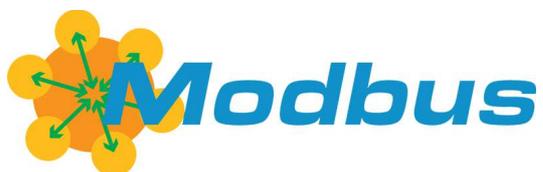


Manuale d'uso

RD1A

RD12A



Versione RS-485



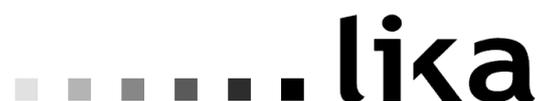
Smart encoders & actuators

Questa pubblicazione è edita da Lika Electronic s.r.l. 2015. 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 (includere 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.



Indice generale

| | |
|--|-----------|
| Manuale d'uso..... | 1 |
| Indice generale..... | 3 |
| Indice analitico..... | 6 |
| Convenzioni grafiche e iconografiche..... | 7 |
| Informazioni preliminari..... | 8 |
| 1 Norme di sicurezza..... | 9 |
| 1.1 Sicurezza..... | 9 |
| 1.2 Avvertenze elettriche..... | 9 |
| 1.3 Avvertenze meccaniche..... | 10 |
| 2 Identificazione..... | 11 |
| 3 Installazione meccanica..... | 12 |
| 4 Connessioni elettriche..... | 16 |
| 4.1 Collegamento messa a terra (Figura 1 e Figura 2)..... | 16 |
| 4.2 Connettori (Figura 4)..... | 18 |
| 4.3 Indicatori LED (Figura 4)..... | 20 |
| 4.4 Selettori e pulsanti (Figura 5)..... | 22 |
| 4.4.1 Indirizzo nodo: Node ID (Figura 5)..... | 23 |
| 4.4.2 Velocità di trasmissione dei dati: Baud rate e bit di parità (Figura 5)..... | 24 |
| 4.4.3 Resistenza di terminazione: RT (Figura 5)..... | 25 |
| 4.4.4 Pulsanti JOG + e JOG - (Figura 5)..... | 25 |
| 4.4.5 Pulsante PRESET (Figura 5)..... | 25 |
| 5 Quick reference..... | 27 |
| 5.1 Configurazione mediante software di Lika Electronic..... | 27 |
| 5.2 Pagina "Configurazione seriale"..... | 28 |
| 5.3 Pagina "Modalità operativa"..... | 31 |
| 5.4 Pagina "Dati macchina"..... | 36 |
| 5.5 Pagina "Monitor messaggi"..... | 37 |
| 5.6 Pagina "Test Lika"..... | 38 |
| 5.7 Pagina "Upgrade Firmware"..... | 39 |
| 5.7.1 In caso di errore..... | 42 |
| 5.8 Getting started..... | 43 |
| 6 Funzioni..... | 44 |
| 6.1 Principio di funzionamento..... | 44 |
| 6.2 Tipi di movimento: jog e posizione..... | 45 |
| Jog: controllo di velocità..... | 45 |
| Posizionamento: controllo di posizione e velocità..... | 45 |
| 6.3 Ingressi e uscite digitali..... | 46 |
| 6.4 Distanza_giro [0x00], Velocità Jog [0x0C], Velocità di lavoro [0x0D], Valore di preset [0x16-0x17], Delta spazio positivo [0x08-0x09] e Delta spazio negativo [0x0A-0x0B]..... | 47 |
| 7 Interfaccia Modbus®..... | 51 |
| 7.1 Principi guida del protocollo Modbus Master / Slave..... | 51 |
| 7.2 Frame Modbus..... | 52 |
| 7.3 Modalità di trasmissione..... | 53 |
| 7.3.1 Modalità di trasmissione RTU..... | 53 |
| 7.4 Codici funzione..... | 56 |
| 7.4.1 Codici funzione implementati..... | 56 |

| | |
|---|-----------|
| 03 Read Holding Registers..... | 56 |
| 04 Read Input Register..... | 58 |
| 06 Write Single Register..... | 60 |
| 16 Write Multiple Registers..... | 62 |
| 8 Parametri di programmazione..... | 66 |
| 8.1 Parametri disponibili..... | 66 |
| 8.1.1 Parametri Dati macchina..... | 66 |
| Distanza_giro [0x00]..... | 66 |
| Tolleranza di posizione [0x01]..... | 67 |
| Tempo asse in tolleranza [0x02]..... | 67 |
| Max errore di inseguimento [0x03]..... | 67 |
| Kp anello di posizione [0x04]..... | 68 |
| Ki anello di posizione [0x05]..... | 68 |
| Accelerazione [0x06]..... | 68 |
| Decelerazione [0x07]..... | 68 |
| Delta spazio positivo [0x08-0x09]..... | 68 |
| Delta spazio negativo [0x0A-0x0B]..... | 69 |
| Velocità Jog [0x0C]..... | 70 |
| Velocità di lavoro [0x0D]..... | 71 |
| Durata corrente di stacco [0x0E]..... | 71 |
| Direzione conteggio [0x0F]..... | 72 |
| Kp anello di corrente [0x10]..... | 72 |
| Ki anello di corrente [0x11]..... | 72 |
| Corrente massima [0x12]..... | 72 |
| Corrente di stacco [0x13]..... | 73 |
| Valore di offset [0x14-0x15]..... | 73 |
| Valore di preset [0x16-0x17]..... | 73 |
| Rapporto di riduzione [0x18]..... | 74 |
| Ampiezza passo jog [0x19]..... | 74 |
| Registro comandi Extra [0x29]..... | 74 |
| Lettura assoluta..... | 74 |
| Control Word [0x2A]..... | 75 |
| Jog +..... | 75 |
| Jog -..... | 75 |
| Stop..... | 75 |
| Reset allarmi..... | 76 |
| Jog incrementale..... | 76 |
| Start..... | 76 |
| Emergenza..... | 76 |
| Abilitazione Watch dog..... | 77 |
| Salva parametri..... | 77 |
| Carica parametri di default..... | 77 |
| Esegui preset conteggio..... | 77 |
| Asse in coppia..... | 77 |
| OUT 1..... | 78 |
| Sblocco freno..... | 78 |
| Target position [0x2B-0x2C]..... | 78 |
| 8.1.2 Parametri Input Register..... | 80 |
| Registro allarmi [0x00]..... | 80 |
| Dati macchina non validi..... | 80 |
| Errore memoria flash..... | 80 |

| | |
|---|-----------|
| Errore di inseguimento..... | 80 |
| Asse non sincronizzato..... | 80 |
| Target non valido..... | 80 |
| Emergenza..... | 81 |
| Sovracorrente..... | 81 |
| Sovratemperatura..... | 81 |
| Sottotensione..... | 81 |
| Watch dog..... | 81 |
| Status word [0x01]..... | 82 |
| Asse in posizione..... | 82 |
| Asse abilitato..... | 82 |
| Finecorsa SW +..... | 82 |
| Finecorsa SW -..... | 82 |
| Allarme..... | 83 |
| Asse in movimento..... | 83 |
| Comando in corso..... | 83 |
| Target raggiunto..... | 83 |
| Pulsante 1 Jog +..... | 83 |
| Pulsante 2 Jog -..... | 83 |
| Pulsante 3 Preset..... | 83 |
| Saturazione DAC..... | 84 |
| IN 1..... | 84 |
| IN 2..... | 84 |
| IN 3..... | 84 |
| Posizione corrente [0x02-0x03]..... | 84 |
| Velocità corrente [0x04]..... | 84 |
| Errore di inseguimento [0x05-0x06]..... | 84 |
| Corrente attuale [0x07]..... | 85 |
| Temperatura [0x08]..... | 85 |
| Elenco DM errati [0x09-0x0A]..... | 85 |
| I2t [0x0B]..... | 86 |
| Dip-switch baud rate [0x0C]..... | 86 |
| Dip-switch indirizzo [0x0D]..... | 86 |
| Versione software [0x0E]..... | 86 |
| Versione hardware [0x0F]..... | 87 |
| 8.2 Codici eccezione..... | 89 |
| 9 Esempi di programmazione..... | 90 |
| 9.1 Utilizzo del codice funzione 03 Read Holding Registers..... | 90 |
| 9.2 Utilizzo del codice funzione 04 Read Input Register..... | 91 |
| 9.3 Utilizzo del codice funzione 06 Write Single Register..... | 93 |
| 9.4 Utilizzo del codice funzione 16 Write Multiple Registers..... | 95 |
| 10 Tabella parametri di default..... | 96 |

Indice analitico

| | | |
|---|--------|--|
| A | | |
| Abilitazione Watch dog..... | 77 | |
| Accelerazione [0x06]..... | 68 | |
| Allarme..... | 83 | |
| Ampiezza passo jog [0x19]..... | 74 | |
| Asse abilitato..... | 82 | |
| Asse in coppia..... | 77 | |
| Asse in movimento..... | 83 | |
| Asse in posizione..... | 82 | |
| Asse non sincronizzato..... | 80 | |
| C | | |
| Carica parametri di default..... | 77 | |
| Comando in corso..... | 83 | |
| Control Word [0x2A]..... | 75 | |
| Corrente attuale [0x07]..... | 85 | |
| Corrente di stacco [0x13]..... | 73 | |
| Corrente massima [0x12]..... | 72 | |
| D | | |
| Dati macchina non validi..... | 80 | |
| Decelerazione [0x07]..... | 68 | |
| Delta spazio negativo [0x0A-0x0B]..... | 69 | |
| Delta spazio positivo [0x08-0x09]..... | 68 | |
| Dip-switch baud rate [0x0C]..... | 86 | |
| Dip-switch indirizzo [0x0D]..... | 86 | |
| Direzione conteggio [0x0F]..... | 72 | |
| Distanza_giro [0x00]..... | 66 | |
| Durata corrente di stacco [0x0E]..... | 71 | |
| E | | |
| Elenco DM errati [0x09-0x0A]..... | 85 | |
| Emergenza..... | 76, 81 | |
| Errore di inseguimento..... | 80 | |
| Errore di inseguimento [0x05-0x06]..... | 84 | |
| Errore memoria flash..... | 80 | |
| Esegui preset conteggio..... | 77 | |
| F | | |
| Finecorsa SW -..... | 82 | |
| Finecorsa SW +..... | 82 | |
| I | | |
| I2t [0x0B]..... | 86 | |
| IN 1..... | 84 | |
| IN 2..... | 84 | |
| IN 3..... | 84 | |
| J | | |
| Jog -..... | 75 | |
| Jog +..... | 75 | |
| Jog incrementale..... | 76 | |
| K | | |
| Ki anello di corrente [0x11]..... | 72 | |
| Ki anello di posizione [0x05]..... | 68 | |
| Kp anello di corrente [0x10]..... | 72 | |
| Kp anello di posizione [0x04]..... | 68 | |
| L | | |
| Lettura assoluta..... | 74 | |
| M | | |
| Max errore di inseguimento [0x03]..... | 67 | |
| O | | |
| OUT 1..... | 78 | |
| P | | |
| Posizione corrente [0x02-0x03]..... | 84 | |
| Pulsante 1 Jog +..... | 83 | |
| Pulsante 2 Jog -..... | 83 | |
| Pulsante 3 Preset..... | 83 | |
| R | | |
| Rapporto di riduzione [0x18]..... | 74 | |
| Registro allarmi [0x00]..... | 80 | |
| Registro comandi Extra [0x29]..... | 74 | |
| Reset allarmi..... | 76 | |
| S | | |
| Salva parametri..... | 77 | |
| Saturazione DAC..... | 84 | |
| Sblocco freno..... | 78 | |
| Sottotensione..... | 81 | |
| Sovracorrente..... | 81 | |
| Sovratemperatura..... | 81 | |
| Start..... | 76 | |
| Status word [0x01]..... | 82 | |
| Stop..... | 75 | |
| T | | |
| Target non valido..... | 80 | |
| Target position [0x2B-0x2C]..... | 78 | |
| Target raggiunto..... | 83 | |
| Temperatura [0x08]..... | 85 | |
| Tempo asse in tolleranza [0x02]..... | 67 | |
| Tolleranza di posizione [0x01]..... | 67 | |
| V | | |
| Valore di offset [0x14-0x15]..... | 73 | |
| Valore di preset [0x16-0x17]..... | 73 | |
| Velocità corrente [0x04]..... | 84 | |
| Velocità di lavoro [0x0D]..... | 71 | |
| Velocità Jog [0x0C]..... | 70 | |
| Versione hardware [0x0F]..... | 87 | |
| Versione software [0x0E]..... | 86 | |
| W | | |
| Watch dog..... | 81 | |

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 **ARANCIONE**;
- 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 ATTENZIONE , 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 NOTA , 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 ESEMPIO quando le istruzioni di impostazione dei parametri siano seguite da esemplificazioni che ne chiarifichino l'utilizzo. |

Informazioni preliminari

Questo manuale ha lo scopo di fornire tutte le informazioni necessarie per un'installazione e un utilizzo corretti e sicuri delle **unità di posizionamento ROTADRIVE modelli RD1A e RD12A**.

Le unità RD1A e RD12A sono dispositivi di movimentazione che integrano in uno un motore brushless con riduttore, un azionamento, un encoder assoluto multigiro e la logica per il posizionamento. Esse trovano impiego in ogni settore industriale e sono adatte a svolgere funzioni di posizionamento di assi secondari come per esempio, tra le tante, cambioformati, battute mobili, cambio utensili, movimento 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.

