ogni messaggio deve essere trasmesso in un flusso continuo di caratteri.
<b>MODBUS Confirmation</b>	Un MODBUS Confirmation è un messaggio di risposta ricevuto sul lato Client.
<b>MODBUS Indication</b>	Un MODBUS Indication è un messaggio di richiesta ricevuto sul lato Server.
<b>MODBUS Request</b>	Un MODBUS Request è un messaggio inviato in rete da un Client per avviare una transazione.
<b>MODBUS Response</b>	Un MODBUS Response è un messaggio di risposta inviato dal Server.
<b>PDU</b>	La Protocol Data Unit (PDU) è il codice funzione e campo dati MODBUS. E' inserito nel pacchetto insieme al campo indirizzo (Address Field) e al CRC (o LRC) a formare la Modbus Serial Line PDU. Il protocollo MODBUS definisce tre PDU, esse sono: <ul style="list-style-type: none"> <li>• MODBUS Request PDU, mb_req_pdu</li> <li>• MODBUS Response PDU, mb_rsp_pdu</li> <li>• MODBUS Exception Response PDU, mb_excep_rsp_pdu</li> </ul>
<b>Protocollo applicativo</b>	MODBUS è un protocollo applicativo o una struttura di trasmissione di messaggi che definisce le regole per organizzare e interpretare i dati indipendentemente dal mezzo fisico di trasmissione.
<b>Read Holding Register (03, 0003hex)</b>	Questo codice funzione è utilizzato per LEGGERE i valori in un blocco di holding register contigui di un dispositivo remoto; in altri termini, permette la lettura dei valori impostati in alcuni parametri di lavoro in successione nel dispositivo.
<b>Read Input Register (04, 0004hex)</b>	Questo codice funzione è utilizzato per LEGGERE registri di input contigui compresi tra 1 e 125 in un dispositivo remoto; in altri termini permette di leggere alcuni valori di risultato e gli stati / allarmi relativi al dispositivo.
<b>Registro</b>	Le funzioni MODBUS operano sui registri di memoria per configurare, monitorare e controllare gli I/O del dispositivo.
<b>Rete</b>	Una rete è un gruppo di computer in un singolo segmento fisico di rete.
<b>Server</b>	Un Server è ogni programma che attende che gli siano inviate le richieste dati. I Server non avviano i contatti con i Client, ma possono solo dare loro risposte. MODBUS segue il modello Client/Server. I Master MODBUS

	sono intesi come Client, mentre gli Slave MODBUS sono intesi come Server.
<b>Service request</b>	E' la MODBUS Request, ossia il messaggio inviato in rete dal Client per avviare una transazione.
<b>Slave</b>	Uno Slave è ogni programma che attende che gli siano inviate le richieste dati. Gli Slave non avviano i contatti con i Masters, ma possono solo dare loro risposte.
<b>Velocità di trasmissione</b>	Velocità di trasferimento dei dati (in bps).
<b>Write Multiple Register (16, 0010hex)</b>	Questo codice funzione è utilizzato per ASSEGNARE (SCRIVERE) DEI VALORI a un blocco di registri contigui (registri da 1 a 123) in un dispositivo remoto.
<b>Write Single Register (06, 0006hex)</b>	Questo codice funzione è utilizzato per ASSEGNARE (SCRIVERE) UN VALORE a un singolo holding register in un dispositivo remoto.

# 1 Norma di sicurezza



## 1.1 Sicurezza

- Durante l'installazione e l'utilizzo del dispositivo osservare le norme di prevenzione e sicurezza sul lavoro previste nel proprio paese;
- l'installazione e le operazioni di manutenzione devono essere eseguite da personale qualificato, in assenza di tensione e parti meccaniche in movimento;
- utilizzare il dispositivo esclusivamente per la funzione per cui è stato costruito: ogni altro utilizzo potrebbe risultare pericoloso per l'utilizzatore;
- alte correnti, tensioni e parti in movimento possono causare lesioni serie o fatali;
- non utilizzare in ambienti esplosivi o infiammabili;
- il mancato rispetto delle norme di sicurezza o delle avvertenze specificate in questo manuale è considerato una violazione delle norme di sicurezza standard previste dal costruttore o richieste dall'uso per cui lo strumento è destinato;
- Lika Electronic non si assume alcuna responsabilità per eventuali danni o lesioni derivanti dall'inosservanza delle norme di sicurezza da parte dell'utilizzatore.



## 1.2 Avvertenze elettriche

- Effettuare le connessioni elettriche esclusivamente in assenza di tensione;
- rispettare le connessioni riportate nella sezione "Connessioni elettriche" a pagina 33;
- è necessario prevedere l'installazione di un pulsante di emergenza per l'interruzione dell'alimentazione al motore in caso di necessità;
- in conformità alla normativa 2014/30/UE sulla compatibilità elettromagnetica rispettare le seguenti precauzioni:
  - prima di maneggiare e installare il dispositivo, eliminare la presenza di carica elettrostatica dal proprio corpo e dagli utensili che verranno in contatto con il dispositivo;
  - alimentare il dispositivo con tensione stabilizzata e priva di disturbi, se necessario, installare appositi filtri EMC all'ingresso dell'alimentazione;
  - utilizzare sempre cavi schermati e possibilmente "twistati";
  - non usare cavi più lunghi del necessario;
  - evitare di far passare il cavo dei segnali del dispositivo vicino a cavi di potenza;
  - installare il dispositivo il più lontano possibile da eventuali fonti di interferenza o schermarlo in maniera efficace;



- per garantire un funzionamento corretto del dispositivo, evitare l'utilizzo di apparecchiature con forte carica magnetica in prossimità dell'unità;
- collegare la calza del cavo e/o la custodia del connettore e/o il corpo del dispositivo a un buon punto di terra; assicurarsi che il punto di terra sia privo di disturbi. Il collegamento a terra può essere effettuato sul lato dispositivo e/o sul lato utilizzatore; è compito dell'utilizzatore valutare la soluzione migliore da adottare per minimizzare i disturbi.



### 1.3 Avvertenze meccaniche

- Montare il dispositivo rispettando rigorosamente le istruzioni riportate nella sezione "Installazione meccanica" a pagina 29;
- effettuare il montaggio meccanico esclusivamente in assenza di parti meccaniche in movimento;
- non disassemblare il dispositivo;
- non eseguire lavorazioni meccaniche sul dispositivo;
- dispositivo elettronico delicato: maneggiare con cura; evitare urti o forti sollecitazioni sia all'albero che al corpo del dispositivo;
- utilizzare il dispositivo in accordo con le caratteristiche ambientali previste dal costruttore.



#### ATTENZIONE

La taratura dell'unità è stata realizzata testandone il funzionamento a pieno carico; i valori di default impostati si riferiscono pertanto a un dispositivo operante in questa condizione. Essi sono altresì previsti per garantire un funzionamento standard e sicuro del dispositivo, che potrà talora non risultare ottimale né prestazionale. Si badi quindi che nella specifica applicazione può essere consigliabile, se non necessaria, la modifica dei parametri di fabbrica e in particolare dei valori di velocità, accelerazione, decelerazione e guadagno.



#### ATTENZIONE

La tensione controelettrica (back EMF) generata dal motore a seguito di una movimentazione manuale dell'asse forzata dall'esterno può provocare danni irreparabili ai componenti interni.

## 2 Identificazione

Il dispositivo è identificato mediante un **codice di ordinazione** e un **numero di serie** stampati sull'etichetta applicata al dispositivo stesso; i dati sono ripetuti anche nei documenti di trasporto che lo accompagnano. Citare sempre il codice di ordinazione e il numero di serie quando si contatta Lika Electronic per l'acquisto di un ricambio o nella necessità di assistenza tecnica. Per ogni informazione sulle caratteristiche tecniche del dispositivo fare riferimento al catalogo del prodotto.

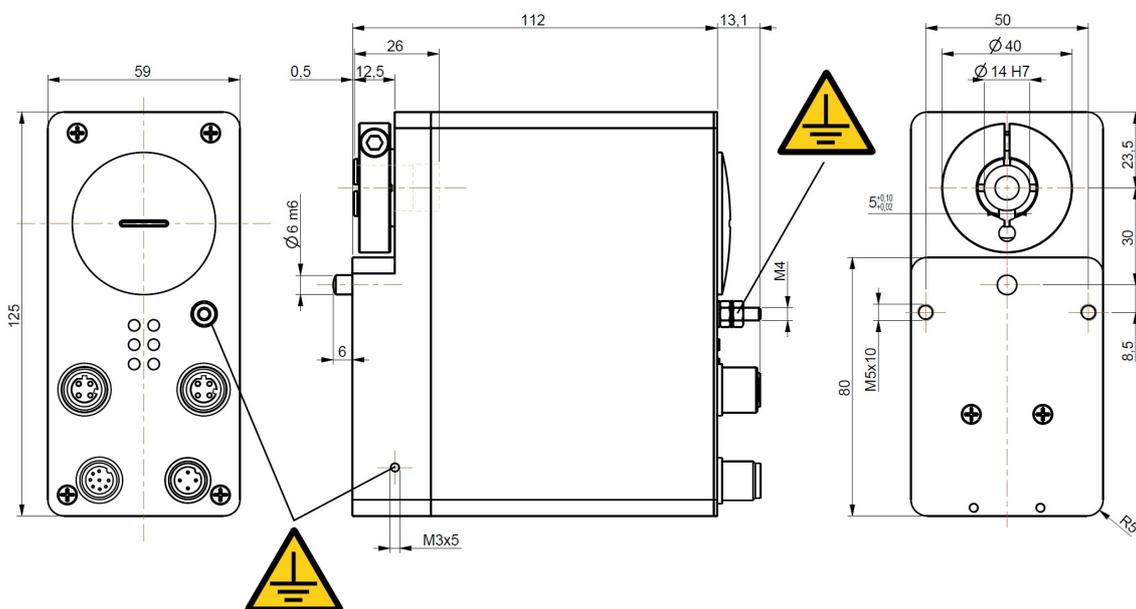


### 3 Installazione meccanica



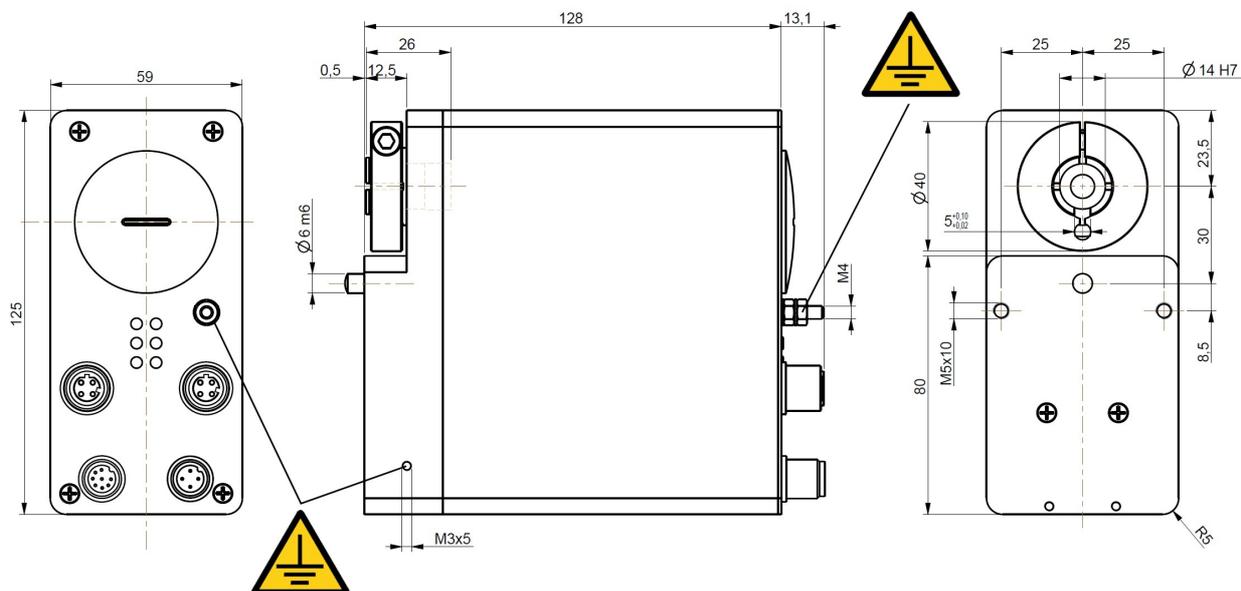
**ATTENZIONE**

L'installazione e le operazioni di manutenzione devono essere eseguite da personale qualificato, in assenza di tensione e movimento del motore e dell'albero.



(i valori sono espressi in mm)

Figura 1 - Unità RD1A – Dimensioni d'ingombro



(i valori sono espressi in mm)

Figura 2 - Unità RD12A – Dimensioni d'ingombro



L'unità DRIVECOD deve essere fissata in maniera solidale esclusivamente all'asse utilizzatore mediante il collare. E' fornita in dotazione di un pin antirotazione (si veda la Figura 3). Consigliamo di inserire il pin antirotazione in un gommino antivibrazione (non fornito). Questo assicura al dispositivo stabilità e contemporaneamente la mobilità necessaria per assorbire le tensioni meccaniche sviluppate durante il funzionamento. Non fissare in maniera rigida il pin antirotazione al supporto fisso lato utilizzatore senza il gommino antivibrazione! Ugualmente non portare in battuta l'unità DRIVECOD contro il supporto fisso lato utilizzatore. Se questo avvenisse le tensioni meccaniche sarebbero completamente concentrate sull'albero motore con conseguente danneggiamento dei cuscinetti!

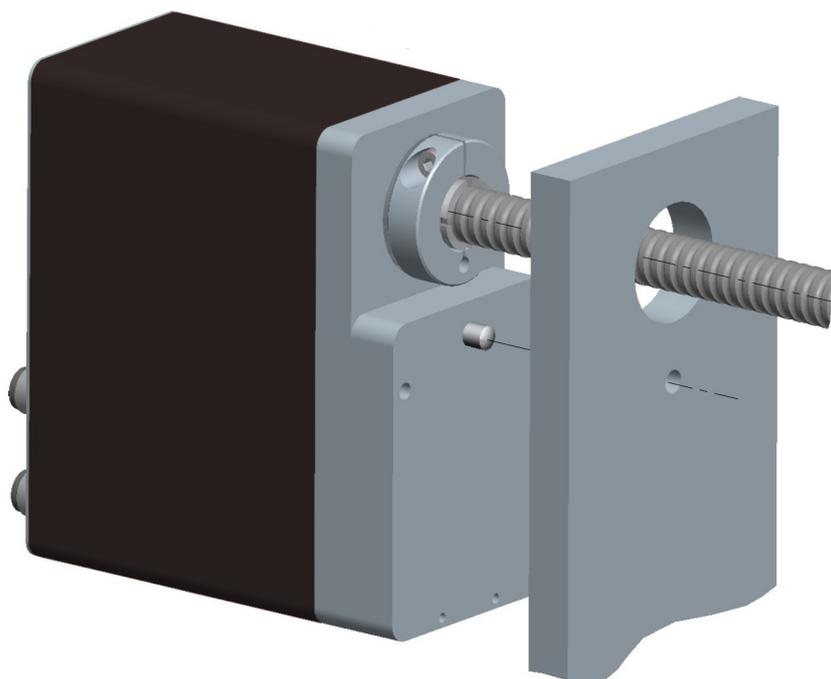
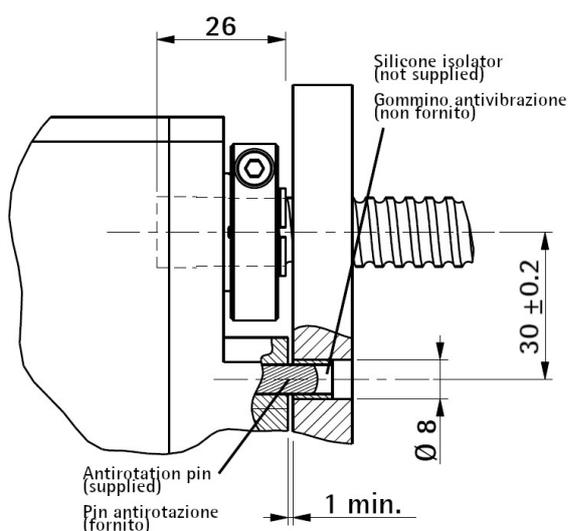


Figura 3 - Esempio di installazione di unità RD1xA su vite senza fine

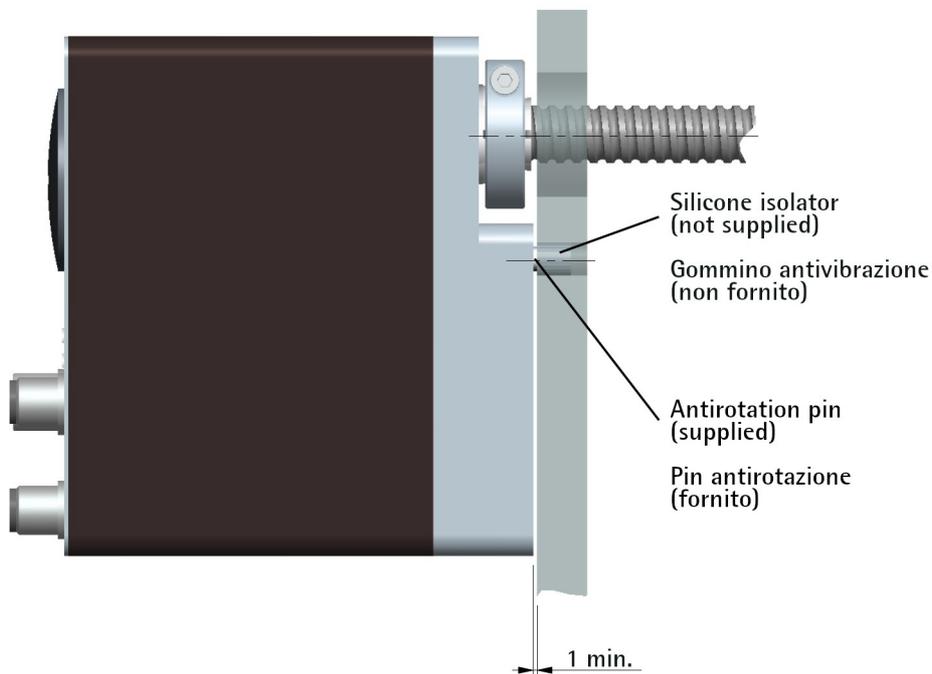
Per installare correttamente l'unità DRIVECOD seguire scrupolosamente le istruzioni riportate qui di seguito, tenendo conto a ogni modo che le possibilità di installazione possono essere molteplici e dipendenti dalla specifica applicazione.

- Eseguire un foro  $\varnothing 8$  mm sulla flangia o sul supporto lato utilizzatore per l'inserimento del pin antirotazione. Consigliamo di aggiungere un gommino antivibrazione (non fornito). La distanza tra l'asse dell'albero e l'asse del foro deve essere di  $30 \pm 0,2$  mm. Assicurarsi che il foro e l'albero siano perfettamente centrati sull'asse verticale. Eventuali disallineamenti risulterebbero in tensioni



meccaniche sull'albero motore con conseguente danneggiamento dei cuscinetti!

- inserire il gommino antivibrazione nel foro realizzato;
- inserire l'albero utilizzatore nell'asse cavo dell'unità DRIVECOD per una profondità massima di 26 mm e assicurarsi che anche il pin antirotazione si vada a inserire nel gommino antivibrazione;
- la distanza minima tra la flangia del dispositivo DRIVECOD e il supporto fisso sul lato utilizzatore deve essere di almeno 1 mm per evitare ogni possibile contatto;
- fissare l'albero utilizzatore mediante il collare e la relativa vite di fissaggio.



#### ATTENZIONE

Non forzare manualmente la rotazione dell'albero al fine di evitare danni permanenti! La tensione controlettromotrice (back EMF) generata dal motore a seguito di una movimentazione manuale dell'asse forzata dall'esterno può provocare danni irreparabili ai componenti interni.

## 4 Connessioni elettriche



### ATTENZIONE

Togliere tensione prima di eseguire qualsiasi collegamento elettrico.

La trasmissione dei comandi **Start**, **Jog +** e **Jog -** procura l'avvio del movimento dell'unità e dell'asse. Assicurarsi che in conseguenza di questo non possano verificarsi lesioni alle persone o danneggiamenti meccanici.

Ogni routine di **Start** deve essere preventivamente valutata con scrupolo!

Non forzare manualmente la rotazione dell'asse al fine di evitare danni permanenti!

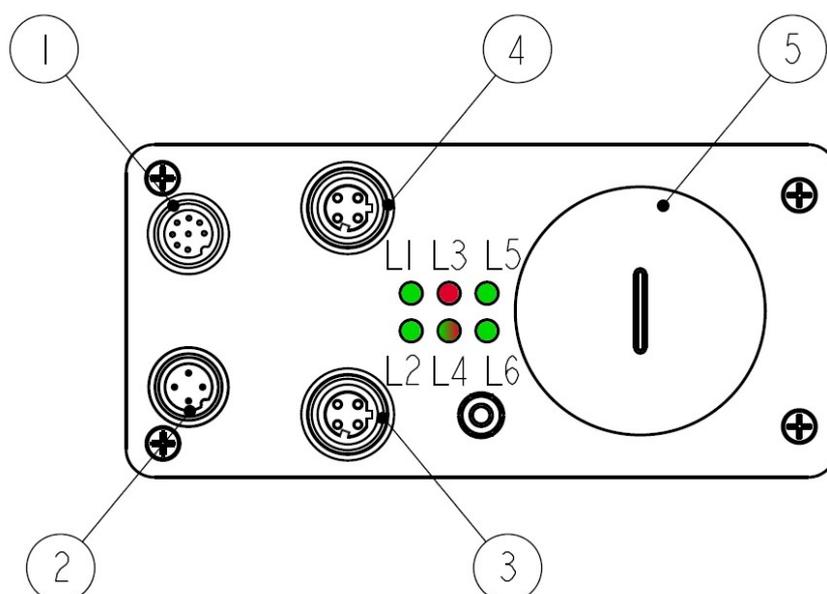


Figura 4: Connettori e led di diagnostica

### Legenda

1	Connettore M12 8 pin maschio ingressi / uscita + MODBUS RS-232
2	Connettore M12 4 pin maschio alimentazione
3	Connettore M12 4 pin femmina PORT IN (ECAT IN)
4	Connettore M12 4 pin femmina PORT OUT (ECAT OUT)
5	Alloggiamento interno pulsanti di Preset / Jog
L1	LED 1 Link / Activity sulla porta PORT OUT
L2	LED 2 Link / Activity sulla porta PORT IN
L3	LED 3 Informazione di errori / guasti attivi

L4	LED 4 Informazione stato interfaccia fieldbus
L5	LED 5 Informazione alimentazione controller
L6	LED 6 Informazione alimentazione motore

#### 4.1 Collegamento messa a terra (Figura 1 e Figura 2)

Per minimizzare i disturbi collegare il corpo del dispositivo a un buon punto di terra; si consiglia di utilizzare il punto di collegamento previsto (si vedano la Figura 1 e la Figura 2). Collegare la calza del cavo a un buon punto di terra sul lato utilizzatore. I cavi intestati EC- di Lika Electronic prevedono il collegamento della calza alla ghiera del connettore per la messa a terra attraverso il corpo del dispositivo. I connettori volanti E- di Lika Electronic utilizzano un connettore plastico; pertanto non è possibile la raccolta calza. Nel caso in cui si utilizzi un connettore metallico collegare opportunamente la calza del cavo attenendosi alle istruzioni del costruttore. Si veda anche la nota al paragrafo successivo. In tutti i casi assicurarsi che il punto di terra sia privo di disturbi e il più vicino possibile al dispositivo.

#### 4.2 Connettori (Figura 4 e Figura 5)

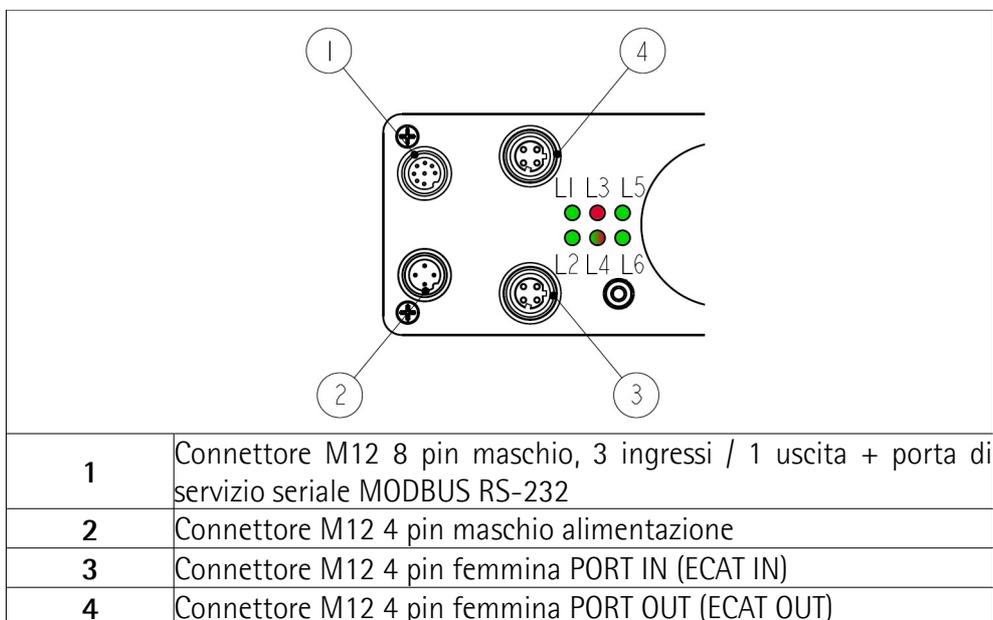


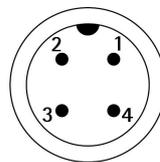
Figura 5: Connettori

#### 4.2.1 Connettore alimentazione

##### Alimentazione

Connettore M12 4 pin maschio

(vista lato contatti)



Pin	Descrizione
1	Alimentazione +24Vdc $\pm$ 10% motore
2	Alimentazione +24Vdc $\pm$ 10% controller
3	Alimentazione 0Vdc motore e controller
4	non collegato

#### 4.2.2 Connettori interfaccia EtherCAT (PORT IN e PORT OUT)

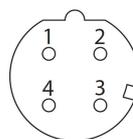
Due connettori M12 4 pin femmina con codifica D per la connessione alla rete Ethernet attraverso le porte PORT IN (ECAT IN) e PORT OUT (ECAT OUT).

##### Interfaccia

Connettori M12 4 pin

Codifica D, femmina

(vista lato contatti)



Pin	Descrizione
1	Tx Data +
2	Rx Data +
3	Tx Data -
4	Rx Data -
Custodia	Calza <sup>1</sup>

<sup>1</sup> Solo con cavi intestati EC- di Lika

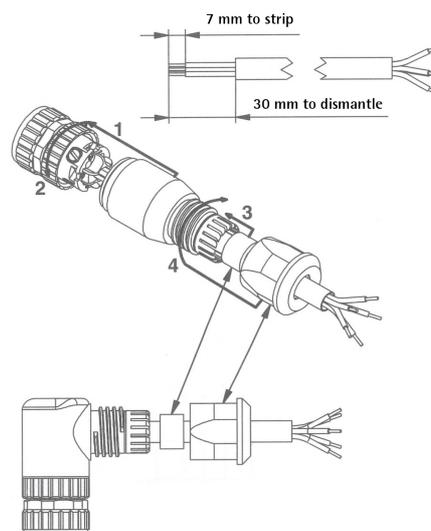
L'interfaccia Ethernet supporta un funzionamento a 100 Mbit/s, fast Ethernet, full duplex.

I connettori M12 PORT IN **3** e PORT OUT **4** hanno pin-out conforme allo standard EtherCAT®. E' perciò possibile utilizzare cavi standard EtherCAT disponibili in commercio.

I connettori d'ingresso (PORT IN **3**) e d'uscita (PORT OUT **4**) non sono intercambiabili! Il connettore PORT IN deve essere collegato in direzione del Master EtherCAT.

**NOTA**

Si consiglia di prevedere sempre il collegamento della calza del cavo a un buon punto di terra sul lato utilizzatore. I cavi intestati EC- di Lika Electronic prevedono il collegamento della calza alla ghiera del connettore per la messa a terra attraverso il corpo del dispositivo. I connettori volanti E- di Lika Electronic utilizzano un connettore plastico; pertanto non è possibile la raccolta calza (si veda la Figura). Nel caso in cui si utilizzi un connettore metallico collegare opportunamente la calza del cavo attenendosi alle istruzioni del costruttore.



#### 4.2.3 Configurazione di rete: topologie, cavi, hub, switch - Raccomandazioni

I cavi e i connettori sono conformi alle specifiche EtherCAT. I cavi sono del tipo CAT-5 schermati.

Line, tree o star: EtherCAT supporta pressoché ogni topologia. La struttura bus o lineare utilizzata negli impianti fieldbus diventa perciò disponibile anche per Ethernet, senza le limitazioni in quantità generate da switch o hub in cascata.

Le caratteristiche fisiche Fast Ethernet (100BASE-TX) permettono una lunghezza dei cavi di 100 m tra due dispositivi. E' possibile connettere fino a 65.535 dispositivi, la dimensione della rete è pressoché illimitata.

Il protocollo Ethernet conforme a IEEE 802.3 rimane integro fin giù al singolo dispositivo; non è richiesto nessun sotto-bus. Al fine di assicurare i requisiti di un dispositivo modulare come un morsetto elettronico, il livello fisico nel dispositivo di attacco può essere convertito da un cavo a intreccio a coppie o fibra ottica in LVDS (livello fisico Ethernet alternativo, standardizzato in [4,5]). Un dispositivo modulare può perciò essere esteso a costi irrisori. La successiva conversione dal livello fisico LVDS di tipo backplane al livello fisico 100BASE-TX è possibile in qualunque momento – come consueto in Ethernet.

Per una lista completa delle prolunghe e dei kit di connessione disponibili riferirsi al datasheet del prodotto (lista "Accessori").

#### 4.2.4 Impostazione indirizzo

Non è necessario assegnare un indirizzo fisico al dispositivo in quanto l'indirizzamento dello Slave avviene in modo automatico all'accensione del sistema durante la fase iniziale di scan della configurazione hardware.

L'indirizzamento è a 32 bit, è supportato l'Auto Increment Addressing.

- Auto Increment Addressing = 16 bit rappresentano la posizione fisica dello Slave nella rete e 16 bit vengono usati per indirizzare la memoria locale; quando riceve il frame, lo Slave incrementa la posizione fisica e il dispositivo che riceve Position = 0 è quello indirizzato.

#### 4.2.5 Terminazione di linea

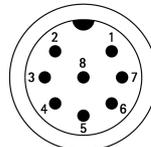
La rete EtherCAT non necessita di terminazione di linea in quanto la chiusura della rete avviene in modo automatico; ogni Slave infatti è in grado di rilevare o meno la presenza di altri Slave a valle.

#### 4.2.6 Ingressi / uscita + porta di servizio RS-232 MODBUS

##### Ingressi / uscita + porta di servizio RS-232 MODBUS

Connettore M12 8 pin

Codifica A, maschio  
(vista lato contatti)



Pin	Descrizione
1	0Vdc
2	Ingresso 1
3	Ingresso 2
4	Ingresso 3
5	Uscita 1
6	TD (RS-232)
7	RD (RS-232)
8	0Vdc (RS-232)

Per ogni informazione sui tre ingressi digitali e l'uscita digitale riferirsi alla sezione "6.3 Ingressi e uscita digitali" a pagina 50.

I parametri di configurazione della porta di servizio seriale MODBUS hanno valori fissi, non è perciò possibile modificarli.

Essi sono:

#### Impostazioni porta di servizio seriale RS-232 MODBUS

	Valore di default
Bit per secondo	9600
Bit di dati	8
Parità	Pari
Bit di stop	1

L'indirizzo MODBUS è "1". Non può essere modificato. Si veda la sezione "8.2 Pagina "Configurazione seriale"" a pagina 117.

Per ogni ulteriore informazione sulla configurazione e l'utilizzo della porta di servizio seriale RS-232 riferirsi alla sezione "Interfaccia MODBUS®" a pagina 115.

### 4.3 Led di diagnostica (Figura 4 e Figura 6)

Sei led installati nella parte posteriore della custodia dell'attuatore segnalano visivamente la condizione di funzionamento delle interfacce EtherCAT e MODBUS e del dispositivo, come esplicitato nelle seguenti tabelle.

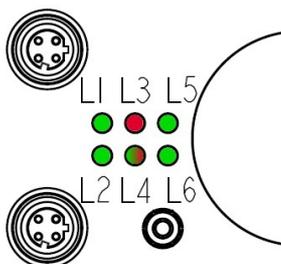


Figura 6: Led di diagnostica

<b>L1</b>	Link / Activity su PORT OUT	<b>L4</b>	Informazione stato interfaccia fieldbus
<b>L2</b>	Link / Activity su PORT IN	<b>L5</b>	Informazione alimentazione controller
<b>L3</b>	Informazione errori / guasti attivi	<b>L6</b>	Informazione alimentazione motore


**NOTA**

Si badi che i led hanno significati diversi a seconda dell'interfaccia attiva.



Il funzionamento dei led è conforme alle specifiche EtherCAT, si veda [ETG1300\\_S\\_R\\_V1i1iO\\_IndicatorLabelingSpecification.pdf](#).

Stati led	Definizione
<b>ON</b>	L'indicatore è costantemente ON.
<b>OFF</b>	L'indicatore è costantemente OFF.
<b>Tremolio</b>	L'indicatore si accende e si spegne con frequenza isofase di 10 Hz: ON per 50 ms e OFF per 50 ms.
<b>Lampeggio</b>	L'indicatore si accende e si spegne con frequenza isofase di 2,5 Hz: ON per 200 ms seguito da OFF per 200 ms.
<b>Singolo flash</b>	L'indicatore presenta un breve flash (200 ms) seguito da una lunga fase in cui è OFF (1000 ms).
<b>Doppio flash</b>	L'indicatore presenta una sequenza di due brevi flash (200 ms), separati da una fase OFF (200 ms), seguiti da una lunga fase di spegnimento (1000 ms).

LED L1 VERDE	Descrizione
Mostra lo stato del link fisico e l'attività sulla porta PORT OUT (connettore 4)	
<b>OFF</b>	Nessun link di rete né attività
<b>ON</b>	Il link di rete è presente, ma non c'è attività
<b>FLASH</b>	Attività (scambio dati)

LED L2 VERDE	Descrizione
Mostra lo stato del link fisico e l'attività sulla porta PORT IN (connettore 3)	
<b>OFF</b>	Nessun link di rete né attività
<b>ON</b>	Il link di rete è presente, ma non c'è attività
<b>FLASH</b>	Attività (scambio dati)

LED L3 ROSSO	Descrizione
Led EtherCAT ERROR, segnala errori di comunicazione EtherCAT	
<b>OFF</b>	Nessun errore
<b>Lampeggio ROSSO</b>	Configurazione non valida. Il cambio di stato ricevuto dal Master non è possibile a causa di impostazioni di registro o oggetto non valide
<b>Singolo flash RED</b>	Cambio di stato non richiesto. L'applicazione dispositivo Slave ha cambiato autonomamente il

	proprio stato EtherCAT
<b>Doppio flash ROSSO</b>	Timeout watchdog dell'applicazione, timeout watchdog Sync manager, comunicazione con il Master interrotta
<b>ON ROSSO</b>	Evento fatale. Se i led RUN ed ERROR si accendono rossi, siamo in presenza di un evento fatale, che forza l'interfaccia bus a uno stato fisicamente passivo. Se si verifica questa condizione, contattare il servizio di assistenza Lika Electronic.
<b>Tremolio ROSSO</b>	Rilevato errore di boot, dovuto per esempio a un problema di download del firmware

<b>LED L4 VERDE / ROSSO</b>	<b>Descrizione</b>
	Led EtherCAT RUN, mostra lo stato della comunicazione EtherCAT
<b>OFF</b>	L'attuatore è in stato <b>INIT</b>
<b>Lampeggio VERDE</b>	L'attuatore è in stato <b>PRE-OPERATIONAL</b>
<b>Single flash VERDE</b>	L'attuatore è in stato <b>SAFE-OPERATIONAL</b>
<b>ON VERDE</b>	L'attuatore è in stato <b>OPERATIONAL</b>
<b>Tremolio VERDE</b>	L'attuatore è in stato <b>BOOT</b> . Non usare, l'unità potrebbe danneggiarsi irreparabilmente
<b>ON ROSSO</b>	Evento fatale. Se i led RUN ed ERROR si accendono rossi, siamo in presenza di un evento fatale, che forza l'interfaccia bus a uno stato fisicamente passivo. Se si verifica questa condizione, contattare il servizio di assistenza Lika Electronic.

<b>LED L5 VERDE</b>	<b>Descrizione</b>
	Indica se il controller è alimentato o no
<b>ON</b>	Segnala che il controller è alimentato
<b>OFF</b>	Segnala che il controller non è alimentato

<b>LED L6 VERDE</b>	<b>Descrizione</b>
	Indica se il motore è alimentato o no
<b>ON</b>	Segnala che il motore è alimentato
<b>OFF</b>	Segnala che il motore non è alimentato



LED L5 VERDE	Descrizione
	Indica se il controller è alimentato o no
ON	Segnala che il controller è alimentato
OFF	Segnala che il controller non è alimentato

LED L6 VERDE	Descrizione
	Indica se il motore è alimentato o no
ON	Segnala che il motore è alimentato
OFF	Segnala che il motore non è alimentato

#### 4.4 Tappo a vite per accesso interno (Figura 4 e Figura 7)



##### ATTENZIONE

Fare attenzione che l'alimentazione è ON quando si utilizzano i pulsanti Preset / Jog!



##### NOTA

Eeguire questa operazione con estrema prudenza per non danneggiare le connessioni interne.

Per accedere ai pulsanti Preset / Jog svitare e rimuovere il tappo a vite con filettatura PG 29. Avere cura di ripristinare il coperchio al termine dell'operazione.

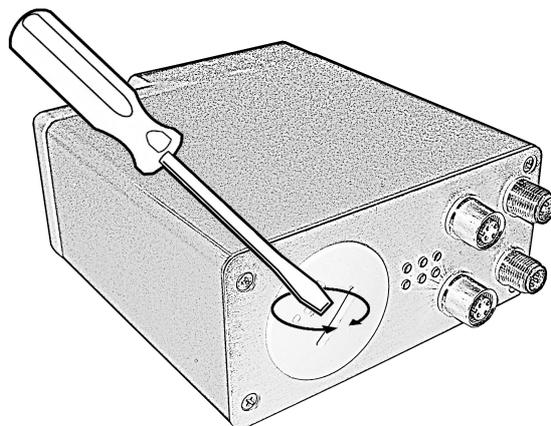


Figura 7: Tappo a vite per accesso interno

#### 4.5 Pulsanti Preset / Jog (Figura 8)

I pulsanti Preset / Jog sono posizionati esattamente sotto il tappo a vite.

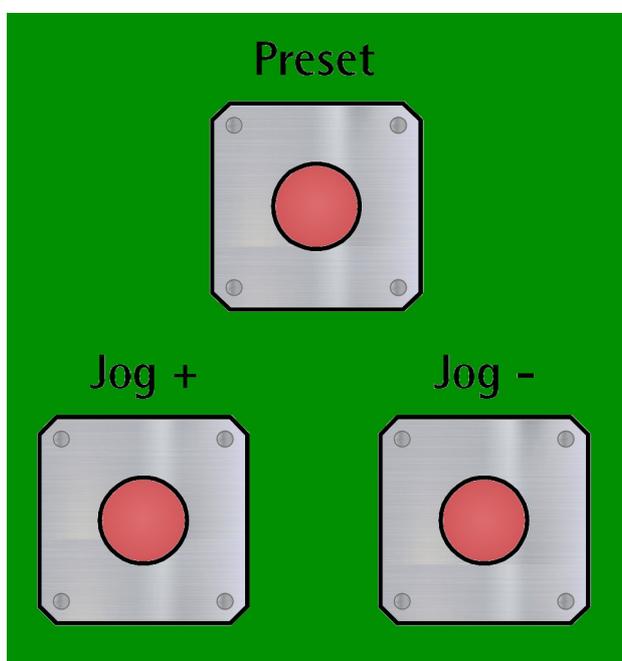


Figura 8: Pulsanti Preset / Jog

#### 4.5.1 Pulsanti JOG + e JOG - (Figura 8)

Premere questi pulsanti per comandare i movimenti del motore in direzione positiva (**Jog +**) o negativa (**Jog -**) nella modalità di funzionamento manuale. Per ogni informazione si consultino i comandi **Jog +** e **Jog -** a pagina 96 e seguenti (interfaccia EtherCAT) o a pagina 158 e seguenti (interfaccia Modbus).



#### ATTENZIONE

I pulsanti JOG e PRESET possono essere attivi e utilizzabili da parte dell'operatore, anche con dispositivo in allarme o in condizione di emergenza. Prima di premere questi pulsanti assicurarsi che il movimento del dispositivo si possa realizzare in condizioni di assoluta sicurezza e che non sussista alcun pericolo di procurare lesioni a persone o danni alle apparecchiature.



#### NOTA

Si badi che nell'utilizzo dei pulsanti manuali la funzione di jog incrementale (si veda **Incremental jog** in **2200 Control Word** a pagina 98 -versione EtherCAT; **Incremental jog** in **Control Word [0x2A]** a pagina 160 -versione Modbus) è disabilitata; non è cioè possibile eseguire passi jog utilizzando i pulsanti manuali. Pertanto i movimenti positivo e negativo sono comandati solo mediante la pressione continuata dei pulsanti e hanno termine al rilascio dei pulsanti o in corrispondenza dei rispettivi finecorsa.



#### NOTA

Le funzioni **Jog +**, **Jog -** e **Start** non possono essere abilitate simultaneamente. Per esempio: se un comando **Jog +** viene inviato allo Slave mentre questo sta eseguendo il movimento verso la posizione target, il comando jog viene ignorato; se i comandi **Jog +** e **Jog -** sono inviati simultaneamente, il dispositivo non si muove o, se già in movimento, si arresta.

#### 4.5.2 Pulsante PRESET (Figura 8)

Utilizzare questo pulsante per assegnare alla posizione attuale dell'asse il valore impostato al parametro **Preset**. Il pulsante deve essere mantenuto premuto per almeno 3 secondi. Consigliamo di impostare il preset quando l'attuatore è in stop. Per ogni informazione sulla funzione di preset si veda il parametro **Preset** a pagina 111 (interfaccia EtherCAT) o a pagina 157 (interfaccia Modbus).