Le interfacce disponibili per la comunicazione in bus di campo sono le seguenti: **Modbus RTU, Profibus-DP e CANopen DS 301**.

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 ROTADRIVE, l'attivazione e il controllo dei movimenti programmati e il monitoraggio del funzionamento da qualsiasi PC con sistema operativo Windows (Windows XP o successivo). Nelle versioni Profibus e CANopen 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 può essere diviso in due 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 invece, intitolata **Interfaccia Modbus**, sono fornite tutte le informazioni sia generali che specifiche relative all'interfaccia Modbus. In questa sezione sono descritte le caratteristiche dell'interfaccia e i registri Modbus che l'unità implementa.

1 Norme 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 meccaniche 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";
- 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 2004/108/CE 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 "Montaggio meccanico";
- 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;
- dispositivo con albero sporgente: utilizzare giunti elastici per collegare ROTADRIVE e asse utilizzatore; rispettare le tolleranze di allineamento ammesse dal giunto elastico;
- dispositivo con albero cavo: l'unità ROTADRIVE può essere montata direttamente su un albero che rispetti le caratteristiche definite nel foglio d'ordine e fissata mediante il collare e l'asola per l'introduzione di un pin antirotazione.



ATTENZIONE

La taratura dell'unità è stata realizzata testandone il funzionamento a vuoto; i valori di default impostati si riferiscono pertanto a un dispositivo operante in questa modalità. 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 controelettromotrice generata dal motore a seguito di una movimentazione manuale dell'asse forzata dall'esterno può provocare danni irreparabili alla circuiteria interna.

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 s.r.l. 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.



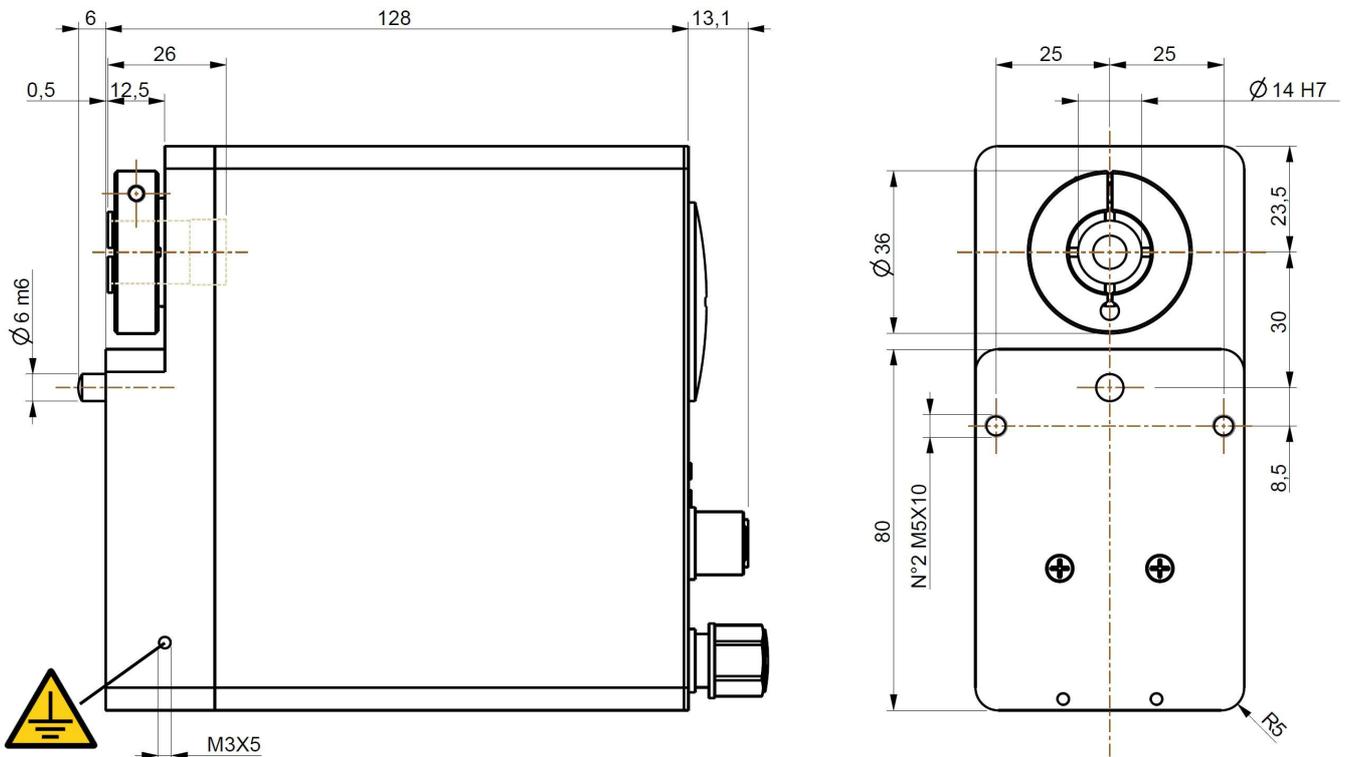


Figura 2 - Lay-out unità RD12A



L'unità ROTADRIVE deve essere fissata in maniera solidale esclusivamente all'asse utilizzatore mediante il collare. E' fornita in dotazione di un gommino antivibrazione sul quale andrà a inserirsi il pin antirotazione, anch'esso fornito in dotazione. 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 ROTARIVE contro il supporto fisso lato utilizzatore. Se questo avvenisse le tensioni meccaniche sarebbero completamente concentrate sull'albero motore con conseguente danneggiamento dei cuscinetti!

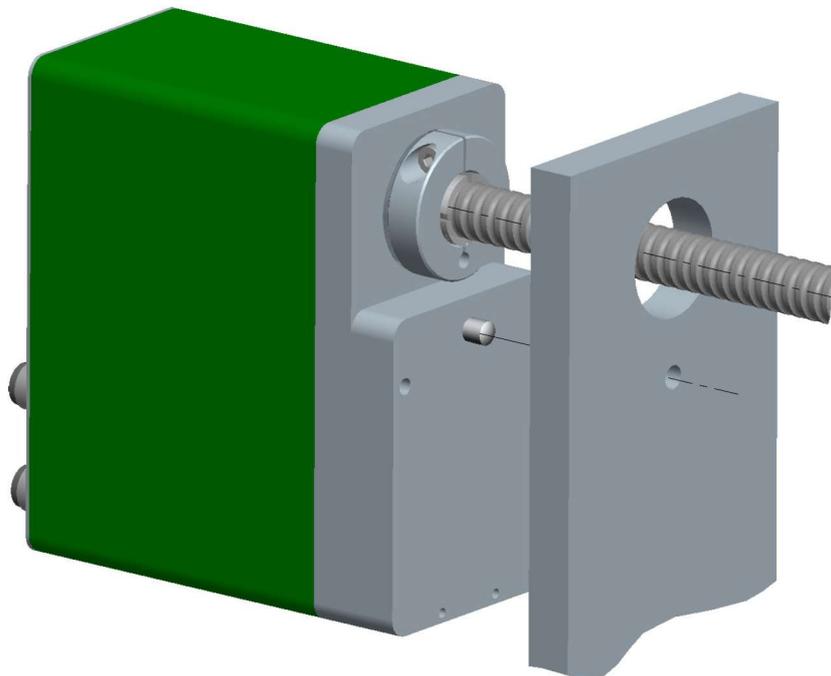
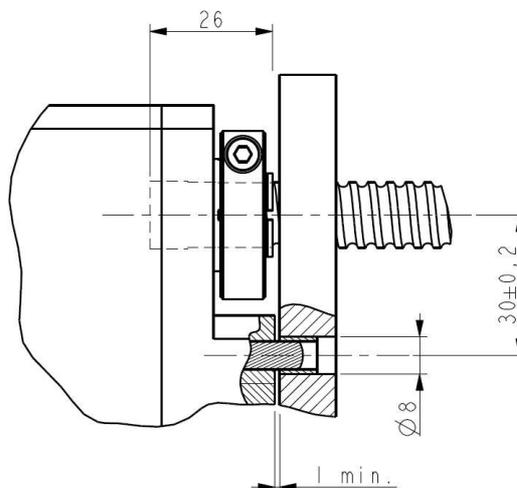


Figura 3 - Esempio di installazione di unità RD1xA su vite senza fine

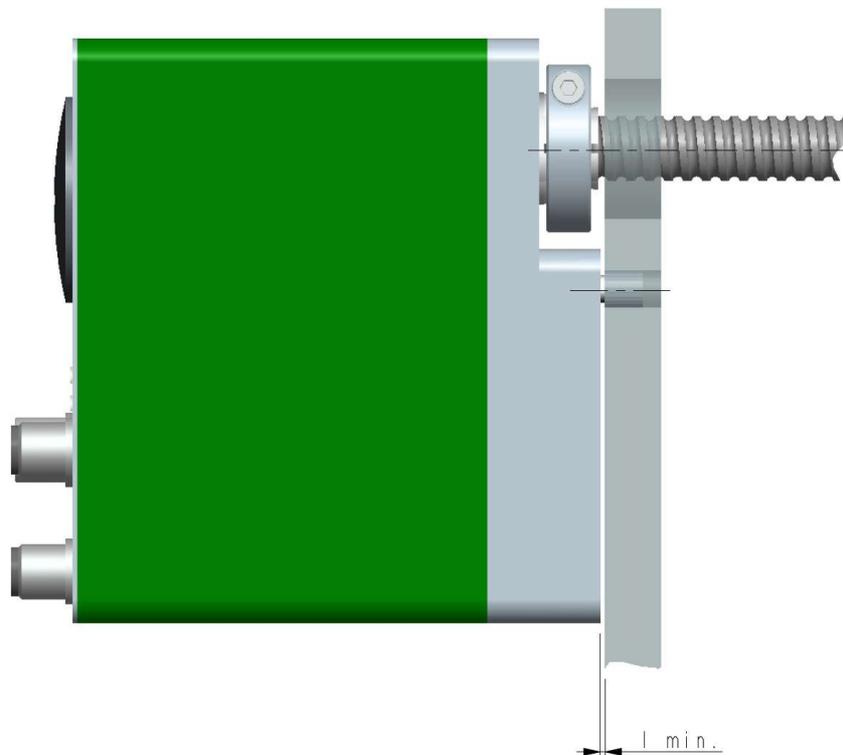
Per installare correttamente l'unità ROTADRIVE seguire scrupolosamente le istruzioni riportate qui di seguito, tenendo conto a ogni modo che le possibilità di installazione possono essere molteplici e dipendenti dalla specifica applicazione.

- Eseguire un foro $\varnothing 8$ mm sulla flangia o sul supporto lato utilizzatore per l'inserimento del gommino antivibrazione e del pin antirotazione. 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 ROTADRIVE 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 ROTADRIVE 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 generata dal motore a seguito di una movimentazione manuale dell'asse forzata dall'esterno può provocare danni irreparabili alla circuiteria interna.

4 Connessioni elettriche



ATTENZIONE

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!

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

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.

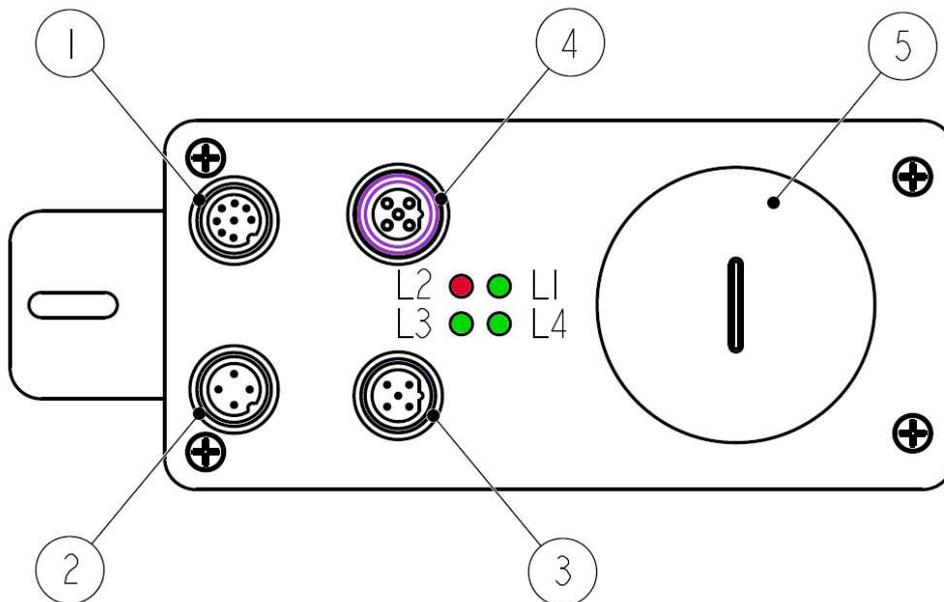


Figura 4: Connessioni elettriche

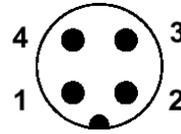
Legenda

| | |
|----|--|
| 1 | Connettore M12 maschio 8 pin ingressi / uscite |
| 2 | Connettore M12 maschio 4 pin alimentazione |
| 3 | Connettore M12 maschio 5 pin BUS IN |
| 4 | Connettore M12 femmina 5 pin BUS OUT |
| 5 | Alloggiamento interno selettori e pulsanti |
| L1 | LED 1 informazione alimentazione controller |
| L2 | LED 2 presenza errori |
| L3 | LED 3 stato bus di campo |
| L4 | LED 4 informazione alimentazione motore |

4.2 Connettori (Figura 4)

Alimentazione

connettore M12 maschio 4 pin
codifica A
(vista lato contatti)

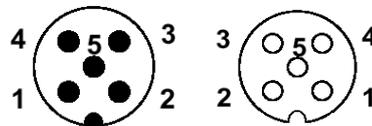


| Pin | Funzione |
|-----|----------------------------|
| 1 | +24VDC ±10% motore |
| 2 | +24VDC ±10% controllore |
| 3 | 0 VDC motore e controllore |
| 4 | n.c. |

n.c. = non connesso

Interfaccia Modbus

connettore M12 5 pin
codifica A
(vista lato contatti)



maschio
(BUS IN)

femmina
(BUS OUT)

| Pin | Funzione |
|----------|--------------------|
| 1 | n.c. |
| 2 | n.c. |
| 3 | GND (RS-485) |
| 4 | Modbus A (RS-485) |
| 5 | Modbus B (RS-485) |
| Custodia | Calza ¹ |

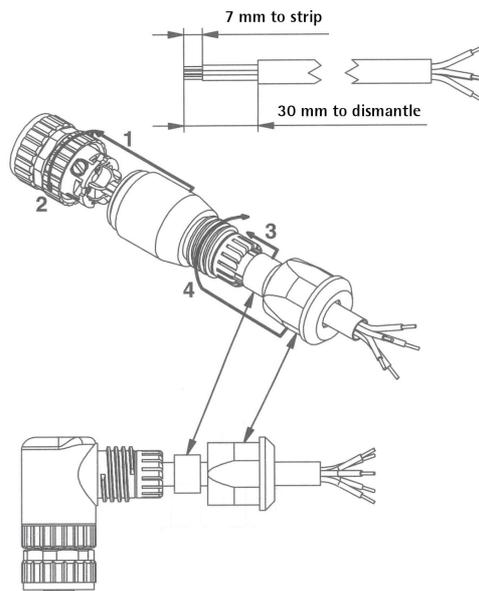
n.c. = non connesso

¹ Solo con cavi intestati EC- Lika Electronic

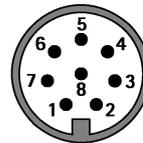


NOTA

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 (Figura sotto). Nel caso in cui si utilizzi un connettore metallico collegare opportunamente la calza del cavo attenendosi alle istruzioni del costruttore.



Ingressi / uscite (opzionale) connettore M12 maschio 8 pin



(vista lato contatti)

| Pin | Funzione |
|-----|------------|
| 1 | 0 VDC |
| 2 | Ingresso 1 |
| 3 | Ingresso 2 |
| 4 | Ingresso 3 |
| 5 | Uscita 1 |
| 6 | n.c. |
| 7 | n.c. |
| 8 | n.c. |

n.c. = non connesso

4.3 Indicatori LED (Figura 4)

Quattro led, posizionati in prossimità dei due connettori BUS (si veda la Figura 4), segnalano visivamente la condizione di funzionamento dell'interfaccia Modbus e del dispositivo, come esplicitato nella seguente tabella:

| LED 1 VERDE | | Descrizione |
|-----------------------|-----------------------|--|
| ON | | Elettronica di controllo alimentata, presenza di tensione |
| OFF | | Elettronica di controllo non alimentata, assenza di tensione |
| LED 2 ROSSO | | Descrizione |
| ON | | Sono presenti allarmi, errore interno |
| OFF | | Nessun allarme presente |
| LED 3 VERDE | | Descrizione |
| Lampeggiante | | Dispositivo in trasmissione o ricezione |
| OFF | | Nessuna trasmissione o ricezione attiva |
| LED 2 | LED 3 | Descrizione |
| VERDE Lampeggiante | VERDE Lampeggiante | Durante il trasferimento dei dati alla memoria flash per l'aggiornamento del firmware del dispositivo (si veda la sezione "5.7 Pagina "Upgrade Firmware"" a pagina 39), i due led lampeggiano verdi a una frequenza di 5 Hz. |
| ROSSO Lampeggiante | ROSSO Lampeggiante | Durante il trasferimento dei dati alla memoria flash per l'aggiornamento del firmware del dispositivo (si veda la sezione "5.7 Pagina "Upgrade Firmware"" a pagina 39), se si verifica un errore che pregiudica l'upgrade (per esempio: la caduta di tensione e lo spegnimento dell'unità ROTADRIVE), l'assenza dello user program all'accensione procura l'attivazione dei due led che lampeggiano rossi a una frequenza di 5 Hz. Per ogni informazione sul ripristino dell'unità riferirsi alla sezione "5.7 Pagina "Upgrade Firmware"" a pagina 39. |
| ROSSO ON | ROSSO ON | Durante il trasferimento dei dati alla memoria flash per l'aggiornamento del firmware del dispositivo (si veda la sezione "5.7 Pagina "Upgrade Firmware"" a pagina 39), la mancata ricezione di record per un tempo superiore a 5 secondi (per esempio, a causa della disconnessione del cavo seriale) procura l'accensione dei due led rossi fissi. Per ogni informazione sul ripristino dell'unità riferirsi alla sezione "5.7 Pagina "Upgrade Firmware"" a pagina 39. |

| LED 4 VERDE | Descrizione |
|-------------|--|
| ON | Motore alimentato, presenza di tensione |
| OFF | Motore non alimentato, assenza di tensione |

Durante l'inizializzazione del dispositivo, il sistema esegue un controllo sul corretto funzionamento degli indicatori led; pertanto gli indicatori led lampeggiano per un istante.

4.4 Selettori e pulsanti (Figura 5)



ATTENZIONE

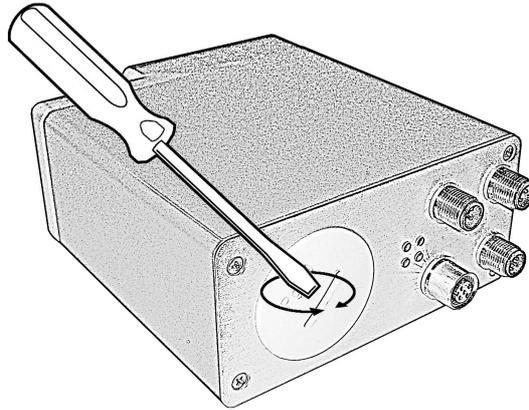
Questa operazione deve essere eseguita con dispositivo non alimentato!



NOTA

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

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



Accedere quindi ai sottostanti selettori e pulsanti.

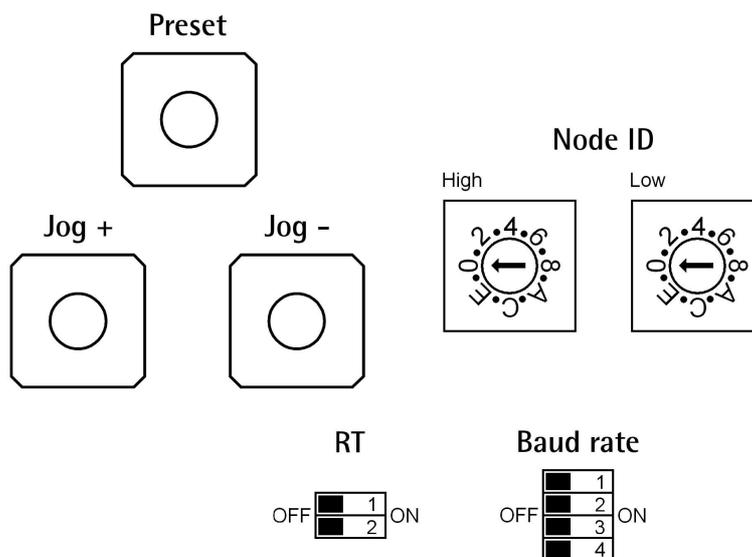


Figura 5: Selettori e pulsanti

4.4.1 Indirizzo nodo: Node ID (Figura 5)



ATTENZIONE

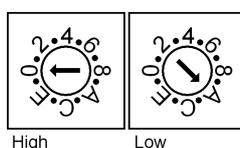
Questa operazione deve essere eseguita con dispositivo non alimentato!

Impostare il valore esadecimale dell'indirizzo del nodo.
L'indirizzo del nodo deve avere un valore compreso tra 1 e 247 (247 = F7 hex).

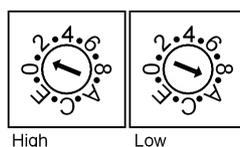


Esempio

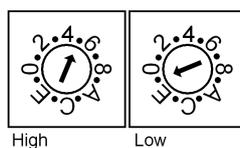
Indirizzo 10 = 0A hex:



Indirizzo 25 = 19 hex:



Indirizzo 95 = 5F hex:



NOTA

L'indirizzo di default è 1.

L'indirizzo 0 è riservato per identificare uno scambio dati in modalità "broadcast" (il Master invia una richiesta contemporaneamente a tutti i nodi connessi nella rete Modbus). Si veda alla sezione "7.1 Principi guida del protocollo Modbus Master / Slave" a pagina 51.

Il Master Modbus non ha un indirizzo specifico, solo i nodi Slave sono tenuti ad avere un indirizzo. L'indirizzo degli Slave deve essere univoco.

Gli indirizzi da 248 a 255 sono riservati.

Se si imposta un indirizzo maggiore di 247, il dispositivo utilizzerà automaticamente l'indirizzo 247.

4.4.2 Velocità di trasmissione dei dati: Baud rate e bit di parità (Figura 5)



ATTENZIONE

Questa operazione deve essere eseguita con dispositivo non alimentato!

Impostare il valore binario della velocità di trasmissione dei dati secondo la seguente tabella, considerando che: ON = 1; OFF = 0.

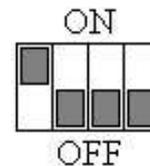
| Switch | Baud rate | Bit di parità |
|----------------|-------------|---------------|
| 0000 | 9600 bit/s | No parity |
| 1000 (default) | 9600 bit/s | Even |
| 0100 | 9600 bit/s | Odd |
| 1100 | 19200 bit/s | No parity |
| 0010 | 19200 bit/s | Even |
| 1010 | 19200 bit/s | Odd |



Esempio

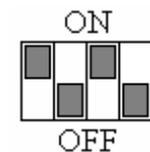
Impostare il baud rate a 9600 bit/s, bit di parità Even:

| Switches | 1 | 2 | 3 | 4 |
|-----------|----|-----|-----|-----|
| Posizione | ON | OFF | OFF | OFF |
| Valore | 1 | 0 | 0 | 0 |



Impostare il baud rate a 19200 bit/s, bit di parità Odd:

| Switches | 1 | 2 | 3 | 4 |
|-----------|----|-----|----|-----|
| Posizione | ON | OFF | ON | OFF |
| Valore | 1 | 0 | 1 | 0 |



4.4.3 Resistenza di terminazione: RT (Figura 5)

La resistenza di terminazione RT deve essere utilizzata come linea di terminazione sull'ultimo dispositivo della rete. Per attivarla si agisce sullo switch RT.

| RT | Descrizione |
|-------------|---|
| 1 = 2 = ON | Attiva: se il dispositivo è l'ultimo della linea |
| 1 = 2 = OFF | Disattiva: se il dispositivo non è l'ultimo della linea |

4.4.4 Pulsanti JOG + e JOG - (Figura 5)

Premere questi pulsanti per comandare i movimenti in direzione positiva (JOG +) e negativa (JOG -) nella modalità di funzionamento manuale. Per ogni informazione si consultino i comandi **Jog +** e **Jog -** a pagina 75.



NOTA

Si badi che nell'utilizzo dei pulsanti manuali la funzione di jog incrementale (si veda **Jog incrementale** in **Control Word [0x2A]** a pagina 76) 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 in corrispondenza dei rispettivi finecorsa.



ATTENZIONE

I pulsanti JOG e PRESET sono sempre attivi, 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.

4.4.5 Pulsante PRESET (Figura 5)

Utilizzare questo pulsante per assegnare alla posizione attuale dell'asse il valore impostato al parametro **Valore di preset [0x16-0x17]**. Il pulsante deve essere mantenuto premuto per almeno 3 secondi. Per ogni informazione si veda il parametro **Valore di preset [0x16-0x17]** a pagina 73.



ATTENZIONE

I pulsanti JOG e PRESET sono sempre attivi, 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

5.1 Configurazione mediante software di Lika Electronic

Le unità RD1A / RD12A Modbus sono fornite con un software sviluppato da Lika Electronic per la programmazione semplificata del dispositivo. Il programma permette di impostare i parametri di lavoro e testare manualmente e monitorare il funzionamento del dispositivo. Il software è fornito gratuitamente e può essere installato in qualsiasi PC con sistema operativo Windows (Windows XP o successivo). Il file di esecuzione del programma **SW_RDX_MODBUS_X.EXE** è contenuto nel supporto informatico allegato oppure disponibile all'indirizzo www.lika.it > **ATTUATORI ROTATIVI** > **CAMBIAFORMATI (DRIVECOD)** > **RD1A / RD12A**). Il programma non richiede installazione; per lanciarlo è sufficiente fare un doppio click sull'icona del file. Per chiudere poi il programma, premere il pulsante **DISCONNETTI** nella pagina **Configurazione seriale**, quindi il pulsante **CHIUDI** nella barra del titolo.