#### ATTENZIONE

I pulsanti JOG e PRESET possono essere attivi e utilizzabili da parte dell'operatore, anche con dispositivo in allarme o in condizione di emergenza.

Prima di premere questi pulsanti assicurarsi che il movimento del dispositivo si possa realizzare in condizioni di assoluta sicurezza e che non sussista alcun pericolo di procurare lesioni a persone o danni alle apparecchiature.

## 5 Quick reference

Le istruzioni che seguono sono fornite per permettere un set up rapido dell'unità in una modalità di funzionamento standard e sicura.

- Procedere all'installazione meccanica, si veda a pagina 29 e segg.;
- effettuare le connessioni elettriche, si veda a pagina 33 e segg.;
- alimentare l'unità con alimentazione +24Vdc (sia il motore che il controller);
- verificare le condizioni di funzionamento segnalate mediante i led;
- per riprendere le normali condizioni di lavoro resettare l'emergenza attiva: portare alto ("=1") il bit 7 **Emergenza** nella **Control Word** (si veda a pagina 96 -interfaccia EtherCAT; si veda a pagina 158 -interfaccia MODBUS); resettare gli allarmi attivi: portare alto ("=1") il bit 3 **Reset allarmi** nella **Control Word** (si veda a pagina 96 -interfaccia EtherCAT; si veda a pagina 158 -interfaccia MODBUS). Verificare le condizioni di funzionamento segnalate mediante i led;
- impostare un valore adeguato in **Distance per revolution** (si veda a pagina 105 -interfaccia EtherCAT; si veda a pagina 151 -interfaccia MODBUS);
- impostare un valore adeguato in **Jog speed** (si veda a pagina 110 -interfaccia EtherCAT; si veda a pagina 155 -interfaccia MODBUS);
- impostare un valore adeguato in **Work speed** (si veda a pagina 110 -interfaccia EtherCAT; si veda a pagina 156 -interfaccia MODBUS);
- se richiesto, impostare un valore adeguato in **Preset** (si veda a pagina 111 -interfaccia EtherCAT; si veda a pagina 157 -interfaccia MODBUS);
- impostare i valori di finecorsa in **Max delta pos / Positive delta** e **Max delta neg / Negative delta** (si veda a pagina 107 e seguenti -interfaccia EtherCAT; si veda a pagina 153 e seguenti -interfaccia MODBUS);
- impostare la posizione comandata in **Target position** (si veda a pagina 101 -interfaccia EtherCAT; si veda a pagina 163 -interfaccia MODBUS);
- salvare le nuove impostazioni (comando **Save parameters**; si veda a pagina 99 -interfaccia EtherCAT; si veda a pagina 161 -interfaccia MODBUS).

Utilizzare i comandi **Jog +**, **Jog -**, **Start** e **Stop** nella **Control Word** (si veda a pagina 96 -interfaccia EtherCAT; si veda a pagina 158 -interfaccia MODBUS ) per muovere l'asse e raggiungere la posizione comandata.



### NOTA

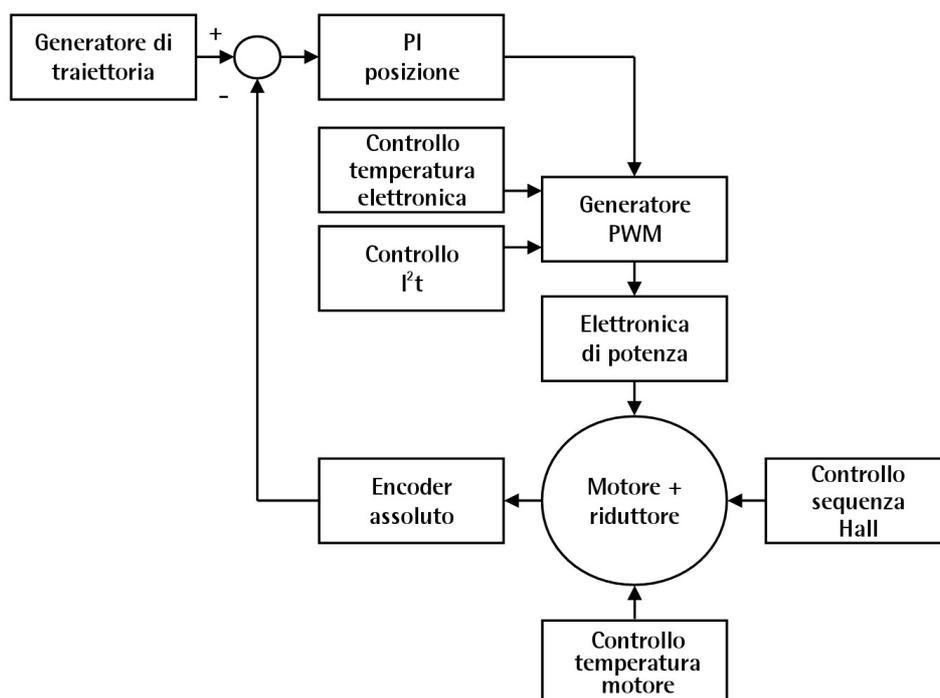
I parametri **Distance per revolution**, **Jog speed**, **Work speed**, **Preset**, **Max delta pos / Positive delta** e **Max delta neg / Negative delta** sono in stretta relazione, si deve quindi prestare particolare attenzione quando si procede alla

modifica anche di uno solo di essi. Per maggiori informazioni riferirsi a pagina 51.

## 6 Funzioni

### 6.1 Principio di funzionamento

Il seguente schema a blocchi illustra schematicamente la logica di controllo del sistema.



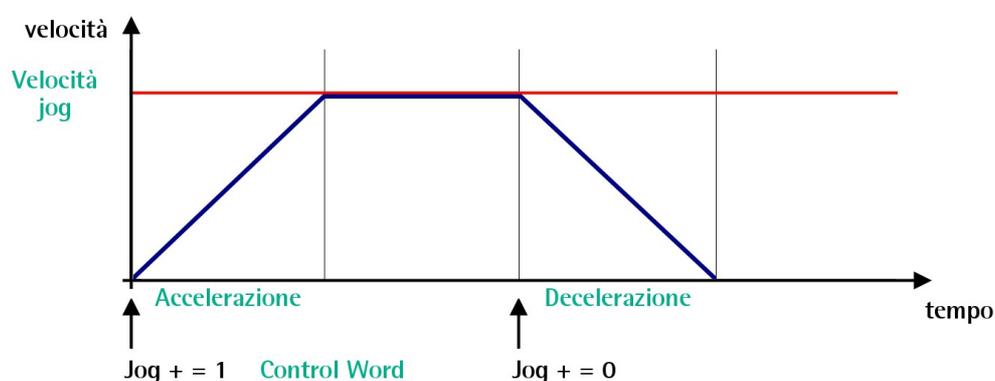
## 6.2 Tipi di movimento: jog e posizione

Nell'unità DRIVECOD sono previsti i seguenti due tipi di movimento:

- Jog: controllo di velocità;
- Posizione: controllo di posizione e velocità.

### Jog: controllo di velocità

Questo tipo di controllo genera una traiettoria di velocità tale da imporre a regime una velocità di rotazione dell'asse del dispositivo DRIVECOD uguale a **220E-00 Jog speed / Jog speed [0x0D]**.



Quando il bit 0 **Jog +** in **2200 Control Word / Control Word [0x2A]** è "1", il motore accelera verso la direzione positiva conformemente al valore impostato in **220A-00 Acceleration / Acceleration [0x07]**; se la corsa disponibile è sufficientemente lunga, esso raggiunge la velocità impostata in **220E-00 Jog speed / Jog speed [0x0D]**. Non appena il bit 0 **Jog +** in **2200 Control Word / Control Word [0x2A]** va basso ("0"), il motore decelera conformemente al valore impostato in **220B-00 Deceleration / Deceleration [0x08]** fino ad arrestarsi.

Se si imposta il bit 1 **Jog -** in **2200 Control Word / Control Word [0x2A]** a "1" si procura il movimento del motore nella direzione opposta (direzione negativa) secondo le fasi di lavoro già descritte sopra.

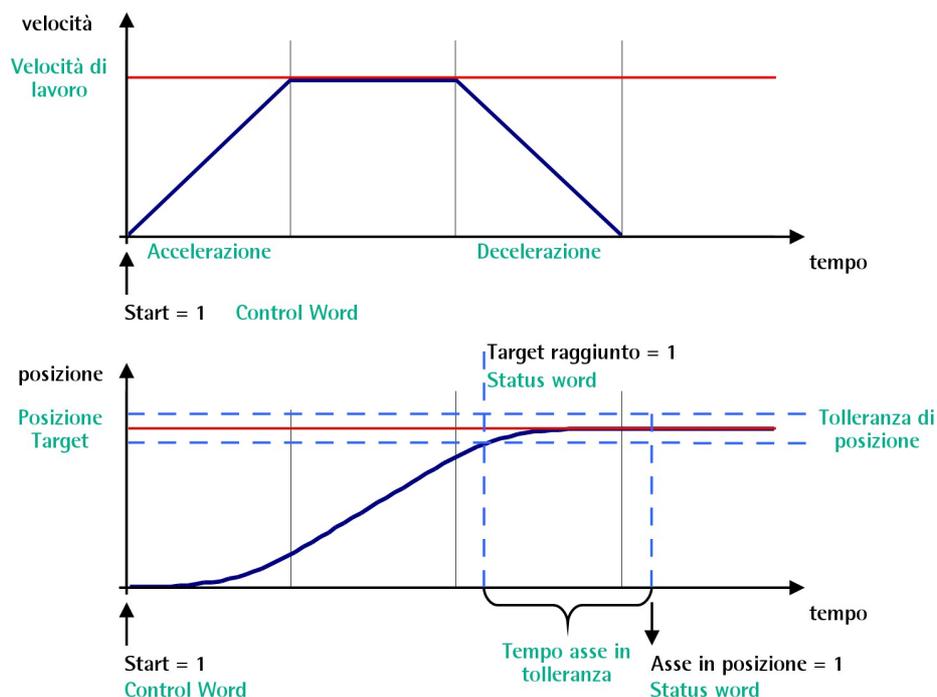


#### NOTA

Si badi che il valore in **220E-00 Jog speed / Jog speed [0x0D]** rappresenta la velocità del motore, non la velocità dell'asse in uscita dopo il riduttore.

### Posizione: controllo di posizione e velocità

Questo tipo di controllo è un movimento punto-punto, nel quale la massima velocità raggiunta è pari a **220F-00 Work speed / Work speed [0x0E]**, tale velocità sarà raggiunta solo se lo spazio da percorrere è sufficientemente grande.



Quando il bit 6 **Start** in **2200 Control Word / Control Word [0x2A]** è "1", il motore inizia a muoversi e accelera conformemente al valore in **220A-00 Acceleration / Acceleration [0x07]** al fine di raggiungere la posizione target impostata in **2201-00 Target Position / Target position [0x2B-0x2C]**. Se la corsa disponibile è sufficientemente lunga, esso raggiunge la velocità impostata in **220F-00 Work speed / Work speed [0x0E]**. La direzione del movimento può essere sia positiva che negativa a seconda della posizione target da raggiungere. Non appena l'asse si trova all'interno dei limiti di tolleranza impostati in **2205-00 Position tolerance / Position window [0x01]**, il bit 8 **Target position reached** in **2202 Status Word / Status word [0x01]** va alto ("1"). Quando la posizione è all'interno dei limiti di tolleranza impostati in **2205-00 Position tolerance / Position window [0x01]**, dopo il ritardo impostato in **2206-00 Settling time / Position window time [0x02]**, il bit 0 **Axis in position** in **2202 Status Word / Status word [0x01]** va alto ("1"). Il motore decelera conformemente al valore impostato in **220B-00**

**Deceleration / Deceleration [0x08]** in modo da raggiungere la posizione di stop in corrispondenza della posizione di target impostata.



**NOTA**

**Funzione di position override**

E' possibile modificare il valore della posizione di target anche al volo, mentre il dispositivo sta ancora raggiungendo una posizione di target comandata in precedenza e senza inviare un nuovo comando di **Start**. Per fare questo, semplicemente impostare un nuovo valore di target in **2201-00 Target Position / Target position [0x2B-0x2C]**.



**NOTA**

Si badi che il valore in **220F-00 Work speed / Work speed [0x0E]** rappresenta la velocità del motore, non la velocità dell'asse in uscita dopo il riduttore.

**6.3 Ingressi e uscita digitali**

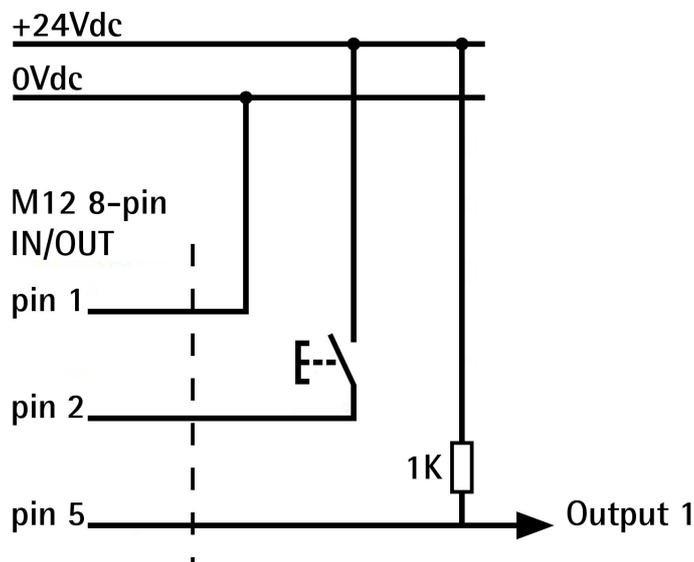
L'unità RD1xA è dotata di **tre ingressi digitali e un'uscita digitale**.

Gli **ingressi** sono letti dal dispositivo Slave e trasmessi al Master attraverso la **Status word** (bit 13-15; si veda a pagina 104 e seguenti -interfaccia EtherCAT- o a pagina 170 e seguenti -interfaccia Modbus) quando il dispositivo si trova nello stato **Operational** -interfaccia EtherCAT- / stato **Idle** -interfaccia Modbus. Il valore logico "alto" è letto con tensione pari a +24Vdc ±10%.

L'**uscita 1** è comandata dal Master allo Slave attraverso la **Control word** (bit 13; si veda a pagina 96 -interfaccia EtherCAT- o a pagina 158 -interfaccia Modbus) quando il dispositivo si trova nello stato **Operational** -interfaccia EtherCAT- / stato **Idle** -interfaccia Modbus.

E' un'uscita open collector con  $I_{max} = 150mA$ .

Esempio di schema di collegamento:



#### 6.4 Distance per revolution, Preset, Max delta pos / Positive delta e Max delta neg / Negative delta

Le variabili **Distance per revolution**, **Preset**, **Max delta pos / Positive delta** e **Max delta neg / Negative delta** sono in stretta correlazione, si deve quindi prestare particolare attenzione quando si procede alla modifica anche di uno solo di essi.

Qualora si rendesse necessaria una modifica dei parametri, la sequenza corretta sarà la seguente:

- impostare un valore adeguato in **Distance per revolution** (si veda a pagina 105 -interfaccia EtherCAT; si veda a pagina 151 -interfaccia MODBUS);
- impostare un valore adeguato in **Jog speed** (si veda a pagina 110 -interfaccia EtherCAT; si veda a pagina 155 -interfaccia MODBUS);
- impostare un valore adeguato in **Work speed** (si veda a pagina 110 -interfaccia EtherCAT; si veda a pagina 156 -interfaccia MODBUS);
- impostare un valore adeguato in **Preset** (si veda a pagina 100 -interfaccia EtherCAT; si veda a pagina 157 -interfaccia MODBUS);
- verificare il valore in **Max delta pos / Positive delta** (si veda a pagina 107 -interfaccia EtherCAT; si veda a pagina 153 -interfaccia MODBUS);

- verificare il valore in **Max delta neg / Negative delta** (si veda a pagina 108 -interfaccia EtherCAT; si veda a pagina 154 -interfaccia MODBUS);
- salvare le nuove impostazioni (comando **Save parameters**, bit 9 in **2200 Control Word / Control Word [0x2A]**, si veda a pagina 96 -interfaccia EtherCAT; si veda a pagina 161 -interfaccia MODBUS).

Ogniqualevolta si modifica il valore in **Distance per revolution** è poi necessario aggiornare il valore in **Preset** in modo da definire lo zero asse in quanto il sistema di riferimento è variato.

Dopo la modifica del parametro **Preset** non occorre invece reimpostare il valore dei finecorsa in quanto la funzione di Preset provvede a ricalcolarli automaticamente reinizializzando i limiti positivo e negativo sulla base dei valori impostati in **Max delta pos / Positive delta** e **Max delta neg / Negative delta**.

Il numero di giri gestiti dal dispositivo è di 512 in direzione negativa e 512 in direzione positiva assumendo il valore di **Preset** come riferimento (il numero massimo di giri è 1.024).

Il valore del parametro **Max delta pos / Positive delta** sommato al valore in **Preset** definisce il massimo spostamento in avanti (corsa positiva) rispetto al preset (valore espresso in impulsi).

Il valore del parametro **Max delta neg / Negative delta** sottratto al valore in **Preset** definisce il massimo spostamento all'indietro (corsa negativa) rispetto al preset (valore espresso in impulsi).



#### ATTENZIONE

Si badi che i parametri elencati qui a seguire sono in stretta correlazione con il parametro **Distance per revolution**; di conseguenza la modifica del valore nel parametro **Distance per revolution** comporta necessariamente una ridefinizione dei valori da essi espressi. I parametri sono: **Position tolerance / Position window**, **Max following error**, **Max delta pos / Positive delta**, **Max delta neg / Negative delta**, **Target position**, **Real position / Current position** e **Following error [pulse] / Position following error**.



#### ESEMPIO 1

Valori di default:

**Distance per revolution** = 1024 impulsi/giro

**Work speed**: 2000 rpm

**Preset** = 0



## 7 Interfaccia EtherCAT®

Gli attuatori Lika sono dispositivi Slave e supportano la modalità "CANopen Over EtherCAT" (COE) per la trasmissione dei dati. In particolare, essi supportano il "CANopen DS301 Communication profile".

Per ogni specifica omessa sul protocollo CANopen®, riferirsi al documento "CiA Draft Standard Proposal 301 CANopen Application layer and communication profile" disponibile all'indirizzo [www.can-cia.org](http://www.can-cia.org).

Per ogni specifica omessa sul protocollo EtherCAT®, riferirsi al documento "ETG.1000 EtherCAT Specification" disponibile all'indirizzo [www.ethercat.org](http://www.ethercat.org).

### 7.1 Configurazione con ambiente di sviluppo TwinCAT di Beckhoff

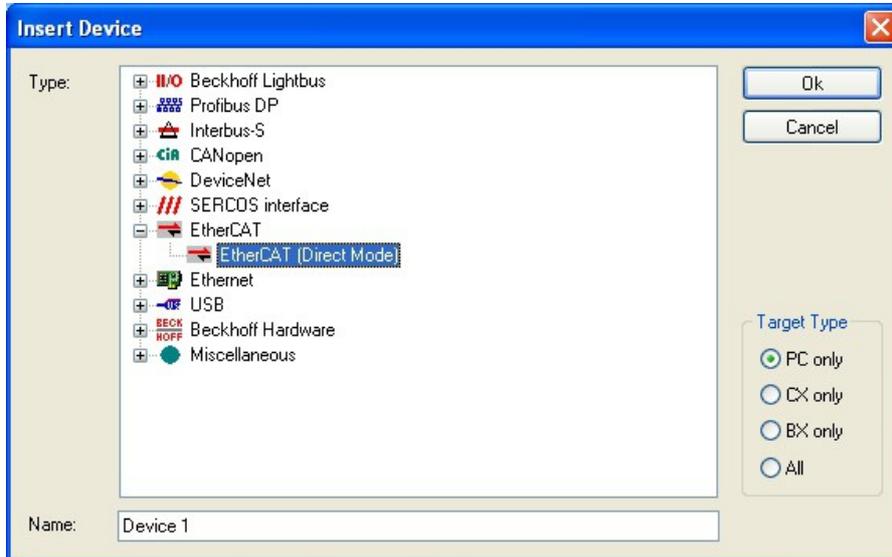
#### 7.1.1 Impostazione della scheda di rete

Avviare **TwinCAT System Manager**.

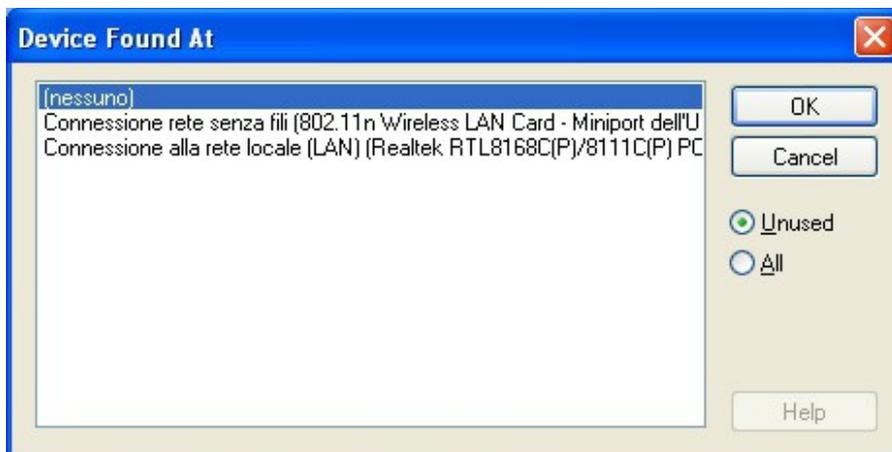
Nella parte sinistra della finestra principale estendere l'albero dei dispositivi e selezionare **I/O Devices**; quindi premere il tasto destro del mouse, aprire il menu a tendina e selezionare il comando **Append Device...**



Nella finestra **Insert Device** selezionare il dispositivo **EtherCAT (Direct Mode)**, quindi confermare premendo il pulsante **OK**.



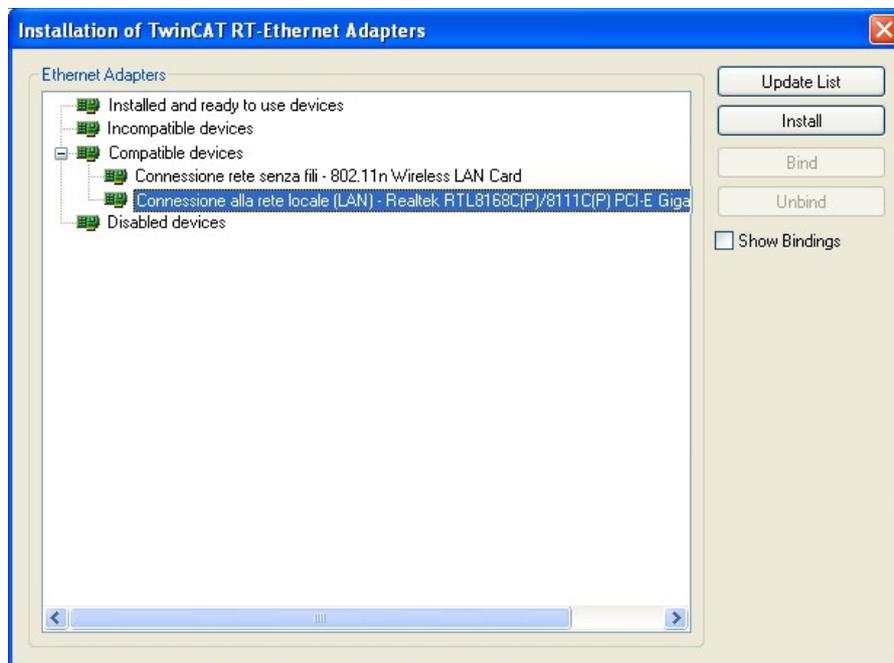
Se vi sono schede di rete già installate apparirà una finestra come quella riportata di seguito dove saranno elencati i dispositivi installati.



Selezionare la scheda di rete che si vuole utilizzare e confermare la scelta premendo il pulsante **OK**.

Se invece non vi sono schede di rete installate bisogna prima installarne una. Per fare questo aprire il menu **Options** nella barra di menu della finestra principale di **TwinCAT System Manager** e selezionare poi il comando **Show Real Time**

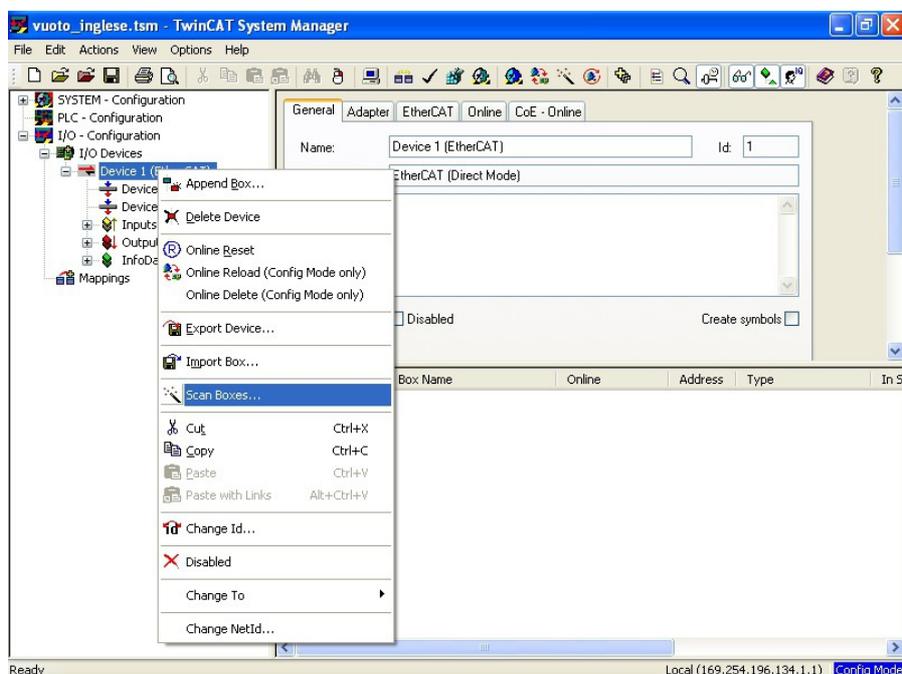
Ethernet Compatible Devices.... Apparirà la finestra **Installation of TwinCAT RT – Ethernet Adapter**.



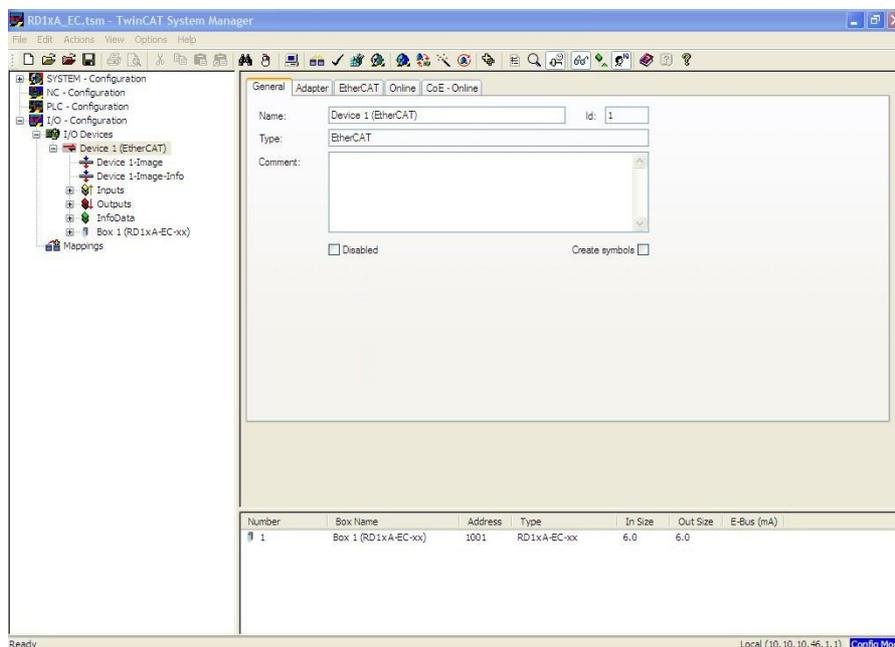
Selezionare ora **Compatible devices** e scegliere poi la scheda di rete che si vuole installare. Confermare premendo il pulsante **Install**.

### 7.1.2 Aggiunta di nuovi moduli I/O (Box)

Se i dispositivi sono già collegati alla rete e sono alimentati, cliccare il tasto destro del mouse sulla voce **Device 1 (EtherCAT)** nella finestra principale di **TwinCAT System Manager** e premere il comando **Scan Boxes...**

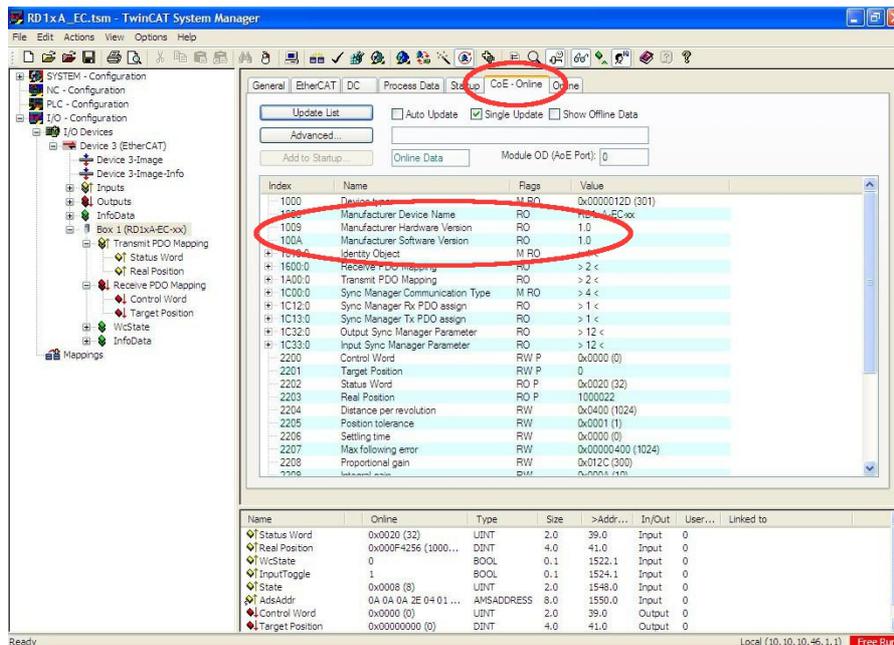


Al termine si avrà una situazione simile a quella descritta nella Figura seguente, alcune informazioni saranno elencate nella pagina.

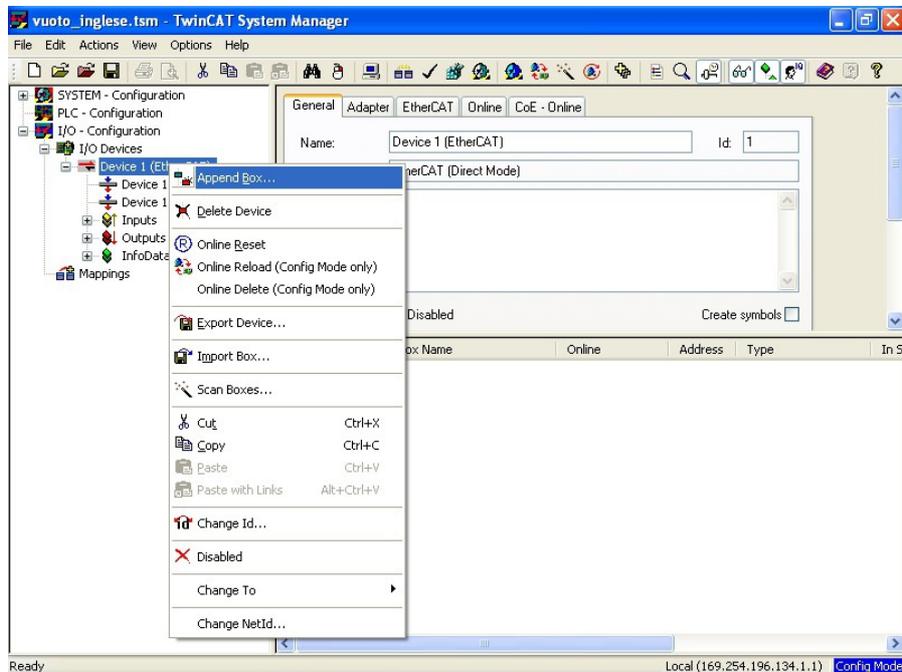


Se i dispositivi non sono già collegati alla rete è necessario utilizzare il file XML fornito con l'attuatore: **Lika\_RDxx\_EC\_Vx.xml** (versione V2 o superiore; si veda all'indirizzo [www.lika.it](http://www.lika.it) > **PRODOTTI** > **ATTUATORI ROTATIVI**).

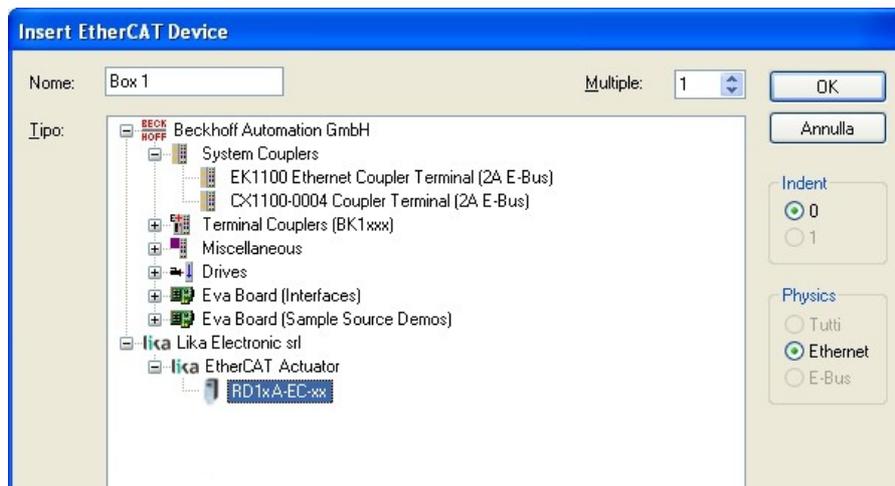
Se si vuole conoscere la versione firmware del dispositivo, premere **Box (RD1xA-EC-xx)** nella finestra principale di **TwinCAT System Manager**: appariranno alcune pagine a schede per la configurazione e la gestione del dispositivo. Accedere alla pagina **CoE - Online** e poi fare riferimento agli indici **1009-00 Manufacturer Hardware version** e **100A-00 Manufacturer Software version**.



Premere il tasto destro del mouse su **Device 1 (EtherCAT)** nel menu ad albero della finestra principale di **TwinCAT System Manager** e premere il comando **Append Box...**



Si aprirà la finestra **Insert EtherCAT Device**.



Nella finestra **Insert EtherCAT Device** selezionare **Lika Electronic srl** e poi **EtherCAT Actuator**; scegliere ora nella lista l'attuatore da installare (nel caso in cui fossero disponibili più attuatori).  
Premere il pulsante **OK** per confermare la scelta.

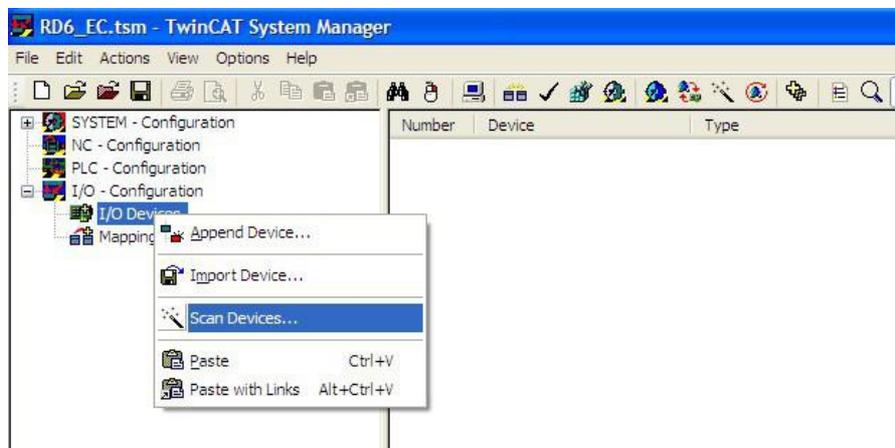
## 7.2 Impostazione della modalità di comunicazione

Gli attuatori Lika con interfaccia EtherCAT supportano esclusivamente la modalità di comunicazione FreeRun.

### 7.2.1 Modalità di comunicazione FreeRun

Quando si avvia per la prima volta uno scan della rete bus alla ricerca dei dispositivi EtherCAT connessi, viene richiesta l'attivazione della modalità di comunicazione FreeRun.

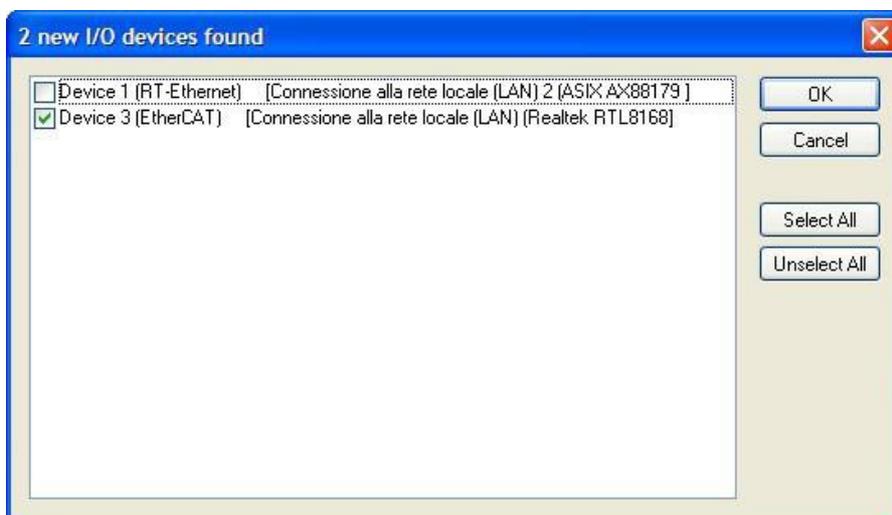
Per fare questo selezionate **I/O Devices** e premere poi il tasto destro del mouse per aprire il menu a tendina. Selezionare il comando **Scan Devices...**



Sullo schermo appare una casella di dialogo con messaggio di avvertenza: premere **OK** per avviare la scansione dei dispositivi EtherCAT.



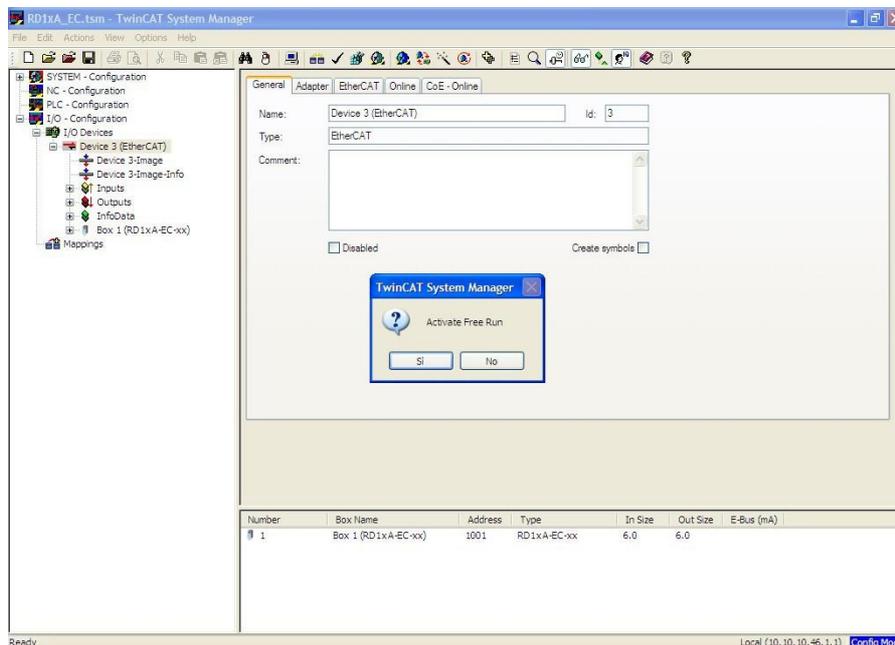
Selezionare una scheda di interfaccia EtherCAT tra quelle rilevate.



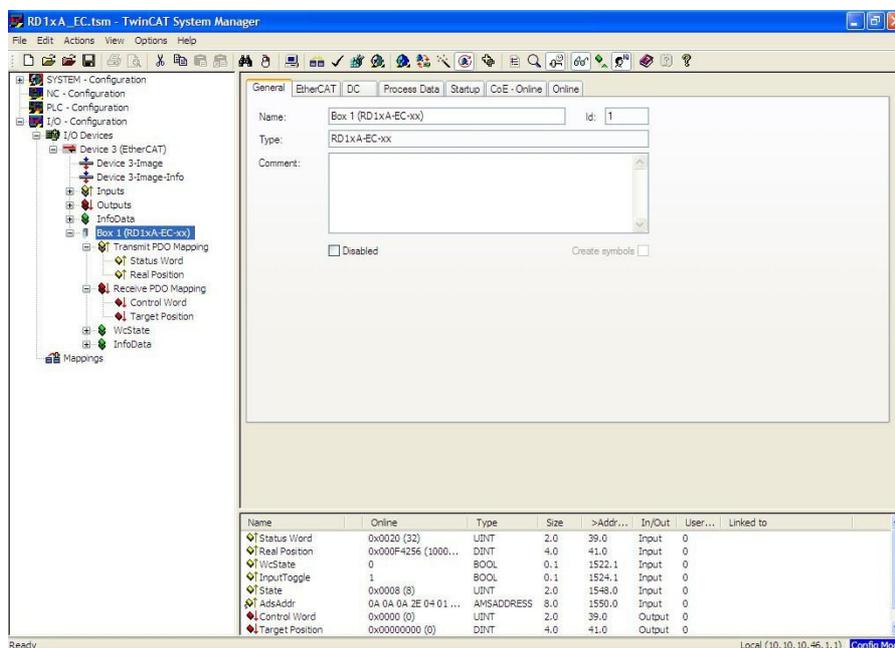
Premere **SI** e confermare l'avvio della procedura "Scan for boxes" e la ricerca dei dispositivi collegati.



Premere **SI** per attivare la modalità di comunicazione FreeRun.



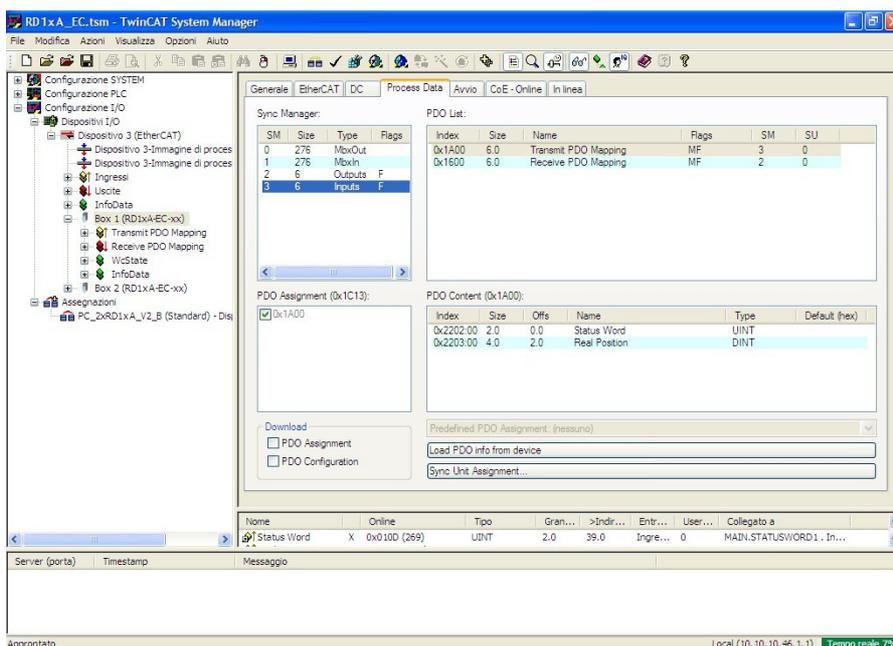
Dopo l'attivazione della modalità di comunicazione FreeRun la pagina apparirà come segue:



Per ogni informazione sulla modalità di comunicazione FreeRun riferirsi alla sezione "FreeRun" a pagina 73 e all'oggetto **1C33 Input Sync Manager Parameter** a pagina 89.

### 7.3 Process Data Object

Nel menu ad albero della finestra principale **TwinCAT System Manager** premere l'elemento **Box (RD1xA-EC-xx)**. Espanderlo per avere accesso ai Process Data Input (PDI) e Output (PDO): nella schermata principale compariranno alcune pagine a scheda per la configurazione e la gestione del dispositivo. Accedere alla pagina **Process Data**. In questa pagina sono disponibili i process data object (dati di processo, Mappatura RxPDO e TxPDO).

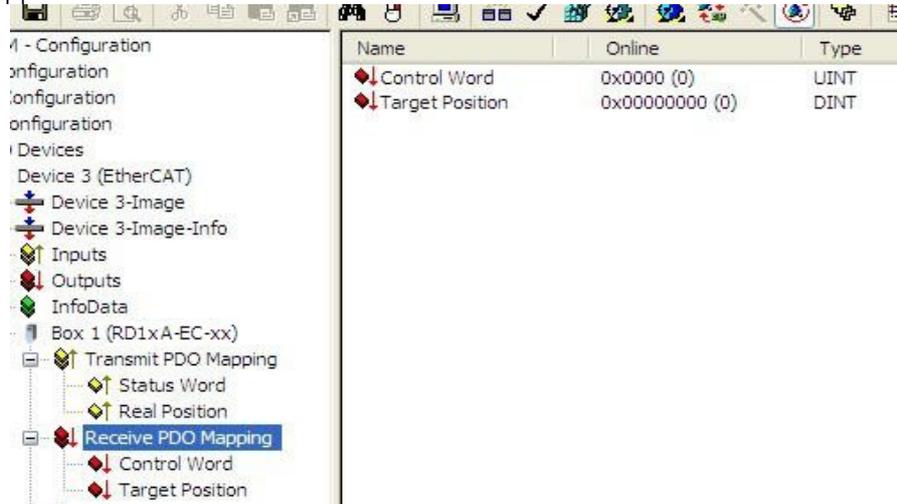


I dati di processo possono essere visualizzati anche premendo gli elementi **Transmit PDO Mapping** e **Receive PDO Mapping** nel menu ad albero della finestra principale **TwinCAT System Manager**; i dati sono elencati nella schermata principale.

#### Mappatura Transmit PDO

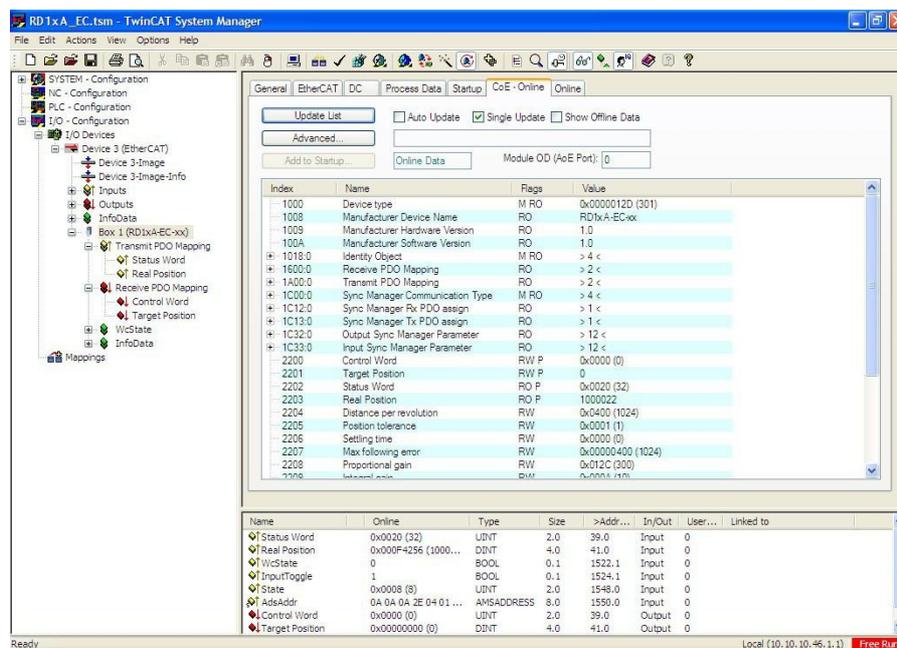


## Mappatura Receive PDO

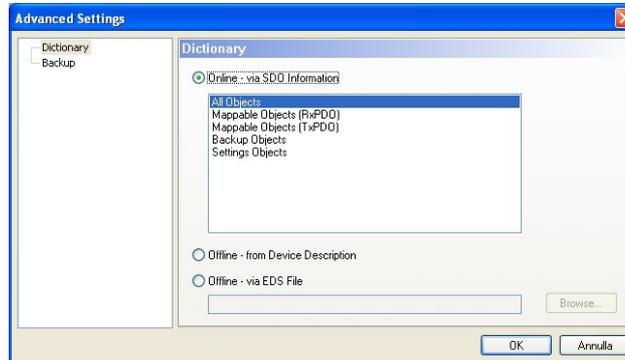


## 7.4 Dizionario oggetti COE

Nel menu ad albero della finestra principale **TwinCAT System Manager** premere l'elemento **Box (RD1xA-EC-xx)**: nella schermata principale compariranno alcune pagine a scheda per la configurazione e la gestione del dispositivo. Accedere alla pagina **CoE - Online**. In questa pagina è visualizzata la lista degli oggetti del dizionario. Questa è la versione offline del dizionario oggetti come viene letta dal file xml.

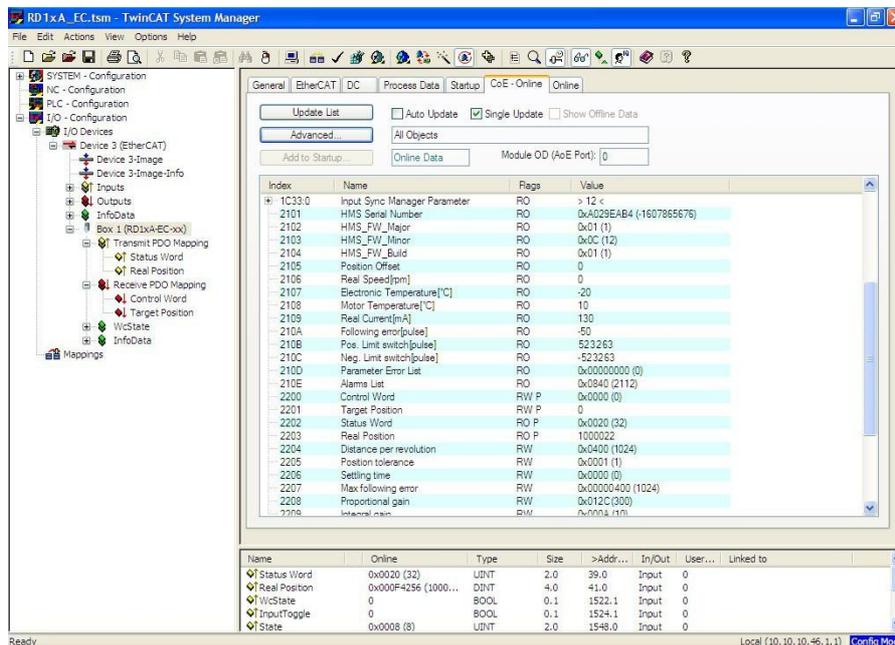


Gli oggetti possono essere letti anche direttamente dall'attuatore; per fare questo cliccare il pulsante **Advanced...**: apparirà la finestra **Advanced Settings**.



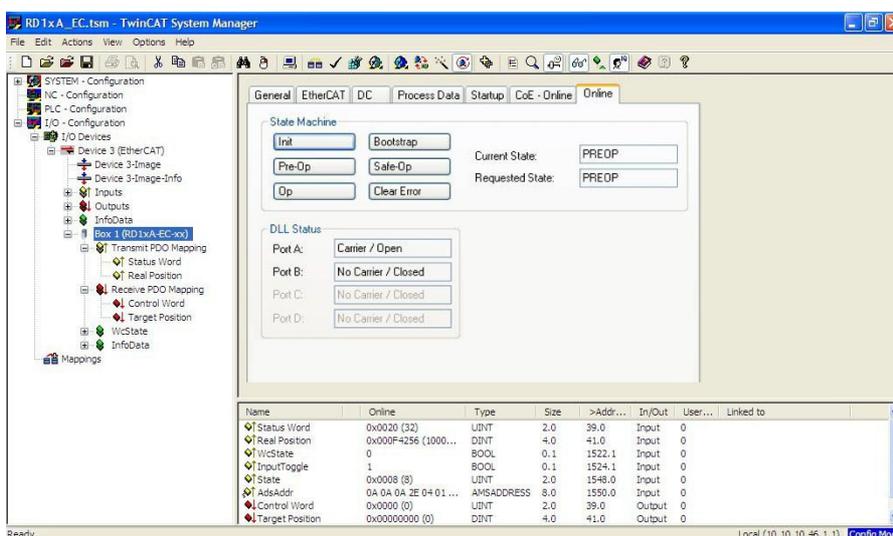
Selezionare l'elemento **Dictionary** nel riquadro di sinistra e scegliere poi l'opzione **Online - via SDO Information** nella pagina **Dictionary**; premere il pulsante **OK** per confermare.

Versione Online del dizionario oggetti come viene letta dall'attuatore.



## 7.5 Dati Online

Nel menu ad albero della finestra principale **TwinCAT System Manager** premere l'elemento **Box (RD1xA-EC-xx)**: nella schermata principale compariranno alcune pagine a scheda per la configurazione e la gestione del dispositivo. Accedere alla pagina **Online** per controllare lo stato dell'attuatore.



Per visualizzare in tempo reale i dati di processo dell'attuatore cliccare il pulsante **Safe-OP** per visualizzare solamente gli input; oppure cliccare il pulsante **OP** per visualizzare anche gli output.



### ATTENZIONE

La struttura dei Data Object (PDO e SDO) prevede l'inserimento dei byte di dati dal meno significativo (LSB, Least Significant Byte) al più significativo (MSB, Most Significant Byte).

Nell'utilizzo di TwinCAT invece i Data byte devono essere scritti e letti da MSB a LSB.

Inoltre anche le stringhe devono essere inserite al contrario.

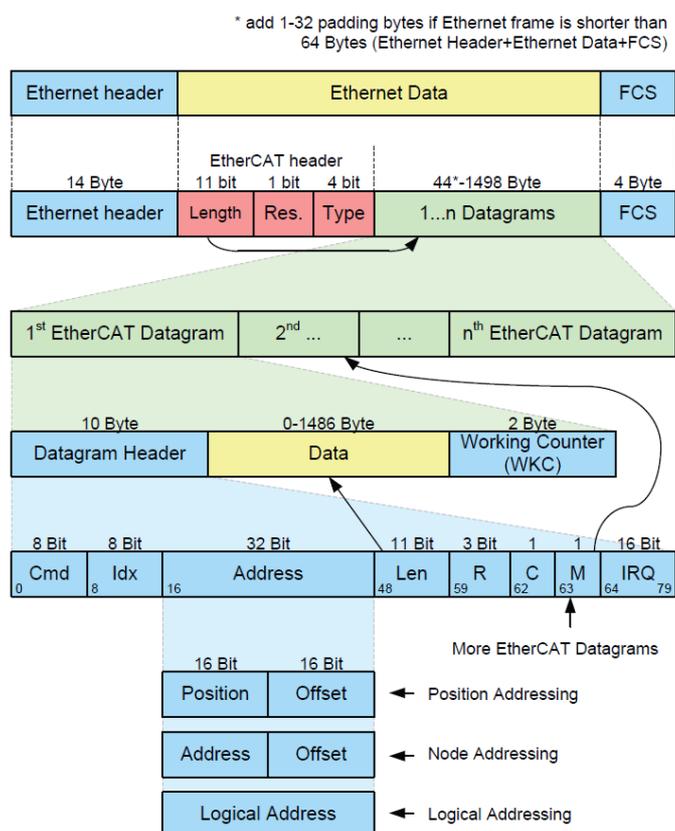
## 7.6 Nozioni di base sul protocollo EtherCAT®

Il protocollo EtherCAT si appoggia direttamente ai dataframe Ethernet standard per la trasmissione dei dati; inoltre dal punto di vista hardware ha il vantaggio di non richiedere l'utilizzo di Master dedicati in quanto vengono utilizzate le schede di rete normalmente utilizzate in qualsiasi rete Ethernet.

Un bus EtherCAT può essere visto come una singola e grande subnet Ethernet che invia e riceve telegrammi Ethernet appoggiandosi alla struttura dei dataframe Ethernet senza alterarne la struttura.

All'interno di questa subnet possono tuttavia essere presenti solo un controller Master e un certo numero di Slave EtherCAT, ma nessun Ethernet controller che invii dati in rete.

Struttura del frame Ethernet con EtherCAT:



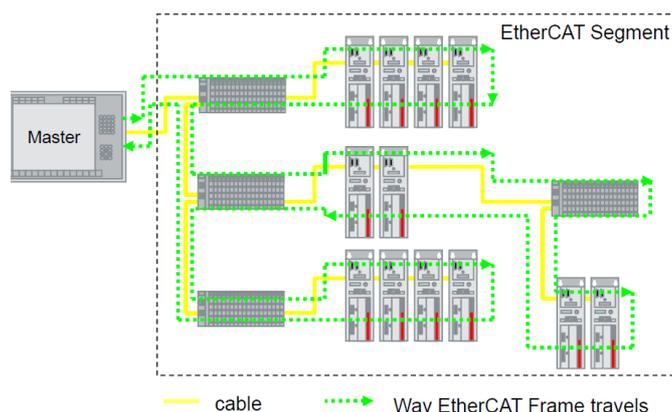
I dati all'interno del frame Ethernet vengono trasmessi tra Master e Slave sotto forma di dati di processo (PDO - Process data object). Ciascun PDO ha associato un indirizzo verso uno o più Slave; l'associazione dati + indirizzo (unitamente ad altri elementi tra cui una checksum di validazione) forma un telegramma EtherCAT (Datagram). Un frame EtherCAT può contenere più telegrammi e spesso un ciclo completo di controllo può richiedere anche più di un frame.

### 7.6.1 Trasmissione dati

Generalmente in una struttura di trasmissione dati a Bus, il controller Master invia una richiesta dati in linea e attende che questi vengano elaborati e poi restituiti da ogni nodo Slave; questo comporta una difficoltà nel rispettare le caratteristiche tipiche di un sistema "real time" in quanto il controller Master acquisisce i dati in istanti diversi -non sincroni- dai diversi Slave.

Con EtherCAT questo problema viene superato in quanto i dati vengono processati "al volo" in tempo reale utilizzando un unico frame di chiamata per tutti gli Slave della rete.

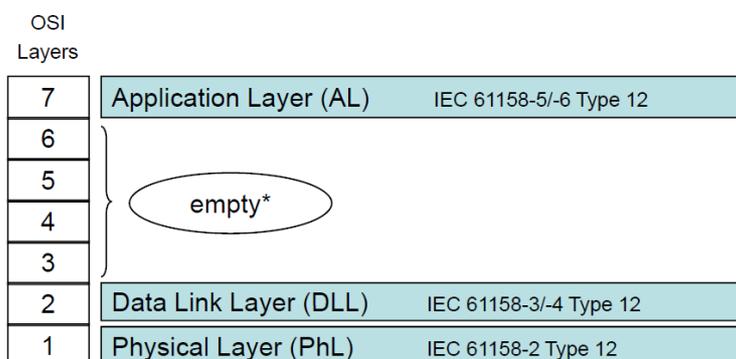
Infatti lo stesso pacchetto di richiesta dati dal Master circola su tutti gli Slave e ciascun dispositivo, se indirizzato, inserisce o preleva i dati richiesti e trasmette il frame al dispositivo successivo per ulteriori elaborazioni. I telegrammi hanno un ritardo di solo alcuni nanosecondi. L'ultimo Slave provvede a restituire a tutti gli Slave e infine al Master il frame completo con tutti i dati richiesti.



Tutto ciò è reso possibile dalla struttura 100BASE-TX full-duplex della rete EtherCAT che presenta linee separate per la trasmissione e la ricezione dei dati (le linee non sono intercambiabili, si veda a pagina 35).

Inoltre l'elaborazione del protocollo avviene all'interno dell'hardware ed è pertanto indipendente dalla CPU e dalla parte di elaborazione software.

### 7.6.2 Modello a livelli ISO/OSI



\* "empty" significa che la funzionalità del layer (livello) esiste, ma non è mostrata esplicitamente.

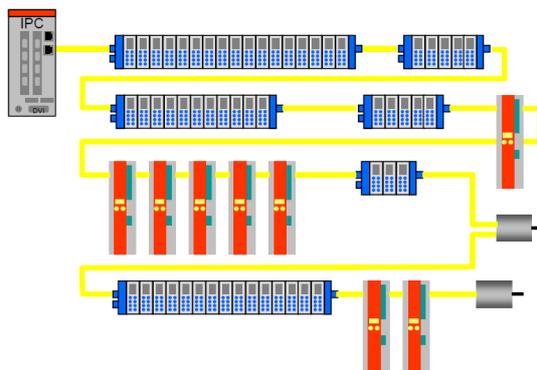
### 7.6.3 Topologia

E' possibile utilizzare diverse topologie di connessione: line, tree, daisy chain + drop line, star; esse possono essere implementate in qualunque combinazione. La lunghezza massima del cavo tra due Slave è di 100 m e si utilizzano cavi standard Ethernet.

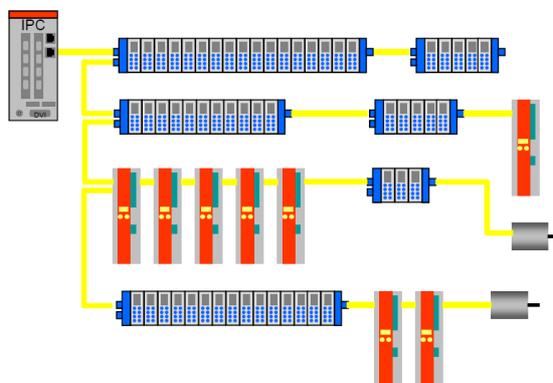
La scelta dell'una o dell'altra topologia viene fatta in base alle caratteristiche della struttura dell'impianto e alla semplicità di cablaggio.

All'interno di una rete EtherCAT si possono collegare fino a 65.535 dispositivi. Alcuni esempi di topologia sono riportati nelle Figure qui sotto:

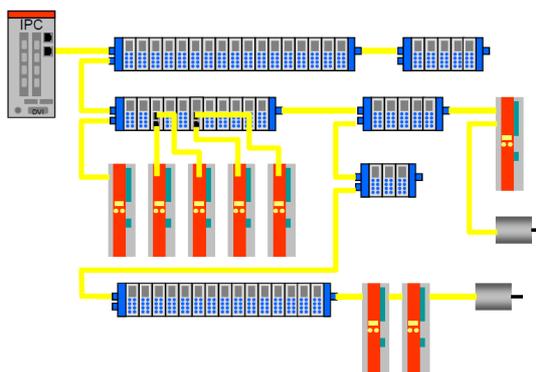
Topologia LINE:



Topologia TREE:



Topologia DAISY CHAIN con drop line:



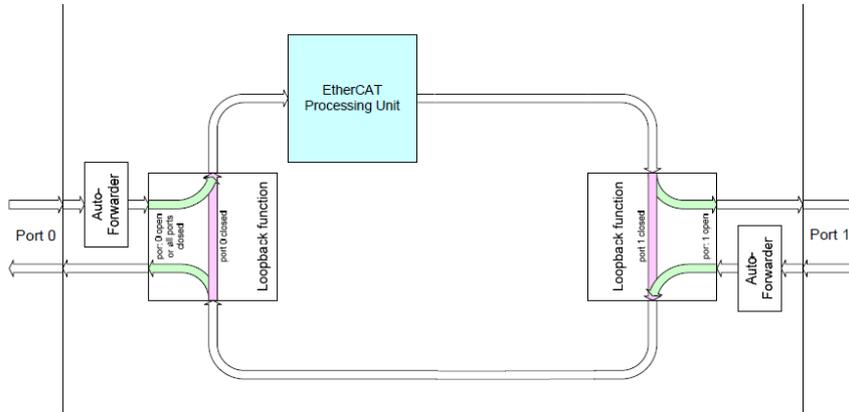
#### 7.6.4 Terminazione di linea

Non c'è la necessità di usare terminazioni di linea in quanto la chiusura della rete EtherCAT avviene in modo automatico; ogni Slave infatti è in grado di rilevare o meno la presenza di altri Slave a valle.

Uno Slave EtherCAT è in grado di rilevare la presenza del segnale sulla linea uscente (outgoing line Port 0) o sulla linea di ritorno (Return line Port 1).

Se lo Slave non rileva più il segnale sulla sua linea di ritorno allora cortocircuita il segnale TX della linea uscente con il segnale RX della linea di ritorno; in questo modo un telegramma ricevuto sulla linea uscente viene processato e rimandato indietro attraverso il TX della linea di ritorno.

Lo Slave invia senza interruzione un "carrier signal" o un telegramma sul TX della linea uscente; non appena è ripristinato lo Slave a valle viene di nuovo rilevato un segnale su RX della linea di ritorno per cui il cortocircuito viene rimosso e i telegrammi sono inviati al TX della linea uscente.



### 7.6.5 Indirizzo dispositivo

Non c'è bisogno di assegnare un indirizzo fisico al dispositivo (tramite per esempio un dip-switch) in quanto l'indirizzamento dello Slave avviene in modo automatico all'accensione del sistema durante la fase di scan della configurazione hardware.

8 Bit	8 Bit	32 Bit		11 Bit	2	1	1	1	16 Bit
Cmd	Idx	Address		Len	R	C	R	M	IRQ
		16 Bit	16 Bit						
APxx		Position	Offset	← Auto Increment Addressing (Position addressing)					
FPxx		Address	Offset	← Fixed Physical Addressing (Node addressing)					
Lxx		Logical Address		← Logical Addressing					

Il campo per l'indirizzamento ha 32 bit; EtherCAT permette tre modalità di indirizzamento:

- Auto Increment Addressing = 16 bit rappresentano la posizione fisica dello Slave nella rete e 16 bit vengono usati per indirizzare la memoria locale; quando riceve il frame, lo Slave incrementa la posizione fisica e il dispositivo che riceve Position = 0 è quello indirizzato;
- Fixed Addressing = 16 bit rappresentano l'indirizzo fisico dello Slave nella rete e 16 bit vengono usati per indirizzare la memoria locale – non utilizzato;
- Logical Address = lo Slave non è indirizzato individualmente, ma legge o scrive dati in una sezione dell'intero spazio di 4Gigabyte disponibile – non utilizzato.

### 7.6.6 Modalità di comunicazione

L'interfaccia EtherCAT permette le seguenti modalità di funzionamento:

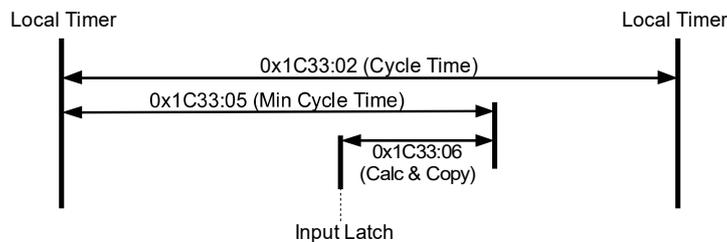
- FreeRun: modalità asincrona;
- SM3 event: modalità sincrona;
- DC: modalità di sincronismo a clock distribuiti (modalità sincrona).

**Gli attuatori Lika con interfaccia EtherCAT supportano solamente la modalità di funzionamento FreeRun.**

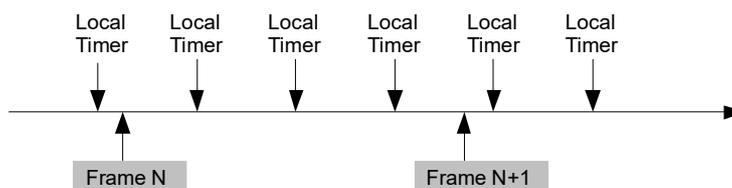
Per un sistema che necessita di alte prestazioni in tempo reale (applicazioni ad anello chiuso) si può usare la modalità DC; nel caso in cui la necessità di real time sia un requisito di minore importanza si possono usare SM3 e FreeRun. In particolare un parametro di riferimento è rappresentato dal "Jitter" ossia la variazione nel tempo dell'istante di campionamento del dato; in altre parole, il dato campionato dal controllore è reso disponibile nella memoria DPRAM dell'EtherCAT controller dopo un certo tempo con una fascia di incertezza pari al jitter.

#### FreeRun

Modalità di funzionamento asincrona in cui i TxPDO e gli RxPDO sono prelevati direttamente dal frame EtherCAT inviato dal Master; il loro aggiornamento è effettuato da un timer interno al controllore ogni 1 millisecondo.



Questa modalità di funzionamento presenta un jitter di campionamento elevato che al massimo può valere 1 ms e può essere usata con tempi di ciclo sensibilmente maggiori rispetto al jitter se si vuole garantire un sistema real time sufficientemente prestante.



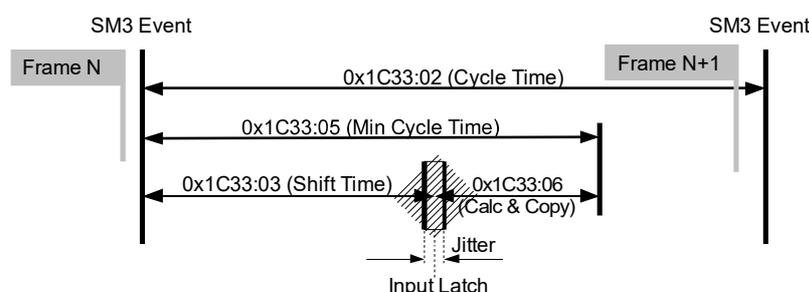
Descrizione	Min	Typ	Max	
Jitter	0		1	msec
Tempo di ciclo	1		64	msec

Si veda l'oggetto [1C33 Input Sync Manager Parameter](#) a pagina 89.

### Sincrono con SM3

**Questa modalità di funzionamento non è supportata dagli attuatori Lika.**

In questa modalità i dati sono campionati e successivamente copiati nel buffer Sync Manager non appena i dati precedenti sono stati letti dal Master (evento SM); quindi i nuovi valori campionati risultano sincroni con le letture da parte del Master.



I nuovi dati saranno letti dal Master col ciclo successivo rispetto a quello che ha generato l'evento SM per cui, se il tempo di ciclo è troppo grande, avremo dei valori relativamente vecchi per un sistema real time.

Il vantaggio principale è che l'aggiornamento dei dati avviene in modo sincrono con la lettura da parte del Master.

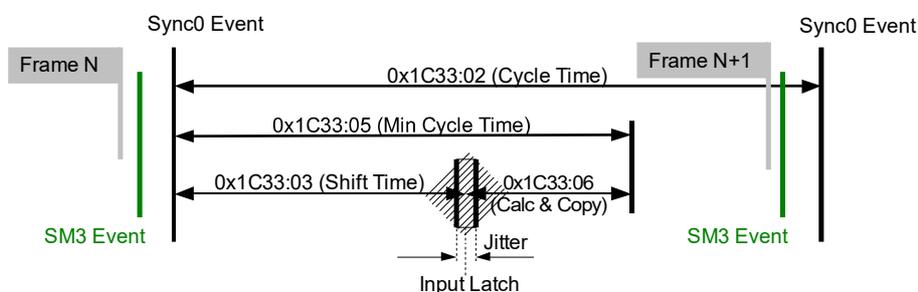
Descrizione	Min	Typ	Max	
Jitter	0		7.2	ns
Tempo di ciclo	62,5		64000	µs

### Sincrono con DC SYNC0

**Questa modalità di funzionamento non è supportata dagli attuatori Lika.**

In questa modalità di funzionamento i dati sono campionati e successivamente copiati nel buffer in corrispondenza del segnale SYNC0 generato dall'unità di capture/compare dell'ESC.

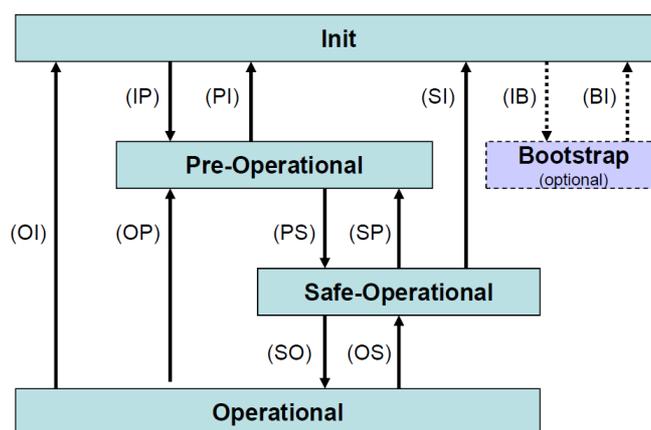
Il tempo necessario per queste operazioni è definito nell'oggetto **1C33 Input Sync Manager Parameter**, in particolare da **03 Shift Time** (1C33hex, sub3) e **06 Calc and Copy Time** (1C33hex, sub6).



Un fattore importante da considerare in questa modalità di funzionamento è il jitter che si ha nel campionamento tra due dati successivi. Il vantaggio principale di questa modalità è quello di avere una relazione diretta tra l'istante di campionamento del dato e il tempo assoluto del sistema per cui, conoscendo gli shift time dei vari dispositivi, si può avere un'esatta fotografia del sistema in un determinato istante (con incertezza pari al jitter).

Descrizione	Min	Typ	Max	
Jitter	0	100	200	µsec
Tempo di ciclo	62.5		64000	µsec

### 7.6.7 EtherCAT State Machine (ESM)



Lo Slave EtherCAT è una macchina a stati; le caratteristiche di comunicazione e di funzionamento dipendono cioè dallo stato in cui si trova lo Slave:

- **INIT**: è lo stato di default dopo l'accensione; in questo stato non c'è comunicazione diretta tra Master e Slave sull'Application Layer; è inizializzata una serie di registri di configurazione ed effettuata la configurazione dei Sync Manager;
- **PRE-OPERATIONAL** (PREOP): in questo stato è attiva la mailbox; il Master e lo Slave possono usare la mailbox e i corrispondenti protocolli per scambiare specifici parametri di inizializzazione per l'applicazione. Non è possibile lo scambio di dati di processo (PDO);
- **SAFE-OPERATIONAL** (SAFEOP): in questo stato Master e Slave possono scambiarsi dati di processo solo per quanto riguarda gli input, mentre gli output rimangono nello stato **SAFE-OPERATIONAL**;
- **OPERATIONAL** (OP): in questo stato Master e Slave possono scambiarsi dati di processo, sia di input che di output.
- **BOOTSTRAP** (BOOT). Opzionale. Nessuna comunicazione di dati di processo. E' disponibile solo la comunicazione via mailbox sull'Application Layer. E' possibile una configurazione specifica della mailbox, ossia una dimensione maggiore della mailbox. In questo stato generalmente è utilizzato il protocollo FoE per il download del firmware. **Si prega di non utilizzare questo stato, l'unità potrebbe danneggiarsi irreparabilmente.**

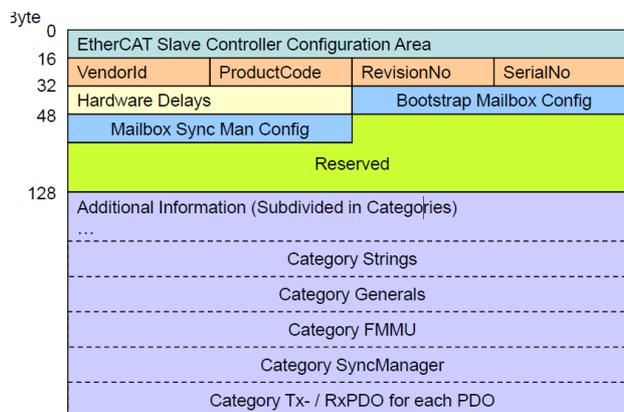


Lo stato corrente del dispositivo Slave è segnalato attraverso il LED L4, si veda pagina 38.

### 7.6.8 Configurazione dello Slave

La configurazione delle caratteristiche di comunicazione dello Slave (Configurazione Sync Manager, indirizzi, modi di sincronizzazione, PDO mapping, ecc.) può avvenire tramite il relativo file XML (EtherCAT Slave Information ESI) oppure tramite caricamento dei dati direttamente dalla EEPROM (Slave Information Interface SII).

Contenuto EEPROM (SII):



### 7.6.9 Temporizzazione e sincronizzazione

La caratteristica principale di EtherCAT è quella di rappresentare in maniera quasi ideale un sistema real time.

Per far ciò il Master deve essere in grado di sincronizzare contemporaneamente tutti i dispositivi Slave in modo tale da avere un sistema in cui tutti i nodi hanno lo stesso tempo di riferimento; questo è realizzato mediante l'uso di "clock distribuiti".

Uno degli Slave (di solito il primo) riceve dal Master il proprio clock che diventa in questo modo il clock di riferimento per tutti gli altri Slave nella rete; questo Slave è quindi incaricato di sincronizzare i clock degli altri dispositivi. Il controller Master invia periodicamente uno speciale telegramma di sincronizzazione in cui lo Slave di riferimento scrive il proprio "tempo attuale". Questo telegramma viene inviato poi a tutti gli altri Slave che provvedono in questo modo a risincronizzare il proprio clock in modo da evitare fenomeni di deriva.

Questa sincronizzazione del tempo di riferimento è di fondamentale importanza per avere una fotografia istantanea del sistema e poter così effettuare delle azioni simultanee in applicazioni particolarmente delicate quali la coordinazione nelle operazioni di controllo assi.

L'EtherCAT Slave Controller (ESC) dispone inoltre di un'unità di comparazione (capture/compare unit) in grado di generare accurati segnali di sincronismo in direzione del controllore locale (SYNCO o interrupt) che permettono al controllore stesso di sincronizzare il proprio tempo locale con quello degli Slave.

#### Sync Manager

Il Sync Manager è responsabile della sincronizzazione della trasmissione dei dati tra Master e Slave ed evita che la stessa area di memoria sia scritta contemporaneamente da diversi eventi.

Sono possibili due modalità di sincronizzazione:

- 3-Buffer Mode;
- 1-Buffer Mode.

L'inizializzazione della modalità di sincronizzazione usata avviene attraverso il file XML o caricando direttamente i dati da EEPROM (SII).

#### Buffered Mode (3-Buffer Mode)

In questa modalità si garantisce un accesso ai nuovi dati in qualsiasi momento, da parte di entrambe le parti (sia il Master EtherCAT che il controller ESC) senza nessuna restrizione di tempistiche. Sono necessari tre buffer (tre aree di memoria consecutive); un buffer è sempre disponibile per la scrittura da parte del controller ESC e un buffer contiene sempre dati aggiornati in lettura da parte del Master.

E' solitamente usato per lo scambio ciclico dei dati di processo.

### Mailbox Mode (1-Buffer Mode)

In questa modalità si deve utilizzare un "handshake" tra Master e Slave in quanto Master e Slave utilizzano un unico buffer di memoria per la scrittura e la lettura; la scrittura da parte del Master (o da parte dello Slave) può avvenire solo quando il buffer è vuoto ossia la controparte (Slave o Master) ha completato la lettura dei dati contenuti nel buffer; analogamente per quanto riguarda la lettura, essa deve avvenire solo quando il buffer è stato completamente scritto dalla controparte. Il Mailbox Mode è tipicamente utilizzato per i protocolli dell'Application Layer e lo scambio di dati aciclici (per esempio l'impostazione di parametri).

L'attuatore presenta quattro Sync Manager, si veda all'oggetto **1C00 Sync Manager Communication Type**:

- **Sync Manager 0 (SM MailBox Receive, SM0)**  
Utilizzato per trasmissioni di scrittura mailbox (da Master a Slave).  
Il modulo ha una dimensione della mailbox in scrittura configurabile, la dimensione di default è 276 byte, corrispondenti a 255 byte più i relativi protocol header e padding.
- **Sync Manager 1 (SM MailBox Send, SM1)**  
Utilizzato per trasmissioni di lettura mailbox (da Slave a Master).  
Il modulo ha una dimensione della mailbox in lettura configurabile, la dimensione di default è 276 byte, corrispondenti a 255 byte più i relativi protocol header e padding.
- **Sync Manager 2 (SM PDO output, SM2)**  
Contiene gli RxPDO (ossia, Sync Manager 2 contiene i Read Process Data).
- **Sync Manager 3 (SM PDO input, SM3)**  
Contiene i TxPDO (ossia, Sync Manager 3 contiene i Write Process Data).

### 7.6.10 Watchdog

Gli attuatori RD1xA EtherCAT dispongono di una caratteristica di sicurezza (watchdog) che blocca le uscite dopo il tempo impostato per esempio nel caso di interruzione del traffico dei dati di processo, in funzione del dispositivo e delle sue impostazioni, per esempio nello stato OFF. L'unità è ripristinata nello stato **SAFE-OPERATIONAL**. Oltre alla funzionalità watchdog standard, sono principalmente implementati tre tipi di watchdog:

- SM watchdog (default: 100 ms)
- Output I/O watchdog (default: 100 ms)
- PDI watchdog (default: 100 ms)

### SM watchdog (SyncManager Watchdog)

Il SyncManager watchdog è resettato dopo ogni comunicazione di dati di processo EtherCAT con il terminale andata a buon fine. Se non si verifica

nessuna comunicazione di dati di processo EtherCAT con l'attuatore per un tempo di SM watchdog maggiore di quello impostato e attivato, per esempio nell'eventualità di un'interruzione della rete, il watchdog si innesca e le uscite sono impostate a FALSE. Lo stato OP del terminale non viene toccato. Il watchdog si resetta solo dopo che un accesso ai dati di processo EtherCAT è riuscito con successo.

Il SyncManager watchdog monitora la comunicazione dei dati di processo corretta e puntuale con L'ESC dal lato EtherCAT.

#### **Output I/O Sync Manager Watchdog**

Se abilitato, questo watchdog monitora la comunicazione PDO verso il modulo. Se il Master non aggiorna i Read Process Data entro il lasso di tempo specificato, questo attiverà una condizione di timeout nel modulo, che lo porterà a passare dallo stato **OPERATIONAL** allo stato **SAFE-OPERATIONAL**.

#### **PDI watchdog (Process Data Watchdog)**

Se non si realizza nessuna comunicazione PDI con l'EtherCAT Slave Controller (ESC) per un tempo di PDI watchdog maggiore di quello impostato e attivato, si innesca questo watchdog. PDI (Process Data Interface) è l'interfaccia interna tra l'ESC e i processori locali nello Slave EtherCAT, per esempio. Il PDI watchdog può essere utilizzato per monitorare la presenza di errori in questa comunicazione.

Il PDI watchdog monitora la comunicazione dei dati di processo corretta e puntuale con L'ESC dal lato applicazione.

Le impostazioni del watchdog SM e PDI devono essere fatte separatamente per ciascun Slave nel TwinCAT System Manager.

## 7.7 CANopen Over EtherCAT (CoE)

Gli attuatori Lika sono dispositivi Slave e utilizzano il protocollo "CANopen Over EtherCAT (CoE)" per la trasmissione dei dati; in particolare supportano il "CANopen DS301 Communication profile".

Per ogni specifica omessa relativa al protocollo CANopen® fare riferimento al documento "CiA Draft Standard Proposal 301 CANopen Application layer and communication profile" disponibile sul sito [www.can-cia.org](http://www.can-cia.org).