ATTENZIONE

Si badi di rispettare sempre le seguenti compatibilità tra la versione hardware-software del dispositivo e la versione software dell'eseguibile Modbus.

| Compatibilità | HW-SW | EXE Modbus |
|---------------|-------------------------|---------------|
| | 1-1 | fino a V2.1 |
| | 1-2, 1-3, 1-4, 1-5 | fino a V2.4 |
| | 2-6, 3-0, 3-1, 3-2, 3-3 | da V2.5 a ... |



NOTA

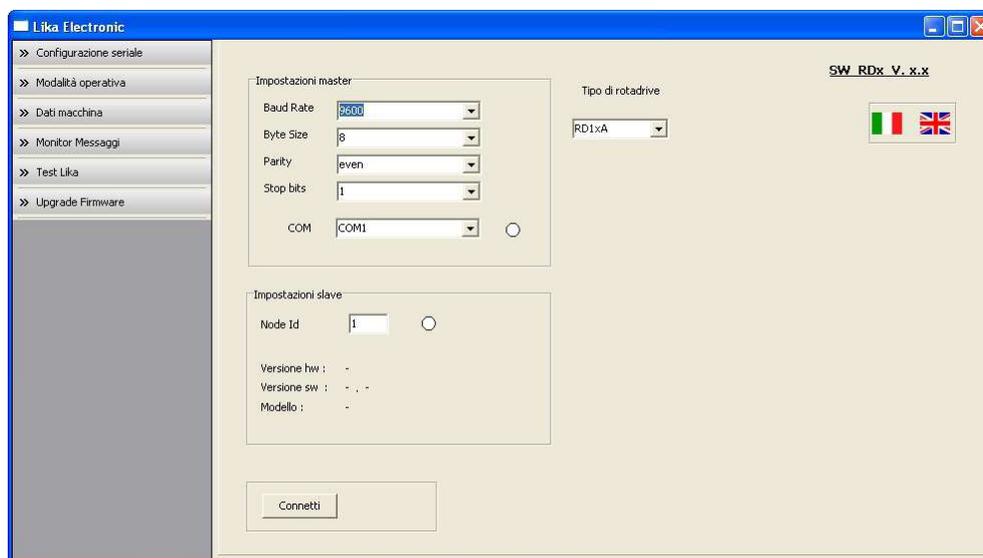
Prima di connettersi al dispositivo è necessario collegarlo serialmente al personal computer. L'interfaccia seriale dell'unità ROTADRIVE è del tipo RS-485, mentre lo standard seriale dei personal computer (quando previsto) è del tipo RS-232. Bisogna quindi prevedere l'utilizzo di un convertitore RS-485 / RS-232, facilmente reperibile in commercio. Nel caso in cui il personal computer non sia provvisto di porta seriale (RS-232 o RS-485), sarà necessario installare un convertitore USB / RS-485, anch'esso facilmente reperibile in commercio. Per ogni informazione sullo schema di collegamento e il pinout del cavo riferirsi al foglio informativo del convertitore utilizzato.

Sul lato ROTADRIVE, il cavo deve essere collegato al connettore M12 5 pin maschio (BUS IN).

Il cavo precablato con connettori M12 5 pin / USB e convertitore RS-485 integrato è disponibile su richiesta contattando il servizio di Assistenza Tecnica di Lika Electronic s.r.l. e citando il seguente codice: **EXC-USB4-S54-GN-2-M12MC-S54**.

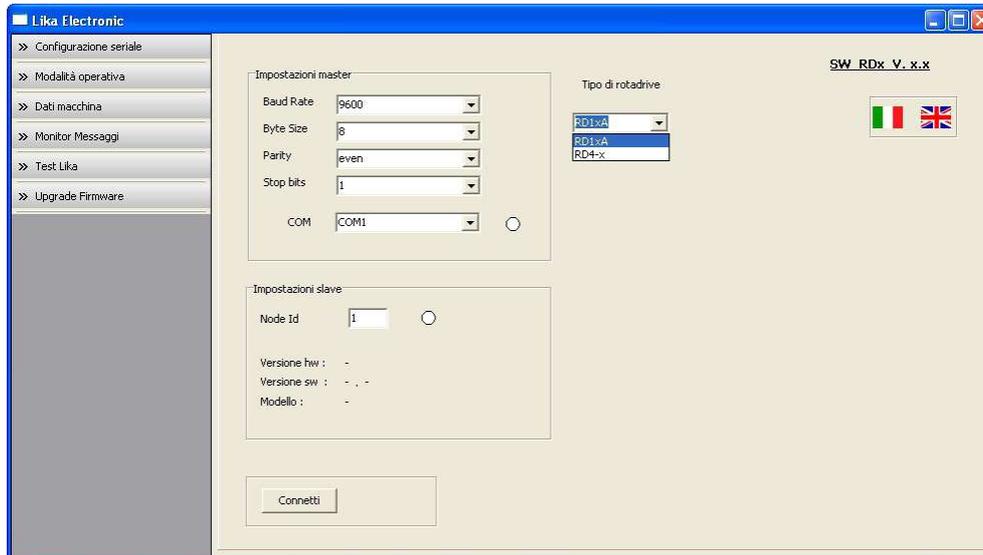
5.2 Pagina "Configurazione seriale"

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

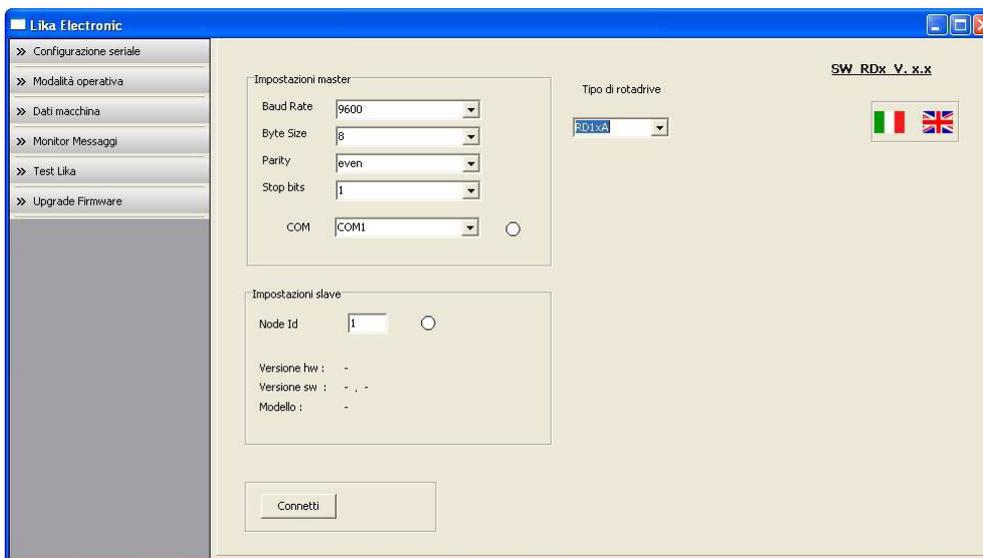


Questa pagina permette anzitutto la scelta della lingua di visualizzazione dei testi. Cliccare sul pulsante  **Bandiera italiana** per scegliere la lingua di visualizzazione italiana; cliccare sul pulsante  **Bandiera inglese** per scegliere la lingua di visualizzazione inglese.

Prima di accedere alle pagine successive attraverso i pulsanti del menu sul lato sinistro, è necessario attivare la connessione seriale con il dispositivo. Per fare questo aprire il menu a tendina **Tipo di Rotadrive** e selezionare il modello **RD1xA**.



Sul lato sinistro del menu a tendina si attiverà il box **Impostazioni master** che permette di scegliere la porta seriale del personal computer cui il dispositivo è connesso (menu a tendina **COM**) e impostarne quindi i relativi parametri di configurazione. Essi dovranno corrispondere ai parametri della porta seriale del dispositivo collegato.



I valori di default impostati da Lika Electronic per i dispositivi RD1xA-Modbus sono i seguenti:

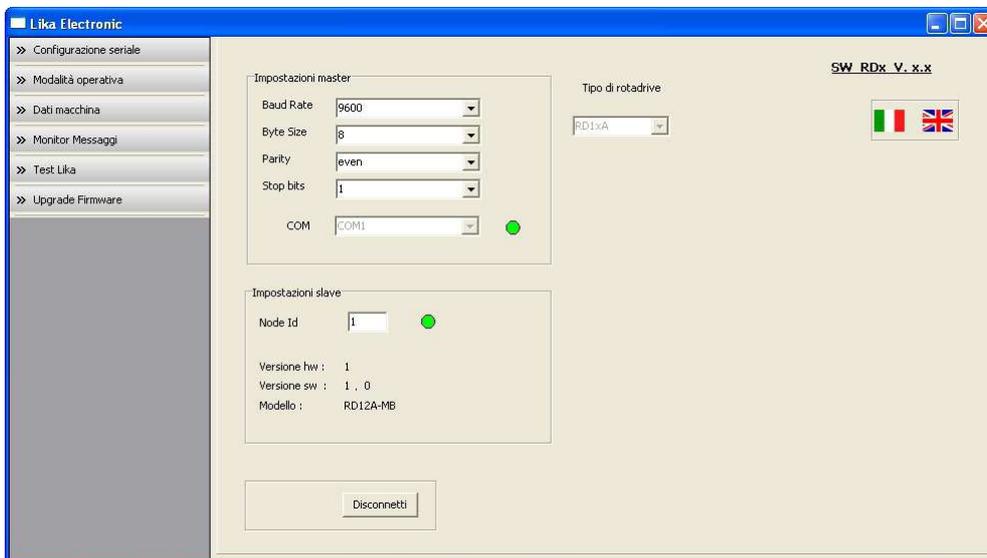
Impostazioni porta seriale

| RS-485 RD1xA-Modbus | Valore di default |
|---------------------|-------------------|
| Baud rate | 9600 |
| Byte size | 8 |
| Parity | Even |
| Stop bits | 1 |

Per impostare i parametri della porta seriale del dispositivo riferirsi al paragrafo "4.4.2 Velocità di trasmissione dei dati: Baud rate e bit di parità (Figura 5)" a pagina 24.

Impostare quindi l'indirizzo del nodo cui ci si vuole connettere nel box **Impostazioni slave** (valore di default dei dispositivi RD1xA-Modbus = 1). Per impostare l'indirizzo del nodo riferirsi al paragrafo "4.4.1 Indirizzo nodo: Node ID (Figura 5)" a pagina 23.

E' ora possibile attivare la connessione premendo il pulsante **CONNETTI** in basso nella pagina.

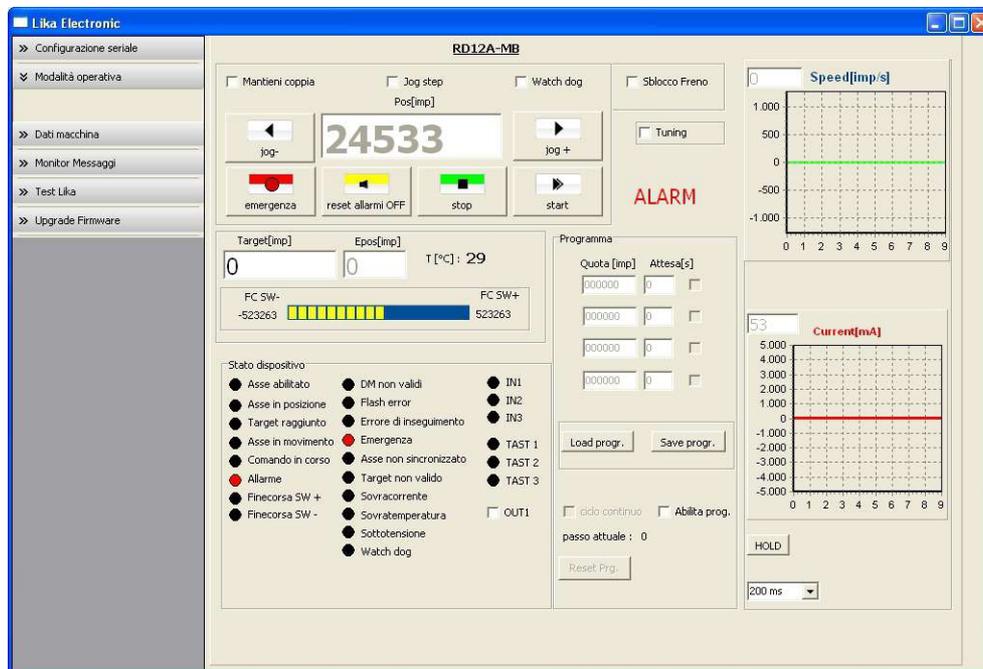


Se la connessione ha esito positivo, le due segnalazioni in corrispondenza dei campi di impostazione della **porta seriale** e dell'**indirizzo del nodo** si illuminano colorandosi di verde, mentre il pulsante **CONNETTI** sparisce sostituito

dal pulsante **DISCONNETTI**. Compaiono inoltre la versione hardware e quella software, nonché il modello del dispositivo nel box **Impostazioni slave**.

5.3 Pagina "Modalità operativa"

Premere ora il pulsante **MODALITA' OPERATIVA** nel menu sul lato sinistro per avviare l'attività di programmazione, gestione manuale e monitoraggio del dispositivo. Compare la seguente schermata.



All'attivazione della modalità operativa il dispositivo si trova necessariamente in condizione di emergenza: il pulsante **Emergenza** è evidenziato in rosso e la segnalazione **ALARM** lampeggia sulla destra, mentre nel box **Stato dispositivo** in basso a sinistra sono attive le due segnalazioni **Allarme** e **Emergenza**; inoltre il LED 2 rosso del dispositivo è acceso fisso. Per ripristinare lo stato **Idle** del dispositivo, premere il pulsante **EMERGENZA** prima e il pulsante **RESET ALLARMI** poi. Le segnalazioni di allarme scompaiono e il LED 3 verde del dispositivo inizia a lampeggiare.

Nel box **RD1xA-MB** in alto a sinistra sono disponibili le seguenti funzioni.

Mantieni coppia

Si veda **Asse in coppia** a pagina 77. Disponibile solo per la versione RD1A, nella versione RD12A il check box non è visualizzato.

Jog step

Si veda **Jog incrementale** a pagina 76.

Watch dog

Si veda **Abilitazione Watch dog** a pagina 77.

Jog -

Si veda **Jog -** a pagina 75.

Pos [imp]

Si veda **Posizione corrente [0x02-0x03]** a pagina 84.

Jog +

Si veda **Jog +** a pagina 75.

Emergenza

Se è presente una condizione di emergenza, il pulsante **Emergenza** è evidenziato in rosso; premere il pulsante per ripristinare lo stato normale di lavoro. Con dispositivo in movimento, comanda un arresto di emergenza. Si veda **Emergenza** a pagina 76.

Reset allarmi

Se è presente un allarme, il pulsante **RESET ALLARMI** è evidenziato in giallo; premere il pulsante per resettare l'allarme. Si veda **Reset allarmi** a pagina 76.

Stop

Comanda l'arresto normale del dispositivo. Si veda **Stop** a pagina 75.

Start

Comanda il movimento del dispositivo per il raggiungimento della posizione impostata in **Target [imp]**; sul lato destro lampeggia l'avvertenza **MOVING**. Al raggiungimento della quota impostata il dispositivo si arresta attivando i bit di stato **Asse in posizione** e **Target raggiunto**. Per un arresto normale del dispositivo premere **STOP**; per un arresto di emergenza premere **EMERGENZA**. Si veda **Start** a pagina 76.

Nel box sottostante sono disponibili le seguenti funzioni:

Target [imp]

Si veda **Target position [0x2B-0x2C]** a pagina 78. Impostare il valore desiderato e confermare con il tasto **INVIO** della tastiera. Alla pressione del pulsante **START** il dispositivo avvia il posizionamento alla quota impostata in questo campo **Target [imp]**, quindi si arresta attivando i bit di stato **Asse in posizione** e **Target raggiunto**.

E pos [imp]

Si veda **Errore di inseguimento [0x05-0x06]** a pagina 84.

T [°C]

Si veda **Temperatura [0x08]** a pagina 85.

FC SW- / FC SW +

Visualizzazione grafica dei valori di finecorsa positivo e negativo. Si vedano **Delta spazio positivo [0x08-0x09]** a pagina 68 e **Delta spazio negativo [0x0A-0x0B]** a pagina 69.

Nel box **Stato dispositivo** in basso nella pagina è visualizzato l'elenco degli stati e degli allarmi disponibili per il dispositivo. Gli stati attivi sono evidenziati in verde; gli allarmi attivi sono evidenziati in rosso. Per una descrizione completa degli stati si veda **Status word [0x01]** a pagina 82; per una descrizione completa degli allarmi si veda **Registro allarmi [0x00]** a pagina 80. Il check box **OUT1** attiva / disattiva l'uscita digitale 1 del dispositivo. Per ogni informazione si veda a pagina 78.

Le funzioni disponibili nel box **Programma** permettono la creazione e il salvataggio di un programma di lavoro del dispositivo. Le funzioni del box **Programma** sono per default disabilitate; per abilitarle selezionare il check box **ABILITA PROG**.

Nei campi **QUOTA [IMP]** si impostano le quote che il dispositivo deve raggiungere (posizione target); si possono impostare fino a quattro quote successive. Nei campi **ATTESA [S]** si imposta la pausa tra un posizionamento e il successivo. I valori impostati devono essere confermati premendo il tasto **INVIO** della tastiera. Ogni campo deve essere preventivamente abilitato selezionando il check box a lato.

Il check box **CICLO CONTINUO** in basso permette di attivare l'esecuzione continua del ciclo di posizionamento impostato.

Se **CICLO CONTINUO** è selezionato, alla pressione del pulsante **START**, il dispositivo avvia il posizionamento alla prima quota impostata; i campi **QUOTA [IMP]** e **ATTESA [S]** si evidenziano in giallo; al raggiungimento della quota impostata, il dispositivo si arresta per la pausa impostata e il campo **QUOTA [IMP]** si evidenzia in verde; allo scadere della pausa impostata (si visualizza un contatore a ritroso) anche il campo **ATTESA [S]** si evidenzia in verde. Allo scadere della pausa, immediatamente il dispositivo avvia il posizionamento alla seconda quota impostata; e così di seguito dalla prima alla quarta quota (se abilitate) e di nuovo dalla prima alla quarta senza soluzione di continuità fino alla pressione del pulsante **STOP**.

Se invece **CICLO CONTINUO** non è selezionato, alla pressione del pulsante **START**, il dispositivo avvia il posizionamento alla prima quota impostata e al raggiungimento della quota si arresta e attiva il contatore per l'esecuzione della pausa; quindi, allo scadere della pausa, rimane in attesa di un nuovo comando **START**; e così di seguito.

E' possibile salvare il programma di lavoro creato. Per fare questo premere il pulsante **SAVE PROGR**. Alla pressione del pulsante si visualizza il dialog box **Salva con nome**: l'operatore deve digitare il nome del file .prg dove i dati saranno salvati e indicarne il repository. Alla conferma mediante il pulsante **SALVA**, il box si chiude.

Per caricare un programma precedentemente salvato, premere il pulsante **LOAD PROGR**. Alla pressione del pulsante si visualizza il dialog box **Apri**: l'operatore deve selezionare il file .prg precedentemente salvato e confermare la scelta mediante il pulsante **APRI**, il box si chiude.

Il pulsante **RESET PRG**. resetta l'esecuzione del programma in corso: alla successiva pressione del pulsante **START** il dispositivo avvierà il posizionamento alla prima quota impostata, quale che fosse la quota raggiunta al precedente arresto.

Per disattivare l'esecuzione di un programma deselezionare il check box **ABILITA PROG**.

Sul lato destro della pagina sono visualizzati graficamente la velocità del dispositivo (espressa in impulsi/secondo) e il valore della corrente assorbita dal motore (espresso in milliampere). Premere il pulsante **HOLD** per disattivare la visualizzazione; premere lo stesso pulsante (che presenta poi l'etichetta **GO**) per

riattivarla. Il menu a tendina in basso permette di scegliere la scala di visualizzazione temporale.

In alto, a destra del box **RD1xA-MB** è presente il check box **TUNING**. Alla selezione del check box, i diagrammi della velocità e della corrente assorbita presenti sulla destra della pagina scompaiono sostituiti da quattro slider per l'impostazione dei valori di guadagno proporzionale e integrale relativi all'anello di posizione e all'anello di corrente. Per ogni informazione riferirsi alle rispettive variabili in questo manuale (sezione "8.1.1 Parametri Dati macchina" a pagina 66).

Quando ci si collega a un dispositivo modello RD12A, appena sopra il check box **TUNING** compare un ulteriore check box: **SBLOCCO FRENO**. Il modello RD12A si differenzia dal modello RD1A per la presenza di un freno che, all'arresto del dispositivo, si attiva inibendo la possibilità di qualsiasi movimento dell'asse. Selezionando questo check box, il freno è momentaneamente disattivato, per cui, per esempio, è possibile muovere manualmente l'asse dell'unità ROTADRIVE. Il freno non rimane disattivato per un tempo indefinito, il sistema prevede un time-out di 20 secondi allo scadere del quale il freno si riattiva automaticamente.

5.4 Pagina "Dati macchina"

Premendo il pulsante **DATI MACCHINA** nel menu sul lato sinistro si accede alla pagina **Dati Macchina**.

In questa pagina è disponibile l'elenco dei parametri di impostazione del dispositivo (dati macchina). A sinistra di ciascun campo di impostazione è visualizzato il valore attualmente caricato sul dispositivo; a destra invece sono indicati i valori minimo e massimo ammessi. Per ogni informazione sulla funzione e la programmazione di ciascun parametro riferirsi alla sezione "8.1.1 Parametri Dati macchina" a pagina 66.

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, alla conferma il campo di impostazione si colora di rosso e il dispositivo è forzato in condizione di allarme (attivazione del bit di stato **Allarme** e visualizzazione dei messaggi di errore **Dati macchina non validi** e/o **Emergenza**). Per ripristinare la normale modalità operativa del dispositivo, impostare un valore valido, quindi premere il pulsante **RESET ALLARMI** nella pagina **Modalità operativa**.

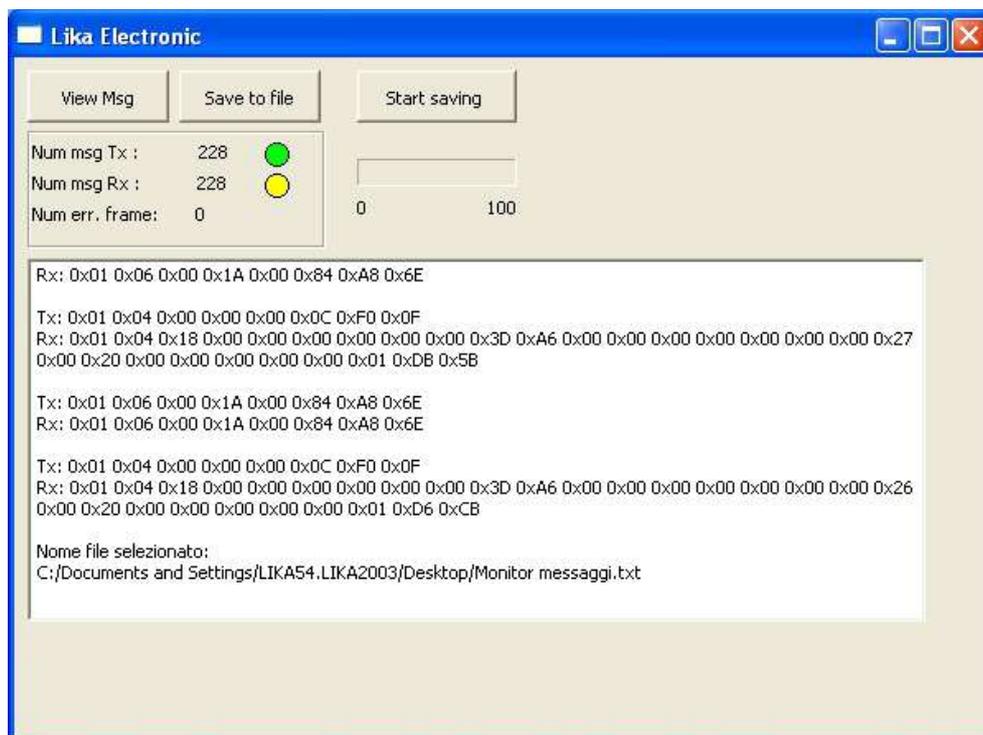
Per salvare i parametri modificati nella memoria non volatile del dispositivo premere il pulsante **SALVA PARAMETRI**. Nel caso di spegnimento del

dispositivo i dati non salvati andranno persi! Per ogni informazione sul salvataggio dei parametri riferirsi alla variabile **Salva parametri** a pagina 77.

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 **CARICA PARAMETRI DI DEFAULT**. Per ogni informazione sull'impostazione dei parametri di default riferirsi alla variabile **Carica parametri di default** a pagina 77. A pagina 96 è disponibile l'elenco dei dati macchina e il rispettivo valore di default preimpostato da Lika Electronic.

5.5 Pagina "Monitor messaggi"

Alla pressione del pulsante **MONITOR MESSAGGI** nel menu sul lato sinistro si accede alla pagina **Monitor Messaggi**.



Tramite questa pagina l'operatore può monitorare la comunicazione tra Master e Slave, visualizzando i messaggi Request PDU (Tx) e Response PDU (Rx). Alla prima apertura della pagina, il campo di visualizzazione dei messaggi è vuoto. Nel box posto immediatamente sotto i pulsanti è visualizzato il numero di

messaggi trasmessi (Num msg TX) e il numero di messaggi ricevuti (Num msg Rx).

Per visualizzare i messaggi è necessario premere il pulsante **VIEW MSG**. Non appena il pulsante viene premuto la pagina inizia a visualizzare il flusso di trasmissione tra Master e Slave. I messaggi sono visualizzati in formato esadecimale. Dopo la pressione del pulsante **VIEW MSG**, la label dello stesso pulsante cambia in **HOLD MSG**. Premere dunque il pulsante **HOLD MSG** per arrestare la visualizzazione dei messaggi.