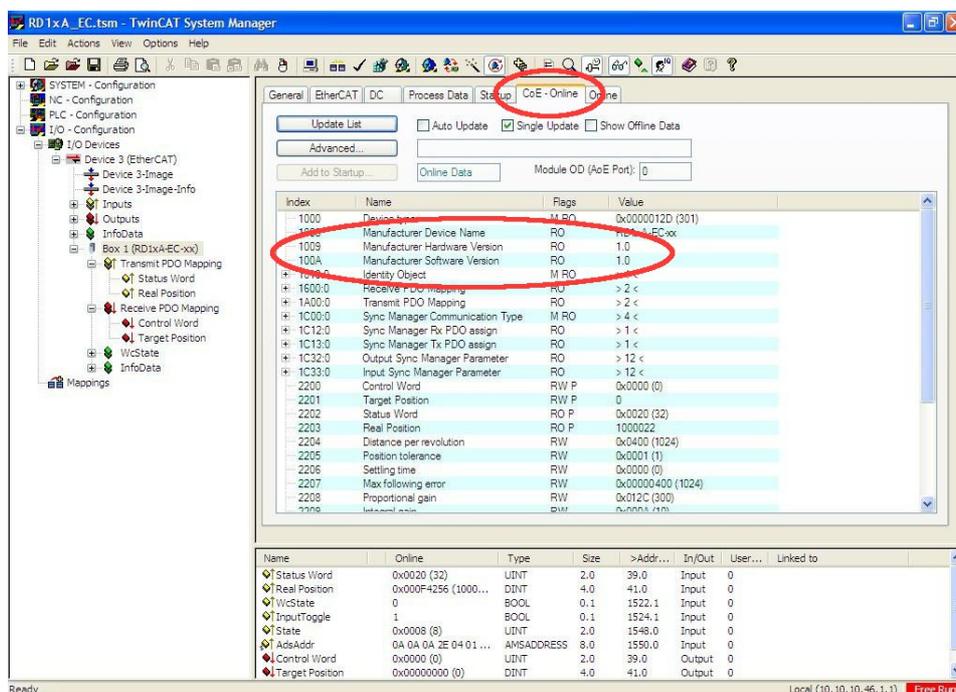
Per ogni specifica omessa relativa al protocollo EtherCAT fare riferimento al documento "ETG.1000 EtherCAT Specification" disponibile sul sito [www.ethercat.org](http://www.ethercat.org).

### 7.7.1 File XML

Gli attuatori EtherCAT® sono forniti con un loro file XML **Lika\_RDxx\_EC\_Vx.xml** (versione V2 o superiore; si veda all'indirizzo [www.lika.it](http://www.lika.it) > **PRODOTTI** > **ATTUATORI ROTATIVI**).

Il file XML deve essere installato sul dispositivo Master EtherCAT®.

Per conoscere la versione del firmware del dispositivo, nella finestra principale di **TwinCAT System Manager** selezionare il **Box (RD1xA-EC-xx)**: compaiono le finestre a scheda dedicate alla configurazione del Box; accedere alla pagina **CoE – Online** e riferirsi agli indici **1009-00 Manufacturer Hardware version** e **100A-00 Manufacturer Software version**.



### 7.7.2 Tipi di messaggi

La modalità CoE prevede la seguente struttura dell'EtherCAT Datagram:

Mbx Header	CoE Cmd			Cmd specific data
type = 3	Number	res	Type	
6 byte	9 bit	3 bit	3 bit	0 ... 1478 byte

Mbx Header = 3 modalità CoE

Number = 0 in caso di messaggi SDO

≠ 0 in caso di messaggi PDO, specifica il tipo di servizio

res bit riservati

Type = 0 valore riservato

= 1 messaggio Emergency

= 2 richiesta SDO

= 3 risposta SDO

= 4 PDO trasmessi (TxPDO)

= 5 PDO ricevuti (RxPDO)

= 6 Remote transmission request dei PDO trasmessi

= 7 Remote transmission request dei PDO ricevuti

= 8 informazione SDO

= 9 ... 15 valori riservati

Cmd specific data messaggi PDO: sono i dati di processo, per esempio il valore di posizione

messaggi SDO: frame CANopen standard

"Type" si intende trasmesso (tx) o ricevuto (rx) dal punto di vista del nodo Slave.

### 7.7.3 Process Data Object (PDO)

I messaggi PDO sono usati per trasmettere o ricevere dati di processo in tempo reale; i dati da trasmettere o da ricevere sono definiti nella PDO Mapping e gestiti dalla Sync Manager PDO Mapping.

### 7.7.4 Service Data Object (SDO)

I messaggi SDO sono trasmessi tramite Mailbox (dati a bassa priorità); non sono supportati il Segmented SDO Service e l'SDO Complete Access (trasmissione di dati con dimensioni contenute e un sotto-indice per volta). Sono utilizzati per accedere al "Dizionario Oggetti" per leggere o modificare i parametri in esso contenuti.

"CoE Cmd type" = 2 o 3

Struttura del "Cmd specific data":

Cmd specific data				
SDO control	Index	Sub index	Data	Data optional
8 bit	16 bit	8 bit	32 bit	1 ... 1470 byte

SDO control	comando per SDO CANopen standard
Index	indice del parametro da leggere o scrivere
Sub index	secondo indice (sotto-indice) del parametro da leggere o scrivere
Data	valore letto o scritto del parametro
Data optional	in via opzionale è possibile inviare in un frame più di 4 byte di dati. E' utilizzabile l'intera dimensione della mailbox.

I valori di indice e sotto-indice sono descritti nel "Dizionario oggetti".

## 7.8 Dizionario oggetti

La parte più importante del profilo di un dispositivo è il Dizionario Oggetti (Object Dictionary). Il Dizionario Oggetti è essenzialmente un insieme di oggetti accessibili attraverso la rete in maniera ordinata e predefinita. Ciascun oggetto all'interno del dizionario è indirizzato utilizzando un indice a 16 bit.

Il Dizionario Oggetti può contenere un massimo di 65.536 voci.

Gli oggetti che hanno rilevanza per l'utilizzatore sono raggruppati in due aree principali: la Communication Profile Area e la Manufacturer Specific Profile Area. Tutti gli oggetti sono descritti nel file XML.

La **Communication Profile Area** agli indici da 1000h a 1FFFh contiene i parametri specifici di comunicazione nella rete EtherCAT. Queste voci sono comuni a tutti i dispositivi. Gli oggetti PDO e gli oggetti SDO sono descritti in questa sezione. Gli oggetti nella Communication Profile Area sono conformi al "CiA Draft Standard Proposal 301 CANopen Application layer and communication profile". Riferirsi alla sezione "7.8.1 Oggetti della Communication Profile Area (DS 301)" a pagina 84.

La **Manufacturer Specific Profile Area** agli indici da 2000h a 22FFh è liberamente disponibile per l'aggiunta di funzionalità da parte del costruttore. Riferirsi alla sezione "7.8.2 Oggetti della Manufacturer Specific Profile Area" a pagina 91.

Di seguito sono riportati gli oggetti implementati nel dispositivo, per ciascuno è indicato:

### Indice - sotto-indice Nome dell'oggetto

[tipo di dato, attributo]

- Indice e sotto-indice sono espressi in notazione esadecimale.
- Attributo:
  - ro = accesso in sola lettura
  - rw = accesso in lettura e scrittura

Tipo di dato Signed8 / Unsigned8:

Process data byte							
byte 4							
7	6	5	4	3	2	1	0
MSbit		...				LSbit	

Tipo di dato Signed16 / Unsigned16:

Process data byte					
byte 4			byte 5		
7	...	0	15	...	8
LSByte			MSByte		

Tipo di dato Signed32 / Unsigned32:

Process data byte											
byte 4			byte 5			byte 6			byte 7		
7	...	0	15	...	8	23	...	16	31	...	24
LSByte			...			...			MSByte		



#### NOTA

Salvare sempre i nuovi valori dopo l'impostazione al fine di memorizzarli permanentemente nella memoria non volatile. Usare la funzione **Save parameters** disponibile nell'oggetto **2200 Control Word**, si veda a pagina 96. Nel caso di spegnimento del dispositivo i dati non salvati andranno persi.

## 7.8.1 Oggetti della Communication Profile Area (DS 301)

### 1000-00 Device type

[Unsigned32, ro]

Fornisce l'informazione sul tipo di dispositivo. Questo oggetto descrive tipo e funzionalità del dispositivo.

Default = 0000 012Dh = attuatore rotativo, DS 301

### 1008-00 Manufacturer Device Name

[String, ro]

Contiene il nome del dispositivo del costruttore, espresso in codifica ASCII.

Default = 52443178412D45432D7878 = "RD1xA-EC-xx" = attuatore rotativo RD1xA con interfaccia EtherCAT

### 1009-00 Manufacturer Hardware version

[String, ro]

Mostra la descrizione della versione hardware del costruttore, espressa in codifica ASCII.

Default = 312E30 = "1.0" = versione hardware 1.0

### 100A-00 Manufacturer Software version

[String, ro]

Mostra la descrizione della versione software del costruttore, espressa in codifica ASCII.

Default = 312E30 = "1.0" = versione software 1.0

### 1018 Identity Object

[Unsigned8, ro]

Questo oggetto fornisce informazioni generiche sul dispositivo. Il sotto-indice 00 specifica il numero di voci.

Default = 4

#### 01 Vendor ID

[Unsigned32, ro]

Fornisce il vendor ID specifico del costruttore. Il vendor ID EtherCAT coincide con il vendor ID CANopen.

Default = 0000 012Eh

#### 02 Product code

[Unsigned32, ro]

Il codice prodotto proprio del costruttore identifica una versione specifica del dispositivo.

Default = 0000 3000h = attuatore rotativo RD1xA

### 03 Revision number

[Unsigned32, ro]

Il numero di revisione proprio del costruttore è formato da una parte Major e da una parte Minor. La parte Major identifica la caratteristica specifica del dispositivo. La parte Minor identifica differenti versioni con la stessa caratteristica.

Default = dipendente dal dispositivo

7	...	0	15	...	8	23	...	16	31	...	24
Parte Minor del numero di revisione						Parte Major del numero di revisione					
LSB			...			...			MSB		

### 04 Serial number

[Unsigned32, ro]

Fornisce il numero di serie del dispositivo. E' "0" se non è fornito nessun numero di serie.

Default = 0000 0000h

## 1600 Receive PDO Mapping

[Unsigned8, ro]

Questo oggetto contiene i parametri di mappatura dei PDO che il dispositivo EtherCAT è in grado di ricevere. Il sotto-indice 00 specifica il numero di voci.

### 01 Mapped Object 001

[Unsigned32, ro]

Il sotto-indice 01 contiene l'informazione dell'oggetto dell'applicazione mappato 001. L'oggetto descrive il contenuto del PDO specificando l'indice, il sotto-indice e la lunghezza.

Il campo lunghezza contiene la lunghezza dell'oggetto dell'applicazione in bit. Può essere usato per verificare la mappatura.

7	0	15	8	23	16	31	24
Lunghezza		Sotto-indice		Indice			
LSB				MSB			

Default = 2200 0010h = oggetto **2200 Control Word**, lunghezza 16 bit

## 02 Mapped Object 002

[Unsigned32, ro]

Il sotto-indice 02 contiene l'informazione dell'oggetto dell'applicazione mappato 002. L'oggetto descrive il contenuto del PDO specificando l'indice, il sotto-indice e la lunghezza.

Il campo lunghezza contiene la lunghezza dell'oggetto dell'applicazione in bit. Può essere usato per verificare la mappatura.

7	0	15	8	23	16	31	24
Lunghezza		Sotto-indice		Indice			
LSB				MSB			

Default = 2201 0020h = oggetto **2201-00 Target Position**, lunghezza 32 bit

## 1A00 Transmit PDO Mapping

[Unsigned8, ro]

Questo oggetto contiene i parametri di mappatura dei PDO che il dispositivo EtherCAT è in grado di trasmettere. Il sotto-indice 00 specifica il numero di voci.

### 01 Mapped Object 001

[Unsigned32, ro]

Il sotto-indice 01 contiene l'informazione dell'oggetto dell'applicazione mappato 001. L'oggetto descrive il contenuto del PDO specificando l'indice, il sotto-indice e la lunghezza.

Il campo lunghezza contiene la lunghezza dell'oggetto dell'applicazione in bit. Può essere usato per verificare la mappatura.

7	0	15	8	23	16	31	24
Lunghezza		Sotto-indice		Indice			
LSB				MSB			

Default = 2202 0010h = oggetto **2202 Status Word**, lunghezza 16 bit

### 02 Mapped Object 002

[Unsigned32, ro]

Il sotto-indice 02 contiene l'informazione dell'oggetto dell'applicazione mappato 002. L'oggetto descrive il contenuto del PDO specificando l'indice, il sotto-indice e la lunghezza.

Il campo lunghezza contiene la lunghezza dell'oggetto dell'applicazione in bit. Può essere usato per verificare la mappatura.

7	0	15	8	23	16	31	24
Lunghezza		Sotto-indice		Indice			
LSB						MSB	

Default = 2203 0020h = oggetto **2203-00 Real Position**, lunghezza 32 bit

### 1C00 Sync Manager Communication Type

[Unsigned8, ro]

Questo oggetto contiene il numero e il tipo dei Sync Manager Communication Type supportati dall'attuatore. Il sotto-indice 00 specifica il numero di canali Sync Manager. Riferirsi anche alla sezione "Sync Manager" a pagina 77.

#### 01 SM MailBox Receive (SM0)

[Unsigned8, ro]

Usato per le trasmissioni di scrittura mailbox (da Master a Slave).

Default = 01

#### 02 SM MailBox Send (SM1)

[Unsigned8, ro]

Usato per le trasmissioni di lettura mailbox (da Slave a Master).

Default = 02

#### 03 SM PDO output (SM2)

[Unsigned8, ro]

Contiene gli RxPDO (ossia, Sync Manager 2 contiene i Read Process Data).

Default = 03

#### 04 SM PDO input (SM3)

[Unsigned8, ro]

Contiene i TxPDO (ossia, Sync Manager 3 contiene i Write Process Data).

Default = 04

### 1C12-00 Sync Manager RxPDO assign

[Unsigned8, ro]

Questo oggetto specifica se il dispositivo utilizza messaggi Receive PDO. Il sotto-indice 00 specifica il numero di voci, cioè il numero di RxPDO stabiliti.

#### 01 SubIndex 001

[Unsigned16, ro]

Default = 1600h = oggetto **1600 Receive PDO Mapping**

### 1C13-01 Sync Manager TxPDO assign

[Unsigned8, ro]

Questo oggetto specifica se il dispositivo utilizza messaggi Transmit PDO. Il sotto-indice 00 specifica il numero di voci, cioè il numero di TxPDO stabiliti.

**01 SubIndex 001**

[Unsigned16, ro]

Default = 1A00h = oggetto **1A00 Transmit PDO Mapping****1C32 Output Sync Manager Parameter**

[Unsigned8, ro]

L'oggetto **1C32 Output Sync Manager Parameter** contiene i parametri di sincronizzazione in uscita. Alcuni di questi sono calcolati dinamicamente e dipendono dalla configurazione dell'attuatore (risoluzione programmata, direzione di conteggio, ...) e dalla modalità di sincronizzazione scelta (nel caso in cui siano possibili più opzioni). Il sotto-indice 00 specifica il numero di voci.

**01 Synchronization Type**

[Unsigned16, ro]

Permette di scegliere la modalità di sincronizzazione. Gli attuatori Lika supportano solamente la modalità di comunicazione FreeRun. Per maggiori informazioni si veda a pagina 73.

Default = 0 = FreeRun

**02 Cycle time**

[Unsigned32, rw]

Tempo di ciclo dell'applicazione, ossia intervallo tra due campionamenti di quota (timer interno). Il valore è espresso in nanosecondi (ns).

Default = 1000000 = 1 ms

**03 Shift Time**

[Unsigned32, rw]

Intervallo tra l'evento di sincronizzazione e l'istante in cui viene fatto il latch delle uscite per l'acquisizione del dato. Questo parametro è calcolato dinamicamente ed espresso in nanosecondi (ns).

Default = 0

**04 Synchronization Types supported**

[Unsigned16, ro]

Mostra l'elenco delle modalità di sincronizzazione supportate.

Bit 0 = 1 = modalità FreeRun supportata

Default = 0001h

**05 Minimum Cycle Time**

[Unsigned32, ro]

Durata minima del tempo di ciclo interno dell'attuatore. Questo parametro è calcolato dinamicamente e dipende dai parametri operativi e dal valore di posizione. E' espresso in nanosecondi (ns).

Default = 1000000 = 1 ms

**06 Calc and Copy Time**

[Unsigned32, ro]

Tempo necessario al micro-controller interno (DSP) per effettuare i calcoli sul dato campionato e copiarlo dalla memoria locale a quella ESC (Sync Manager) prima di renderlo disponibile in EtherCAT. Questo parametro è calcolato

dinamicamente e dipende dai parametri operativi e dal valore di posizione. E' espresso in nanosecondi (ns).

Default = 0000 0000

#### **09 Delay Time**

[Unsigned32, ro]

Ritardo espresso in nanosecondi (ns). E' sempre impostato a 0.

Default = 0000 0000

#### **0C Cycle Time Too Small**

[Unsigned16, ro]

Contatore degli errori per tempi di ciclo troppo brevi.

Default = 0000

### **1C33 Input Sync Manager Parameter**

[Unsigned8, ro]

L'oggetto **1C33 Input Sync Manager Parameter** contiene i parametri di sincronizzazione in ingresso. Alcuni di questi sono calcolati dinamicamente e dipendono dalla configurazione dell'attuatore (risoluzione programmata, direzione di conteggio, ...) e dalla modalità di sincronizzazione scelta (nel caso in cui siano possibili più opzioni). Il sotto-indice 00 specifica il numero di voci.

#### **01 Synchronization Type**

[Unsigned16, ro]

Permette di scegliere la modalità di sincronizzazione. Gli attuatori Lika supportano solamente la modalità di comunicazione FreeRun. Per maggiori informazioni si veda a pagina 73.

Default = 0 = FreeRun

#### **02 Cycle time**

[Unsigned32, rw]

Tempo di ciclo dell'applicazione, ossia intervallo tra due campionamenti di quota (timer interno). Il valore è espresso in nanosecondi (ns).

Default = 1000000 = 1 ms

#### **03 Shift Time**

[Unsigned32, rw]

Intervallo tra l'evento di sincronizzazione e l'istante in cui viene fatto il latch degli ingressi per l'acquisizione del dato. Questo parametro è calcolato dinamicamente ed espresso in nanosecondi (ns).

Default = 0

#### **04 Synchronization Types supported**

[Unsigned16, ro]

Mostra l'elenco delle modalità di sincronizzazione supportate.

Bit 0 = 1 = modalità FreeRun supportata

Default = 0001h

**05 Minimum Cycle Time**

[Unsigned32, ro]

Durata minima del tempo di ciclo interno dell'attuatore. Questo parametro è calcolato dinamicamente e dipende dai parametri operativi e dal valore di posizione. E' espresso in nanosecondi (ns).

Default = 1000000 = 1 ms

**06 Calc and Copy Time**

[Unsigned32, ro]

Tempo necessario al micro-controller interno (DSP) per effettuare i calcoli sul dato campionato e copiarlo dalla memoria locale a quella ESC (Sync Manager) prima di renderlo disponibile in EtherCAT. Questo parametro è calcolato dinamicamente e dipende dai parametri operativi e dal valore di posizione. E' espresso in nanosecondi (ns).

Default = 0000 0000

**0C Cycle Time Too Small**

[Unsigned16, ro]

Contatore degli errori per tempi di ciclo troppo brevi.

Default = 0000

**NOTA**

Salvare sempre i nuovi valori dopo l'impostazione al fine di memorizzarli permanentemente nella memoria non volatile. Usare la funzione **Save parameters** disponibile nell'oggetto **2200 Control Word**, si veda a pagina 96.

Nel caso di spegnimento del dispositivo i dati non salvati andranno persi.

## 7.8.2 Oggetti della Manufacturer Specific Profile Area

### 2101-00 HMS Serial Number

[Unsigned32, ro]

Visualizza il numero di serie del modulo HMS.

Valore = dipendente dal dispositivo

### 2102-00 HMS\_FW\_Major

[Unsigned8, ro]

Il numero della revisione firmware HMS è formato da una parte Major e da una parte Minor. In questo oggetto viene visualizzata la parte Major del numero di revisione.

Valore = dipendente dal dispositivo

### 2103-00 HMS\_FW\_Minor

[Unsigned8, ro]

Il numero della revisione firmware HMS è formato da una parte Major e da una parte Minor. In questo oggetto viene visualizzata la parte Minor del numero di revisione.

Valore = dipendente dal dispositivo

### 2104-00 HMS\_FW\_Build

[Unsigned8, ro]

Visualizza il numero build del firmware del modulo HMS.

Valore = dipendente dal dispositivo

### 2105-00 Position Offset

[Signed32, ro]

Questa variabile definisce la differenza tra il valore di posizione trasmesso dal dispositivo e la posizione reale: posizione reale – preset. Il valore è espresso in impulsi.

### 2106-00 Real Speed [rpm]

[Signed32, ro]

Velocità del dispositivo espressa in giri al minuto [rpm], aggiornata ogni secondo. Questa è la velocità del motore, non la velocità dell'albero in uscita dopo il riduttore. La velocità in uscita sarà come segue:

Velocità del motore = 2000 rpm

Velocità in uscita: T12 = 166 rpm

T24 = 83 rpm

T48 = 41 rpm

T92 = 21 rpm

### 2107-00 Electronics Temperature [°C]

[Signed8, ro]

Questa variabile visualizza la temperatura dell'elettronica rilevata dalle sonde interne. Il valore è espresso in gradi Celsius (°C). La temperatura minima rilevabile è -20°C.

### 2108-00 Motor Temperature [°C]

[Signed8, ro]

Questa variabile visualizza la temperatura del motore rilevata dalle sonde interne. Il valore è espresso in gradi Celsius (°C). La temperatura minima rilevabile è -20°C.

### 2109-00 Real Current

[Signed32, ro]

Questa variabile visualizza il valore della corrente assorbita dal motore (corrente nominale). Il valore è espresso in milliamper (mA).

### 210A-00 Following error [pulse]

[Signed32, ro]

Questa variabile contiene la differenza tra la posizione richiesta e la posizione attuale istante per istante. Se questo valore è superiore a quello impostato nell'oggetto **Max following error [0x03-0x04]**, il dispositivo genera l'allarme **Following error** e arresta il proprio movimento. Il valore è espresso in impulsi.

### 210B-00 Pos. Limit Switch [pulse]

[Signed32, ro]

È il valore **SW limit switch +** (massimo spostamento positivo) calcolato sulla scorta dei valori impostati negli oggetti **2211-00 Preset** e **220C-00 Max delta pos**. Se si raggiunge il massimo spostamento in avanti viene attivata la segnalazione sul bit di stato 3 **SW limit switch +** della **2202 Status Word**.

**SW limit switch + = 2211-00 Preset + 220C-00 Max delta pos.**

Il valore è espresso in impulsi.

Riferirsi anche all'ESEMPIO 1 nella sezione "6.4 Distance per revolution, Preset, Max delta pos / Positive delta e Max delta neg / Negative delta" a pagina 51.

Default = 523.263

### 210C-00 Neg. Limit Switch [pulse]

[Signed32, ro]

E' il valore **SW limit switch** - (massimo spostamento negativo) calcolato sulla scorta dei valori impostati negli oggetti **2211-00 Preset** e **220D-00 Max delta neg.** Se si raggiunge il massimo spostamento all'indietro viene attivata la segnalazione sul bit di stato 4 **SW limit switch** - della **2202 Status Word**.

**SW limit switch** - = **2211-00 Preset** - **220D-00 Max delta neg.**

Il valore è espresso in impulsi.

Riferirsi anche all'ESEMPIO 1 nella sezione "6.4 Distance per revolution, Preset, Max delta pos / Positive delta e Max delta neg / Negative delta" a pagina 51.

Default = - 523.263

### 210D Parameter Error List

[Unsigned32, ro]

L'operatore ha impostato valori non validi e il dispositivo ha generato l'allarme **Machine data not valid**. Questa variabile visualizza quali parametri contengono valori errati, secondo la lista riportata nella seguente tabella.

Si badi che si può ripristinare il normale stato di lavoro solo impostando dati macchina validi.

Bit	Parametro
0	Non usato
1	<b>2204-00 Distance per revolution</b>
2	<b>220A-00 Acceleration</b>
3	<b>220B-00 Deceleration</b>
4	<b>220C-00 Max delta pos</b>
5	<b>220D-00 Max delta neg</b>
6	<b>220E-00 Jog speed</b>
7	<b>220F-00 Work speed</b>
8	<b>2210-00 Count direction</b>
9	<b>2211-00 Preset</b>
10	<b>2212-00 Step jog</b>
11	<b>2208-00 Proportional gain</b>
12	<b>2209-00 Integral gain</b>
13	<b>2206-00 Settling time</b>
14	<b>2207-00 Max following error</b>
15	Non usato

## 210E Alarms List

[Unsigned16, ro]

Questa variabile visualizza gli allarmi correntemente attivi nel dispositivo.

Struttura del byte allarmi:

byte	LSB			MSB		
bit	7	...	0	15	...	8
	msb		lsb	msb		lsb

Codici registro allarmi previsti:

### Byte 0

#### Machine data not valid

bit 0 Uno o più parametri non sono validi, impostare valori corretti per ristabilire la normale condizione di lavoro. Controllare la lista dei parametri errati nell'oggetto **210D Parameter Error List**.

#### Flash memory error

bit 1 Errore interno non ripristinabile.

#### Counting error

bit 2 Per motivi di sicurezza, la posizione assoluta e quella incrementale dell'encoder integrato sono lette e salvate in due registri separati. Nel caso in cui si riscontri una differenza tra i valori nei due registri viene segnalato l'allarme.

#### Following error

bit 3 La differenza tra la posizione reale e quella teorica è maggiore di quella impostata nell'oggetto **2207-00 Max following error**; si consiglia di ridurre la velocità di lavoro.

#### Axis not synchronized

bit 4 Errore interno non ripristinabile.

#### Target not valid

bit 5 E' stata confermata una posizione comandata che supera i finecorsa.

#### Emergency

bit 6 Il bit 7 **Emergency** nella **2200 Control Word** è stato forzato a 0; oppure sono attivi allarmi nel dispositivo.

#### Overcurrent

bit 7 Sovracorrente motore.

**Byte 1****Electronics Overtemperature**

bit 8 La temperatura dei MOSFET rilevata da una sonda interna supera il valore massimo ammesso (si veda l'oggetto **2107-00 Electronics Temperature [°C]** a pagina 92). Attendere alcuni minuti per permettere all'attuatore di raffreddarsi. Assicurarsi che la temperatura operativa sia compresa nel range permesso.

**Motor Overtemperature**

bit 9 La temperatura del motore rilevata da una sonda interna supera il valore massimo ammesso (si veda l'oggetto **2108-00 Motor Temperature [°C]** a pagina 92). Attendere alcuni minuti per permettere all'attuatore di raffreddarsi. Assicurarsi che la temperatura operativa sia compresa nel range permesso.

**Undervoltage**

bit 10 Il valore della tensione di alimentazione è inferiore al minimo consentito. Assicurarsi che la tensione di alimentazione sia compresa nel range ammesso.

**Watch dog**

bit 11 Il watchdog è una caratteristica di sicurezza che blocca le uscite dopo il tempo impostato per esempio nel caso di interruzione del traffico dei dati di processo, in funzione del dispositivo e delle sue impostazioni, per esempio nello stato OFF. Quando la caratteristica di sicurezza Watchdog è abilitata nel tool di configurazione per il Master, se nessuna comunicazione di dati di processo EtherCAT ha luogo con l'attuatore oppure non si attiva nessuna comunicazione PDI con l'EtherCAT Slave Controller (ESC) per un tempo superiore a quello impostato (100 millisecondi), il sistema forza una condizione di allarme (attivazione del bit di allarme **Watch dog**). Quando si genera il SyncManager watchdog, le uscite sono impostate a FALSE, lo stato OP del terminale invece non viene toccato. Il watchdog PDI può essere utilizzato per monitorare la presenza di errori di comunicazione. Per maggiori informazioni riferirsi alla sezione "7.6.10 Watchdog" a pagina 78.

bit 12 e bit 13 Non usati.

**Hall sequence**

bit 14 Si è verificato un errore nella sequenza di commutazione dei sensori di Hall.

**Overvoltage**

bit 15

Il valore della tensione di alimentazione è superiore al massimo consentito. Assicurarsi che la tensione di alimentazione sia compresa nel range ammesso.

Se l'allarme si genera durante l'operazione di frenatura, considerare la tensione controelettromotrice (back EMF). Per prevenire il verificarsi di questa evenienza, diminuire la rampa della velocità o valutare attentamente le caratteristiche dell'alimentatore 24V (modulo condensatore).

Per ripristinare una condizione di errore utilizzare il comando **Alarm reset**, bit 3 della **2200 Control Word**. In una normale condizione di lavoro il bit **Alarm reset** ha valore "0". Impostando il valore a "1" si ristabilisce il normale stato operativo del dispositivo. Il normale stato operativo è ripristinato nella commutazione da "0" a "1". Questo comando libera lo Slave dalla condizione di allarme solo se sono state rimosse le cause che ne hanno provocato l'errore.



Si badi che, se l'allarme è relativo a dati macchina non validi (si veda **Machine data not valid** e **210D Parameter Error List**), si può ripristinare lo stato normale di funzionamento solo impostando dati macchina validi. Gli allarmi **Flash memory error** e **Axis not synchronized** non sono ripristinabili.

**2200 Control Word**

[Unsigned16, rw]

Questa variabile contiene i comandi da inviare in tempo reale allo Slave per controllarlo. E' mappato nell'oggetto **1600 Receive PDO Mapping** che contiene i parametri di mappatura dei PDO che il dispositivo EtherCAT è in grado di ricevere, si veda a pagina 85. E' aggiornato ogni 1 msec.

Struttura byte dell'oggetto **2200 Control Word**:

byte	LSB			MSB		
bit	7	...	0	15	...	8
	msb		lsb	msb		lsb

**Byte 0**

**Jog +**

bit 0

Se il bit 4 **Incremental jog** = 0, lo Slave si muove in direzione positiva fintanto che **Jog +** = 1; se invece il bit 4 **Incremental jog** = 1, in corrispondenza del fronte di salita di **Jog +** lo Slave esegue un singolo passo in direzione

positiva la cui ampiezza, espressa in impulsi, è impostata nell'oggetto **2212-00 Step jog**; quindi si arresta in attesa di un nuovo comando. La velocità, l'accelerazione e la decelerazione sono definite negli oggetti **220E-00 Jog speed**, **220A-00 Acceleration** e **220B-00 Deceleration** rispettivamente. Per una descrizione dettagliata del controllo jog si veda a pagina 48.



Le funzioni **Jog +**, **Jog -** e **Start** non possono essere abilitate simultaneamente. Per esempio: se un comando **Jog +** viene inviato allo Slave mentre questo sta eseguendo il movimento verso la posizione target, il comando jog viene ignorato; se i comandi **Jog +** e **Jog -** sono inviati simultaneamente, il dispositivo non si muove o, se già in movimento, si arresta.

**Jog -**  
bit 1

Se il bit 4 **Incremental jog** = 0, lo Slave si muove in direzione negativa fintanto che **Jog -** = 1; se invece il bit 4 **Incremental jog** = 1, in corrispondenza del fronte di salita di **Jog -** lo Slave esegue un singolo passo in direzione negativa la cui ampiezza, espressa in impulsi, è impostata nell'oggetto **2212-00 Step jog**; quindi si arresta in attesa di un nuovo comando. La velocità, l'accelerazione e la decelerazione sono definite negli oggetti **220E-00 Jog speed**, **220A-00 Acceleration** e **220B-00 Deceleration** rispettivamente. Per una descrizione dettagliata del controllo jog si veda a pagina 48.



Le funzioni **Jog +**, **Jog -** e **Start** non possono essere abilitate simultaneamente. Per esempio: se un comando **Jog +** viene inviato allo Slave mentre questo sta eseguendo il movimento verso la posizione target, il comando jog viene ignorato; se i comandi **Jog +** e **Jog -** sono inviati simultaneamente, il dispositivo non si muove o, se già in movimento, si arresta.

**Stop**  
bit 2

Se impostato a "=1" lo Slave è libero di eseguire i comandi di movimento ricevuti. Se durante il movimento questo bit commuta a "=0" allora lo Slave si arresta eseguendo la decelerazione impostata in **220B-00 Deceleration**. Per un arresto immediato del movimento, utilizzare il bit 7 **Emergency**.

**Alarm reset**  
bit 3

Questo comando è usato per ripristinare una condizione di allarme dello Slave, ma solo se sono state rimosse le cause

che hanno provocato l'errore. In una normale condizione di lavoro il bit ha valore "0". Impostando questo bit a "1" si ristabilisce il normale stato di lavoro del dispositivo. Il normale stato operativo è ripristinato nella commutazione da "0" a "1".



Si badi che, se l'allarme è relativo a dati macchina non validi (si vedano **Machine data not valid** e **210D Parameter Error List**), si può ripristinare lo stato normale di funzionamento solo impostando dati macchina validi. Gli allarmi **Flash memory error** e **Axis not synchronized** non sono ripristinabili

### Incremental jog

bit 4

Se il bit 4 è "=0", l'attivazione dei bit **Jog +** e **Jog -** procura il movimento dello Slave fintanto che **Jog + / Jog - = 1**. Impostando questo bit a 1 si abilita la funzione di jog incrementale (jog a passo), ossia: l'attivazione dei bit **Jog +** e **Jog -** procura, in corrispondenza del fronte di salita, l'esecuzione di un singolo passo in direzione positiva o negativa la cui ampiezza, espressa in impulsi, è impostata all'oggetto **2212-00 Step jog**; quindi lo Slave si arresta in attesa di un nuovo comando.



Si badi che nell'utilizzo dei pulsanti manuali di jog (si veda la sezione "4.5.1 Pulsanti JOG + e JOG - (Figura 8)" a pagina 43) la funzione di jog incrementale è disabilitata; non è cioè possibile eseguire passi jog utilizzando i pulsanti manuali.

bit 5

Non usato.

### Start

bit 6

Se impostato a "=1" il dispositivo si muove allo scopo di raggiungere la posizione di target specificata (si veda **2201-00 Target Position** a pagina 101). Per una descrizione completa del controllo di posizione si veda a pagina 49.



Le funzioni **Jog +**, **Jog -** e **Start** non possono essere abilitate simultaneamente. Per esempio: se un comando **Jog +** viene inviato allo Slave mentre questo sta eseguendo il movimento verso la posizione target, il comando jog viene ignorato; se i comandi **Jog +** e **Jog -** sono inviati simultaneamente, il dispositivo non si muove o, se già in movimento, si arresta.

### Emergency

bit 7

Questo bit deve essere normalmente alto ("=1") altrimenti il dispositivo bloccherà istantaneamente ogni proprio movimento. Per un arresto normale (non immediato) nel rispetto della decelerazione programmata usare il bit 2 **Stop**.

All'avvio il bit è forzato basso (=“0”) per ragioni di sicurezza. Commutarlo al valore alto (=“1”) per ripristinare il normale funzionamento.

**Byte 1**

bit 8

Non usato.

**Save parameters**

bit 9

I dati sono salvati nella memoria non volatile in corrispondenza di ogni fronte di salita del bit; in altre parole, il salvataggio è eseguito ogniqualvolta il bit commuta dal livello logico basso (“0”) al livello logico alto (“1”).



Salvare sempre i nuovi valori dopo l'impostazione al fine di memorizzarli permanentemente nella memoria non volatile. Nel caso di spegnimento del dispositivo i dati non salvati andranno persi.

**Load default parameters**

bit 10

I parametri di default (parametri impostati in azienda dai tecnici Lika durante la messa a punto del dispositivo che ne permettono un funzionamento standard e sicuro) sono ripristinati in corrispondenza di ogni fronte di salita del bit; in altre parole, l'operazione di caricamento dei parametri di default è eseguita ogniqualvolta il bit commuta dal livello logico basso (“0”) al livello logico alto (“1”). La lista completa dei dati macchina e dei relativi parametri di default preimpostati da Lika Electronic sono disponibili a pagina 186.



Salvare sempre i nuovi valori dopo l'impostazione al fine di memorizzarli permanentemente nella memoria non volatile. Nel caso di spegnimento del dispositivo i dati non salvati andranno persi.

**ATTENZIONE**

La taratura dell'unità è stata realizzata testandone il funzionamento a pieno carico; i valori di default impostati si riferiscono pertanto a un dispositivo operante in questa condizione. Essi sono altresì previsti per garantire un funzionamento standard e sicuro del dispositivo, che potrà talora non risultare ottimale né prestazionale. Si badi quindi che nella specifica applicazione può essere consigliabile, se non necessaria, la modifica dei parametri di fabbrica e in particolare dei valori di velocità, accelerazione, decelerazione e guadagno.

**Setting the preset**

bit 11 Imposta la quota attuale al valore impostato nell'oggetto **2211-00 Preset**. L'operazione è realizzata in corrispondenza di ogni fronte di salita del bit, ossia ogniqualvolta il bit commuta dal livello logico basso ("0") al livello logico alto ("1"). Consigliamo di attivare il preset quando l'attuatore è in stop. Per maggiori informazioni riferirsi alla pagina 111.

**Axis torque**

bit 12 Questa funzione è disponibile solo nella versione RD1A (modello senza freno integrato); nella versione RD12A (modello con freno) il bit 12 non è usato. Quando l'asse ha raggiunto la posizione comandata, mantiene la coppia. Se impostato "=1", con asse in posizione, il PWM è mantenuto attivo (asse in coppia a fine posizionamento). Se impostato "=0", con asse in posizione, il PWM è disattivato (la coppia è rilasciata).

**OUT 1**

bit 13 Permette di attivare / disattivare il funzionamento dell'uscita digitale 1. Il significato dell'uscita disponibile è descritto alla sezione "6.3 Ingressi e uscita digitali" a pagina 50.

**OUT 1 = 0** uscita 1 bassa (non attiva)

**OUT 1 = 1** uscita 1 alta (attiva)

**Brake disabled**

bit 14 Questa funzione è disponibile solo nella versione RD12A (modello con freno integrato); nella versione RD1A (modello senza freno) il bit 14 non è usato. Il modello RD12A è equipaggiato con un freno che si attiva all'arresto del motore inibendo così la possibilità di qualsiasi movimento dell'asse. Se si imposta questo bit "=1" il freno è disattivato e non operativo; se si imposta questo bit "=0" il freno è attivo e il suo funzionamento è gestito in modo automatico dal sistema.



Si badi che lo sblocco del freno è possibile solo in assenza di allarmi.

bit 15 Non usato.

### 2201-00 Target Position

[Signed32, rw]

Questo oggetto è mappato nell'oggetto **1600 Receive PDO Mapping** che contiene i parametri di mappatura dei PDO che il dispositivo EtherCAT è in grado di ricevere, si veda a pagina 85. E' aggiornato ogni 1 msec.

Imposta la posizione da raggiungere (target), altrimenti detta posizione comandata. Il dispositivo si muove allo scopo di raggiungere la posizione target impostata in questo oggetto quando si invia il comando **Start**, i bit **Stop** ed **Emergency** sono "=1" e il dispositivo non è in stato di allarme.

Non appena l'asse si trova all'interno dei limiti di tolleranza impostati nell'oggetto **2205-00 Position tolerance**, il bit 8 **Target position reached** in **2202 Status Word** va alto ("=1"). Quando la posizione è all'interno dei limiti di tolleranza impostati nell'oggetto **2205-00 Position tolerance**, dopo il ritardo impostato nell'oggetto **2206-00 Settling time**, il bit 0 **Axis in position** in **2202 Status Word** va alto ("=1").

Per maggiori informazioni riferirsi anche alla sezione "Posizione: controllo di posizione e velocità" a pagina 49.

Default = 0 (min. = 0, max. = compreso entro **210B-00 Pos. Limit Switch [pulse]** / **210C-00 Neg. Limit Switch [pulse]**)



#### NOTA

##### Funzione di position override

E' possibile modificare il valore della posizione di target anche al volo, mentre il dispositivo sta ancora raggiungendo una posizione di target comandata in precedenza e senza inviare un nuovo comando di **Start**. Per fare questo, semplicemente impostare un nuovo valore di target nell'oggetto **2201-00 Target Position**. Si veda anche a pagina 49.



**NOTA**

Le funzioni **Jog +**, **Jog -** e **Start** non possono essere abilitate simultaneamente. Per esempio: se un comando **Jog +** viene inviato allo Slave mentre questo sta eseguendo il movimento verso la posizione di target, il comando jog viene ignorato; se i comandi **Jog +** e **Jog -** sono inviati simultaneamente, il dispositivo non si muove o, se già in movimento, si arresta.

Se la funzione di Watchdog è abilitata, nel caso in cui il dispositivo fosse disconnesso dalla rete EtherCAT mentre sta eseguendo un movimento (per esempio a causa dell'interruzione di un cavo o di un cablaggio errato), il dispositivo arresta immediatamente il proprio movimento e attiva il bit di allarme **Watch dog**.

**2202 Status Word**

[Unsigned16, ro]

Questa variabile fornisce informazioni sullo stato attuale del dispositivo. E' mappato nell'oggetto **1A00 Transmit PDO Mapping** che contiene i parametri di mappatura dei PDO che il dispositivo EtherCAT è in grado di trasmettere, si veda a pagina 86. E' aggiornato ogni 1 msec.

Struttura byte dell'oggetto **2202 Status Word**:

byte	LSB			MSB		
bit	7	...	0	15	...	8
	msb		lsb	msb		lsb

**Byte 0**

**Axis in position**

bit 0

Il valore è "=1" quando il dispositivo raggiunge e mantiene la posizione comandata (**2201-00 Target Position**) per il tempo impostato nell'oggetto **2206-00 Settling time**. E' mantenuto attivo fintanto che l'errore di posizione è inferiore a **2205-00 Position tolerance**. Per ulteriori informazioni riferirsi alla sezione "Posizione: controllo di posizione e velocità" a pagina 49.

bit 1

Non usato.

**Drive enabled**

bit 2

Visualizza la condizione di abilitazione del motore. Questo bit è "=1" quando il motore è abilitato, ossia: il PWM è attivo e l'asse è in controllo ad anello chiuso (per esempio durante un posizionamento o un jog). E' "=0" quando il motore è disabilitato, vale a dire

quando l'azionamento che controlla il motore viene spento al termine di un posizionamento o di un jog o a seguito di un allarme.

**SW limit switch +**

bit 3

Il valore è "=1" quando si verifica che il dispositivo raggiunge il limite positivo massimo (finecorsa positivo). Per maggiori informazioni si veda all'oggetto **220C-00 Max delta pos** a pagina 107.

**SW limit switch -**

bit 4

Il valore è "=1" quando si verifica che il dispositivo raggiunge il limite negativo massimo (finecorsa negativo). Per maggiori informazioni si veda all'oggetto **220D-00 Max delta neg** a pagina 108.

**Alarm**

bit 5

Il valore è "=1" quando si attiva un allarme, per dettagli si veda la variabile **210E Alarms List** a pagina 94.

**Axis running**

bit 6

Il valore è "=0" quando il dispositivo non si sta muovendo.

Il valore è "=1" mentre il dispositivo è in movimento.

**Executing a command**

bit 7

Il valore è "=0" quando il controller non sta eseguendo nessun comando.

Il valore è "=1" mentre il controller sta eseguendo un comando.