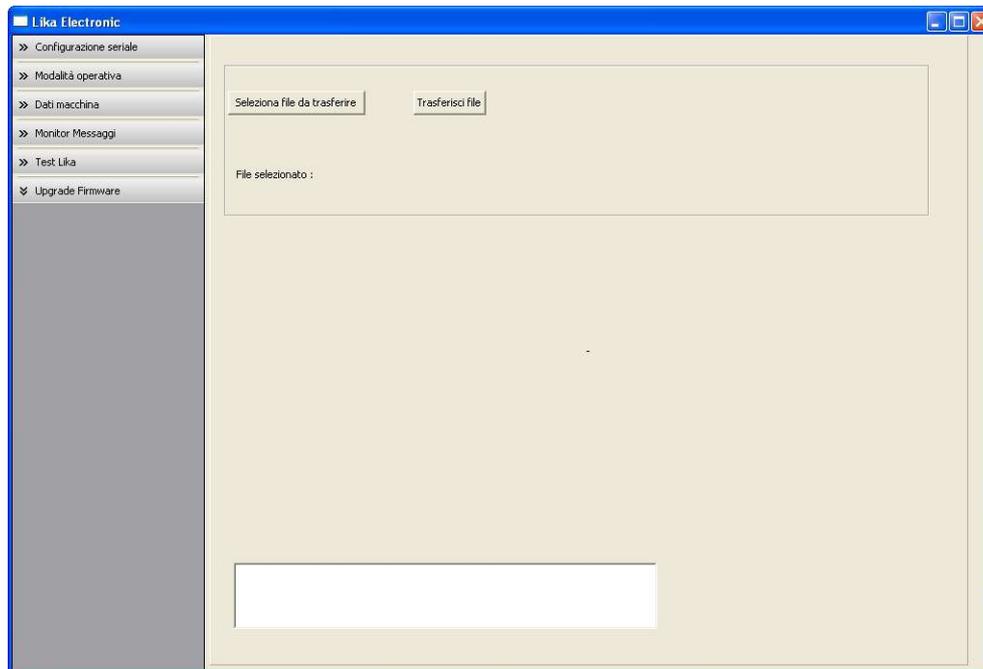
E' possibile memorizzare i messaggi in un file di testo. Per fare questo premere il pulsante **SAVE TO FILE**. Alla pressione del pulsante si apre il dialog box **Apertura file di log**: l'operatore deve digitare il nome del file .txt dove i dati saranno salvati e indicarne il repository. Alla conferma mediante il pulsante **APRI**, il box si chiude e nel campo di visualizzazione della pagina **Monitor Messaggi** compare il path completo del file indicato. Premere quindi il pulsante **START SAVING** per iniziare il salvataggio dei messaggi, nel campo di visualizzazione della pagina **Monitor Messaggi** compare il messaggio "File aperto correttamente". Dopo la pressione del pulsante **START SAVING**, la label dello stesso pulsante cambia in **STOP SAVING**. Premere dunque il pulsante **STOP SAVING** per arrestare la memorizzazione dei messaggi.

5.6 Pagina "Test Lika"

La pagina **Test Lika** è riservata all'utilizzo da parte dei tecnici di Lika Electronic e non è accessibile all'utente.

5.7 Pagina "Upgrade Firmware"

Alla pressione del pulsante **UPGRADE FIRMWARE** nel menu sul lato sinistro si accede alla pagina **Upgrade Firmware**.



Le funzioni disponibili in questa pagina permettono l'aggiornamento del firmware dell'unità ROTADRIVE 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, qui chiamato anche "user program" o "programma utente", è memorizzato nella memoria flash integrata all'interno dell'unità ROTADRIVE. I dispositivi ROTADRIVE 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 ROTADRIVE 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 ROTADRIVE, non è necessario procedere con l'installazione di alcun aggiornamento. La versione firmware correntemente installata può essere verificata alla voce **VERSIONE SW** nel box **Impostazioni slave** della pagina **Configurazione seriale** dopo la corretta connessione al dispositivo (si veda a pagina 30).

La funzione di aggiornamento firmware è implementata:

- a partire dalla versione firmware 1.0 nelle unità ROTADRIVE con interfaccia CANopen e Modbus;
- a partire dalla versione firmware 2.0 nelle unità ROTADRIVE con interfaccia Profibus.

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



NOTA

La funzione di aggiornamento del firmware è implementata a partire dalla versione 2.0 del software di programmazione via seriale in rete Modbus **SW_RDX_MODBUS_2.0.EXE**. Prima di procedere all'aggiornamento del firmware assicurarsi di possedere la versione citata del programma oppure scaricarla all'indirizzo **www.lika.it > ATTUATORI ROTATIVI > CAMBIAFORMATI (DRIVECOD) > RD1A / RD12A**.

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

1. assicurarsi che siano impostati i seguenti parametri di configurazione della porta seriale dell'unità ROTADRIVE: baud rate = 9600 bit/s; parity = even; in caso contrario modificarli seguendo le istruzioni al paragrafo "4.4.2 Velocità di trasmissione dei dati: Baud rate e bit di parità (Figura 5)" a pagina 24;
2. assicurarsi che l'unità ROTADRIVE da aggiornare sia l'unico nodo collegato al personal computer;
3. assicurarsi di avviare la versione 2.0 o successiva del software di programmazione via seriale in rete Modbus **SW_RDX_MODBUS_X.EXE**;
4. collegarsi al dispositivo e accedere alla pagina **Upgrade Firmware**;
5. se, all'accensione, i LED 2 e 3 lampeggiassero rossi a una frequenza di 5 Hz (assenza dello user program nella memoria flash), non sarà possibile collegarsi al dispositivo mediante la pagina **Configurazione seriale**; in questo caso accedere direttamente alla pagina **Upgrade**

Firmware; accertarsi che nella pagina **Configurazione seriale** sia selezionata la corretta porta seriale del personal computer collegata all'unità ROTADRIVE; per maggiori informazioni si veda qui sotto alla sezione "5.7.1 In caso di errore";

6. premere il pulsante **SELEZIONA FILE DA TRASFERIRE** e, mediante la finestra **APRI**, individuare il repository dove è contenuto il file di aggiornamento firmware .BIN fornito da Lika Electronic;



ATTENZIONE

Si badi che, per ogni modello di unità ROTADRIVE 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: RD1xA_PB_H1S2.0.BIN, dove:

- RD1xA = modello unità ROTADRIVE;
- PB = interfaccia bus unità ROTADRIVE (MB = Modbus; CB = CANopen; PB = Profibus);
- H1 = versione hardware;
- S2.0 = versione firmware.

7. selezionare il file .BIN fornito da Lika Electronic e confermare la scelta mediante la pressione del pulsante **APRI** in basso nella finestra;
8. in corrispondenza dell'etichetta **FILE SELEZIONATO** compare il percorso completo relativo al file appena confermato;
9. premere ora il pulsante **TRASFERISCI FILE** per procedere al download dell'aggiornamento firmware;
10. al centro della pagina è visualizzato lo stato di avanzamento del processo;
11. durante l'aggiornamento della memoria flash i LED 2 e 3 lampeggiano verdi a una frequenza di 5 Hz;



ATTENZIONE

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

12. al completamento dell'operazione, se l'esito è positivo, viene visualizzato il messaggio **TRASFERIMENTO COMPLETATO CON SUCCESSO**;
13. l'unità ROTADRIVE si trova ora in condizione di emergenza;
14. chiudere il programma SW_RDX_MODBUS_X.EXE e riavviarlo; riconnettersi all'unità ROTADRIVE e, accedendo alla pagina **Modalità operativa**, ripristinare la normale condizione di lavoro.

5.7.1 In caso di errore

Durante il download dell'aggiornamento firmware potrebbero verificarsi delle condizioni inattese che pregiudicano l'esito positivo del processo. In questo caso il download non può essere completato e la condizione di fault è segnalata mediante l'accensione dei LED 2 e 3 (pagina 20), come esplicitato qui di seguito.

LED 2 E 3 LAMPEGGIANTI ROSSI A 5 Hz

Durante il trasferimento dei dati alla memoria flash, se si verifica, per esempio, una caduta di tensione e lo spegnimento dell'unità, sarà impossibile ripristinare il firmware che si è proceduto a cancellare dalla memoria flash all'inizio del processo. Nel box in basso nella pagina **Upgrade Firmware** compare il messaggio: **ERRORE DI COMUNICAZIONE. AGGIORNAMENTO FLASH ANNULLATO**. L'assenza dello user program all'accensione procura quindi l'attivazione dei due LED che lampeggiano rossi a una frequenza di 5 Hz. Occorre chiudere il programma SW_RDX_MODBUS_X.EXE e poi riavviarlo. Non è possibile collegarsi al dispositivo mediante la pagina **Configurazione seriale**, ma si deve accedere direttamente alla pagina **Upgrade Firmware**; iniziare quindi la procedura di ripristino a partire dal punto 6. Accertarsi sempre che nella pagina **Configurazione seriale** sia selezionata la corretta porta seriale del personal computer collegata all'unità ROTADRIVE.

LED 2 E 3 FISSI ROSSI

Durante il trasferimento dei dati alla memoria flash, la mancata ricezione di record per un tempo superiore a 5 secondi (per esempio, a causa della disconnessione del cavo seriale) procura l'accensione dei LED 2 e 3 rossi fissi. In questo caso sarà impossibile ripristinare il firmware che si è proceduto a cancellare dalla memoria flash all'inizio del processo. Nel box in basso nella pagina **Upgrade Firmware** compare il messaggio: **ERRORE DI COMUNICAZIONE. AGGIORNAMENTO FLASH ANNULLATO**. E' necessario spegnere e riaccendere l'unità ROTADRIVE. L'assenza dello user program all'accensione procura quindi l'attivazione dei due LED che ora lampeggiano rossi a una frequenza di 5 Hz. Occorre inoltre chiudere il programma SW_RDX_MODBUS_X.EXE e poi riavviarlo. Non è possibile collegarsi al dispositivo mediante la pagina **Configurazione seriale**, ma si deve accedere direttamente alla pagina **Upgrade Firmware**; iniziare quindi la procedura di ripristino a partire dal punto 6. Accertarsi sempre che nella pagina **Configurazione seriale** sia selezionata la corretta porta seriale del personal computer collegata all'unità ROTADRIVE.

5.8 Getting started

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

- Procedere all'installazione meccanica;
- effettuare le connessioni elettriche;
- impostare la velocità di trasmissione dei dati (baud rate, si veda a pagina 24); il valore di default preimpostato da Lika Electronic è "baud rate = 9600 bit/s, parità = Even";
- impostare l'indirizzo del nodo (node ID, si veda a pagina 23); il valore di default preimpostato da Lika Electronic è "1";
- alimentare l'unità con alimentazione +24VDC (sia il motore che il controllore);
- impostare il valore in **Distanza_giro [0x00]** (registro 1; si veda a pagina 66);
- impostare il valore in **Velocità Jog [0x0C]** (registro 13, si veda alla pagina 70);
- impostare il valore in **Velocità di lavoro [0x0D]** (registro 14, si veda alla pagina 71);
- impostare il valore in **Valore di preset [0x16-0x17]** (registri 23-24; si veda a pagina 73);
- impostare i valori finecorsa in **Delta spazio positivo [0x08-0x09]** e **Delta spazio negativo [0x0A-0x0B]**; si veda a pagina 68);
- procedere al salvataggio dei nuovi valori (registro **Salva parametri**; si veda a pagina 77).



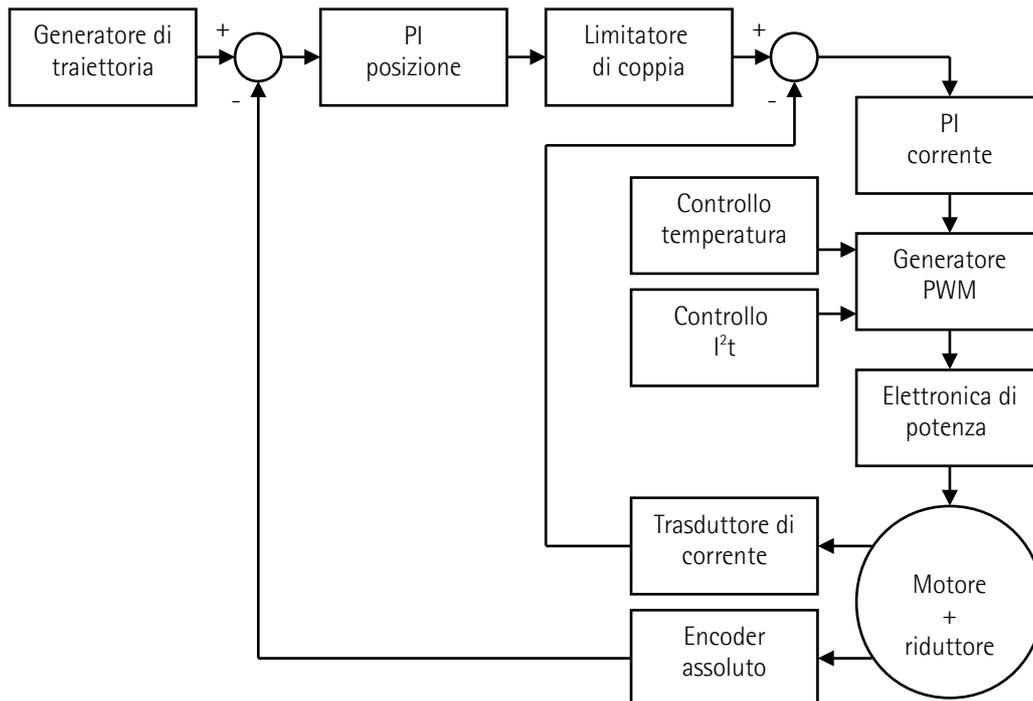
NOTA

I dati macchina **Distanza_giro [0x00]**, **Velocità Jog [0x0C]**, **Velocità di lavoro [0x0D]**, **Valore di preset [0x16-0x17]**, **Delta spazio positivo [0x08-0x09]** e **Delta spazio negativo [0x0A-0x0B]** sono strettamente relazionati tra loro; si deve quindi prestare particolare attenzione quando si procede alla modifica anche di uno solo di essi. Per maggiori informazioni riferirsi a pagina 47.

6 Funzioni

6.1 Principio di funzionamento

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



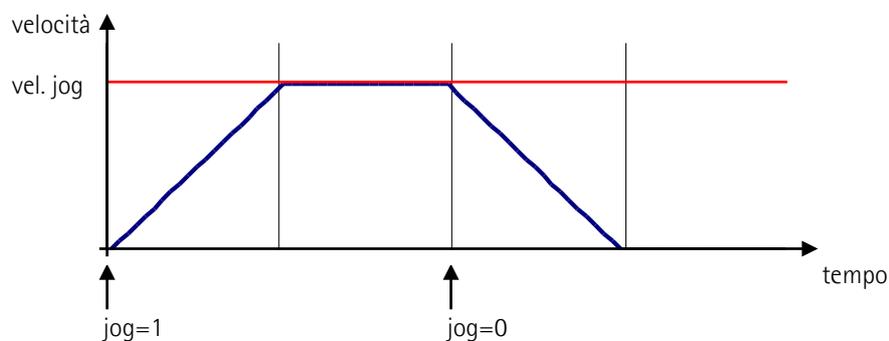
6.2 Tipi di movimento: jog e posizione

I tipi di movimento previsti nell'unità ROTADRIVE sono i seguenti:

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

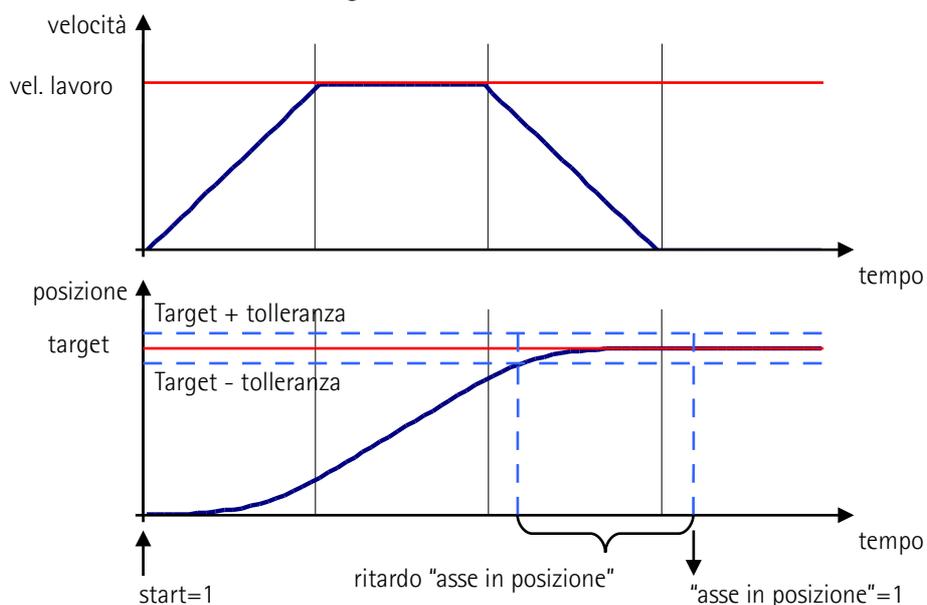
Jog: controllo di velocità

Questo tipo di controllo genera una traiettoria di velocità tale da imporre a regime una velocità di rotazione dell'asse del dispositivo uguale a **Velocità Jog [0x0C]** (si veda l'informazione relativa al parametro a pagina 70).



Posizionamento: controllo di posizione e velocità

Questo tipo di controllo è un movimento punto-punto, nel quale la massima velocità raggiunta è pari a **Velocità di lavoro [0x0D]** (si veda l'informazione relativa al parametro a pagina 71) che verrà raggiunta solo se lo spazio da percorrere è sufficientemente grande.



6.3 Ingressi e uscite digitali

Il dispositivo è provvisto di **tre ingressi e un'uscita digitali**.

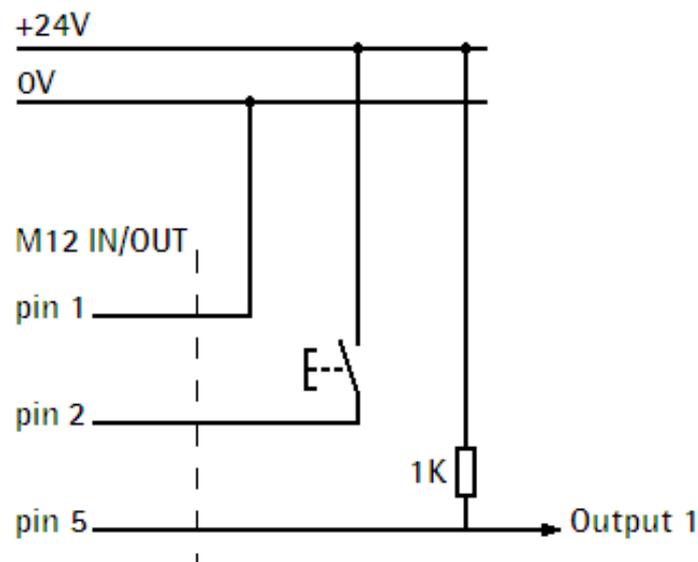
Gli ingressi sono letti dal dispositivo e trasmessi al Master attraverso **Status word [0x01]** (bit 13-15, si veda alla pagina 84) quando il dispositivo si trova nello stato **Idle**.

Il valore logico "Alto" è letto con tensione pari a $+24\text{VDC} \pm 10\%$.

L'uscita è comandata dal Master allo Slave attraverso **Control Word [0x2A]** (bit 13, si veda alla pagina 78) quando il dispositivo si trova nello stato **Idle**.

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

Esempio schema di collegamento:



6.4 Distanza_giro [0x00], Velocità Jog [0x0C], Velocità di lavoro [0x0D], Valore di preset [0x16-0x17], Delta spazio positivo [0x08-0x09] e Delta spazio negativo [0x0A-0x0B]

I dati macchina **Distanza_giro [0x00]**, **Velocità Jog [0x0C]**, **Velocità di lavoro [0x0D]**, **Valore di preset [0x16-0x17]**, **Delta spazio positivo [0x08-0x09]** e **Delta spazio negativo [0x0A-0x0B]** sono strettamente relazionati tra loro; si deve quindi prestare particolare attenzione quando si procede alla modifica anche di uno solo di essi.

La sequenza corretta per la modifica dei parametri è la seguente:

- impostazione del valore di **Distanza_giro [0x00]** (registro 1, si veda alla pagina 66);
- impostazione del valore di **Velocità Jog [0x0C]** (registro 13, si veda alla pagina 70);
- impostazione del valore di **Velocità di lavoro [0x0D]** (registro 14, si veda alla pagina 71);
- impostazione di **Valore di preset [0x16-0x17]** (registri 23-24, si veda alla pagina 73);
- verifica della correttezza di **Delta spazio positivo [0x08-0x09]** (registri 9-10, si veda alla pagina 68);
- verifica della correttezza di **Delta spazio negativo [0x0A-0x0B]** (registri 11-12, si veda alla pagina 69);
- salvataggio dei nuovi valori (**Salva parametri** bit 9 del registro **Control Word [0x2A]**, si veda alla pagina 77).



ATTENZIONE

A ogni modifica del parametro **Distanza_giro [0x00]** si devono poi reimpostare **Velocità Jog [0x0C]** e **Velocità di lavoro [0x0D]** in quanto le velocità sono espresse in impulsi al secondo. Nel calcolo delle velocità si deve sempre rispettare la seguente relazione:

$$\frac{vel_{\min} * Distanza / giro}{1024} \leq Velocità \leq \frac{vel_{\max} * Distanza / giro}{1024}$$

dove:

- **Distanza/giro**: nuovo valore di **Distanza_giro [0x00]** impostato dall'utilizzatore ed espresso in impulsi
- **vel_{min}**: velocità minima 1 [imp/s] per tutti i dispositivi RD1xA
- **vel_{max}**: velocità massima
 - 4266 [imp/s] per RD1xA-...-T12-...
 - 2133 [imp/s] per RD1xA-...-T24-...
 - 1066 [imp/s] per RD1xA-...-T48-...
 - 556 [imp/s] per RD1xA-...-T92-...

- **1024**: è il valore massimo ammesso per **Distanza_giro [0x00]** (espresso in impulsi).

Dopo ogni modifica del parametro **Distanza_giro [0x00]** si deve reimpostare anche **Valore di preset [0x16-0x17]** in modo da definire lo zero asse in quanto il sistema di riferimento è variato.

Dopo la modifica del parametro **Valore di preset [0x16-0x17]** 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 di **Delta spazio positivo [0x08-0x09]** e **Delta spazio negativo [0x0A-0x0B]** impostati.

Il numero di giri gestiti dal dispositivo è di 511 in direzione negativa e 511 in direzione positiva rispetto al preset.

Il valore del parametro **Delta spazio positivo [0x08-0x09]** sommato a **Valore di preset [0x16-0x17]** definisce il massimo spostamento in avanti (positivo) rispetto al preset (valore espresso in impulsi).

Il valore del parametro **Delta spazio negativo [0x0A-0x0B]** sottratto a **Valore di preset [0x16-0x17]** definisce il massimo spostamento all'indietro (negativo) rispetto al preset (valore espresso in impulsi).



ATTENZIONE

Si badi inoltre che i parametri di seguito elencati sono tutti espressi in relazione al parametro **Distanza_giro [0x00]**; di conseguenza la modifica del valore nel parametro **Distanza_giro [0x00]** comporta necessariamente una ridefinizione dei valori da essi espressi. I parametri sono: **Tolleranza di posizione [0x01]**, **Max errore di inseguimento [0x03]**, **Accelerazione [0x06]**, **Decelerazione [0x07]**, **Delta spazio positivo [0x08-0x09]**, **Delta spazio negativo [0x0A-0x0B]**, **Target position [0x2B-0x2C]**, **Posizione corrente [0x02-0x03]**, **Velocità corrente [0x04]** e **Errore di inseguimento [0x05-0x06]**. Si veda per esempio la relazione che intercorre tra **Distanza_giro [0x00]** e i valori di velocità, illustrata alla pagina precedente.



Esempio 1

Valori di default:

Distanza_giro [0x00] = 1024 impulsi/giro

Velocità di lavoro [0x0D] massima:

= 4266 impulsi/secondo per RD1xA-...T12-... ($4266 \cdot 1024 / 1024 = 4266$)

= 2133 impulsi/secondo per RD1xA-...T24-... ($2133 \cdot 1024 / 1024 = 2133$)

= 1066 impulsi/secondo per RD1xA-...T48-... ($1066 \cdot 1024 / 1024 = 1066$)

= 556 impulsi/secondo per RD1xA-...T92-... ($556 \cdot 1024 / 1024 = 556$)

Valore di preset [0x16-0x17] = 0

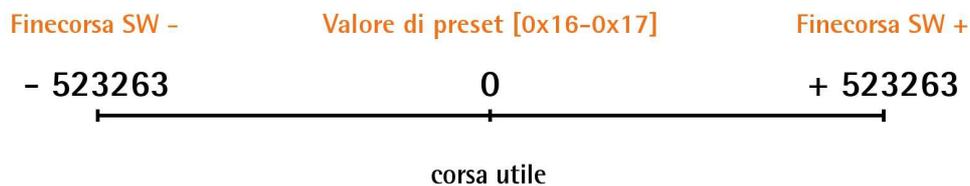
Delta spazio positivo [0x08-0x09] e **Delta spazio negativo [0x0A-0x0B]**

massimi = $523263 = (1024 \text{ impulsi/giro} \times 511 \text{ giri}) - 1$ con **Valore di preset [0x16-0x17]** = 0

Finecorsa SW + massimo = $0 + 523263 = + 523263$ impulsi (in avanti)

Finecorsa SW - massimo = $0 - 523263 = - 523263$ impulsi (indietro).

Con **Valore di preset [0x16-0x17]** = 0, la corsa utile dell'asse sarà perciò compresa tra i due limiti **Finecorsa SW +** massimo + 523263 e **Finecorsa SW -** massimo - 523263.





Esempio 2

L'unità RD1xA-...T12-... è montata su una vite senza fine con passo 1 mm e si desidera mantenere una risoluzione al centesimo di millimetro.

Distanza_giro [0x00] = 100 impulsi/giro

Velocità di lavoro [0x0D] massima = 417 impulsi/secondo ($4266 \cdot 100 / 1024 = 417$, arrotondato al numero intero)

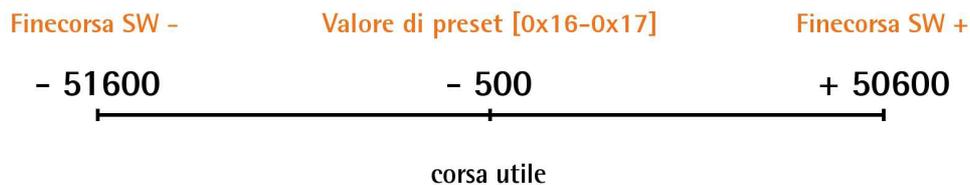
Valore di preset [0x16-0x17] = -500 (es. spessore utensile)

Delta spazio positivo [0x08-0x09] e **Delta spazio negativo [0x0A-0x0B]** massimi = 100 impulsi/giro x 511 giri = 51100 impulsi

Finecorsa SW + massimo = (-500) + 51100 = 50600 impulsi (in avanti)

Finecorsa SW - massimo = (-500) - 51100 = -51600 impulsi (indietro)

Con **Valore di preset [0x16-0x17]** = - 500, la corsa utile dell'asse sarà perciò compresa tra i due limiti **Finecorsa SW +** massimo + 50600 e **Finecorsa SW -** massimo - 51600.