**Byte 1****Target position reached**

bit 8

Il valore è "=1" quando il dispositivo raggiunge la posizione di target impostata nell'oggetto **2201-00 Target Position** (si trova all'interno della finestra di tolleranza impostata in **2205-00 Position tolerance**). Il bit è mantenuto attivo fino a quando non si comanda una nuova **2201-00 Target Position** o si invia un comando **Alarm reset**. Per maggiori informazioni riferirsi anche alla sezione "Posizione: controllo di posizione e velocità" a pagina 49.

**Button 1 Jog +**

bit 9

L'unità di posizionamento RD1xA è equipaggiata con tre pulsanti alloggiati all'interno della custodia e

accessibili previa rimozione di un tappo a vite. Fintanto che il pulsante 1 JOG + è mantenuto premuto, il bit 9 è forzato alto "=1"; quando il pulsante 1 non è premuto, il bit 9 è basso "=0". Per ulteriori informazioni si veda la sezione "4.4 Tappo a vite per accesso interno (Figura 4 e Figura 7)" a pagina 41 e la sezione "4.5.1 Pulsanti JOG + e JOG - (Figura 8)" a pagina 43.

#### **Button 2 Jog -** bit 10

L'unità di posizionamento RD1xA è equipaggiata con tre pulsanti alloggiati all'interno della custodia e accessibili previa rimozione di un tappo a vite. Fintanto che il pulsante 2 JOG - è mantenuto premuto, il bit 10 è forzato alto "=1"; quando il pulsante 2 non è premuto, il bit 10 è basso "=0". Per ulteriori informazioni si veda la sezione "4.4 Tappo a vite per accesso interno (Figura 4 e Figura 7)" a pagina 41 e la sezione "4.5.1 Pulsanti JOG + e JOG - (Figura 8)" a pagina 43.

#### **Button 3 Preset** bit 11

L'unità di posizionamento RD1xA è equipaggiata con tre pulsanti alloggiati all'interno della custodia e accessibili previa rimozione di un tappo a vite. Non appena si preme il pulsante 3 PRESET, il bit 11 è forzato alto "=1"; quando il pulsante 3 non è premuto, il bit 11 è basso "=0". Per ulteriori informazioni si veda la sezione "4.4 Tappo a vite per accesso interno (Figura 4 e Figura 7)" a pagina 41 e la sezione "4.5.2 Pulsante PRESET (Figura 8)" a pagina 43.

#### **PWM saturation** bit 12

La corrente erogata dall'elettronica di potenza per il controllo delle fasi del motore ha raggiunto il livello massimo (saturazione) e non può essere ulteriormente aumentata. Il funzionamento del motore è gravato da un'eccessiva dinamica o qualcosa sta impedendo il movimento.

#### **IN 1** bit 13

Visualizza lo stato dell'ingresso digitale 1. Il significato degli ingressi disponibili è descritto nella sezione "6.3 Ingressi e uscita digitali" a pagina 50.

**IN 1 = 0**            ingresso 1 basso (non attivo)

**IN 1 = 1**            ingresso 1 alto (attivo)

**IN 2**

bit 14

Visualizza lo stato dell'ingresso digitale 2. Il significato degli ingressi disponibili è descritto nella sezione "6.3 Ingressi e uscita digitali" a pagina 50.

**IN 2 = 0**          ingresso 2 basso (non attivo)

**IN 2 = 1**          ingresso 2 alto (attivo)

**IN 3**

bit 15

Visualizza lo stato dell'ingresso digitale 3. Il significato degli ingressi disponibili è descritto nella sezione "6.3 Ingressi e uscita digitali" a pagina 50.

**IN 3 = 0**          ingresso 3 basso (non attivo)

**IN 3 = 1**          ingresso 3 alto (attivo)

**2203-00 Real Position**

[Signed32, ro]

Posizione attuale del dispositivo espressa in impulsi. Questo oggetto è mappato nell'oggetto **1A00 Transmit PDO Mapping** che contiene i parametri di mappatura dei PDO che il dispositivo EtherCAT è in grado di trasmettere, si veda a pagina 86. E' aggiornato ogni 1 msec.

**2204-00 Distance per revolution**

[Unsigned16, rw]

Questo parametro imposta il numero di impulsi per ogni giro completo dell'albero. Si rivela utile per relazionare un giro dell'asse con una grandezza lineare; per esempio, se il dispositivo è montato su una vite senza fine con passo 5 mm, impostando **2204-00 Distance per revolution** = 500, si ottiene che a ogni giro dell'asse il sistema trasla di 5 mm con una risoluzione al centesimo di millimetro.

Default = 1024 (min. = 1, max. = 1024)


**ATTENZIONE**

Dopo la modifica di questo oggetto è necessario reimpostare anche l'oggetto **2211-00 Preset**. Per una spiegazione dettagliata si veda a pagina 51 e i relativi oggetti.

Si badi inoltre che gli oggetti di seguito elencati sono in stretta correlazione con l'oggetto **2204-00 Distance per revolution**; pertanto la modifica del valore dell'oggetto **2204-00 Distance per revolution** comporta necessariamente una ridefinizione dei valori da essi espressi. Essi sono: **2205-00 Position tolerance**, **2207-00 Max following error**, **220C-00 Max delta pos**, **220D-00 Max**

delta neg, 2201-00 Target Position, 2203-00 Real Position e 210A-00 Following error [pulse].

**NOTA**

Se **2204-00 Distance per revolution** non è una potenza di 2 (2, ..., 512, 1024), durante il controllo di posizione potrebbe verificarsi un errore di posizionamento pari a un impulso.

**2205-00 Position tolerance**

[Unsigned16, rw]

Questo oggetto definisce la finestra di tolleranza da applicare al valore impostato in **2201-00 Target Position**. Non appena l'asse si trova all'interno dei limiti di tolleranza, il bit 8 **Target position reached** della **2202 Status Word** va alto ("=1"). Quando la posizione dell'asse è all'interno dei limiti di tolleranza per il tempo impostato nell'oggetto **2206-00 Settling time**, il bit 0 **Axis in position** della **2202 Status Word** va alto ("=1"). Il parametro è espresso in impulsi. Si veda anche la sezione "Posizione: controllo di posizione e velocità" a pagina 49.

Default = 1 (min. = 0, max. = 65535)

**2206-00 Settling time**

[Unsigned16, rw]

Rappresenta il tempo per la durata del quale la posizione dell'asse deve essere compresa all'interno dei limiti di tolleranza impostati nell'oggetto **2205-00 Position tolerance** prima che lo stato sia segnalato attraverso il bit di stato **Axis in position** della **2202 Status Word**. Il parametro è espresso in millisecondi (ms). Si veda anche la sezione "Posizione: controllo di posizione e velocità" a pagina 49.

Default = 0 (min. = 0, max. = 10000)

**2207-00 Max following error**

[Unsigned32, rw]

Questo oggetto definisce la differenza massima ammissibile tra la posizione reale del dispositivo e quella teorica. Se il dispositivo rileva un valore superiore a quello impostato in questo parametro, viene segnalato l'allarme **Following error** e il dispositivo arresta il proprio movimento. Il parametro è espresso in impulsi.

Default = 1024 (min. = 0, max. = 65535)

### 2208-00 Proportional gain

[Unsigned16, rw]

Questo parametro contiene il guadagno proporzionale usato dal controllore PI relativo all'anello di posizione. Tale valore è già ottimizzato da Lika Electronic in relazione alle caratteristiche del dispositivo.

Default = 300 (min. = 0, max. = 1000)

### 2209-00 Integral gain

[Unsigned16, rw]

Questo parametro contiene il guadagno integrale usato dal controllore PI relativo all'anello di posizione. Tale valore è già ottimizzato da Lika Electronic in relazione alle caratteristiche del dispositivo.

Default = 10 (min. = 0, max. = 1000)

### 220A-00 Acceleration

[Unsigned16, rw]

Questo parametro definisce il valore di accelerazione di cui si serve il dispositivo per raggiungere la velocità di jog **220E-00 Jog speed** o la velocità di lavoro **220F-00 Work speed**. Il parametro è espresso in giri al secondo<sup>2</sup> [giri/s<sup>2</sup>]. Si veda anche la sezione "6.2 Tipi di movimento: jog e posizione" a pagina 48.

Default = 10 (min. = 1, max. = 500)

### 220B-00 Deceleration

[Unsigned16, rw]

Questo parametro definisce il valore di decelerazione di cui si serve il dispositivo quando si arresta. Il parametro è espresso in giri al secondo<sup>2</sup> [giri/s<sup>2</sup>]. Si veda anche la sezione "6.2 Tipi di movimento: jog e posizione" a pagina 48.

Default = 10 (min. = 1, max. = 500)

### 220C-00 Max delta pos

[Unsigned32, rw]

Questo valore è utilizzato per calcolare il massimo spostamento in avanti (positivo) che il dispositivo può raggiungere a partire dal preset. Non appena si raggiunge il massimo spostamento in avanti, la condizione è segnalata attraverso il bit di stato **SW limit switch +** della **2202 Status Word** (il bit è forzato alto). Il parametro è espresso in impulsi.

**SW limit switch +** = **2211-00 Preset** + **220C-00 Max delta pos**. E' possibile leggere il valore del massimo spostamento positivo all'oggetto **210B-00 Pos. Limit Switch [pulse]**.

Per ulteriori informazioni riferirsi alla sezione "6.4 Distance per revolution, Preset, Max delta pos / Positive delta e Max delta neg / Negative delta" a pagina 51.

Default = 523.263 (min. = 0, max. = 523.263)



#### ATTENZIONE

Si badi che il valore massimo di volta in volta ammesso in questo parametro dipende dallo scaling impostato.



#### ESEMPIO

Se **2204-00 Distance per revolution** = 1.024 e **2211-00 Preset** = 0, il massimo valore accettabile per **220C-00 Max delta pos** è:

$(1.024 \text{ imp./giro} * 512 \text{ giri}) - 1 \text{ imp.} - 1.024 \text{ imp.}$  (ossia 1 giro per motivi di sicurezza) = 523.263

Se **2204-00 Distance per revolution** = 256 e **2211-00 Preset** = 0, il massimo valore accettabile per **220C-00 Max delta pos** è:

$(256 \text{ imp./giro} * 512 \text{ giri}) - 1 \text{ imp.} - 256 \text{ imp.}$  (ossia 1 giro per motivi di sicurezza) = 130.815

Altri esempi alla sezione "6.4 Distance per revolution, Preset, Max delta pos / Positive delta e Max delta neg / Negative delta" a pagina 51.



#### ATTENZIONE

Ogniquale volta si modificano gli oggetti **2204-00 Distance per revolution** e **2211-00 Preset**, occorre poi verificare attentamente i valori in **220C-00 Max delta pos** e **220D-00 Max delta neg**. Ogniquale volta si modifica il valore in **2204-00 Distance per revolution**, occorre poi reimpostare il valore in **2211-00 Preset** in modo da definire lo zero asse in quanto è cambiata l'unità di misura di riferimento del sistema.

Dopo la modifica del parametro **2211-00 Preset** non occorre invece reimpostare il valore dei finecorsa in quanto la funzione di preset provvede a ricalcolarli automaticamente reinizializzando i limiti positivo e negativo sulla base dei valori impostati negli oggetti **220C-00 Max delta pos** e **220D-00 Max delta neg**. Per una spiegazione dettagliata si veda a pagina 51.

#### 220D-00 Max delta neg

[Unsigned32, rw]

Questo valore è utilizzato per calcolare il massimo spostamento all'indietro (negativo) che il dispositivo può raggiungere a partire dal preset. Non appena si raggiunge il massimo spostamento all'indietro, la condizione è segnalata

attraverso il bit di stato **SW limit switch** - della **2202 Status Word** (il bit è forzato alto). Il parametro è espresso in impulsi.

**SW limit switch** - = **2211-00 Preset** - **220D-00 Max delta neg**. E' possibile leggere il valore del massimo spostamento negativo all'oggetto **210C-00 Neg. Limit Switch [pulse]**.

Per ulteriori informazioni riferirsi alla sezione "6.4 Distance per revolution, Preset, Max delta pos / Positive delta e Max delta neg / Negative delta" a pagina 51.

Default = 523.263 (min. = 0, max. = 523.263)



#### ATTENZIONE

Si badi che il valore massimo di volta in volta ammesso in questo parametro dipende dallo scaling impostato.



#### ESEMPIO

Se **2204-00 Distance per revolution** = 1.024 e **2211-00 Preset** = 0, il massimo valore accettabile per **220D-00 Max delta neg** è:

$(1.024 \text{ imp./giro} * 512 \text{ giri}) - 1 \text{ imp.} - 1.024 \text{ imp.}$  (ossia 1 giro per motivi di sicurezza) = 523.263

Se **2204-00 Distance per revolution** = 256 e **2211-00 Preset** = 0, il massimo valore accettabile per **220D-00 Max delta neg** è:

$(256 \text{ imp./giro} * 512 \text{ giri}) - 1 \text{ imp.} - 256 \text{ imp.}$  (ossia 1 giro per motivi di sicurezza) = 130.815

Altri esempi alla sezione "6.4 Distance per revolution, Preset, Max delta pos / Positive delta e Max delta neg / Negative delta" a pagina 51.



#### ATTENZIONE

Ogniqualevolta si modificano gli oggetti **2204-00 Distance per revolution** e **2211-00 Preset**, occorre poi verificare attentamente i valori in **220C-00 Max delta pos** e **220D-00 Max delta neg**. Ogniqualevolta si modifica il valore in **2204-00 Distance per revolution**, occorre poi reimpostare il valore in **2211-00 Preset** in modo da definire lo zero asse in quanto è cambiata l'unità di misura di riferimento del sistema.

Dopo la modifica del parametro **2211-00 Preset** non occorre invece reimpostare il valore dei finecorsa in quanto la funzione di preset provvede a ricalcolarli automaticamente reinizializzando i limiti positivo e negativo sulla base dei valori impostati negli oggetti **220C-00 Max delta pos** e **220D-00 Max delta neg**. Per una spiegazione dettagliata si veda a pagina 51.

### 220E-00 Jog speed

[Unsigned16, rw]

Questo oggetto definisce la velocità massima che il motore può raggiungere quando si utilizzano le funzioni **Jog +** e **Jog -** (si veda l'oggetto **2200 Control Word**). Il parametro è espresso in giri al minuto (rpm). Si veda anche la sezione "Jog: controllo di velocità" a pagina 48.

Default = 2000 (min. = 1, max. = 3000)



#### NOTA

Si badi che questa è la velocità del motore, non la velocità dell'albero in uscita dopo il riduttore.

La velocità in uscita sarà come segue:

Velocità del motore	= 2000 rpm
Velocità in uscita:	T12 = 166 rpm
	T24 = 83 rpm
	T48 = 41 rpm
	T92 = 21 rpm

### 220F-00 Work speed

[Unsigned16, rw]

Questo oggetto definisce la velocità massima che il motore può raggiungere in funzionamento automatico (i movimenti sono controllati mediante i comandi **Start** e **Stop** -si veda l'oggetto **2200 Control Word**- e sono eseguiti al fine di raggiungere la posizione impostata in **2201-00 Target Position**). Il parametro è espresso in giri al minuto (rpm). Si veda anche la sezione "Posizione: controllo di posizione e velocità" a pagina 49.

Default = 2000 (min. = 1, max. = 3000)



#### NOTA

Si badi che questa è la velocità del motore, non la velocità dell'albero in uscita dopo il riduttore.

La velocità in uscita sarà come segue:

Velocità del motore	= 2000 rpm
Velocità in uscita:	T12 = 166 rpm
	T24 = 83 rpm
	T48 = 41 rpm
	T92 = 21 rpm

**2210-00 Count direction**

[Unsigned16, rw]

Imposta se il valore di posizione provvisto dal dispositivo incrementa (conteggio crescente) quando l'albero ruota in senso orario (0) o antiorario (1). La direzione della rotazione oraria e antioraria è stabilita guardando il dispositivo dall'estremità dell'asse.

0 = informazione crescente con rotazione oraria (default)

1 = informazione crescente con rotazione antioraria

**ATTENZIONE**

La modifica di questo parametro influenza la posizione calcolata dal controllore. Si deve quindi reimpostare poi l'oggetto **2211-00 Preset** e verificare i valori negli oggetti **220C-00 Max delta pos** e **220D-00 Max delta neg**.

**2211-00 Preset**

[Signed32, rw]

Usare questo oggetto per impostare il valore di Preset. La funzione di Preset è utilizzata per assegnare un valore desiderato a una posizione fisica dell'asse. La posizione fisica prescelta assumerà pertanto il valore assegnato in questo oggetto e tutte le posizioni precedenti e successive assumeranno un valore conseguente. Il valore di preset sarà assegnato alla posizione dell'asse al momento dell'invio del parametro. Il valore di preset è attivato quando il bit 11 **Setting the preset** nell'oggetto **2200 Control Word** commuta dal livello logico basso ("0") al livello logico alto ("1").

Default = 0 (min. = -1 048 576, max. = +1 048 576)

**NOTA**

Consigliamo di attivare il preset quando l'attuatore è in stop. Si veda il comando **Setting the preset** a pagina 100.

**ATTENZIONE**

Occorre poi reimpostare un nuovo valore nell'oggetto **2211-00 Preset** ogniqualvolta si modifica il valore in **2204-00 Distance per revolution**. Dopo la modifica del parametro **2211-00 Preset** non occorre invece reimpostare il valore dei finecorsa in quanto la funzione di preset provvede a ricalcolarli automaticamente reinizializzando i limiti positivo e negativo sulla base dei valori impostati negli oggetti **220C-00 Max delta pos** e **220D-00 Max delta neg**. Per una spiegazione dettagliata si veda a pagina 51.

**2212-00 Step jog**

[Unsigned16, rw]

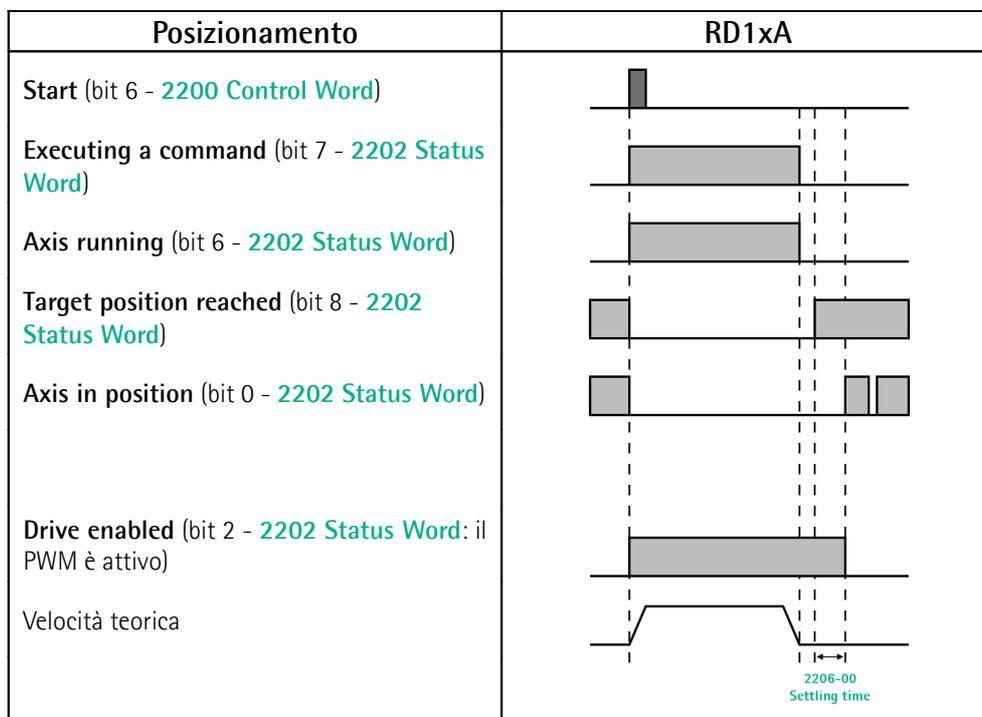
Se il controllo del jog a passo (funzione di jog incrementale) è abilitato (bit 4 **Incremental jog** in **2200 Control Word** = 1), l'attivazione dei bit **Jog +** e **Jog -** procura, in corrispondenza del fronte di salita, l'esecuzione di un singolo passo in direzione positiva o negativa la cui ampiezza, espressa in impulsi, è impostata in questo oggetto; l'attuatore poi si arresta in attesa di un nuovo comando.  
Default = 1000 (min. = 1, max. = 10000).

**NOTA**

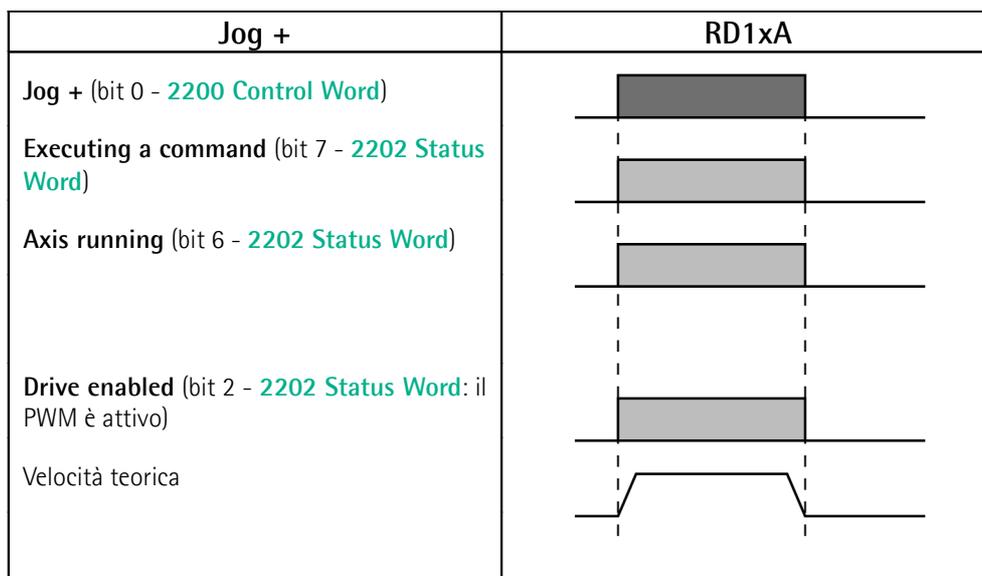
Salvare sempre i nuovi valori dopo l'impostazione al fine di memorizzarli permanentemente nella memoria non volatile. Usare la funzione **Save parameters** disponibile nell'oggetto **2200 Control Word**, si veda a pagina 96. Nel caso di spegnimento del dispositivo i dati non salvati andranno persi.



ESEMPIO 1



ESEMPIO 2



## 7.9 SDO Abort code

Una trasmissione SDO potrebbe fallire; le cause dell'errore sono elencate e descritte negli SDO Abort Code (si veda ETG1000.6 "EtherCAT Specification – Part 6. Application Layer protocol Specification", par. 5.6.2.7.2, tabella 40).

## 7.10 Emergency Error Code

L'Emergency Service è usato dal Server per trasmettere al Client messaggi di diagnostica tramite MailBox; gli Error Code sono elencati e descritti in ETG1000.6 "EtherCAT Specification – Part 6. Application Layer protocol Specification", par. 5.6.4.2, tabella 50.

Error Code		Error Register	Diagnostic Data				
Byte (0)	Byte (1)	Byte (2)	Byte (3)	Byte (4)	Byte (5)	Byte (6)	Byte (7)

Error Code	State Transition Error dello stato macchina: (per una descrizione dettagliata vedi ETG1000.6 par. 5.6.4.3) A000hex: transition error da <b>PRE-OPERATIONAL</b> a <b>SAFE-OPERATIONAL</b> A001hex: transition error da <b>SAFE-OPERATIONAL</b> a <b>OPERATIONAL</b>
Error Register	stato attuale dell'EtherCAT state machine (ESM)
Diagnostic Data	informazione sulle possibili cause di errore (si veda ETG1000.6 par. 5.6.4.3.2-5).

## 7.11 AL Status Error Code

Se la transizione di stato richiesta dal Master attraverso l'"AL Control Register" fallisce, lo Slave imposta a 1 l'"Error Indicator Bit" nell'"AL Status Register" e scrive la causa dell'errore nell'"AL Status Code Register".

I valori e le descrizioni dell'"AL Status Code" sono disponibili in ETG1000.6 "EtherCAT Specification – Part 6. Application Layer protocol Specification", par.5.3.2, Tabella 11.

## 8 Interfaccia MODBUS®

Le unità di posizionamento DRIVECOD Lika sono dispositivi Slave e implementano il Modbus application protocol (livello 7 del modello OSI) e il protocollo "Modbus over Serial Line" (livelli 1 e 2 del modello OSI).

Per ulteriori informazioni o specifiche omesse riferirsi ai documenti "Modbus Application Protocol Specification V1.1b" e "Modbus over Serial Line. Specification and Implementation Guide V1.02" disponibili all'indirizzo [www.modbus.org](http://www.modbus.org).

### 8.1 Configurazione del dispositivo mediante software Lika

Le unità di posizionamento DRIVECOD RD1xA possono disporre di una varietà di interfacce di comunicazione come EtherNet/IP, EtherCAT, POWERLINK, MODBUS RTU, Profibus-DP, CANopen DS 301 ecc. Tutte le versioni eccetto la MODBUS RTU sono equipaggiate con una porta seriale di servizio RS-232 conforme al protocollo MODBUS. Può essere usata per configurare l'attuatore. A questo scopo tutte le versioni sono fornite con un software sviluppato da Lika Electronic per la programmazione semplificata del dispositivo. Il programma permette di impostare i parametri di lavoro, testare manualmente alcuni movimenti e funzioni e monitorare il corretto funzionamento del dispositivo. Il software è fornito gratuitamente e può essere installato in qualsiasi PC con sistema operativo Windows (Windows XP o successivo). L'eseguibile da utilizzare per lanciare il programma è **ROTADRIVE\_INTERFACE.EXE** ed è disponibile con la documentazione allegata oppure scaricabile all'indirizzo [www.lika.it](http://www.lika.it) > **PRODOTTI > ATTUATORI ROTATIVI**. Il programma non richiede installazione, basta copiare l'eseguibile nel repository desiderato; per lanciarlo è poi sufficiente fare un doppio click sull'icona del file. Per chiudere il programma, premere il pulsante **DISCONNETTI** nella pagina **Configurazione seriale**, quindi il pulsante **CHIUDI** nella barra del titolo.



#### NOTA

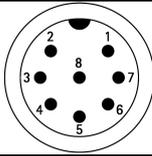
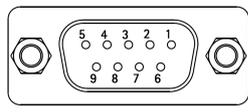
Prima di lanciare il programma è necessario collegare il dispositivo al personal computer mediante una porta seriale RS-232. L'interfaccia seriale dell'unità DRIVECOD per collegamento in rete Modbus RTU è del tipo RS-232. Nel caso in cui il personal computer non sia provvisto di porta seriale RS-232, sarà necessario installare un convertitore USB / RS-232, facilmente reperibile in commercio. Per ogni informazione sullo schema di collegamento e il pinout del cavo riferirsi alle istruzioni del convertitore.

Sul lato DRIVECOD il cavo deve essere collegato al connettore M12 8 pin maschio della porta seriale di servizio. Si veda la sezione "Connessioni elettriche" a pagina 33.

Un kit di connessione con connettori M12 8 pin / USB è disponibile su richiesta; contattare il servizio di assistenza tecnica e post vendita di Lika Electronic riferendosi al seguente codice: **IF92 + EC-RD1A-M12M8**.


**NOTA**

Quando si utilizza il kit IF92 + cavo di connessione, è necessario installare preventivamente i driver del Convertitore Seriale USB e della Porta Seriale USB. I driver sono disponibili nella cartella Software dell'attuatore e scaricabili dal sito di Lika.

			
<b>Funzione</b>	<b>Connettore RS-232 M12 8 pin maschio</b>	<b>Connettore D-SUB 9 pin femmina</b>	<b>Funzione</b>
TD	6	2	RD
RD	7	3	TD
0Vdc	8	5	0Vdc

Assicurarsi che RD dell'unità DRIVECOD sia connesso con TD del PC e che TD del PC sia connesso con RD dell'unità DRIVECOD.

Si badi che i parametri di configurazione della porta seriale di servizio MODBUS sono fissi e perciò non modificabili.

Essi sono:

**Impostazioni porta seriale di servizio RS-232 MODBUS**

	<b>Valore di default</b>
Bit per secondo	9600
Bit di dati	8
Parità	Pari
Bit di stop	1

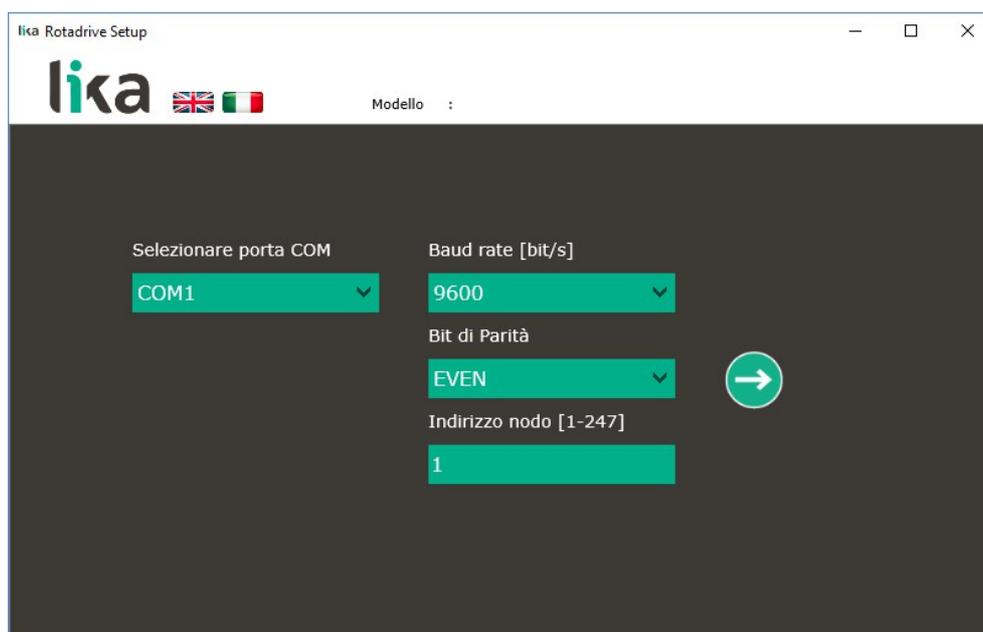
L'indirizzo MODBUS è "1". Non può essere modificato. Si veda la sezione "8.2 Pagina "Configurazione seriale"" qui a seguire.

**NOTA**

Si badi che con interfaccia MODBUS solo il LED L5 (alimentazione controllore) e il LED L6 (alimentazione motore) sono operativi.

**8.2 Pagina "Configurazione seriale"**

All'avvio del programma, si visualizza la pagina **Configurazione seriale**.



Questa pagina permette anzitutto la scelta della lingua di visualizzazione dei testi nell'interfaccia. Cliccare sul pulsante **Bandiera italiana**  per scegliere la lingua di visualizzazione italiana; cliccare sul pulsante **Bandiera inglese**  per scegliere la lingua di visualizzazione inglese.

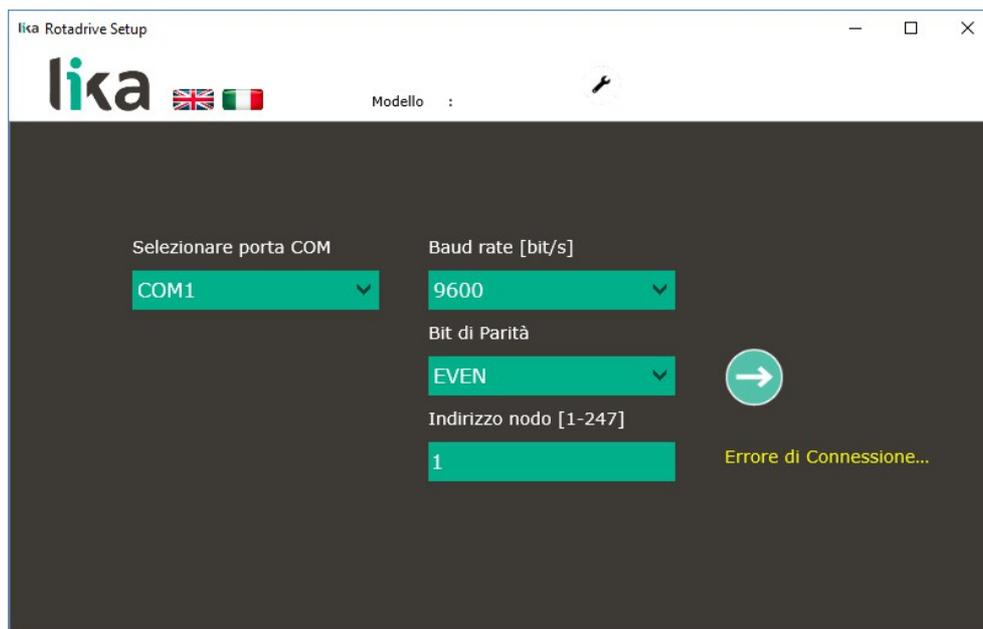
Inoltre premendo sul logo Lika in alto a sinistra si accede al sito web di Lika [www.lika.biz](http://www.lika.biz).

La pagina **Configurazione seriale** permette di selezionare la porta seriale del personal computer cui l'unità RD1xA è collegata (menu a tendina **Selezionare porta COM**) e poi impostare i parametri di configurazione. Le impostazioni della porta seriale nel personal computer devono necessariamente corrispondere ai parametri della porta seriale del dispositivo Lika collegato.

**Per le impostazioni della porta seriale si veda la sezione precedente.**

Impostare quindi l'indirizzo del nodo cui ci si vuole connettere tramite il menu a tendina **Indirizzo nodo [1-247]** (valore di default dei dispositivi RD1xA con interfaccia EtherCAT = 1). Si veda a pagina 37.

E' ora possibile attivare la connessione con lo Slave: premere il pulsante di **CONNESSIONE** a destra per iniziare la ricerca dei dispositivi collegati.



Se la connessione non si realizza (per esempio, a causa di impostazioni errate della porta seriale, o dell'indirizzo del nodo non corretto), sotto il pulsante di **CONNESSIONE** posso comparire i messaggi **Errore di connessione, Il dispositivo non risponde, Errore apertura COM, Selezionare porta COM o Errore Node ID**. Verificare le impostazioni e ritentare il collegamento.



#### NOTA

Si badi che a seguito del tentativo di connessione fallito compare sulla barra bianca in alto (toolbar) il pulsante **FIRMWARE** . Esso permette di accedere alla pagina **Programming firmware**. Per informazioni complete riferirsi alla sezione "8.7 Pagina "Programming firmware"" a pagina 126.

### 8.3 Pagina "Principale"

Non appena si preme il pulsante di **CONNESSIONE**  nella pagina **Configurazione seriale**, se il collegamento ha esito positivo si accede alla pagina **Principale**.



Nella barra bianca in alto nella pagina (toolbar) a fianco dell'etichetta **Modello** viene visualizzato il modello DRIVECOD cui ci si è collegati.

Il messaggio di avvertenza **ALARM** lampeggia in quanto l'unità si trova in condizione di emergenza.

Nella stessa barra diventano disponibili i seguenti pulsanti.

#### **FIRMWARE**

Il pulsante **FIRMWARE**  permette di accedere alla pagina **Programming firmware**. Questa pagina permette all'operatore l'aggiornamento del firmware dell'unità DRIVECOD mediante il download dei dati di aggiornamento alla memoria flash. Per informazioni complete riferirsi alla sezione "8.7 Pagina "Programming firmware"" a pagina 126.

#### **SETUP**

Il pulsante **SETUP**  permette di accedere alla pagina **Parametri**. In questa pagina sono elencati i parametri (dati macchina) disponibili per l'unità di posizionamento RD1xA. Per informazioni complete riferirsi alla sezione "8.8 Pagina "Parametri"" a pagina 129.

### SCHEDULE

Il pulsante **SCHEDULE**  permette l'accesso alla pagina **Programma**. Le funzioni disponibili nella pagina **Programma** permettono all'operatore la creazione e il salvataggio di un programma di lavoro al fine di testare il funzionamento dell'unità RD1xA. Per informazioni complete riferirsi alla sezione "8.9 Pagina "Programma"" a pagina 132.

### FRECCIA INDIETRO

Il pulsante **FRECCIA INDIETRO**  permette di ritornare alla pagina visualizzata in precedenza.

### HOME

Pulsante **HOME**. Quando è visualizzata la pagina **Principale**, premere il pulsante **HOME**  per accedere alla pagina **Configurazione seriale**. Quando invece è visualizzata una qualsiasi delle altre pagine, premere il pulsante **HOME**  per accedere alla pagina **Principale**.

In questa pagina sono presenti alcune ulteriori informazioni sulla posizione e gli stati dell'unità DRIVECOD.

I seguenti elementi appaiono a sinistra nella pagina.

#### **Posizione attuale**

Si veda **Current position [0x02-0x03]** a pagina 170.

#### **Posizione target**

Si veda **Target position [0x2B-0x2C]** a pagina 163. Impostare il valore di posizione che il dispositivo deve raggiungere e confermare con il tasto **INVIO** della tastiera. Alla pressione del pulsante **START**  il dispositivo avvia il posizionamento alla quota impostata in questo campo **Posizione target**, quindi si arresta attivando i bit di stato **Axis in position** e **Target position reached** (si veda la sezione "8.5 Box "Stato"" a pagina 123). Per informazioni dettagliate sulla creazione di un ciclo completo di posizionamenti e il controllo del funzionamento dell'attuatore RD1xA (velocità, accelerazione, decelerazione, ecc.) riferirsi alla sezione "8.9 Pagina "Programma"" a pagina 132.

#### **Velocità attuale [rpm]**

Si veda **Current velocity [0x04]** a pagina 170.

#### **Corrente attuale [mA]**

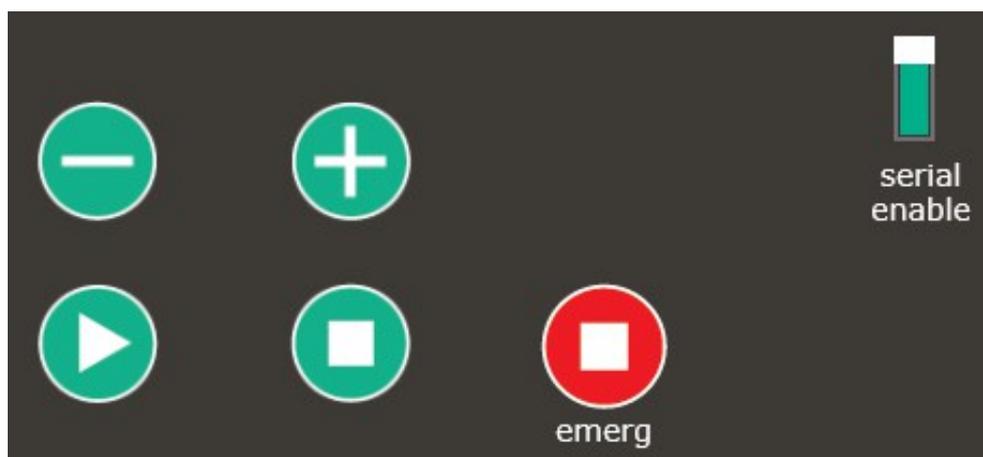
Visualizza il valore della corrente assorbita dal motore (corrente nominale). Il valore è espresso in milliampere (mA). Si veda **Current value [0x0B]** a pagina 172.

## Temperatura interna [°C]

Si veda [Temperature value \[0x07\]](#) a pagina 171.

### 8.4 Comandi MODBUS

In alto a destra nella pagina **Principale** sono disponibili alcuni comandi.



Questi comandi permettono di controllare manualmente e monitorare il dispositivo collegato. Quando si accede alla pagina **Principale**, tutti i comandi sono disabilitati in quanto il dispositivo si trova ancora sotto il controllo della rete EtherCAT. Per avviare l'attività di programmazione, gestione manuale e monitoraggio del dispositivo attraverso la seriale di servizio RS-232 e il protocollo MODBUS è necessario abilitare i comandi acquisendo prima il controllo del dispositivo in rete MODBUS mediante il PC. Per fare questo attivare

il cursore **ABILITA SERIALE**  (si veda [Extra commands register \[0x29\]](#) a pagina 158).

Tutti i comandi diventano immediatamente disponibili all'uso.

#### JOG -

Si veda **Jog -** a pagina 159. Se il parametro **JOG Incrementale** è abilitato = ON, il valore **Increm.** (parametro **Ampiezza passo JOG**) che è attualmente impostato appare tra i due pulsanti **JOG -**  e **JOG +** . Questo pulsante è disponibile solamente se i comandi MODBUS sono abilitati (si veda il cursore

**ABILITA SERIALE** ). Si veda la sezione "8.8 Pagina "Parametri"" a pagina 129.

### JOG +

Si veda **Jog +** a pagina 158. Se il parametro **JOG Incrementale** è abilitato = ON, il valore **Increm.** (parametro **Ampiezza passo JOG**) che è attualmente impostato appare tra i due pulsanti **JOG -** e **JOG +** . Questo pulsante è disponibile solamente se i comandi MODBUS sono abilitati (si veda il cursore

**ABILITA SERIALE** ). Si veda la sezione "8.8 Pagina "Parametri"" a pagina 129.

### START

Premendo il pulsante **START** si comanda il movimento del dispositivo per il raggiungimento della posizione impostata in **Posizione target**. Al raggiungimento della quota impostata il dispositivo si arresta attivando i bit di stato **Axis in position** e **Target position reached** (si veda la sezione "8.5 Box "Stato"" a pagina 123). Per un arresto normale del dispositivo premere il pulsante **STOP** ; per un arresto di emergenza premere il pulsante **EMERGENZA** . Questo pulsante è disponibile solamente se i comandi

MODBUS sono abilitati (si veda il cursore **ABILITA SERIALE** ). Si veda il bit **Start** a pagina 160.

### STOP

Premere il pulsante **STOP** per comandare un arresto normale del dispositivo, nel rispetto del valore di decelerazione. Questo pulsante è disponibile solamente

se i comandi MODBUS sono abilitati (si veda il cursore **ABILITA SERIALE** ). Si veda il bit **Stop** a pagina 159.

### EMERGENZA

Con dispositivo in movimento, premere il pulsante **EMERGENZA** per comandare un arresto immediato in condizione di emergenza. Premere il pulsante **RESET** per ripristinare lo stato normale di lavoro (si veda la sezione "8.5 Box "Stato"" a pagina 123). Questo pulsante è disponibile solamente se i

comandi MODBUS sono abilitati (si veda il cursore **ABILITA SERIALE** ). Si veda **Emergency** a pagina 160.

### Abilita seriale

Come detto in precedenza, quando si accede alla pagina **Principale**, tutti i comandi sono disabilitati in quanto il dispositivo si trova ancora sotto il controllo della rete EtherCAT. Per avviare l'attività di programmazione, gestione manuale e monitoraggio del dispositivo attraverso la seriale di servizio RS-232 e

il protocollo MODBUS è necessario abilitare i comandi acquisendo prima il controllo del dispositivo in rete MODBUS mediante il PC. Per fare questo

impostare il cursore **ABILITA SERIALE**  (si veda [Extra commands register \[0x29\]](#) a pagina 158).

### 8.5 Box "Stato"

Il box **Stato** è disponibile in basso a destra nella pagina **Principale**.



Le funzioni disponibili all'interno di questo box forniscono un'informazione essenziale sullo stato dell'attuatore rotativo e permettono di accedere ad ulteriori pagine che contengono informazioni più dettagliate.

Gli allarmi e gli stati gravi attualmente attivi sono visualizzati in rosso. Gli stati ordinari attualmente attivi sono visualizzati in verde.

Si badi che al momento dell'accensione per ragioni di sicurezza l'unità RD1xA si trova necessariamente in una condizione di emergenza: quando perciò ci si connette e si accede alla pagina **Principale** il messaggio di warning **Allarme in corso** è acceso rosso. Per saperne di più sullo specifico allarme attivo accedere alla pagina **Allarmi e stati** premendo il pulsante **STATUS & INFO** , riferirsi alla sezione "8.6 Pagina "Allarmi e stati"" a pagina 124. Per ripristinare lo stato **Idle** del dispositivo, premere il pulsante **RESET**  nel box. Gli allarmi e le emergenze saranno resettati e sarà ripristinata la normale condizione di lavoro del dispositivo.

Per informazioni complete sugli stati e gli allarmi che appaiono in questo box riferirsi al registro **Status word [0x01]** a pagina 167; e al registro **Alarms register [0x00]** a pagina 165.

### STATUS & INFO

Premere questo pulsante per accedere alla pagina **Allarmi e stati** dove sono disponibili informazioni specifiche sugli stati e gli allarmi correntemente attivi nell'attuatore rotativo. Riferirsi alla sezione "8.6 Pagina "Allarmi e stati"" a pagina 124.

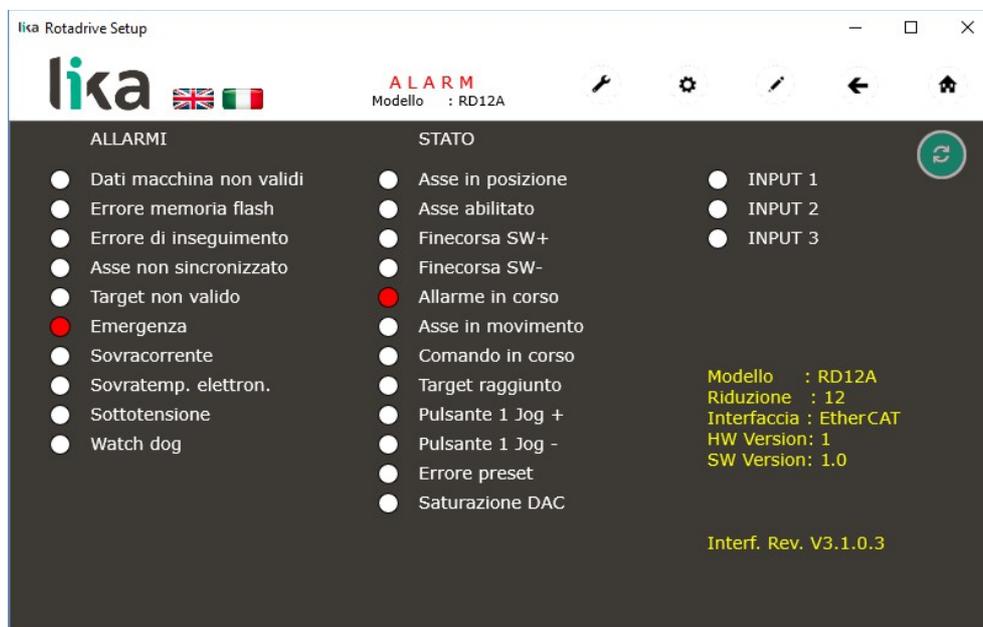
### RESET

Se un allarme è attivo, è segnalato mediante i warning generici nel box **Stato**. Per saperne di più sullo specifico allarme attivo accedere alla pagina **Allarmi e stati** premendo il pulsante **STATUS & INFO** , riferirsi alla sezione "8.6 Pagina "Allarmi e stati"" a pagina 124. Premere questo pulsante per resettare l'allarme e ripristinare la normale condizione di lavoro del dispositivo. Questo pulsante è disponibile solamente se i comandi MODBUS sono abilitati (si veda il cursore

**ABILITA SERIALE** ). Si veda **Alarm reset** a pagina 159.

## 8.6 Pagina "Allarmi e stati"

Quando si preme il pulsante **STATUS & INFO** nel box **Stato** si accede alla pagina **Allarmi e stati**.



Come detto in precedenza il box **Stato** fornisce un'informazione essenziale sugli stati e gli allarmi correntemente attivi nell'attuatore rotativo. Questa pagina **Allarmi e stati** fornisce informazioni più dettagliate sugli stati e sugli allarmi

correntemente attivi. Gli allarmi e gli stati gravi correntemente attivi sono visualizzati in rosso. Gli stati ordinari correntemente attivi sono visualizzati in verde.

Per informazioni complete sugli stati e gli allarmi che appaiono in questa pagina riferirsi al registro **Status word [0x01]** a pagina 167; e al registro **Alarms register [0x00]** a pagina 165.

Inoltre la pagina offre informazioni dettagliate sull'attuatore collegato e sul tool software. Queste informazioni sono elencate in giallo in basso a destra nella pagina.

In particolare è possibile trovare:

- **Modello:** il modello dell'attuatore collegato;
- **Riduzione:** il rapporto di riduzione dell'attuatore collegato;
- **Interfaccia:** il protocollo dell'attuatore collegato;
- **HW version:** la versione hardware dell'attuatore collegato;
- **SW version:** la versione software dell'attuatore collegato;
- **Interf. Rev.:** la release del tool software.

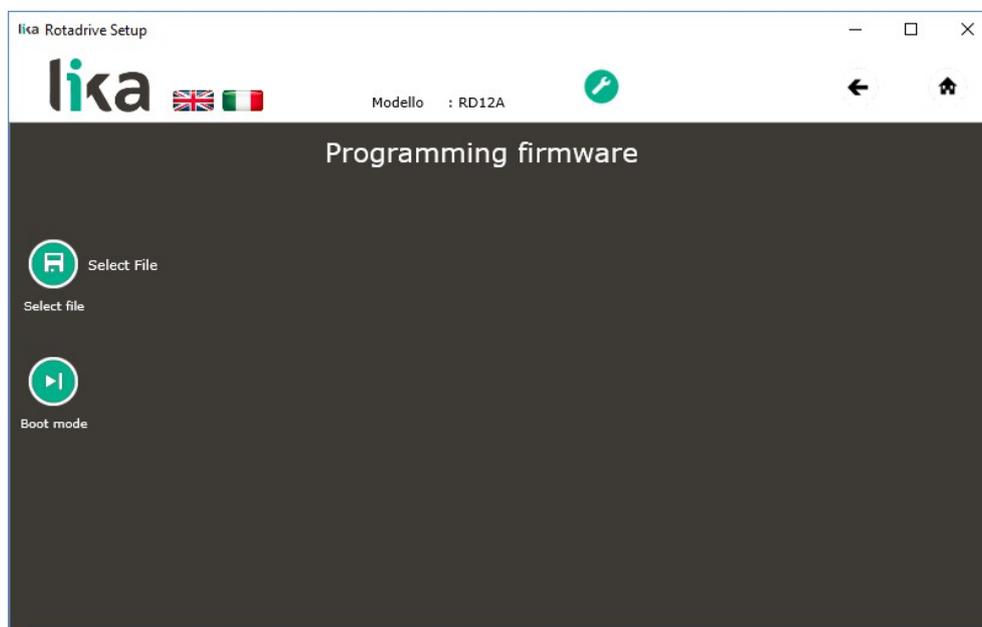
#### RESET

Pulsante **RESET** . Si veda la pagina precedente.

Premere il pulsante **HOME**  per ritornare alla pagina **Principale**.

## 8.7 Pagina "Programming firmware"

Premendo il pulsante **FIRMWARE**  nella toolbar si accede alla pagina **Programming firmware**.



Le funzioni disponibili in questa pagina permettono l'aggiornamento del firmware dell'unità DRIVECOD mediante il trasferimento dei dati di upgrade alla memoria flash.

Il firmware è un programma software che permette la gestione e il controllo del funzionamento di un dispositivo; il programma firmware, talora chiamato anche "user program" o "programma utente", è memorizzato nella memoria flash integrata all'interno dell'unità DRIVECOD. I dispositivi DRIVECOD sono progettati in modo che il firmware possa essere aggiornato agevolmente e direttamente dall'utente finale. Questo permette di rendere disponibili nuovi e più aggiornati firmware durante tutto il corso di vita del prodotto.

Le tipiche motivazioni che procurano il rilascio di un nuovo firmware derivano dalla necessità di correggere, migliorare o talora aggiungere nuove funzionalità al dispositivo.

L'aggiornamento firmware consiste in un file con estensione .BIN fornito direttamente dal Servizio di Assistenza Tecnica di Lika Electronic.



### ATTENZIONE

Il processo di aggiornamento del firmware di un dispositivo DRIVECOD deve essere eseguito da personale esperto e competente. L'applicazione di un aggiornamento errato o incompatibile pregiudica il funzionamento del dispositivo stesso.

Se la versione firmware più recente è già installata nel dispositivo DRIVECOD, non è necessario procedere con l'installazione di alcun aggiornamento. La versione firmware correntemente installata può essere verificata alla voce **SW Version** nella pagina **Allarmi e Stati** dopo la corretta connessione al dispositivo (si veda a pagina 124).

Qualora sussistano dei dubbi sull'aggiornamento del firmware, si prega di contattare il Servizio di Assistenza Tecnica di Lika Electronic.

Per l'installazione dell'aggiornamento firmware procedere come segue:

1. assicurarsi che siano impostati i seguenti parametri di configurazione (essi non sono modificabili nella porta seriale dell'unità DRIVECOD): bit per secondo = 9600 bit/s; bit di parità = pari; se sono impostati diversamente nel vostro PC, modificarli; si veda la sezione "4.2.6 Ingressi / uscita + porta di servizio RS-232 MODBUS" a pagina 37;
2. assicurarsi che l'unità DRIVECOD da aggiornare sia l'unico nodo collegato al personal computer;
3. collegarsi all'unità, andare online e accedere alla pagina **Programming firmware**;
4. se, all'accensione, lo user program non è presente nella memoria flash, non è possibile collegarsi all'unità attraverso la pagina **Configurazione seriale**; quando questo avviene bisogna accedere direttamente alla pagina **Programming firmware**; dopo che il tentativo di connessione al dispositivo è fallito, nella toolbar della pagina **Configurazione seriale** diventa disponibile il pulsante **FIRMWARE** ; assicurarsi che nella pagina **Configurazione seriale** sia selezionata la corretta porta seriale del personal computer collegato all'unità DRIVECOD;
5. premere il pulsante **SELECT FILE** ; non appena si preme il pulsante appare la finestra di dialogo **APRI**; individuare quindi il repository dove è contenuto il file di aggiornamento firmware .BIN fornito da Lika Electronic;



### ATTENZIONE

Si badi che, per ogni modello di unità DRIVECOD con diversa interfaccia bus, è disponibile uno specifico firmware. Assicurarsi di possedere l'aggiornamento appropriato per il proprio modello. Il file .BIN rilasciato da Lika Electronic presenta un nome che deve essere interpretato nel modo seguente.

Per esempio: RD12A\_PL\_157\_H2S1.BIN, dove:

- RD12A = modello unità DRIVECOD;
- PL = interfaccia bus dell'unità DRIVECOD (CB = CANopen; EC = EtherCAT; EP = EtherNet/IP; MB = MODBUS RTU; MT = MODBUS TCP; PB = Profibus; PL = POWERLINK; PT = Profinet)
- 157 = potenza motore

- H2 = versione hardware
  - S1 = versione software
6. selezionare il file .BIN fornito da Lika Electronic e confermare la scelta mediante la pressione del pulsante **APRI**, la finestra di dialogo si chiude;
  7. in corrispondenza del pulsante **SELECT FILE**  compare il percorso completo relativo al file appena confermato;
  8. premere ora il pulsante **BOOT MODE**  per portare il dispositivo nello stato **Boot Mode**;
  9. in basso nella pagina compare il pulsante **UPLOAD** ; premere il pulsante per procedere al download dell'aggiornamento firmware;
  10. una barra di avanzamento verde con indicazione della percentuale viene visualizzata a fianco del pulsante **UPLOAD**  e mostra lo stato di avanzamento del processo;



#### ATTENZIONE

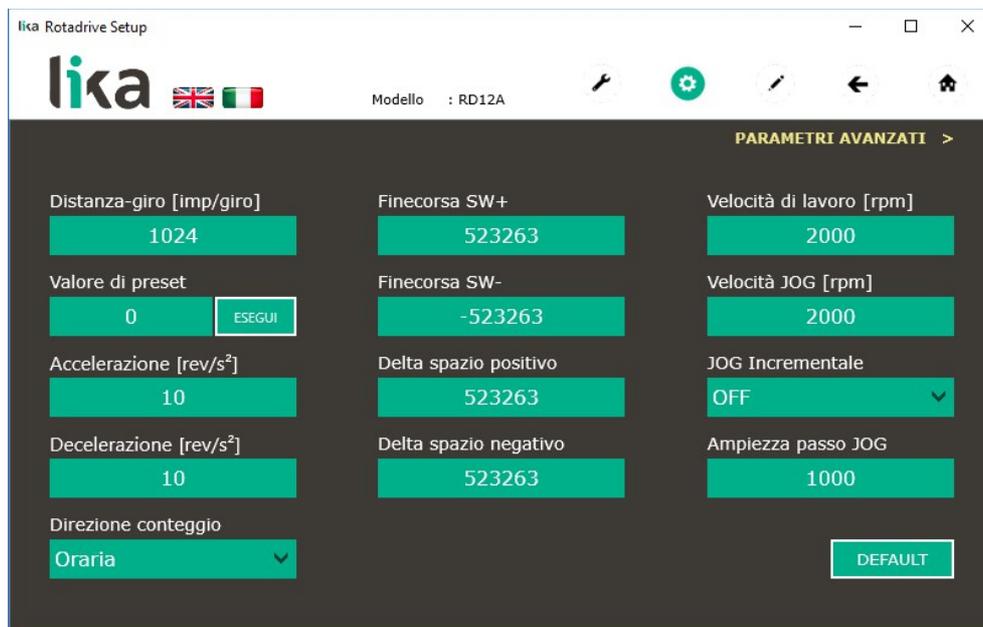
Non uscire dalla pagina **Programming firmware** durante l'installazione, altrimenti il processo sarà interrotto!

11. al completamento dell'operazione, se l'esito è positivo, viene visualizzato per un attimo il messaggio **OK**; quindi l'attuatore si spegne e si riaccende automaticamente e il tool software mostra la pagina **Configurazione seriale**;
12. riconnettersi all'unità DRIVECOD e ripristinare la normale condizione di lavoro.

Premere il pulsante **HOME**  per ritornare alla pagina **Principale**.

## 8.8 Pagina "Parametri"

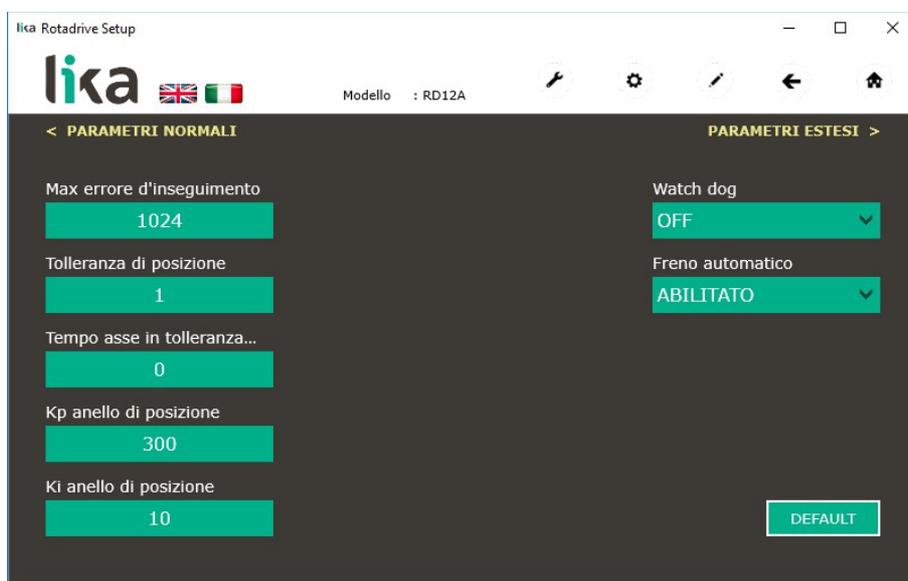
Premendo il pulsante **SETUP**  nella toolbar si accede alla pagina **Parametri**. Permette di configurare il dispositivo e impostare i parametri. Appare la pagina seguente.



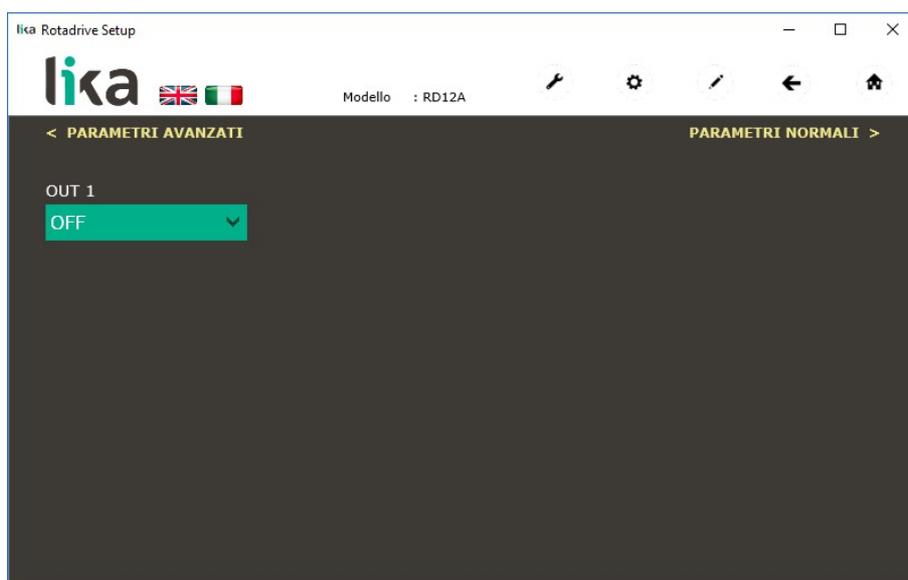
In questa pagina è disponibile l'elenco dei parametri cosiddetti "normali" per l'impostazione dell'unità di posizionamento RD1xA (dati macchina). Si tratta dei parametri d'uso più comune, quelli cioè che più frequentemente sono utilizzati per la configurazione del dispositivo e la creazione dei programmi dell'attuatore.

**PARAMETRI AVANZATI > PARAMETRI AVANZATI >**

Un'ulteriore pagina è accessibile premendo il pulsante **PARAMETRI AVANZATI > PARAMETRI AVANZATI >**; visualizza i parametri "avanzati" che più raramente richiedono un'impostazione.


**PARAMETRI ESTESI > PARAMETRI ESTESI >**

Premere il pulsante **PARAMETRI ESTESI > PARAMETRI ESTESI >** per accedere alla pagina che permette l'abilitazione delle uscite disponibili dell'attuatore. Per maggiori informazioni riferirsi alla sezione "6.3 Ingressi e uscita digitali" a pagina 50.



< PARAMETRI NORMALI 

Premere il pulsante < **PARAMETRI NORMALI**  per ritornare alla prima pagina dei **Parametri**.

Per ogni informazione sulla funzione e la programmazione di ciascun parametro riferirsi alla sezione "8.14.1 Parametri Holding Register" a pagina 150.

In corrispondenza di ciascun campo di impostazione è visualizzato il valore attualmente caricato sul dispositivo.

Per impostare un nuovo valore in un parametro digitare il valore desiderato, quindi premere il tasto **INVIO** della tastiera. Se si imposta un valore non conforme (fuori range), alla conferma è forzato il valore minimo o massimo. In alcuni casi il valore è selezionabile tramite il menu a tendina. I nuovi valori sono scaricati nel dispositivo e memorizzati automaticamente non appena si conferma l'impostazione.

**DEFAULT** 

Per caricare i parametri di default (parametri impostati durante la messa a punto in azienda del dispositivo che permettono un funzionamento standard e sicuro del dispositivo) premere il pulsante **DEFAULT** . Per ogni informazione sull'impostazione dei parametri di default riferirsi alla variabile **Load default parameters** a pagina 161. A pagina 186 è disponibile l'elenco dei dati macchina e il rispettivo valore di default preimpostato da Lika Electronic.

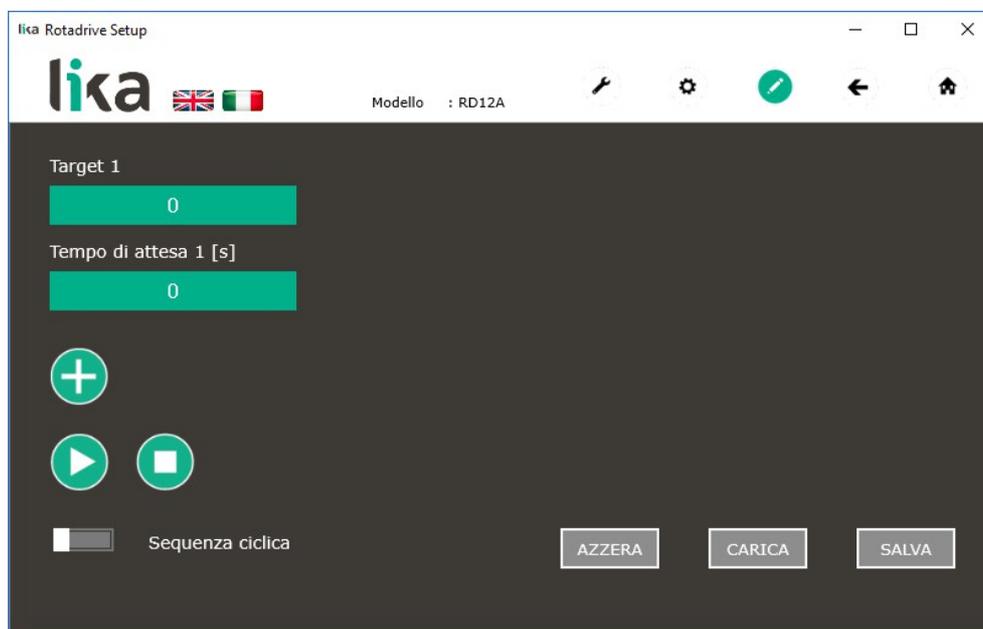
Nelle pagine **Parametri** sono disponibili anche le seguenti funzioni.

**ESEGUI** 

Il pulsante **ESEGUI**  è posizionato a fianco del parametro **Valore di preset** nella prima pagina dei **Parametri**. Quando si imposta un nuovo valore nel parametro **Valore di preset**, il preset è memorizzato, ma non attivato. Occorre premere il pulsante **ESEGUI**  per attivare il valore di preset per la posizione corrente dell'albero dell'attuatore. Per ogni ulteriore informazione sull'attivazione del valore di preset riferirsi al bit 11 **Setting the preset** nel registro **Control Word [0x2A]** a pagina 162. Riferirsi anche ai registri **Preset [0x12-0x13]** a pagina 157.

## 8.9 Pagina "Programma"

Premere il pulsante **SCHEDULE**  nella toolbar per accedere alla pagina **Programma**. Viene visualizzata la pagina seguente.

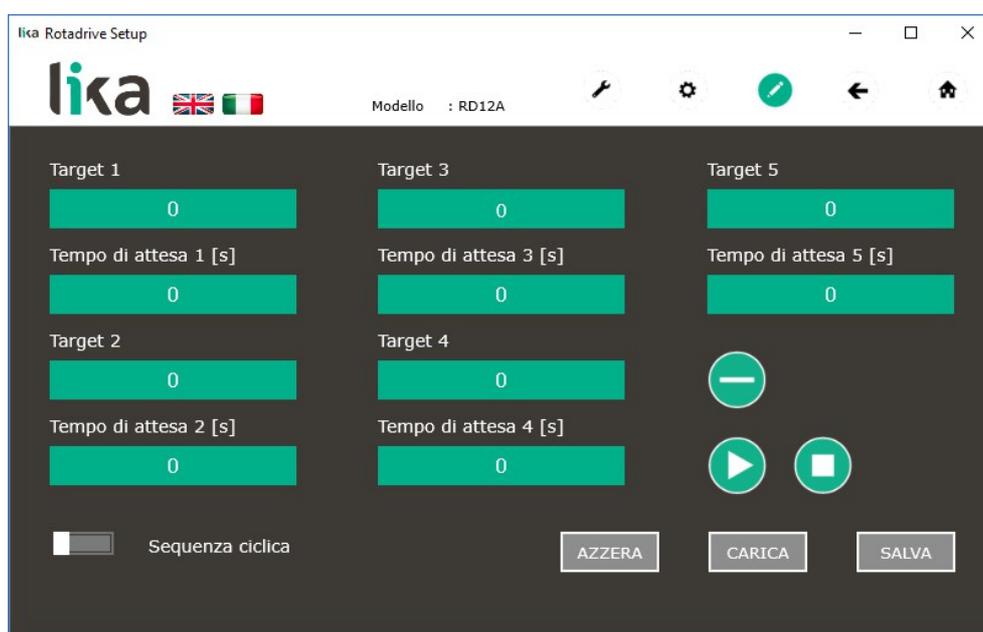


Le funzioni disponibili nella pagina **Programma** permettono all'operatore la creazione e il salvataggio di un programma di lavoro dell'unità RD1xA. Permettono cioè di testare il funzionamento dell'attuatore verificandone velocità, accelerazione, decelerazione, ecc. mediante l'impostazione di alcune quote da raggiungere (fino a cinque) e l'esecuzione delle relative corse.

### Target 1 ... Target 5

Nei campi **Target 1 ... Target 5** si impostano le quote che il dispositivo deve raggiungere (posizione target); si possono impostare fino a cinque quote successive. **Target 1** è la prima quota comandata. Premere il tasto **INVIO** della tastiera per confermare.

E' possibile aggiungere fino a cinque quote in sequenza che l'attuatore deve raggiungere, per fare questo premere il pulsante **PLUS** .



Premere il pulsante **MINUS**  per cancellare l'ultima quota target impostata.

### Tempo di attesa 1 [s]

Nei campi **Tempo di attesa [s]** si imposta la pausa tra un posizionamento e il successivo. I valori impostati devono essere confermati premendo il tasto **INVIO** della tastiera.

### Sequenza ciclica Sequenza ciclica

Il cursore **SEQUENZA CICLICA**  Sequenza ciclica permette di attivare l'esecuzione continua del ciclo di posizionamento impostato, ossia il dispositivo continua a muoversi e a eseguire i posizionamenti comandati senza interruzione. In altri termini, abilita la funzione di "loop": dopo la pressione del pulsante **START**  il dispositivo esegue i movimenti e i posizionamenti comandati senza interruzione, dalla quota **Target 1** alla quota **Target 5** (se abilitate) e poi di nuovo dalla quota **Target 1** alla quota **Target 5**, fino a quando non si preme il pulsante **STOP** .

Se il cursore **SEQUENZA CICLICA**  Sequenza ciclica è attivato, alla pressione del pulsante **START** , il dispositivo avvia il posizionamento alla prima quota impostata **Target 1**; un semaforo verde si accende a fianco del campo e in alto nella pagina compare una barra di avanzamento verde; al raggiungimento della quota impostata **Target 1**, il dispositivo si arresta per la pausa impostata in **Tempo di attesa 1 [s]**. Allo scadere della pausa impostata, un semaforo verde si accende a fianco del campo **Target 2** e il dispositivo avvia il posizionamento alla seconda quota impostata **Target 2**; di nuovo una barra di avanzamento verde

compare in alto nella pagina; e così di seguito dalla prima alla quinta quota (se abilitate) e di nuovo dalla prima alla quinta senza soluzione di continuità fino alla pressione del pulsante **STOP** .

Se invece il cursore **SEQUENZA CICLICA**  Sequenza ciclica non è abilitato, alla pressione del pulsante **START** , il dispositivo avvia il posizionamento alla prima quota impostata **Target 1** e al raggiungimento della quota si arresta e attiva il contatore per l'esecuzione della pausa impostata in **Tempo di attesa 1 [s]**; quindi, allo scadere della pausa, rimane in attesa di un nuovo comando **START**  per avviare il posizionamento alla seconda quota **Target 2**; e così di seguito.

Nella pagina **Programma** sono disponibili i seguenti pulsanti:

**PLUS** 

Premere il pulsante **PLUS**  per aggiungere ulteriori quote (posizionamenti) che l'attuatore dovrà raggiungere.

**MINUS** 

Premere il pulsante **MINUS**  per cancellare l'ultima quota target impostata.

**START** 

Premere il pulsante **START**  per avviare la sequenza dei posizionamenti impostati. Se il cursore **SEQUENZA CICLICA**  Sequenza ciclica è attivato, è necessario premere il pulsante **START**  una sola volta: l'attuatore raggiungerà in sequenza tutte le quote che sono state abilitate, senza interruzione. Se invece il cursore **SEQUENZA CICLICA**  Sequenza ciclica non è abilitato, occorre premere il pulsante **START**  dopo ogni sequenza per poter continuare.

**STOP** 

Premere il pulsante **STOP**  per arrestare la sequenza dei movimenti.

**AZZERA** 

Il pulsante **AZZERA**  resetta l'esecuzione del programma in corso, azzerando cioè il contatore delle sequenze del programma: alla successiva pressione del pulsante **START**  il dispositivo partirà dalla prima sequenza e avvierà il posizionamento alla prima quota impostata **Target 1**, quale che fosse la quota raggiunta prima che il contatore fosse azzerato.

**CARICA** 

Premere il pulsante **CARICA**  per caricare un programma precedentemente salvato. Alla pressione del pulsante si visualizza il dialog box **Apri**: l'operatore deve aprire la cartella dove si trova il file .DAT precedentemente salvato, selezionarlo e confermare la scelta mediante il pulsante **APRI**, il box si chiude e i valori di lavoro sono caricati automaticamente.

**SALVA** 

Premere il pulsante **SALVA** per salvare il programma di lavoro creato. Alla pressione del pulsante si visualizza il dialog box **Salva con nome**: l'operatore deve digitare il nome del file .DAT dove i dati saranno salvati e indicarne il repository. Alla conferma mediante il pulsante **SALVA**, il box si chiude. I valori impostati sono salvati automaticamente.

## 8.10 Principi guida del protocollo MODBUS Master / Slave

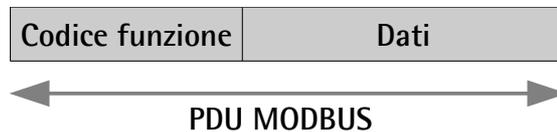
Il protocollo seriale Modbus è un protocollo Master – Slave. Un solo Master può essere connesso contemporaneamente alla rete Modbus; mentre il numero di Slave connessi allo stesso bus seriale può essere compreso tra 1 e 247. Una comunicazione Modbus è sempre iniziata dal Master. I nodi Slave non sono abilitati alla trasmissione di dati se non a seguito di una richiesta da parte del nodo Master. Inoltre i nodi Slave non solo abilitati a comunicare l'uno con l'altro. Il Master può attivare una sola transazione Modbus per volta.

Il nodo Master può inviare una richiesta Modbus ai nodi Slave in due modalità:

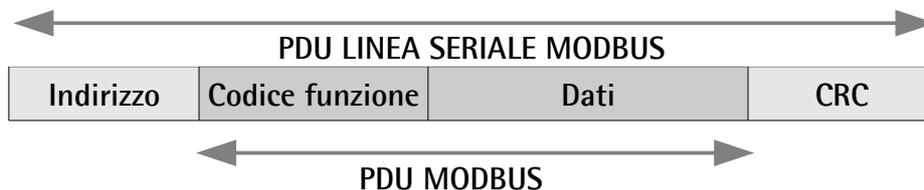
- **modalità UNICAST**: il Master invia la richiesta a un singolo Slave. Dopo aver ricevuto e processato la richiesta, lo Slave invia una risposta al Master. In questa modalità, la transazione Modbus consiste di due messaggi: una richiesta da parte del Master e una risposta da parte dello Slave. Ogni Slave deve avere il proprio indirizzo (da 1 a 247) di modo che la richiesta possa essere inviata specificatamente. Nei dispositivi Lika sono previsti comandi solo in modalità "unicast".
- **modalità BROADCAST**: il Master può inviare una richiesta a tutti gli Slave contemporaneamente. Gli Slave non inviano nessuna risposta a seguito di una richiesta di tipo "broadcast". Ne consegue che le richieste di tipo "broadcast" sono necessariamente dei comandi di scrittura. L'indirizzo 0 è riservato per identificare uno scambio dati in modalità "broadcast". Nei dispositivi Lika non sono previsti comandi in modalità "broadcast".

### 8.11 Frame MODBUS

Il protocollo applicativo Modbus definisce una semplice Protocol Data Unit (PDU) indipendentemente dal livello di comunicazione:



La mappatura del protocollo Modbus in specifici bus o reti introduce ulteriori campi nella PDU. Il Client che avvia una transazione Modbus prepara la PDU Modbus, quindi aggiunge i campi al fine di ottenere la PDU di comunicazione appropriata.



- **INDIRIZZO:** contiene l'indirizzo dello Slave. Come detto in precedenza (sezione "8.10 Principi guida del protocollo MODBUS Master / Slave" a pagina 135), gli indirizzi dei nodi Slave devono essere compresi tra 1 e 247. Il Master invia un messaggio a uno Slave impostando l'indirizzo nel campo INDIRIZZO del messaggio. Nella risposta, lo Slave pone a sua volta il proprio indirizzo, per far sì che il Master possa riconoscere da chi proviene il messaggio.
- **CODICE FUNZIONE:** indica al server il tipo di azione da eseguire. Si deve impostare il valore esadecimale del codice funzione voluto. Il codice funzione può essere seguito da un campo **DATI** che contiene i parametri di interrogazione e risposta. Per maggiori informazioni sui codici funzione implementati riferirsi alla sezione "8.13 Codici funzione" a pagina 140.
- **DATI:** byte dedicati alle informazioni aggiuntive e alla trasmissione dei dati, il numero di byte e la struttura dipendono da ciascun **CODICE FUNZIONE**. Il campo **DATI** include valori come per esempio indirizzi di registro, numero di registri da processare, numero di data byte presenti nel campo, ecc. (si veda alla sezione "8.13 Codici funzione" a pagina 140).
- **CRC (Cyclical Redundancy Checking, controllo a ridondanza ciclica):** campo di verifica della corretta trasmissione del frame, basato sul metodo del controllo a ridondanza ciclica. E' utilizzato per verificare se la trasmissione è stata realizzata correttamente. Il campo CRC ha una dimensione di 2 byte, contenenti un valore binario di 16 bit. Il valore CRC è calcolato dal dispositivo trasmittente che lo allega al messaggio. Il

dispositivo che riceve il messaggio ricalcola il valore del CRC alla ricezione e lo confronta con quello ricevuto. Se i due valori non sono uguali, il dispositivo attiva un allarme.

Il protocollo Modbus definisce tre PDU. Esse sono:

- **Modbus Request PDU;**
- **Modbus Response PDU;**
- **Modbus Exception Response PDU.**

La **Modbus Request PDU** consiste di {function\_code, request\_data}, dove:  
function\_code = codice funzione Modbus, 1 byte;  
request\_data = questo campo dipende dal codice funzione utilizzato e solitamente contiene informazioni quali coordinate di variabili, valori di variabili, offset dati, codici di sottofunzioni, ecc., n byte.

La **Modbus Response PDU** consiste di {function\_code, response\_data}, dove:  
function\_code = codice funzione Modbus, 1 byte;  
response\_data = questo campo dipende dal codice funzione utilizzato e solitamente contiene informazioni quali coordinate di variabili, valori di variabili, offset dati, codici di sottofunzioni, ecc., n byte.

La **Modbus Exception Response PDU** consiste di {exception-function\_code, exception\_code}, dove:  
exception-function\_code = codice funzione Modbus + 0x80, 1 byte;  
exception\_code = Modbus Exception code, riferirsi alla tabella "Modbus Exception Codes" nel documento "Modbus Application Protocol Specification V1.1b".

## 8.12 Modalità di trasmissione

Il protocollo seriale Modbus prevede due modalità di trasmissione: la **modalità RTU (Remote Terminal Unit)** e la **modalità ASCII**. La modalità di trasmissione definisce la sequenza dei bit nei campi messaggio trasmessi serialmente. Definisce cioè come le informazioni sono ordinate all'interno dei campi messaggio e codificate. La modalità di trasmissione e i parametri della porta seriale devono essere gli stessi per tutti i dispositivi della linea seriale Modbus. Tutti i dispositivi devono implementare la modalità di trasmissione RTU, mentre la modalità di trasmissione ASCII è opzionale. I dispositivi Lika implementano solamente la modalità di trasmissione RTU, descritta nel successivo paragrafo.

### 8.12.1 Modalità di trasmissione RTU

Quando dei dispositivi comunicano in una linea seriale Modbus utilizzando la modalità di trasmissione RTU, ogni byte di 8 bit del messaggio contiene due

caratteri esadecimali a 4 bit. Ogni messaggio deve essere inviato in una sequenza ininterrotta di caratteri. La sincronizzazione dei messaggi tra trasmettitore e ricevitore è ottenuta interponendo un intervallo tra messaggi successivi (chiamato "silent interval") pari ad almeno 3,5 volte il tempo di un carattere. In questo modo un messaggio Modbus è posto dal dispositivo trasmittente all'interno di un frame che ha un punto d'inizio e un punto di fine noti. Questo permette ai dispositivi che ricevono il nuovo frame di sapere qual è l'inizio del messaggio e quando il messaggio è completato. Se quindi il ricevitore non riceve un messaggio per un tempo di 4 caratteri, ritiene completato il messaggio precedente e considera che il successivo byte ricevuto sarà il primo del nuovo messaggio e quindi un indirizzo.

Il "silent interval" con velocità di trasmissione = 9600 bit/s è pari a 4 ms.

Il "silent interval" con velocità di trasmissione = 19200 bit/s è pari a 2 ms.

Il formato (11 bit) per ogni byte in modalità RTU sarà il seguente:

**Sistema di codifica:** binario a 8 bit  
**Bit per byte:** 1 bit di start;  
 8 bit di dati, lsb inviato per primo;  
 1 bit di parità (= Even);  
 1 bit di stop.

Il protocollo Modbus utilizza il formato Big Endian, questo significa che quando è trasmessa una quantità numerica più grande di un singolo byte, l'MSB è trasmesso per primo.

Ogni carattere o byte è trasmesso nel seguente ordine (da sinistra a destra):

lsb (Least Significant Bit) ... msb (Most Significant Bit)

Start	1	2	3	4	5	6	7	8	Parità*	Stop
-------	---	---	---	---	---	---	---	---	---------	------

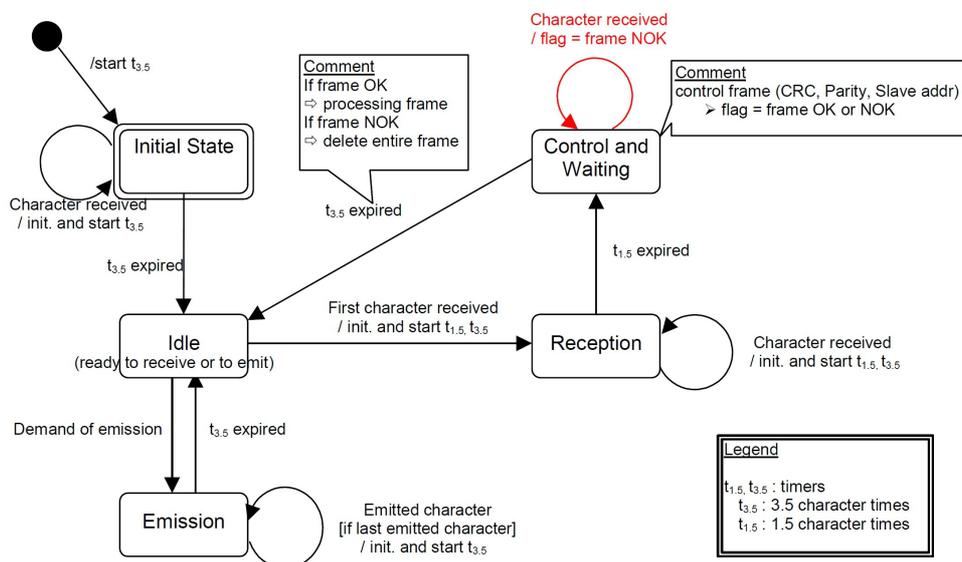
\* Con "No parity", il bit di parità è sostituito da un bit di stop.

Come default bisogna impostare il bit di parità = pari.

Il frame, che avrà una dimensione massima di 256 byte, sarà così composto:

Indirizzo	Codice funzione	Dati	CRC	
1 byte	1 byte	da 0 a 252 byte	CRC Low	CRC Hi

L'immagine che segue visualizza il diagramma degli stati nella modalità di trasmissione RTU.



- La transizione dallo stato **Initiate** allo stato **Idle** necessita di un intervallo di almeno 3,5 volte il tempo di un carattere ( $t_{3,5}$ ).
- Lo stato **Idle** è lo stato normale quando non sono attivi né invii né ricezioni, quando cioè non è presente attività di trasmissione dopo un intervallo di tempo pari ad almeno 3,5 volte il tempo di un carattere ( $t_{3,5}$ ).
- Una richiesta (request) può essere inviata solamente nello stato **Idle**. Dopo aver inviato una richiesta, il Master abbandona lo stato **Idle** e non può inviare una seconda richiesta nello stesso tempo.
- In una condizione **Idle**, la ricezione di un carattere trasmesso è considerata l'inizio di un frame e il link passa allo stato **Active**. Si considera il frame concluso quando non c'è trasmissione di caratteri per un intervallo almeno pari a  $t_{3,5}$ .
- Dopo che il frame è considerato concluso, si calcola e controlla il CRC. Successivamente si analizza il campo INDIRIZZO per determinare se il frame sia indirizzato al dispositivo. Se non è così, il frame viene scartato. Per ridurre il tempo di elaborazione della ricezione, l'INDIRIZZO può essere analizzato immediatamente alla ricezione senza attendere il completamento del frame. In questo caso il CRC viene calcolato e controllato solamente se il frame è effettivamente indirizzato allo Slave.

### 8.13 Codici funzione

Come detto in precedenza, i codici funzione esplicitano al server il tipo di azione da eseguire. Ogni codice funzione è codificato in un byte e i valori disponibili sono compresi tra 1 e 255 (ma i valori tra 128 e 255 sono riservati e utilizzati per le Exception response). Quando un messaggio è inviato dal Client al Server, il codice funzione esplicita al Server il tipo di azione da eseguire. Il codice funzione 0 non è ammesso.

Modbus prevede tre tipologie di codici funzione: i **codici funzione pubblici**, i **codici funzione definiti dall'utente** e i **codici funzione riservati**.

I **codici funzione pubblici** (compresi tra 1 e 64, tra 73 e 99 e tra 111 e 127) sono definiti e approvati da MODBUS-IDA.org che ne gestisce la conformità e ne garantisce l'unicità. I codici funzione compresi tra 65 e 72 e tra 100 e 110 sono a disposizione e possono essere definiti a piacimento dall'utilizzatore (**codici funzione utente**). Naturalmente non c'è alcuna garanzia che un codice utente sia univoco nella rete. I **codici funzione riservati** non sono invece disponibili in alcun modo agli utilizzatori.

#### 8.13.1 Codici funzione implementati

I dispositivi Lika RD1xA Modbus implementano esclusivamente i codici funzione pubblici descritti qui di seguito.

#### 03 Read Holding Register

FC = 03 (Hex = 0x03) ro

Questo codice funzione è utilizzato per LEGGERE i valori in un blocco di holding register contigui di un dispositivo remoto; in altri termini, permette la lettura dei valori impostati in alcuni parametri di lavoro in successione nel dispositivo. Il Request PDU specifica l'indirizzo del primo registro del gruppo e il numero di registri del gruppo. Nel PDU i Register sono indirizzati a partire da 0. Ne consegue che i registri numerati 1-16 sono indirizzati come 0-15.

Il valore del registro nel Response PDU è inviato in due byte per ciascun registro con il valore binario allineato a destra in ogni byte. Per ogni registro, il primo byte contiene i bit msb, mentre il secondo contiene i bit lsb.

Per un elenco dei parametri accessibili mediante il codice funzione **03 Read Holding Register**, si veda la sezione "8.14.1 Parametri Holding Register" a pagina 150.

#### Request PDU

Codice funzione	1 byte	<b>0x03</b>
Indirizzo iniziale	2 byte	da 0x0000 a 0xFFFF
Numero di registri	2 byte	da 1 a 125 (0x7D)

### Response PDU

Codice funzione	1 byte	<b>0x03</b>
Numero byte	1 byte	2 x N*
Valore dei registri	N* x 2 byte	

\*N = Numero di registri

### Exception Response PDU

Codice errore	1 byte	<b>0x83 (=0x03 + 0x80)</b>
Codice eccezione	1 byte	01 o 02 o 03 o 04



Esempio di richiesta di lettura dei parametri **Acceleration [0x07]** (registro 8) e **Deceleration [0x08]** (registro 9).

Request		Response	
Nome campo	(Hex)	Nome campo	(Hex)
Codice funzione	<b>03</b>	Codice funzione	<b>03</b>
Indirizzo iniziale Hi	<b>00</b>	Numero byte	<b>04</b>
Indirizzo iniziale Lo	<b>07</b>	Valore registro 8 Hi	<b>00</b>
Numero di registri Hi	<b>00</b>	Valore registro 8 Lo	<b>0A</b>
Numero di registri Lo	<b>02</b>	Valore registro 9 Hi	<b>00</b>
		Valore registro 9 Lo	<b>0A</b>

Come si evince dalla tabella, il parametro **Acceleration [0x07]** (registro 8) ha valore 00 0A hex, cioè 10 in notazione decimale; il parametro **Deceleration [0x08]** (registro 9) ha valore 00 0A hex, cioè 10 in notazione decimale.

Il frame completo di richiesta di lettura dei parametri **Acceleration [0x07]** (registro 8) e **Deceleration [0x08]** (registro 9) allo Slave con indirizzo 1 è il seguente:

**Request PDU** (in formato esadecimale)

[01][03][00][07][00][02][75][CA]

dove:

[01] = indirizzo dello Slave

[03] = codice funzione **03 Read Holding Register**  
 [00][07] = indirizzo iniziale (parametro **Acceleration [0x07]**, registro 8)  
 [00][02] = numero di registri richiesti  
 [75][CA] = CRC

Il frame completo di invio dei valori nei parametri **Acceleration [0x07]** (registro 8) e **Deceleration [0x08]** (registro 9) da parte dello Slave con indirizzo 1 è il seguente:

**Response PDU** (in formato esadecimale)

[01][03][04][00][0A][00][0A][5A][36]

dove:

[01] = indirizzo dello Slave  
 [03] = codice funzione **03 Read Holding Register**  
 [04] = numero di byte (2 byte per ciascun registro)  
 [00][0A] = valore del registro 8 **Acceleration [0x07]**, 00 0A hex = 10 dec  
 [00][0A] = valore del registro 9 **Deceleration [0x08]**, 00 0A hex = 10 dec  
 [5A][36] = CRC

#### 04 Read Input Register

FC = 04 (Hex = 0x04)

Questo codice funzione è utilizzato per LEGGERE registri di input contigui compresi tra 1 e 125 in un dispositivo remoto; in altri termini permette di leggere alcuni valori di risultato e gli stati / allarmi relativi al dispositivo. Il Request PDU specifica l'indirizzo del primo registro da leggere e il numero di registri da leggere. Nel PDU i registri sono indirizzati a partire da 0. Ne consegue che i registri di ingresso numerati 1-16 sono indirizzati come 0-15.

Il valore del registro nel Response PDU è inviato in due byte per ciascun registro con il valore binario allineato a destra in ogni byte. Per ogni registro, il primo byte contiene i bit msb, mentre il secondo contiene i bit lsb.

Per un elenco dei parametri accessibili mediante il codice funzione **04 Read Input Register**, si veda la sezione "8.14.2 Parametri Input Register" a pagina 165.

#### Request PDU

Codice funzione	1 byte	<b>0x04</b>
Indirizzo iniziale	2 byte	da 0x0000 a 0xFFFF
Numero di registri	2 byte	da 0x0000 a 0x007D

### Response PDU

Codice funzione	1 byte	<b>0x04</b>
Numero byte	1 byte	2 x N*
Valore dei registri	N* x 2 byte	

\*N = Numero di registri

### Exception Response PDU

Codice errore	1 byte	<b>0x84 (=0x04 + 0x80)</b>
Codice eccezione	1 byte	01 o 02 o 03 o 04



Esempio di richiesta di lettura del parametro **Current position [0x02-0x03]** (registri 3 e 4).

Request		Response	
Nome campo	(Hex)	Nome campo	(Hex)
Codice funzione	<b>04</b>	Codice funzione	<b>04</b>
Indirizzo iniziale Hi	<b>00</b>	Numero byte	<b>04</b>
Indirizzo iniziale Lo	<b>02</b>	Valore registro 3 Hi	<b>00</b>
Numero di registri Hi	<b>00</b>	Valore registro 3 Lo	<b>00</b>
Numero di registri Lo	<b>02</b>	Valore registro 4 Hi	<b>2F</b>
		Valore registro 4 Lo	<b>F0</b>

Come si evince dalla tabella, il parametro **Current position [0x02-0x03]** (registri 3 e 4) ha valore 00 00 2F F0 hex, cioè 12272 in notazione decimale.

Il frame completo di richiesta di lettura del parametro **Current position [0x02-0x03]** (registri 3 e 4) allo Slave con indirizzo 1 è il seguente:

**Request PDU** (in formato esadecimale)

[01][04][00][02][00][02][D0][0B]

dove:

[01] = indirizzo dello Slave

[04] = codice funzione **04 Read Input Register**

[00][02] = indirizzo iniziale (parametro **Current position [0x02-0x03]**, registro 3)

[00][02] = numero di registri richiesti

[D0][0B] = CRC

Il frame completo di invio dei valori del parametro **Current position [0x02-0x03]** (registri 3 e 4) da parte dello Slave con indirizzo 1 è il seguente:

**Response PDU** (in formato esadecimale)

[01][04][04][00][00][2F][F0][E7][F0]

dove:

[01] = indirizzo dello Slave

[04] = codice funzione **04 Read Input Register**

[04] = numero di byte (2 byte per ciascun registro)

[00][00] = valore del registro 3 **Current position [0x02-0x03]**, 00 00 hex = 0 dec

[2F][F0] = valore del registro 4 **Current position [0x02-0x03]**, 2F F0 hex = 12272 dec

[E7][F0] = CRC

## 06 Write Single Register

FC = 06 (Hex = 0x06)

Questo codice funzione è utilizzato per ASSEGNARE UN VALORE a un singolo holding register in un dispositivo remoto. Il Request PDU specifica l'indirizzo del registro da scrivere. I registri sono indirizzati a partire da 0. Ne consegue che i registri numerati 1-16 sono indirizzati come 0-15.

La risposta positiva rispecchia nella sua struttura la domanda ed è inviata dopo che il valore richiesto è stato scritto.

Per un elenco dei parametri accessibili mediante il codice funzione **06 Write Single Register**, si veda la sezione "8.14.1 Parametri Holding Register" a pagina 150.

### Request PDU

Codice funzione	1 byte	<b>0x06</b>
Indirizzo del registro	2 byte	da 0x0000 a 0xFFFF
Valore del registro	2 byte	da 0x0000 a 0xFFFF

### Response PDU

Codice funzione	1 byte	<b>0x06</b>
Indirizzo del registro	2 byte	da 0x0000 a 0xFFFF
Valore del registro	2 byte	da 0x0000 a 0xFFFF

### Exception Response PDU

Codice errore	1 byte	<b>0x86 (=0x06 + 0x80)</b>
Codice eccezione	1 byte	01 o 02 o 03 o 04



Esempio di scrittura del valore 00 96 hex (= 150 dec) nel parametro **Acceleration [0x07]** (registro 8).

Request		Response	
Nome campo	(Hex)	Nome campo	(Hex)
Codice funzione	<b>06</b>	Codice funzione	<b>06</b>
Indirizzo registro Hi	<b>00</b>	Indirizzo registro Hi	<b>00</b>
Indirizzo registro Lo	<b>07</b>	Indirizzo registro Lo	<b>07</b>
Valore del registro Hi	<b>00</b>	Valore del registro Hi	<b>00</b>
Valore del registro Lo	<b>96</b>	Valore del registro Lo	<b>96</b>

Come si evince dalla tabella, nel parametro **Acceleration [0x07]** (registro 8) si imposta il valore 00 96 hex, cioè 150 in notazione decimale.

Il frame completo di richiesta di scrittura del valore 00 96 hex (= 150 dec) nel parametro **Acceleration [0x07]** (registro 8) dello Slave con indirizzo 1 è il seguente:

**Request PDU** (in formato esadecimale)

[01][06][00][07][00][96][B8][65]

dove:

[01] = indirizzo dello Slave

[06] = codice funzione **06 Write Single Register**

[00][07] = indirizzo del registro (parametro **Acceleration [0x07]**, registro 8)

[00][96] = valore da impostare nel registro

[B8][65] = CRC

Il frame completo di risposta alla richiesta di scrittura del valore 00 96 hex (= 150 dec) nel parametro **Acceleration [0x07]** (registro 8) da parte dello Slave con indirizzo 1 è il seguente:

**Response PDU** (in formato esadecimale)

[01][06][00][07][00][96][B8][65]

dove:

[01] = indirizzo dello Slave

[06] = codice funzione **06 Write Single Register**

[00][07] = indirizzo del registro (parametro **Acceleration [0x07]**, registro 8)

[00][96] = valore impostato nel registro

[B8][65] = CRC

## 16 Write Multiple Register

FC = 16 (Hex = 0x10)

Questo codice funzione è utilizzato per ASSEGNARE DEI VALORI a un blocco di registri contigui (registri da 1 a 123) in un dispositivo remoto.

I valori da impostare sono specificati nel campo dati della richiesta. Il valore da assegnare a ogni registro è inviato in due byte per ciascun registro.

La risposta positiva restituisce il codice funzione, l'indirizzo iniziale e il numero di registri su cui si è scritto.

Per un elenco dei parametri accessibili mediante il codice funzione **16 Write Multiple Register**, si veda la sezione "8.14.1 Parametri Holding Register" a pagina 150.

## Request PDU

Codice funzione	1 byte	<b>0x10</b>
Indirizzo iniziale	2 byte	da 0x0000 a 0xFFFF
Numero di registri	2 byte	da 0x0001 a 0x007B
Numero byte	1 byte	2 x <b>N*</b>
Valore dei registri	<b>N*</b> x 2 byte	valore

\*N = Numero di registri

### Response PDU

Codice funzione	1 byte	<b>0x10</b>
Indirizzo iniziale	2 byte	da 0x0000 a 0xFFFF
Numero di registri	2 byte	da 1 a 123 (0x7B)

### Exception Response PDU

Codice errore	1 byte	<b>0x90 (= 0x10 + 0x80)</b>
Codice eccezione	1 byte	01 o 02 o 03 o 04



Esempio di richiesta di scrittura dei valori 150 e 100 nei parametri **Acceleration [0x07]** (registro 8) e **Deceleration [0x08]** (registro 9).

Request		Response	
Nome campo	(Hex)	Nome campo	(Hex)
Codice funzione	<b>10</b>	Codice funzione	<b>10</b>
Indirizzo iniziale Hi	<b>00</b>	Indirizzo iniziale Hi	<b>00</b>
Indirizzo iniziale Lo	<b>07</b>	Indirizzo iniziale Lo	<b>07</b>
Numero di registri Hi	<b>00</b>	Numero di registri Hi	<b>00</b>
Numero di registri Lo	<b>02</b>	Numero di registri Lo	<b>02</b>
Numero byte	<b>04</b>		
Valore registro 8 Hi	<b>00</b>		
Valore registro 8 Lo	<b>96</b>		
Valore registro 9 Hi	<b>00</b>		
Valore registro 9 Lo	<b>64</b>		

Come si evince dalla tabella, nel parametro **Acceleration [0x07]** (registro 8) si imposta il valore 00 96 hex, cioè 150 in notazione decimale; nel parametro **Deceleration [0x08]** (registro 9) si imposta il valore 00 64 hex, cioè 100 in notazione decimale.

Il frame completo di richiesta di scrittura dei valori 00 96 hex (= 150 dec) e 00 64 hex (= 100 dec) nei parametri **Acceleration [0x07]** (registro 8) e **Deceleration [0x08]** (registro 9) dello Slave con indirizzo 1 è il seguente:

**Request PDU** (in formato esadecimale)

[01][10][00][07][00][02][04][00][96][00][64][53][8E]

dove:

[01] = indirizzo dello Slave

[10] = codice funzione **16 Write Multiple Register**

[00][07] = indirizzo iniziale (parametro **Acceleration [0x07]**, registro 8)

[00][02] = numero di registri richiesti

[04] = numero di byte (2 byte per ciascun registro)

[00][96] = valore da impostare nel registro 8 **Acceleration [0x07]**, 00 96 hex = 150 dec

[00][64] = valore da impostare nel registro 9 **Deceleration [0x08]**, 00 64 hex = 100 dec

[53][8E] = CRC

Il frame completo di risposta alla richiesta di impostazione dei valori 00 96 hex (= 150 dec) e 00 64 hex (= 100 dec) nei parametri **Acceleration [0x07]** (registro 8) e **Deceleration [0x08]** (registro 9) da parte dello Slave con indirizzo 1 è il seguente:

**Response PDU** (in formato esadecimale)

[01][10][00][07][00][02][F0][09]

dove:

[01] = indirizzo Slave

[10] = codice funzione **16 Write Multiple Register**

[00][07] = indirizzo iniziale (parametro **Acceleration [0x07]**, registro 8)

[00][02] = numero di registri su cui si è scritto

[F0][09] = CRC



#### NOTA

Per ulteriori esempi si veda anche alla sezione "8.16 Esempi di programmazione" a pagina 180.



#### ATTENZIONE

Per motivi di sicurezza, durante il movimento dell'unità DRIVECOD si deve sempre prevedere un continuo scambio di dati tra Master e Slave per monitorare l'effettiva presenza di comunicazione; questo si rende necessario per evitare situazioni di pericolo nel caso in cui fossero presenti guasti nella rete di comunicazione.

A questo scopo il sistema prevede l'attivazione opzionale della funzione Watch dog. Il Watch dog è un sistema di sicurezza che, grazie a un timeout, permette di rilevare condizioni di loop o di deadlock. Per esempio, nel caso in cui si interrompesse la comunicazione seriale mentre è attivo un comando -a esempio un comando di jog- il Watch dog interverrebbe comandando lo stop in sicurezza del dispositivo e l'attivazione di un allarme. Per abilitare il Watch dog impostare a "1" il bit **Watch dog enable** in **Control Word [0x2A]**. Se impostato a "0" il Watch dog non è attivo; se impostato a "1" il Watch dog è attivo. Con Watch dog attivo, se il dispositivo non riceve un messaggio dal Server entro 1 secondo, il sistema forza una condizione di allarme (visualizzazione dell'allarme **Watch dog** al ripristino della comunicazione con la rete Modbus).

## 8.14 Parametri di programmazione

Di seguito sono riportati i parametri disponibili per il dispositivo RD1xA Modbus, per ognuno è indicato:

### Nome parametro [Indirizzo registro]

[Numero registro, tipo variabile, attributo]

- L'indirizzo del registro è espresso in valore esadecimale.
- Il numero del registro è espresso in valore decimale.
- Attributo:
  - ro = variabile accessibile in sola lettura
  - rw = variabile accessibile in lettura e scrittura

I registri MODBUS hanno una dimensione di 16 bit; i parametri dell'attuatore hanno la dimensione di 1 registro, ossia 16 bit, o di 2 registri, ossia 32 bit.

#### Struttura parametri Unsigned16:

byte	MSB			LSB		
bit	15	...	8	7	...	0
	msb		lsb	msb		lsb

#### Struttura parametri Unsigned32:

word	MSW			LSW		
bit	31	...	16	15	...	0
	msb		lsb	msb		lsb

### 8.14.1 Parametri Holding Register

Gli **Holding register** (parametri **Dati macchina**) sono accessibili sia in lettura che scrittura; per leggere il valore di un parametro utilizzare il codice funzione **03 Read Holding Register** (lettura multipla dei registri); per scrivere il valore in un parametro utilizzare il codice funzione **06 Write Single Register** (scrittura di un singolo registro) oppure il codice funzione **16 Write Multiple Register** (scrittura di più registri); per ogni informazione sui codici funzione implementati riferirsi alla sezione "8.13.1 Codici funzione implementati" a pagina 140.

**NOTA**

Salvare sempre i nuovi valori dopo l'impostazione al fine di memorizzarli permanentemente nella memoria non volatile. Usare la funzione **Save parameters** disponibile nell'oggetto **Control Word [0x2A]**, si veda a pagina 158.

Nel caso di spegnimento del dispositivo i dati non salvati andranno persi.

**ATTENZIONE**

Per motivi di sicurezza i seguenti parametri holding register **Extra commands register [0x29]**, **Control Word [0x2A]** e **Target position [0x2B-0x2C]** non vengono salvati in memoria. Perciò devono essere reimpostati dopo ogni accensione.

**Distance per revolution [0x00]**

[Registro 1, Unsigned16, rw]

Questo parametro imposta il numero di impulsi per ogni giro completo dell'albero. Si rivela utile per relazionare un giro dell'asse con una grandezza lineare. Per esempio: se il dispositivo è montato su una vite senza fine con passo 5 mm, impostando **Distance per revolution [0x00]** = 500, si ottiene che a ogni giro dell'asse il sistema trasla di 5 mm con una risoluzione al centesimo di millimetro.

Default = 1024 (min. = 1, max. = 1024)

**ATTENZIONE**

Dopo la modifica di questo parametro è necessario reimpostare anche il parametro **Preset [0x12-0x13]**. Per una spiegazione dettagliata si veda a pagina 51 e i relativi parametri.

Si badi inoltre che i parametri di seguito elencati sono in stretta correlazione con il parametro **Distance per revolution [0x00]**; pertanto la modifica del valore del parametro **Distance per revolution [0x00]** comporta necessariamente una ridefinizione dei valori da essi espressi. Essi sono: **Position window [0x01]**, **Max following error [0x03-0x04]**, **Positive delta [0x09-0x0A]**, **Negative delta [0x0B-0x0C]**, **Target position [0x2B-0x2C]**, **Current position [0x02-0x03]** e **Position following error [0x05-0x06]**.

**NOTA**

Se **Distance per revolution [0x00]** non è una potenza di 2 (2, 4, ..., 512, 1024), durante il controllo di posizione potrebbe verificarsi un errore di posizionamento pari a un impulso.

### Position window [0x01]

[Registro 2, Unsigned16, rw]

Questo parametro definisce la finestra di tolleranza da applicare al valore impostato in **Target position [0x2B-0x2C]**. Non appena l'asse si trova all'interno dei limiti di tolleranza, il bit 8 **Target position reached** della **Status word [0x01]** va alto ("=1"). Quando la posizione dell'asse è all'interno dei limiti di tolleranza per il tempo impostato nel parametro **Position window time [0x02]**, il bit 0 **Axis in position** della **Status word [0x01]** va alto ("=1"). Il parametro è espresso in impulsi. Si veda anche la sezione "Posizione: controllo di posizione e velocità" a pagina 49.

Default = 1 (min. = 0, max. = 65535)

### Position window time [0x02]

[Registro 3, Unsigned16, rw]

Rappresenta il tempo per la durata del quale la posizione dell'asse deve essere compresa all'interno dei limiti di tolleranza impostati nei parametri **Position window [0x01]** prima che lo stato sia segnalato attraverso il bit di stato **Axis in position** della **Status word [0x01]**. Il parametro è espresso in millisecondi (ms). Si veda anche la sezione "Posizione: controllo di posizione e velocità" a pagina 49.

Default = 0 (min. = 0, max. = 10000)

### Max following error [0x03-0x04]

[Registri 4-5, Unsigned32, rw]

Questo parametro definisce la differenza massima ammissibile tra la posizione reale del dispositivo e quella teorica. Se il dispositivo rileva un valore superiore a quello impostato in questo parametro, viene segnalato l'allarme **Following error** e il dispositivo arresta il proprio movimento. Il parametro è espresso in impulsi.

Default = 1024 (min. = 0, max. = 65535)

### Kp position loop [0x05]

[Registro 6, Unsigned16, rw]

Questo parametro contiene il guadagno proporzionale usato dal controllore PI relativo all'anello di posizione. Tale valore è già ottimizzato da Lika Electronic in relazione alle caratteristiche del dispositivo.

Default = 300 (min. = 0, max. = 1000)

**Ki position loop [0x06]**

[Registro 7, Unsigned16, rw]

Questo parametro contiene il guadagno integrale usato dal controllore PI relativo all'anello di posizione. Tale valore è già ottimizzato da Lika Electronic in relazione alle caratteristiche del dispositivo.

Default = 10 (min. = 0, max. = 1000)

**Acceleration [0x07]**

[Registro 8, Unsigned16, rw]

Questo parametro definisce il valore di accelerazione di cui si serve il dispositivo per raggiungere la velocità di jog **Jog speed [0x0D]** o la velocità di lavoro **Work speed [0x0E]**. Il parametro è espresso in giri al secondo<sup>2</sup> [giri/s<sup>2</sup>]. Si veda anche la sezione "6.2 Tipi di movimento: jog e posizione" a pagina 48.

Default = 10 (min. = 1, max. = 500)

**Deceleration [0x08]**

[Registro 9, Unsigned16, rw]

Questo parametro definisce il valore di decelerazione di cui si serve il dispositivo quando si arresta. Il parametro è espresso in giri al secondo<sup>2</sup> [giri/s<sup>2</sup>]. Si veda anche la sezione "6.2 Tipi di movimento: jog e posizione" a pagina 48.

Default = 10 (min. = 1, max. = 500)

**Positive delta [0x09-0x0A]**

[Registri 10-11, Unsigned32, rw]

Questo valore è utilizzato per calcolare il massimo spostamento in avanti (positivo) che il dispositivo può raggiungere a partire dal preset. Non appena si raggiunge il massimo spostamento in avanti, la condizione è segnalata attraverso il bit di stato **SW limit switch +** della **Status word [0x01]** (il bit è forzato alto). Il parametro è espresso in impulsi.

**SW limit switch + = Preset [0x12-0x13] + Positive delta [0x09-0x0A].**

Per ulteriori informazioni riferirsi alla sezione "6.4 Distance per revolution, Preset, Max delta pos / Positive delta e Max delta neg / Negative delta" a pagina 51.

Default = 523 263 (min. = 0, max. = 523 263)

**ATTENZIONE**

Si badi che il valore massimo di volta in volta ammesso in questo parametro dipende dallo scaling impostato.



### ESEMPIO

Se **Distance per revolution [0x00]** = 1.024 e **Preset [0x12-0x13]** = 0, il massimo valore accettabile per **Positive delta [0x09-0x0A]** è:

$(1.024 \text{ imp./giro} * 512 \text{ giri}) - 1 \text{ imp.} - 1.024 \text{ imp.}$  (ossia 1 giro per motivi di sicurezza) = 523.263

Se **Distance per revolution [0x00]** = 256 e **Preset [0x12-0x13]** = 0, il massimo valore accettabile per **Positive delta [0x09-0x0A]** è:

$(256 \text{ imp./giro} * 512 \text{ giri}) - 1 \text{ imp.} - 256 \text{ imp.}$  (ossia 1 giro per motivi di sicurezza) = 130.815

Altri esempi alla sezione "6.4 Distance per revolution, Preset, Max delta pos / Positive delta e Max delta neg / Negative delta" a pagina 51.



### ATTENZIONE

Ogniqualvolta si modificano i parametri **Distance per revolution [0x00]** e **Preset [0x12-0x13]**, occorre poi verificare attentamente i valori in **Positive delta [0x09-0x0A]** e **Negative delta [0x0B-0x0C]**. Ogniqualvolta si modifica il valore in **Distance per revolution [0x00]**, occorre poi reimpostare il valore in **Preset [0x12-0x13]** in modo da definire lo zero asse in quanto è cambiata l'unità di misura di riferimento del sistema.

Dopo la modifica del parametro **Preset [0x12-0x13]** non occorre invece reimpostare il valore dei finecorsa in quanto la funzione di preset provvede a ricalcolarli automaticamente reinizializzando i limiti positivo e negativo sulla base dei valori impostati nei parametri **Positive delta [0x09-0x0A]** e **Negative delta [0x0B-0x0C]**. Per una spiegazione dettagliata si veda a pagina 51.

### Negative delta [0x0B-0x0C]

[Registri 12-13, Unsigned32, rw]

Questo valore è utilizzato per calcolare il massimo spostamento all'indietro (negativo) che il dispositivo può raggiungere a partire dal preset. Non appena si raggiunge il massimo spostamento all'indietro, la condizione è segnalata attraverso il bit di stato **SW limit switch** - della **Status word [0x01]** (il bit è forzato alto). Il parametro è espresso in impulsi.

**SW limit switch** - = **Preset [0x12-0x13]** - **Negative delta [0x0B-0x0C]**.

Per ulteriori informazioni riferirsi alla sezione "6.4 Distance per revolution, Preset, Max delta pos / Positive delta e Max delta neg / Negative delta" a pagina 51.

Default = 523 263 (min. = 0, max. = 523 263)



### ATTENZIONE

Si badi che il valore massimo di volta in volta ammesso in questo parametro dipende dallo scaling impostato.



### ESEMPIO

Se **Distance per revolution [0x00]** = 1.024 e **Preset [0x12-0x13]** = 0, il massimo valore accettabile per **Negative delta [0x0B-0x0C]** è:

$(1.024 \text{ imp./giro} * 512 \text{ giri}) - 1 \text{ imp.} - 1.024 \text{ imp.}$  (ossia 1 giro per motivi di sicurezza) = 523.263

Se **Distance per revolution [0x00]** = 256 e **Preset [0x12-0x13]** = 0, il massimo valore accettabile per **Negative delta [0x0B-0x0C]** è:

$(256 \text{ imp./giro} * 512 \text{ giri}) - 1 \text{ imp.} - 256 \text{ imp.}$  (ossia 1 giro per motivi di sicurezza) = 130.815

Altri esempi alla sezione "6.4 Distance per revolution, Preset, Max delta pos / Positive delta e Max delta neg / Negative delta" a pagina 51.



### ATTENZIONE

Ogniqualevolta si modificano i parametri **Distance per revolution [0x00]** e **Preset [0x12-0x13]**, occorre poi verificare attentamente i valori in **Positive delta [0x09-0x0A]** e **Negative delta [0x0B-0x0C]**. Ogniqualevolta si modifica il valore in **Distance per revolution [0x00]**, occorre poi reimpostare il valore in **Preset [0x12-0x13]** in modo da definire lo zero asse in quanto è cambiata l'unità di misura di riferimento del sistema.

Dopo la modifica del parametro **Preset [0x12-0x13]** non occorre invece reimpostare il valore dei finecorsa in quanto la funzione di preset provvede a ricalcolarli automaticamente reinizializzando i limiti positivo e negativo sulla base dei valori impostati nei parametri **Positive delta [0x09-0x0A]** e **Negative delta [0x0B-0x0C]**. Per una spiegazione dettagliata si veda a pagina 51.

### Jog speed [0x0D]

[Registro 14, Unsigned16, rw]

Questo parametro definisce la velocità massima che il motore può raggiungere quando si utilizzano le funzioni **Jog +** e **Jog -** (si veda il parametro **Control Word [0x2A]**). Il parametro è espresso in giri al minuto (rpm). Si veda anche la sezione "Jog: controllo di velocità" a pagina 48.

Default = 2000 (min. = 1, max. = 3000)



### NOTA

Si badi che questa è la velocità del motore, non la velocità dell'albero in uscita dopo il riduttore.

La velocità in uscita sarà come segue:

Velocità del motore	= 2000 rpm
Velocità in uscita:	T12 = 166 rpm
	T24 = 83 rpm
	T48 = 41 rpm
	T92 = 21 rpm

**Work speed [0x0E]**

[Registro 15, Unsigned16, rw]

Questo parametro definisce la velocità massima che il motore può raggiungere in funzionamento automatico (i movimenti sono controllati mediante i comandi **Start** e **Stop** -si veda il parametro **Control Word [0x2A]** - e sono eseguiti al fine di raggiungere la posizione impostata in **Target position [0x2B-0x2C]**). Il parametro è espresso in giri al minuto (rpm). Si veda anche la sezione "Posizione: controllo di posizione e velocità" a pagina 49.

Default = 2000 (min. = 1, max. = 3000)

**NOTA**

Si badi che questa è la velocità del motore, non la velocità dell'albero in uscita dopo il riduttore.

La velocità in uscita sarà come segue:

Velocità del motore	= 2000 rpm
Velocità in uscita:	T12 = 166 rpm
	T24 = 83 rpm
	T48 = 41 rpm
	T92 = 21 rpm

**Code sequence [0x0F]**

[Registro 16, Unsigned16, rw]

Imposta se il valore di posizione provvisto dal dispositivo incrementa (conteggio crescente) quando l'albero ruota in senso orario (0) o antiorario (1). La direzione della rotazione oraria e antioraria è stabilita guardando il dispositivo dall'estremità dell'asse.

0 = informazione crescente con rotazione oraria (default)

1 = informazione crescente con rotazione antioraria

**ATTENZIONE**

La modifica di questo parametro influenza la posizione calcolata dal controllore. Si deve quindi reimpostare poi il parametro **Preset [0x12-0x13]** e verificare i valori nei parametri **Positive delta [0x09-0x0A]** e **Negative delta [0x0B-0x0C]**.

**Offset [0x10-0x11]**

[Registri 17-18, Integer32, ro]

Questa variabile definisce la differenza tra il valore di posizione trasmesso dal dispositivo e la posizione reale: posizione reale - preset. Il valore è espresso in impulsi.

Default = 0

### Preset [0x12-0x13]

[Registri 19-20, Integer32, rw]

Usare questo parametro per impostare il valore di Preset. La funzione di Preset è utilizzata per assegnare un valore desiderato a una posizione fisica dell'asse. La posizione fisica prescelta assumerà pertanto il valore assegnato in questo parametro e tutte le posizioni precedenti e successive assumeranno un valore conseguente. Il valore di preset sarà assegnato alla posizione dell'asse al momento dell'invio del parametro. Il valore di preset è attivato quando il bit 11 **Setting the preset** nel registro **Control Word [0x2A]** commuta dal livello logico basso ("0") al livello logico alto ("1").

Default = 0 (min. = -1 048 576, max. = +1 048 576)



#### NOTA

Consigliamo di attivare il preset quando l'attuatore è in stop. Si veda il comando **Setting the preset** a pagina 162.



#### ATTENZIONE

Occorre poi reimpostare un nuovo valore nei registri **Preset [0x12-0x13]** ogniqualvolta si modifica il valore in **Distance per revolution [0x00]**. Dopo la modifica del parametro **Preset [0x12-0x13]** non occorre invece reimpostare il valore dei finecorsa in quanto la funzione di preset provvede a ricalcolarli automaticamente reinizializzando i limiti positivo e negativo sulla base dei valori impostati nei parametri **Positive delta [0x09-0x0A]** e **Negative delta [0x0B-0x0C]**. Per una spiegazione dettagliata si veda a pagina 51.

### Jog step length [0x14]

[Registro 21, Unsigned16, rw]

Se il controllo del jog a passo (funzione di jog incrementale) è abilitato (bit 4 **Incremental jog** in **Control Word [0x2A]** = 1), l'attivazione dei bit **Jog +** e **Jog -** procura, in corrispondenza del fronte di salita, l'esecuzione di un singolo passo in direzione positiva o negativa la cui ampiezza, espressa in impulsi, è impostata in questo parametro; l'attuatore poi si arresta in attesa di un nuovo comando.

Default = 1000 (min. = 1, max. = 10000).

### Extra commands register [0x29]

[Registro 42, Unsigned16, rw]

Struttura byte del parametro **Extra commands register [0x29]**:

byte	MSB			LSB		
bit	15	...	8	7	...	0
	msb		lsb	msb		lsb

#### Byte 0

bit 0: Non usato.

#### Control by PC

bit 1: Questa funzione è riservata per il solo utilizzo da parte dei tecnici di Lika Electronic (utilizzata esclusivamente con porta di servizio Modbus).

bit 2 ... 7 Non usati.

Byte 1 Non usato.



#### ATTENZIONE

Per motivi di sicurezza il parametro holding register **Extra commands register [0x29]** non viene salvato in memoria. Perciò deve essere reimpostato dopo ogni accensione.

### Control Word [0x2A]

[Registro 43, Unsigned16, rw]

Questa variabile contiene i comandi da inviare in tempo reale allo Slave per controllarlo.

Struttura byte del registro **Control Word [0x2A]**:

byte	MSB			LSB		
bit	15	...	8	7	...	0
	msb		lsb	msb		lsb

#### Byte 0

##### Jog +

bit 0 Se il bit 4 **Incremental jog** = 0, lo Slave si muove in direzione positiva fintanto che **Jog +** = 1; se invece il bit 4 **Incremental jog** = 1, in corrispondenza del fronte di salita di **Jog +** lo Slave esegue un singolo passo in direzione positiva la cui ampiezza, espressa in impulsi, è impostata nel parametro **Jog step length [0x14]**; quindi si arresta in

attesa di un nuovo comando. La velocità, l'accelerazione e la decelerazione sono definite nei parametri **Jog speed [0x0D]**, **Acceleration [0x07]** e **Deceleration [0x08]** rispettivamente. Per una descrizione dettagliata del controllo jog si veda a pagina 48.



Le funzioni **Jog +**, **Jog -** e **Start** non possono essere abilitate simultaneamente. Per esempio: se un comando **Jog +** viene inviato allo Slave mentre questo sta eseguendo il movimento verso la posizione target, il comando jog viene ignorato; se i comandi **Jog +** e **Jog -** sono inviati simultaneamente, il dispositivo non si muove o, se già in movimento, si arresta.

**Jog -**  
bit 1

Se il bit 4 **Incremental jog** = 0, lo Slave si muove in direzione negativa fintanto che **Jog -** = 1; se invece il bit 4 **Incremental jog** = 1, in corrispondenza del fronte di salita di **Jog -** lo Slave esegue un singolo passo in direzione negativa la cui ampiezza, espressa in impulsi, è impostata nel parametro **Jog step length [0x14]**; quindi si arresta in attesa di un nuovo comando. La velocità, l'accelerazione e la decelerazione sono definite nei parametri **Jog speed [0x0D]**, **Acceleration [0x07]** e **Deceleration [0x08]** rispettivamente. Per una descrizione dettagliata del controllo jog si veda a pagina 48.



Le funzioni **Jog +**, **Jog -** e **Start** non possono essere abilitate simultaneamente. Per esempio: se un comando **Jog +** viene inviato allo Slave mentre questo sta eseguendo il movimento verso la posizione target, il comando jog viene ignorato; se i comandi **Jog +** e **Jog -** sono inviati simultaneamente, il dispositivo non si muove o, se già in movimento, si arresta.

**Stop**  
bit 2

Se impostato a "1" lo Slave è libero di eseguire i comandi di movimento ricevuti. Se durante il movimento questo bit commuta a "0" allora lo Slave si arresta eseguendo la decelerazione impostata in **Deceleration [0x08]**. Per un arresto immediato del movimento, utilizzare il bit 7 **Emergency**.

**Alarm reset**  
bit 3

Questo comando è usato per ripristinare una condizione di allarme dello Slave, ma solo se sono state rimosse le cause che hanno provocato l'errore. In una normale condizione di lavoro il bit ha valore "0". Impostando questo bit a "1" si

ristabilisce il normale stato di lavoro del dispositivo. Il normale stato operativo è ripristinato nella commutazione da "0" a "1".



Si badi che, se l'allarme è relativo a dati macchina non validi (si vedano **Machine data not valid** e **Wrong parameters list [0x08-0x09]**), si può ripristinare lo stato normale di funzionamento solo impostando dati macchina validi. Gli allarmi **Flash memory error** e **Axis not synchronized** non sono ripristinabili.

### Incremental jog

bit 4

Se il bit 4 è "=0", l'attivazione dei bit **Jog +** e **Jog -** procura il movimento dello Slave fintanto che **Jog + / Jog - = 1**. Impostando questo bit a 1 si abilita la funzione di jog incrementale (jog a passo), ossia: l'attivazione dei bit **Jog +** e **Jog -** procura, in corrispondenza del fronte di salita, l'esecuzione di un singolo passo in direzione positiva o negativa la cui ampiezza, espressa in impulsi, è impostata nel parametro **Jog step length [0x14]**; quindi lo Slave si arresta in attesa di un nuovo comando.

bit 5

Non usato.

### Start

bit 6

Se impostato a "=1" il dispositivo si muove allo scopo di raggiungere la posizione di target specificata (si veda **Target position [0x2B-0x2C]** a pagina 163). Per una descrizione completa del controllo di posizione di veda a pagina 49.



Le funzioni **Jog +**, **Jog -** e **Start** non possono essere abilitate simultaneamente. Per esempio: se un comando **Jog +** viene inviato allo Slave mentre questo sta eseguendo il movimento verso la posizione target, il comando jog viene ignorato; se i comandi **Jog +** e **Jog -** sono inviati simultaneamente, il dispositivo non si muove o, se già in movimento, si arresta.

### Emergency

bit 7

Questo bit deve essere normalmente alto ("=1") altrimenti il dispositivo bloccherà istantaneamente ogni proprio movimento. Per un arresto normale (non immediato) nel rispetto della decelerazione programmata usare il bit 2 **Stop**. All'avvio il bit è forzato basso ("=0") per ragioni di sicurezza. Commutarlo al valore alto ("=1") per ripristinare il normale funzionamento.

**Byte 1****Watch dog enable**

bit 8

Impostando il bit **Watch dog enable** a "1" si abilita la funzione Watch dog; impostando il bit **Watch dog enable** a "0" si disabilita la funzione Watch dog. Quando la funzione Watch dog è abilitata, se il dispositivo non riceve un messaggio dal Server entro 1 secondo, il sistema forza una condizione di allarme (attivazione del bit di allarme **Watch dog** al ripristino della comunicazione con la rete Modbus). Il Watch dog è un sistema di sicurezza che, grazie a un timeout, permette di rilevare condizioni di loop o di deadlock. Per esempio, nel caso in cui si interrompesse la comunicazione seriale mentre è attivo un comando -a esempio un comando di jog- il Watch dog interverrebbe comandando lo stop in sicurezza del dispositivo e l'attivazione dell'allarme.

**Save parameters**

bit 9

I dati sono salvati nella memoria non volatile in corrispondenza di ogni fronte di salita del bit; in altre parole, il salvataggio è eseguito ogniqualvolta il bit commuta dal livello logico basso ("0") al livello logico alto ("1").



Salvare sempre i nuovi valori dopo l'impostazione al fine di memorizzarli permanentemente nella memoria non volatile. Nel caso di spegnimento del dispositivo i dati non salvati andranno persi.

**Load default parameters**

bit 10

I parametri di default (parametri impostati in azienda dai tecnici Lika durante la messa a punto del dispositivo che ne permettono un funzionamento standard e sicuro) sono ripristinati in corrispondenza di ogni fronte di salita del bit; in altre parole, l'operazione di caricamento dei parametri di default è eseguita ogniqualvolta il bit commuta dal livello logico basso ("0") al livello logico alto ("1"). La lista completa dei dati macchina e dei relativi parametri di default preimpostati da Lika Electronic sono disponibili a pagina 186.



Salvare sempre i nuovi valori dopo l'impostazione al fine di memorizzarli permanentemente nella memoria non volatile. Nel caso di spegnimento del dispositivo i dati non salvati andranno persi.

**ATTENZIONE**

La taratura dell'unità è stata realizzata testandone il funzionamento a pieno carico; i valori di default impostati si

riferiscono pertanto a un dispositivo operante in questa condizione. Essi sono altresì previsti per garantire un funzionamento standard e sicuro del dispositivo, che potrà talora non risultare ottimale né prestazionale. Si badi quindi che nella specifica applicazione può essere consigliabile, se non necessaria, la modifica dei parametri di fabbrica e in particolare dei valori di velocità, accelerazione, decelerazione e guadagno.

### Setting the preset

bit 11 Imposta la quota attuale al valore impostato nei registri **Preset [0x12-0x13]**. L'operazione è realizzata in corrispondenza di ogni fronte di salita del bit, ossia ogniqualvolta il bit commuta dal livello logico basso ("0") al livello logico alto ("1"). Consigliamo di attivare il preset quando l'attuatore è in stop. Per maggiori informazioni riferirsi alla pagina 157.

### Axis torque

bit 12 Questa funzione è disponibile solo nella versione RD1A (modello senza freno integrato); nella versione RD12A (modello con freno) il bit 12 non è usato. Quando l'asse ha raggiunto la posizione comandata, mantiene la coppia. Se impostato "=1", con asse in posizione, il PWM è mantenuto attivo (asse in coppia a fine posizionamento). Se impostato "=0", con asse in posizione, il PWM è disattivato (la coppia è rilasciata).

### OUT 1

bit 13 Permette di attivare / disattivare il funzionamento dell'uscita digitale 1. Il significato dell'uscita disponibile è descritto nella sezione "6.3 Ingressi e uscita digitali" a pagina 50.

**OUT 1 = 0** uscita 1 bassa (non attiva)

**OUT 1 = 1** uscita 1 alta (attiva)

### Brake disabled

bit 14 Questa funzione è disponibile solo nella versione RD12A (modello con freno integrato); nella versione RD1A (modello senza freno) il bit 14 non è usato. Il modello RD12A è equipaggiato con un freno che si attiva all'arresto del motore inibendo così la possibilità di qualsiasi movimento dell'asse. Se si imposta questo bit "=1" il freno è disattivato e non operativo; se si imposta questo bit "=0" il freno è attivo e il suo funzionamento è gestito in modo automatico dal sistema.



Si badi che lo sblocco del freno è possibile solo in assenza di allarmi.

bit 15

Non usato.



#### ATTENZIONE

Per motivi di sicurezza il parametro holding register **Control Word [0x2A]** non viene salvato in memoria. Perciò deve essere reimpostato dopo ogni accensione.

#### Target position [0x2B-0x2C]

[Registri 44-45, Integer32, rw]

Imposta la posizione da raggiungere (target), altrimenti detta posizione comandata. Il dispositivo si muove allo scopo di raggiungere la posizione target impostata in questo parametro quando si invia il comando **Start**, i bit **Stop** ed **Emergency** sono "=1" e il dispositivo non è in stato di allarme.

Non appena l'asse di trova all'interno dei limiti di tolleranza impostati nel registro **Position window [0x01]**, il bit 8 **Target position reached** in **Status word [0x01]** va alto ("=1"). Quando la posizione è all'interno dei limiti di tolleranza impostati nel registro **Position window [0x01]**, dopo il ritardo impostato nel parametro **Position window time [0x02]**, il bit 0 **Axis in position** in **Status word [0x01]** va alto ("=1").

Per maggiori informazioni riferirsi alla sezione "Posizione: controllo di posizione e velocità" a pagina 49.

Default = 0 (min. = 0, max. = compreso entro il massimo limite positivo / massimo limite negativo)



#### NOTA

##### Funzione di position override

E' possibile modificare il valore della posizione di target anche al volo, mentre il dispositivo sta ancora raggiungendo una posizione di target comandata in precedenza e senza inviare un nuovo comando di **Start**. Per fare questo, semplicemente impostare un nuovo valore di target nei registri **Target position [0x2B-0x2C]**. Si veda anche a pagina 49.



#### NOTA

Le funzioni **Jog +**, **Jog -** e **Start** non possono essere abilitate simultaneamente. Per esempio: se un comando **Jog +** viene inviato allo Slave mentre questo sta eseguendo il movimento verso la posizione di target, il comando jog viene

ignorato; se i comandi **Jog +** e **Jog -** sono inviati simultaneamente, il dispositivo non si muove o, se già in movimento, si arresta.

Se la funzione di Watch dog è abilitata (**Watch dog enable** in **Control Word [0x2A]** impostato a "=1"), nel caso in cui il dispositivo fosse disconnesso dalla rete Modbus mentre sta eseguendo un movimento (per esempio a causa dell'interruzione di un cavo o di un cablaggio errato), il dispositivo arresta immediatamente il proprio movimento e attiva il bit di allarme **Watch dog** (l'allarme compare non appena viene ripristinata la comunicazione nella rete Modbus).



#### ATTENZIONE

Per motivi di sicurezza il parametro holding register **Target position [0x2B-0x2C]** non viene salvato in memoria. Perciò deve essere reimpostato dopo ogni accensione.



#### NOTA

Salvare sempre i nuovi valori dopo l'impostazione al fine di memorizzarli permanentemente nella memoria non volatile. Usare la funzione **Save parameters**.

Nel caso di spegnimento del dispositivo i dati non salvati andranno persi.

### 8.14.2 Parametri Input Register

I parametri **Input Register** sono accessibili in sola lettura; per leggere il valore di un parametro utilizzare il codice funzione **04 Read Input Register** (lettura multipla degli input register); per ogni informazione sui codici funzione implementati riferirsi alla sezione "8.13.1 Codici funzione implementati" a pagina 140.

#### Alarms register [0x00]

[Registro 1, Unsigned16, ro]

Questa variabile visualizza gli allarmi correntemente attivi nel dispositivo.

Struttura del byte allarmi:

byte	MSB			LSB		
bit	15	...	8	7	...	0
	msb		lsb	msb		lsb

Codici registro allarmi previsti:

#### Byte 0

##### Machine data not valid

bit 0 Uno o più parametri non sono validi, impostare valori corretti per ristabilire la normale condizione di lavoro. Controllare la lista dei parametri errati nei registri **Wrong parameters list [0x08-0x09]**.

##### Flash memory error

bit 1 Errore interno non ripristinabile.

##### Counting error

bit 2 Per motivi di sicurezza, la posizione assoluta e quella incrementale dell'encoder integrato sono lette e salvate in due registri separati. Nel caso in cui si riscontri una differenza tra i valori nei due registri viene segnalato l'allarme.

##### Following error

bit 3 La differenza tra la posizione reale e quella teorica è maggiore di quella impostata nel parametro **Max following error [0x03-0x04]**; si consiglia di ridurre la velocità di lavoro.

##### Axis not synchronized

bit 4 Errore interno non ripristinabile.

**Target not valid**

bit 5 E' stata confermata una posizione comandata che supera i finecorsa.

**Emergency**

bit 6 Il bit 7 **Emergency** in **Control Word [0x2A]** è stato forzato a 0; oppure sono attivi allarmi nel dispositivo.

**Overcurrent**

bit 7 Sovracorrente motore.

**Byte 1****Electronics Overtemperature**

bit 8 La temperatura dei MOSFET rilevata da una sonda interna supera il valore massimo ammesso (si veda **Temperature value [0x07]** a pagina 171). Attendere alcuni minuti per permettere all'attuatore di raffreddarsi. Assicurarsi che la temperatura operativa sia compresa nel range permesso.

**Motor Overtemperature**

bit 9 La temperatura del motore rilevata da una sonda interna supera il valore massimo ammesso (si veda **Temperature value [0x07]** a pagina 171). Attendere alcuni minuti per permettere all'attuatore di raffreddarsi. Assicurarsi che la temperatura operativa sia compresa nel range permesso.

**Undervoltage**

bit 10 Il valore della tensione di alimentazione è inferiore al minimo consentito. Assicurarsi che la tensione di alimentazione sia compresa nel range ammesso.

**Watch dog**

bit 11 Quando la funzione Watch dog è abilitata (il bit 8 **Watch dog enable** in **Control Word [0x2A]** è "=1"), se il dispositivo non riceve un messaggio dal Server entro 1 secondo, il sistema forza una condizione di allarme (attivazione del bit di allarme **Watch dog**). L'allarme compare non appena viene ripristinata la comunicazione nella rete Modbus. Il Watch dog è un sistema di sicurezza che, grazie a un timeout, permette di rilevare condizioni di loop o di deadlock. Per esempio, nel caso in cui si interrompesse la comunicazione seriale mentre è attivo un comando -a esempio un comando di jog- il Watch dog interverrebbe comandando lo stop in sicurezza del dispositivo e l'attivazione dell'allarme.

bit 12 e 13 Non usati.

### Hall sequence

bit 14 Si è verificato un errore nella sequenza di commutazione dei sensori di Hall.

### Overvoltage

bit 15 Il valore della tensione di alimentazione è superiore al massimo consentito. Assicurarsi che la tensione di alimentazione sia compresa nel range ammesso.  
Se l'allarme si genera durante l'operazione di frenatura, considerare la tensione controlettromotrice (back EMF). Per prevenire il verificarsi di questa evenienza, diminuire la rampa della velocità o valutare attentamente le caratteristiche dell'alimentatore 24V (modulo condensatore).

Per ripristinare una condizione di errore utilizzare il comando **Alarm reset**, bit 3 della **Control Word [0x2A]**. In una normale condizione di lavoro il bit **Alarm reset** ha valore "0". Impostando il valore a "1" si ristabilisce il normale stato operativo del dispositivo. Il normale stato operativo è ripristinato nella commutazione da "0" a "1". Questo comando libera lo Slave dalla condizione di allarme solo se sono state rimosse le cause che ne hanno provocato l'errore.



Si badi che, se l'allarme è relativo a dati macchina non validi (si vedano **Machine data not valid** e **Wrong parameters list [0x08-0x09]**), si può ripristinare lo stato normale di funzionamento solo impostando dati macchina validi. Gli allarmi **Flash memory error** e **Axis not synchronized** non sono ripristinabili.

### Status word [0x01]

[Registro 2, Unsigned16, ro]

Questo registro fornisce informazioni sullo stato attuale del dispositivo.

Struttura byte del registro **Status word [0x01]**:

byte	MSB			LSB		
bit	15	...	8	7	...	0
	msb		lsb	msb		lsb

### Byte 0

#### Axis in position

bit 0 Il valore è "=1" quando il dispositivo raggiunge e mantiene la posizione comandata (**Target position**)

[0x2B-0x2C]) per il tempo impostato nel registro **Position window time [0x02]**. E' mantenuto attivo fintanto che l'errore di posizione è inferiore a **Position window [0x01]**. Per ulteriori informazioni riferirsi alla sezione "Posizione: controllo di posizione e velocità" a pagina 49.

bit 1

Non usato.

#### Drive enabled

bit 2

Visualizza la condizione di abilitazione del motore. Questo bit è "=1" quando il motore è abilitato, ossia: il PWM è attivo e l'asse è in controllo ad anello chiuso (per esempio durante un posizionamento o un jog). E' "=0" quando il motore è disabilitato, vale a dire quando l'azionamento che controlla il motore viene spento al termine di un posizionamento o di un jog o a seguito di un allarme.

#### SW limit switch +

bit 3

Il valore è "=1" quando si verifica che il dispositivo raggiunge il limite positivo massimo (finecorsa positivo). Per maggiori informazioni si veda al parametro **Positive delta [0x09-0x0A]**.

#### SW limit switch -

bit 4

Il valore è "=1" quando si verifica che il dispositivo raggiunge il limite negativo massimo (finecorsa negativo). Per maggiori informazioni si veda al parametro **Negative delta [0x0B-0x0C]**.

#### Alarm

bit 5

Il valore è "=1" quando si attiva un allarme, per dettagli si veda la variabile **Alarms register [0x00]**.

#### Axis running

bit 6

Il valore è "=0" quando il dispositivo non si sta muovendo.

Il valore è "=1" mentre il dispositivo è in movimento.

#### Executing a command

bit 7

Il valore è "=0" quando il controller non sta eseguendo nessun comando.

Il valore è "=1" mentre il controller sta eseguendo un comando.

**Byte 1****Target position reached**

bit 8

Il valore è "=1" quando il dispositivo raggiunge la posizione di target impostata nel parametro **Target position [0x2B-0x2C]** (si trova all'interno della finestra di tolleranza impostata in **Position window [0x01]**). Il bit è mantenuto attivo fino a quando non si comanda una nuova **Target position [0x2B-0x2C]** o si invia un comando **Alarm reset**. Per maggiori informazioni riferirsi anche alla sezione "Posizione: controllo di posizione e velocità" a pagina 49.

**Button 1 Jog +**

bit 9

L'unità di posizionamento RD1xA è equipaggiata con tre pulsanti alloggiati all'interno della custodia e accessibili previa rimozione di un tappo a vite. Fintanto che il pulsante 1 JOG + è mantenuto premuto, il bit 9 è forzato alto "=1"; quando il pulsante 1 non è premuto, il bit 9 è basso "=0". Per ulteriori informazioni si veda la sezione "4.4 Tappo a vite per accesso interno (Figura 4 e Figura 7)" a pagina 41 e la sezione "4.5.1 Pulsanti JOG + e JOG - (Figura 8)" a pagina 43.

**Button 2 Jog -**

bit 10

L'unità di posizionamento RD1xA è equipaggiata con tre pulsanti alloggiati all'interno della custodia e accessibili previa rimozione di un tappo a vite. Fintanto che il pulsante 2 JOG - è mantenuto premuto, il bit 10 è forzato alto "=1"; quando il pulsante 2 non è premuto, il bit 10 è basso "=0". Per ulteriori informazioni si veda la sezione "4.4 Tappo a vite per accesso interno (Figura 4 e Figura 7)" a pagina 41 e la sezione "4.5.1 Pulsanti JOG + e JOG - (Figura 8)" a pagina 43.

**Button 3 Preset**

bit 11

L'unità di posizionamento RD1xA è equipaggiata con tre pulsanti alloggiati all'interno della custodia e accessibili previa rimozione di un tappo a vite. Non appena si preme il pulsante 3 PRESET, il bit 11 è forzato alto "=1"; quando il pulsante 3 non è premuto, il bit 11 è basso "=0". Per ulteriori informazioni si veda la sezione "4.4 Tappo a vite per accesso interno (Figura 4 e Figura 7)" a pagina 41 e la

sezione "4.5.2 Pulsante PRESET (Figura 8)" a pagina 43.

### PWM saturation

bit 12

La corrente erogata dall'elettronica di potenza per il controllo delle fasi del motore ha raggiunto il livello massimo (saturazione) e non può essere ulteriormente aumentata. Il funzionamento del motore è gravato da un'eccessiva dinamica o qualcosa sta impedendo il movimento.

### IN 1

bit 13

Visualizza lo stato dell'ingresso digitale 1. Il significato degli ingressi disponibili è descritto nella sezione "6.3 Ingressi e uscita digitali" a pagina 50.

**IN 1 = 0**      ingresso 1 basso (non attivo)

**IN 1 = 1**      ingresso 1 alto (attivo)

### IN 2

bit 14

Visualizza lo stato dell'ingresso digitale 2. Il significato degli ingressi disponibili è descritto nella sezione "6.3 Ingressi e uscita digitali" a pagina 50.

**IN 2 = 0**      ingresso 2 basso (non attivo)

**IN 2 = 1**      ingresso 2 alto (attivo)

### IN 3

bit 15

Visualizza lo stato dell'ingresso digitale 3. Il significato degli ingressi disponibili è descritto nella sezione "6.3 Ingressi e uscita digitali" a pagina 50.

**IN 3 = 0**      ingresso 3 basso (non attivo)

**IN 3 = 1**      ingresso 3 alto (attivo)

### Current position [0x02-0x03]

[Registri 3-4, Integer32, ro]

Posizione attuale del dispositivo espressa in impulsi.

### Current velocity [0x04]

[Registro 5, Integer16, ro]

Velocità del dispositivo espressa in giri al minuto [rpm], aggiornata ogni secondo. Questa è la velocità del motore, non la velocità dell'albero in uscita dopo il riduttore. La velocità in uscita sarà come segue:

Velocità del motore      = 2000 rpm

Velocità in uscita:      T12 = 166 rpm

                                 T24 = 83 rpm

T48 = 41 rpm

T92 = 21 rpm

### Position following error [0x05-0x06]

[Registri 6-7, Integer32, ro]

Questa variabile contiene la differenza tra la posizione richiesta (target) e la posizione attuale istante per istante. Se questo valore è superiore a quello impostato nel parametro **Max following error [0x03-0x04]**, il dispositivo genera l'allarme **Following error** e arresta il proprio movimento. Il valore è espresso in impulsi.

### Temperature value [0x07]

[Registro 8, Integer16, ro]

Questa variabile visualizza la temperatura del motore e la temperatura dell'elettronica rilevate dalle sonde interne. Il valore è espresso in gradi Celsius (°C). La temperatura minima rilevabile è -20°C.

Il significato dei 16 bit nel registro è il seguente:

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
MSB								LSB							
Numero Major								Numero Minor							
Temperatura del motore								Temperatura dell'elettronica							

Il valore 18 1A hex in notazione esadecimale corrisponde alla rappresentazione binaria 0001 1000 0001 1010 e deve essere letto come segue: temperatura del motore = 24°C; temperatura dell'elettronica = 26°C.

### Wrong parameters list [0x08-0x09]

[Registri 9-10, Unsigned32, ro]

L'operatore ha impostato valori non validi e il dispositivo ha generato l'allarme **Machine data not valid**. Questa variabile visualizza quali parametri contengono valori errati, secondo la lista riportata nella seguente tabella.

Si badi che si può ripristinare il normale stato di lavoro solo impostando dati macchina validi.

Bit	Parametro
0	Non usato
1	<b>Distance per revolution [0x00]</b>
2	<b>Acceleration [0x07]</b>

3	Deceleration [0x08]
4	Positive delta [0x09-0x0A]
5	Negative delta [0x0B-0x0C]
6	Jog speed [0x0D]
7	Work speed [0x0E]
8	Code sequence [0x0F]
9	Preset [0x12-0x13]
10	Jog step length [0x14]
11	Kp position loop [0x05]
12	Ki position loop [0x06]
13	Position window time [0x02]
14	Max following error [0x03-0x04]
15	Non usato

#### Motor voltage [0x0A]

[Registro 11, Unsigned16, ro]

Visualizza la tensione motore espressa in millivolt (mV).

#### Current value [0x0B]

[Registro 12, Unsigned16, ro]

Questa variabile visualizza il valore della corrente assorbita dal motore (corrente nominale). Il valore è espresso in milliampere (mA).

#### Hall [0x0C]

[Registro 13, Unsigned16, ro]

Questa funzione è riservata all'uso dei tecnici di Lika Electronic.

#### Duty cycle [0x0D]

[Registro 14, Unsigned16, ro]

Questa funzione è riservata all'uso dei tecnici di Lika Electronic.

**DIP switch baud rate [0x0E]**

[Registro 15, Unsigned16, ro]

Visualizza la velocità di trasmissione dati (baud rate) della porta seriale di cui è equipaggiata l'unità RD1xA; la velocità di trasmissione dati deve essere impostata mediante il provvisto DIP switch. In questo modello il DIP switch del baud rate ha valore fisso e non è accessibile all'utilizzatore.

**DIP switch node ID [0x0F]**

[Register 16, Unsigned16, ro]

Visualizza l'indirizzo del nodo impostato nell'unità RD1xA; l'indirizzo del nodo deve essere impostato mediante il provvisto DIP switch. In questo modello il DIP switch del node ID ha valore fisso e non è accessibile all'utilizzatore.

**SW Version [0x10]**

[Registro 17, Unsigned16, ro]

Visualizza la versione software dell'unità DRIVECOD.

Il significato dei 16 bit nel registro è il seguente:

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
MSB								LSB							
Numero Major								Numero Minor							

Il valore 01 02 hex in notazione esadecimale corrisponde alla rappresentazione binaria 00000001 00000010 e deve essere interpretato come: versione 1.2.

**HW Version [0x11]**

[Registro 18, Unsigned16, ro]

Visualizza la versione hardware e il modello dell'unità DRIVECOD.

Il significato dei 16 bit nel registro è il seguente:

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Modello DRIVECOD								Freno	Riduzione	Versione hardware					

dove:

00 ... 03	= versione hardware
04 ... 06	= rapporto di riduzione (1 = T12; 2 = T24; 3 = T48; 4 = T92)
07	= freno: 0 = modello RD1A senza freno; 1 = modello RD12A con freno
08 ... 15	= modello RD1xA equipaggiato con la seguente interfaccia: 0x30 = MODBUS RTU; 0x31 = Profibus; 0x32 = CANopen; 0x33 = POWERLINK; 0x34 = EtherCAT; 0x35 = MODBUS TCP; 0x36 =

EtherNet/IP; 0x37 = Profinet
------------------------------

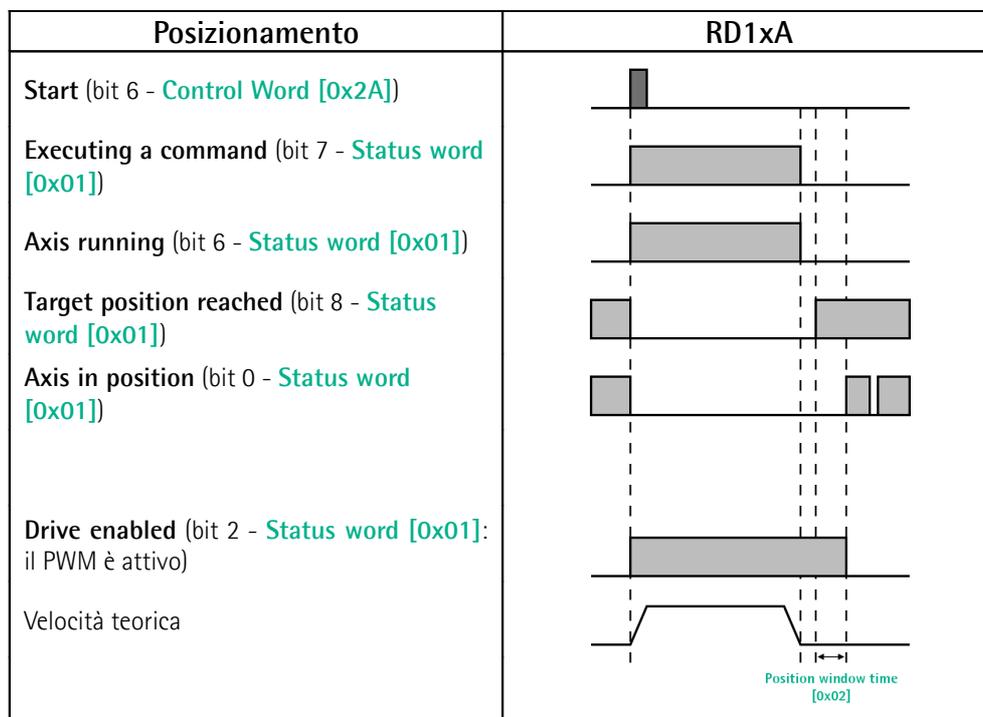
Il valore 30 B1 hex in notazione esadecimale corrisponde alla rappresentazione binaria 0011 0000 1011 0001 e deve essere interpretato come segue: modello RD1xA con interfaccia MODBUS RTU, rapporto di riduzione T48, equipaggiato con freno, versione hardware 1.

**NOTA**

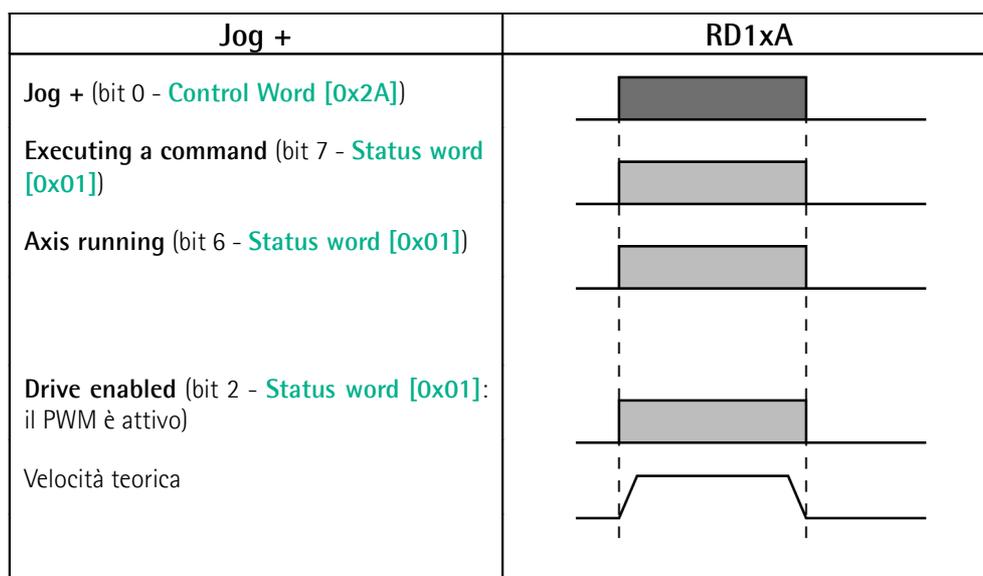
Salvare sempre i nuovi valori dopo l'impostazione per memorizzarli permanentemente nella memoria non volatile. Usare la funzione **Save parameters**. In caso di spegnimento dell'unità i dati non salvati andranno persi.



ESEMPIO 1



ESEMPIO 2



### 8.15 Codici eccezione

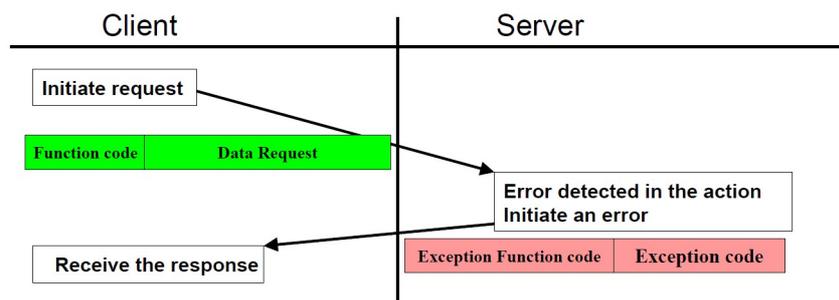
Quando un dispositivo Client invia una richiesta a un dispositivo Server, si attende poi una risposta normale. In realtà si possono verificare uno dei quattro possibili eventi descritti qui di seguito.

- Se il dispositivo Server riceve la richiesta senza che siano presenti errori di comunicazione e di conseguenza può gestire normalmente l'interrogazione, invia di ritorno una risposta normale.
- Se il dispositivo Server non riceve la richiesta a causa di un errore di comunicazione, non sarà possibile ritornare alcuna risposta. Eventualmente il programma client potrà gestire un timeout per la richiesta.
- Se il Server riceve la richiesta, ma rileva un errore di comunicazione (bit di parità, CRC, ...), non sarà possibile ritornare alcuna risposta. Eventualmente il programma client potrà gestire un timeout per la richiesta.
- Se il Server riceve la richiesta senza che siano presenti errori di comunicazione, ma non è in grado di processarla (per esempio, nel caso in cui si trattasse di una richiesta di lettura di un'uscita o di un registro inesistenti), il Server invierà una exception response informando il Client sulla natura dell'errore.

Il messaggio di exception response consta di due campi che lo differenziano da un messaggio normale di risposta:

**CAMPO CODICE FUNZIONE:** in una risposta normale, il Server ripete il codice funzione della richiesta originale riportandolo nel campo codice funzione della risposta. Tutti i codici funzione hanno un most significant bit (msb) di 0 (i loro valori sono tutti inferiori a 80 esadecimale). In una exception response, il Server imposta il most significant bit del codice funzione a 1. Questo fa sì che il valore del codice funzione di una exception response sia maggiore del valore che esso avrebbe assunto in una risposta normale esattamente di 80 esadecimale. Con il most significant bit del function code impostato, il programma di applicazione client può riconoscere la exception response e valutare il campo dati al fine di inviare un exception code.

**CAMPO DATI:** in una risposta normale, il Server può ritornare dati o statistiche nel campo dati (qualsiasi informazione che fosse richiesta nella interrogazione). In un exception code, il Server scrive nel campo dati il codice eccezione. Esso specifica la condizione Server che ha procurato l'eccezione.



**NOTA**

Si badi che quello che segue è un elenco dei codici eccezione (exception code) indicati da MODBUS, ma non necessariamente supportati dal costruttore.

Codici eccezione MODBUS (exception code)		
Codice	Nome	Significato
01	ILLEGAL FUNCTION	Il codice funzione ricevuto nell'interrogazione si configura in un'azione inammissibile per il server. Il motivo potrebbe risiedere nel fatto che il codice funzione è ammesso solo per nuovi dispositivi e non è implementato nell'unità selezionata. Potrebbe anche indicare che il server si trova in uno stato non compatibile per la processazione di una richiesta di questo tipo, per esempio perché non è configurato e riceve una richiesta di restituzione dei valori di registro.
02	ILLEGAL DATA ADDRESS	L'indirizzo dati ricevuto nell'interrogazione non è un indirizzo ammissibile per il server. Più specificamente, la combinazione di reference number e registri richiesti non è valida. Per un controller con 100 registri, la PDU assegna al primo registro il valore 0 e all'ultimo il valore 99. Se viene fatta una richiesta dove l'indirizzo del registro iniziale è 96 e la quantità di registri richiesti è 4, la richiesta viene assolta con successo (indirizzamento corretto) operando sui registri 96, 97, 98, 99. Se invece viene inviata una richiesta dove l'indirizzo del registro iniziale è 96 e la quantità di registri richiesti è 5, in questo caso la richiesta non potrà essere portata a termine e causerà il codice eccezione 0x02 "Illegal Data Address" dato che viene richiesto di operare sui registri 96, 97, 98, 99 e 100, ma non esiste nessun registro con indirizzo 100.
03	ILLEGAL DATA VALUE	Un valore contenuto nel campo dati

		dell'interrogazione non è ammissibile per il server. Indica un errore nella struttura del remainder di una richiesta complessa, come per esempio che la lunghezza indicata non è corretta. Non significa nello specifico che un data item inviato per la memorizzazione in un registro ha un valore fuori dal range atteso dal programma applicativo, dato che il protocollo MODBUS non tiene conto del significato di un qualunque valore di un qualunque registro.
<b>04</b>	SERVER DEVICE FAILURE	Si è verificato un errore non recuperabile durante il tentativo da parte del server di eseguire una azione richiesta.
<b>05</b>	ACKNOWLEDGE	Utilizzo specifico con comandi di programmazione. Il server ha accettato la richiesta e la sta eseguendo, ma per fare questo sarà necessario diverso tempo. Questa risposta viene restituita per evitare che si possa registrare nel client un errore di timeout. Il client può poi inviare un messaggio Poll Program Complete per determinare se la processazione è completata.
<b>06</b>	SERVER DEVICE BUSY	Utilizzo specifico con comandi di programmazione. Il server è impegnato nella processazione di un comando di programmazione che richiede diverso tempo. Pertanto il client è invitato a ritrasmettere il messaggio più tardi quando il server sarà disponibile.
<b>08</b>	MEMORY PARITY ERROR	Utilizzo specifico con i codici funzione 20 e 21 e il reference type 6, indica che l'area di file estesa non ha superato un consistency check. Il server ha tentato di leggere il file in memoria, ma ha rilevato un errore di parità nella memoria. Il client può ritentare la richiesta, ma potrebbe essere necessario un intervento sul dispositivo server.
<b>0A</b>	GATEWAY PATH UNAVAILABLE	Utilizzo specifico con gateway, indica che il gateway non è stato in grado di assegnare un percorso di comunicazione interna dalla porta d'ingresso alla porta d'uscita per la processazione della richiesta. Solitamente significa che il gateway è mal configurato o sovraccarico.
<b>0B</b>	GATEWAY TARGET DEVICE FAILED TO RESPOND	Utilizzo specifico con gateway, indica che non si è avuta risposta dal dispositivo indicato. Solitamente significa che il dispositivo non è presente nella rete.

Per ogni informazione sui codici eccezione (exception code) disponibili e sul loro significato riferirsi al capitolo "MODBUS Exception Responses" alla pagina 48 del documento "MODBUS Application Protocol Specification V1.1b3".

## 8.16 Esempi di programmazione

Di seguito sono riportati alcuni esempi di lettura e impostazione dei parametri. Tutti i valori sono espressi in notazione esadecimale.

### 8.16.1 Utilizzo del codice funzione 03 Read Holding Register



#### ESEMPIO 1

Richiesta di lettura dei parametri **Acceleration [0x07]** (registro 8) e **Deceleration [0x08]** (registro 9) allo Slave con indirizzo 1.

#### Request PDU

[01][03][00][07][00][02][75][CA]

dove

[01] = indirizzo dello Slave

[03] = codice funzione **03 Read Holding Register**

[00][07] = indirizzo iniziale (parametro **Acceleration [0x07]**, registro 8)

[00][02] = numero di registri richiesti

[75][CA] = CRC

#### Response PDU

[01][03][04][00][0A][00][0A][5A][36]

dove:

[01] = indirizzo dello Slave

[03] = codice funzione **03 Read Holding Register**

[04] = numero di byte (2 byte per ciascun registro)

[00][0A] = valore del registro 8 **Acceleration [0x07]**, 00 0A hex = 10 dec

[00][0A] = valore del registro 9 **Deceleration [0x08]**, 00 0A hex = 10 dec

[5A][36] = CRC

Il parametro **Acceleration [0x07]** (registro 8) ha valore 00 0A hex, cioè 10 in notazione decimale; il parametro **Deceleration [0x08]** (registro 9) ha valore 00 0A hex, cioè 10 in notazione decimale.

### 8.16.2 Utilizzo del codice funzione 04 Read Input Register



#### ESEMPIO 1

Richiesta di lettura del parametro **Current position [0x02-0x03]** (registri 3 e 4) allo Slave con indirizzo 1.

#### Request PDU

[01][04][00][02][00][02][D0][0B]

dove:

[01] = indirizzo dello Slave

[04] = codice funzione **04 Read Input Register**

[00][02] = indirizzo iniziale (parametro **Current position [0x02-0x03]**, registro 3)

[00][02] = numero di registri richiesti

[D0][0B] = CRC

#### Response PDU

[01][04][04][00][00][2F][F0][E7][F0]

dove:

[01] = indirizzo dello Slave

[04] = codice funzione **04 Read Input Register**

[04] = numero di byte (2 byte per ciascun registro)

[00][00] = valore del registro 3 **Current position [0x02-0x03]**, 00 00 hex = 0 dec

[2F][F0] = valore del registro 4 **Current position [0x02-0x03]**, 2F F0 hex = 12272 dec

[E7][F0] = CRC

Il parametro **Current position [0x02-0x03]** (registri 3 e 4) ha valore 00 00 2F F0 hex, cioè 12272 in notazione decimale.



#### ESEMPIO 2

Richiesta di lettura della variabile **Alarms register [0x00]** (registro 1) allo Slave con indirizzo 1.

#### Request PDU

[01][04][00][00][00][01][31][CA]

dove:

[01] = indirizzo dello Slave

[04] = codice funzione **04 Read Input Register**

[00][00] = indirizzo iniziale (variabile **Alarms register [0x00]**, registro 1)

[00][01] = numero di registri richiesti  
[31][CA] = CRC

### Response PDU

[01][04][02][00][81][79][50]

dove:

[01] = indirizzo dello Slave

[04] = codice funzione **04 Read Input Register**

[02] = numero di byte (2 byte per ciascun registro)

[00][81] = valore del registro 1 **Alarms register [0x00]**, 00 81 hex = 0000 0000  
1000 0001 bin

[79][50] = CRC

Questo significa che nella variabile **Alarms register [0x00]** (registro 1) i bit 0 e 7 sono attivi (livello logico alto = 1), ossia (si veda a pagina 165): **Machine data not valid** e **Emergency**.

### 8.16.3 Utilizzo del codice funzione 06 Write Single Register



#### ESEMPIO 1

Richiesta di scrittura del valore 00 96 hex (= 150 dec) nel parametro **Acceleration [0x07]** (registro 8) dello Slave con indirizzo 1.

#### Request PDU

[01][06][00][07][00][96][B8][65]

dove:

[01] = indirizzo dello Slave

[06] = codice funzione **06 Write Single Register**

[00][07] = indirizzo del registro (parametro **Acceleration [0x07]**, registro 8)

[00][96] = valore da impostare nel registro

[B8][65] = CRC

#### Response PDU

[01][06][00][07][00][96][B8][65]

dove:

[01] = indirizzo dello Slave

[06] = codice funzione **06 Write Single Register**

[00][07] = indirizzo del registro (parametro **Acceleration [0x07]**, registro 8)

[00][96] = valore impostato nel registro

[B8][65] = CRC

Il valore 00 96 hex, cioè 150 in notazione decimale, è impostato nel parametro **Acceleration [0x07]** (registro 8).



#### ESEMPIO 2

Richiesta di scrittura del valore 00 84 hex nella variabile **Control Word [0x2A]** (registro 43) dello Slave con indirizzo 1.

#### Request PDU

[01][06][00][2A][00][84][A8][61]

dove:

[01] = indirizzo dello Slave

[06] = codice funzione **06 Write Single Register**

[00][2A] = indirizzo del registro (variabile **Control Word [0x2A]**, registro 43)

[00][84] = valore da impostare nel registro

[A8][61] = CRC

### Response PDU

[01][06][00][2A][00][84][A8][61]

dove:

[01] = indirizzo dello Slave

[06] = codice funzione **06 Write Single Register**

[00][2A] = indirizzo del registro (variabile **Control Word [0x2A]**, registro 43)

[00][84] = valore impostato nel registro

[A8][61] = CRC

Il valore 00 84 hex = 0000 0000 1000 0100 in notazione binaria è impostato nella variabile **Control Word [0x2A]** (registro 43). In altre parole, i bit **Stop** e **Emergency** sono forzati al livello logico alto (bit 2 = 1; bit 7 = 1): l'unità è pronta per eseguire il comando di movimento richiesto.



### ESEMPIO 3

Richiesta di scrittura del valore 0A 80 hex nella variabile **Control Word [0x2A]** (registro 43) dello Slave con indirizzo 1.

### Request PDU

[01][06][00][2A][0A][80][AF][02]

dove:

[01] = indirizzo dello Slave

[06] = codice funzione **06 Write Single Register**

[00][2A] = indirizzo del registro (variabile **Control Word [0x2A]**, registro 43)

[0A][80] = valore da impostare nel registro

[AF][02] = CRC

### Response PDU

[01][06][00][2A][0A][80][AF][02]

dove:

[01] = indirizzo dello Slave

[06] = codice funzione **06 Write Single Register**

[00][2A] = indirizzo del registro (variabile **Control Word [0x2A]**, registro 43)

[0A][80] = valore impostato nel registro

[AF][02] = CRC

Il valore 0A 80 hex = 0000 0010 1000 0000 in notazione binaria è impostato nella variabile **Control Word [0x2A]** (registro 43). In altre parole, il dispositivo è forzato in stop (bit 2 **Stop** = 0), ma non in condizione di emergenza (bit 7 **Emergency** = 1); è inoltre richiesto il salvataggio dei dati (bit 9 **Save parameters** = 1).

### 8.16.4 Utilizzo del codice funzione 16 Write Multiple Register



#### ESEMPIO 1

Richiesta di scrittura dei valori 150 e 100 nei parametri **Acceleration [0x07]** (registro 8) e **Deceleration [0x08]** (registro 9) dello Slave con indirizzo 1.

#### Request PDU

[01][10][00][07][00][02][04][00][96][00][64][53][8E]

dove:

[01] = indirizzo dello Slave

[10] = codice funzione **16 Write Multiple Register**

[00][07] = indirizzo iniziale (parametro **Acceleration [0x07]**, registro 8)

[00][02] = numero di registri richiesti

[04] = numero di byte (2 byte per ciascun registro)

[00][96] = valore da impostare nel registro 8 **Acceleration [0x07]**, 00 96 hex = 150 dec

[00][64] = valore da impostare nel registro 9 **Deceleration [0x08]**, 00 64 hex = 100 dec

[53][8E] = CRC

#### Response PDU

[01][10][00][07][00][02][F0][09]

dove:

[01] = indirizzo dello Slave

[10] = codice funzione **16 Write Multiple Register**

[00][07] = indirizzo iniziale (parametro **Acceleration [0x07]**, registro 8)

[00][02] = numero di registri scritti

[F0][09] = CRC

Il valore 00 96 hex, cioè 150 in notazione decimale, è impostato nel parametro **Acceleration [0x07]** (registro 8); il valore 00 64 hex, cioè 100 in notazione decimale, è impostato nel parametro **Deceleration [0x08]** (registro 9).

## 9 Lista parametri di default

### EtherCAT®

Lista parametri	Valore di default		
2200 Control Word	0		
2201-00 Target Position	0		
2204-00 Distance per revolution PPR	1024		
2205-00 Position tolerance P	1		
2206-00 Settling time ms	0		
2207-00 Max following error P	1024		
2208-00 Proportional gain	300		
2209-00 Integral gain	10		
220A-00 Acceleration rev/s <sup>2</sup>	10		
220B-00 Deceleration rev/s <sup>2</sup>	10		
220C-00 Max delta pos P	523263		
220D-00 Max delta neg P	523263		
220E-00 Jog speed rpm	2000		
220F-00 Work speed rpm	2000		
2210-00 Count direction	0		
2211-00 Preset P	0		
2212-00 Step jog P	1000		

### Modbus

Lista parametri	Valore di default		
Distance per revolution [0x00] PPR	1024		
Position window [0x01] P	1		
Position window time [0x02] ms	0		
Max following error [0x03-0x04] P	1024		
Kp position loop [0x05]	300		
Ki position loop [0x06]	10		
Acceleration [0x07] rev/s <sup>2</sup>	10		
Deceleration [0x08] rev/s <sup>2</sup>	10		
Positive delta [0x09-0x0A] P	523263		
Negative delta [0x0B-0x0C] P	523263		
Jog speed [0x0D] rpm	2000		
Work speed [0x0E] rpm	2000		
Code sequence [0x0F]	0		
Preset [0x12-0x13] P	0		
Jog step length [0x14] P	1000		
Control Word [0x2A]	0		
Target position [0x2B-0x2C]	0		

Pagina lasciata bianca intenzionalmente

Versione documento	Data rilascio	Descrizione	HW	SW	File XML	Interfaccia
1.0	15.11.2016	Prima stampa	2.0	1.0	V2	1.1
1.1	19.02.2019	Revisione generale, aggiornamento interfaccia MODBUS	2.0	2.0	V3	3.1.0.3
1.2	05.03.2021	Aggiornamento informazione sull'installazione, correzioni minori	2.0	2.0	V4	3.1.0.3



Smaltire separatamente

**lika**

**Lika Electronic**

Via S. Lorenzo, 25 • 36010 Carrè (VI) • Italy

Tel. +39 0445 806600

Fax +39 0445 806699



info@lika.biz • www.lika.biz