7 Interfaccia Modbus®

Le unità ROTADRIVE Lika sono dispositivi Slave e implementano il protocollo applicativo Modbus (livello OSI 7) e il protocollo "Modbus over Serial Line" (livelli OSI 1 & 2).

Per ogni informazione e specifica omessa fare riferimento ai documenti "Modbus Application Protocol Specification V1.1b" e "Modbus over Serial Line. Specification and Implementation Guide V1.02" disponibili sul sito www.modbus.org.

7.1 Principi guida del protocollo Modbus Master / Slave

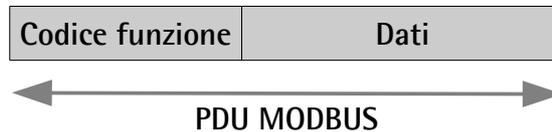
Il protocollo seriale Modbus è un protocollo Master – Slave. Un solo Master può essere connesso contemporaneamente alla rete Modbus; mentre il numero di Slave connessi allo stesso bus seriale può essere compreso tra 1 e 247. Una comunicazione Modbus è sempre iniziata dal Master. I nodi Slave non sono 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:

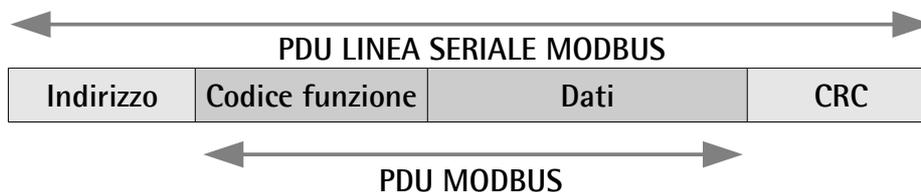
- **modalità UNICAST:** il Master invia la richiesta a un singolo Slave. Dopo aver ricevuto e processato la richiesta, lo Slave invia una risposta al Master. In questa modalità, la transazione Modbus consiste di due messaggi: una richiesta da parte del Master e una risposta da parte dello Slave. Ogni 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".

7.2 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 "4.4.1 Indirizzo nodo: Node ID (Figura 5)" a pagina 23), 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 una campo **DATI** che contiene i parametri di interrogazione e risposta. Per maggiori informazioni sui codici funzione implementati riferirsi alla sezione "7.4 Codici funzione" a pagina 56.
- **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 "7.4 Codici funzione" a pagina 56).
- **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".

7.3 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.

7.3.1 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. 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 |
|-------|---|---|---|---|---|---|---|---|---------|------|

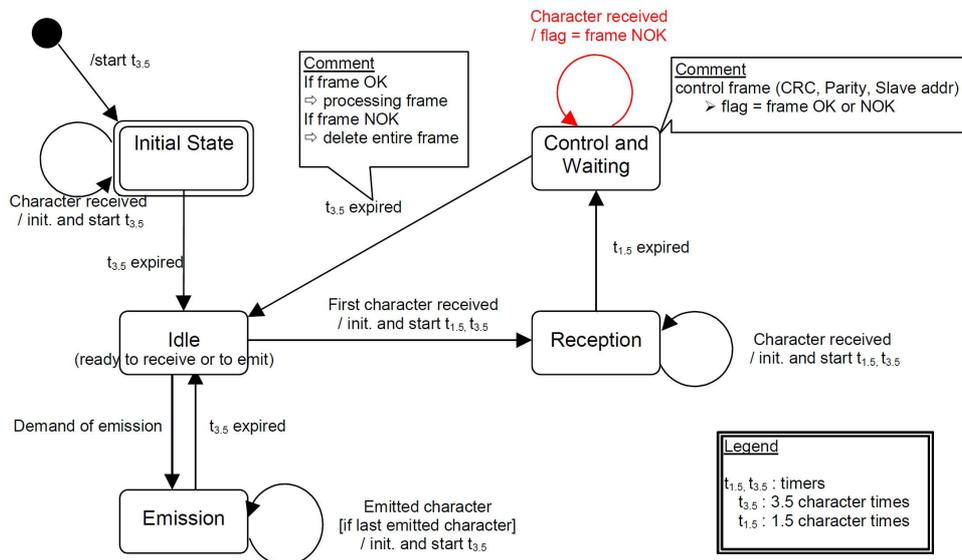
* Con "No parity", il bit di parità è sostituito da un bit di stop.

Come default bisogna impostare il parity bit = even.

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 | 2 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.

7.4 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 funzioni 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.

7.4.1 Codici funzione implementati

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

03 Read Holding Registers

FC = 03 (Hex = 0x03) r0

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 Registers**, si veda la sezione "8.1.1 Parametri Dati macchina" a pagina 66.

Request PDU

| | | |
|--------------------|--------|--------------------|
| Codice funzione | 1 byte | 0x03 |
| Indirizzo iniziale | 2 byte | da 0x0000 a 0xFFFF |
| Numero di registri | 2 byte | da 1 a 125 (0x7D) |

Response PDU

| | | |
|---------------------|-------------|-------------|
| Codice funzione | 1 byte | 0x03 |
| 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 | 0x83 (=0x03 + 0x80) |
| Codice eccezione | 1 byte | 01 o 02 o 03 o 04 |



Esempio di richiesta di lettura dei parametri **Accelerazione [0x06]** (registro 7) e **Decelerazione [0x07]** (registro 8).

| Request | | Response | |
|-----------------------|-----------|----------------------|-----------|
| Nome campo | (Hex) | Nome campo | (Hex) |
| Codice funzione | 03 | Codice funzione | 03 |
| Indirizzo iniziale Hi | 00 | Numero byte | 04 |
| Indirizzo iniziale Lo | 06 | Valore registro 7 Hi | 03 |
| Numero di registri Hi | 00 | Valore registro 7 Lo | E8 |
| Numero di registri Lo | 02 | Valore registro 8 Hi | 05 |
| | | Valore registro 8 Lo | DC |

Come si evince dalla tabella, il parametro **Accelerazione [0x06]** (registro 7) ha valore 03 E8 hex, cioè 1000 in notazione decimale; il parametro **Decelerazione [0x07]** (registro 8) ha valore 05 DC hex, cioè 1500 in notazione decimale.

Il frame completo di richiesta di lettura dei parametri **Accelerazione [0x06]** (registro 7) e **Decelerazione [0x07]** (registro 8) allo Slave con indirizzo 1 è il seguente:

Request PDU (in formato esadecimale)

[01][03][00][06][00][02][24][0A]

dove:

[01] = indirizzo dello Slave

[03] = codice funzione **03 Read Holding Registers**

[00][06] = indirizzo iniziale (parametro **Accelerazione [0x06]**, registro 7)

[00][02] = numero di registri richiesti

[24][0A] = CRC

Il frame completo di invio dei valori nei parametri **Accelerazione [0x06]** (registro 7) e **Decelerazione [0x07]** (registro 8) da parte dello Slave con indirizzo 1 è il seguente:

Response PDU (in formato esadecimale)

[01][03][04][03][E8][05][DC][78][8A]

dove:

[01] = indirizzo dello Slave

[03] = codice funzione **03 Read Holding Registers**

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

[03][E8] = valore del registro 7 **Accelerazione [0x06]**, 03 E8 hex = 1000 dec

[05][DC] = valore del registro 8 **Decelerazione [0x07]**, 05 DC hex = 1500 dec

[78][8A] = 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 "8.1.2 Parametri Input Register" a pagina 80.

Request PDU

| | | |
|--------------------|--------|--------------------|
| Codice funzione | 1 byte | 0x04 |
| Indirizzo iniziale | 2 byte | da 0x0000 a 0xFFFF |
| Numero di registri | 2 byte | da 0x0000 a 0x007D |

Response PDU

| | | |
|---------------------|-------------|-------------|
| Codice funzione | 1 byte | 0x04 |
| 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 | 0x84 (=0x04 + 0x80) |
| Codice eccezione | 1 byte | 01 o 02 o 03 o 04 |



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

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

Come si evince dalla tabella, il parametro **Posizione corrente [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 **Posizione corrente [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 **Posizione corrente [0x02-0x03]**, registro 3)

[00][02] = numero di registri richiesti

[D0][0B] = CRC

Il frame completo di invio dei valori del parametro **Posizione corrente [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 **Posizione corrente [0x02-0x03]**, 00 00 hex = 0 dec

[2F][F0] = valore del registro 4 **Posizione corrente [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 "8.1.1 Parametri Dati macchina" a pagina 66.

Request PDU

| | | |
|------------------------|--------|--------------------|
| Codice funzione | 1 byte | 0x06 |
| Indirizzo del registro | 2 byte | da 0x0000 a 0xFFFF |
| Valore del registro | 2 byte | da 0x0000 a 0xFFFF |

Response PDU

| | | |
|------------------------|--------|--------------------|
| Codice funzione | 1 byte | 0x06 |
| 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 | 0x86 (=0x06 + 0x80) |
| Codice eccezione | 1 byte | 01 o 02 o 03 o 04 |



Esempio di scrittura del valore 05 DC hex (= 1500 dec) nel parametro **Accelerazione [0x06]** (registro 7).

| Request | | Response | |
|------------------------|-----------|------------------------|-----------|
| Nome campo | (Hex) | Nome campo | (Hex) |
| Codice funzione | 06 | Codice funzione | 06 |
| Indirizzo registro Hi | 00 | Indirizzo registro Hi | 00 |
| Indirizzo registro Lo | 06 | Indirizzo registro Lo | 06 |
| Valore del registro Hi | 05 | Valore del registro Hi | 05 |
| Valore del registro Lo | DC | Valore del registro Lo | DC |

Come si evince dalla tabella, nel parametro **Accelerazione [0x06]** (registro 7) si imposta il valore 05 DC hex, cioè 1500 in notazione decimale.

Il frame completo di richiesta di scrittura del valore 05 DC hex (= 1500 dec) nel parametro **Accelerazione [0x06]** (registro 7) dello Slave con indirizzo 1 è il seguente:

Request PDU (in formato esadecimale)

[01][06][00][06][05][DC][6B][02]

dove:

[01] = indirizzo dello Slave

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

[00][06] = indirizzo del registro (parametro **Accelerazione [0x06]**, registro 7)

[05][DC] = valore da impostare nel registro

[6B][02] = CRC

Il frame completo di risposta alla richiesta di scrittura nel parametro **Accelerazione [0x06]** (registro 7) da parte dello Slave con indirizzo 1 è il seguente:

Response PDU (in formato esadecimale)

[01][06][00][06][05][DC][6B][02]

dove:

[01] = indirizzo dello Slave

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

[00][06] = indirizzo del registro (parametro **Accelerazione [0x06]**, registro 7)

[05][DC] = valore impostato nel registro

[6B][02] = CRC

16 Write Multiple Registers

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 Registers**, si veda la sezione "8.1.1 Parametri Dati macchina" a pagina 66.

Request PDU

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

*N = Numero di registri

Response PDU

| | | |
|--------------------|--------|--------------------|
| Codice funzione | 1 byte | 0x10 |
| 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 | 0x90 (= 0x10 + 0x80) |
| Codice eccezione | 1 byte | 01 o 02 o 03 o 04 |



Esempio di richiesta di scrittura dei valori 1500 e 1000 nei parametri **Accelerazione [0x06]** (registro 7) e **Decelerazione [0x07]** (registro 8).

| Request | | Response | |
|-----------------------|-----------|-----------------------|-----------|
| Nome campo | (Hex) | Nome campo | (Hex) |
| Codice funzione | 10 | Codice funzione | 10 |
| Indirizzo iniziale Hi | 00 | Indirizzo iniziale Hi | 00 |
| Indirizzo iniziale Lo | 06 | Indirizzo iniziale Lo | 06 |
| Numero di registri Hi | 00 | Numero di registri Hi | 00 |
| Numero di registri Lo | 02 | Numero di registri Lo | 02 |
| Numero byte | 04 | | |
| Valore registro 7 Hi | 05 | | |

| | |
|----------------------|-----------|
| Valore registro 7 Lo | DC |
| Valore registro 8 Hi | 03 |
| Valore registro 8 Lo | E8 |

Come si evince dalla tabella, nel parametro **Accelerazione [0x06]** (registro 7) si imposta il valore 05 DC hex, cioè 1500 in notazione decimale; nel parametro **Decelerazione [0x07]** (registro 8) si imposta il valore 03 E8 hex, cioè 1000 in notazione decimale.

Il frame completo di richiesta di scrittura dei valori 1500 e 1000 nei parametri **Accelerazione [0x06]** (registro 7) e **Decelerazione [0x07]** (registro 8) dello Slave con indirizzo 1 è il seguente:

Request PDU (in formato esadecimale)

[01][10][00][06][00][02][04][05][DC][03][E8][B2][0D]

dove:

[01] = indirizzo dello Slave

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

[00][06] = indirizzo iniziale (parametro **Accelerazione [0x06]**, registro 7)

[00][02] = numero di registri richiesti

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

[05][DC] = valore da impostare nel registro 7 **Accelerazione [0x06]**, 05 DC hex = 1500 dec

[03][E8] = valore da impostare nel registro 8 **Decelerazione [0x07]**, 03 E8 hex = 1000 dec

[B2][0D] = CRC

Il frame completo di risposta alla richiesta di impostazione dei valori 1000 e 1500 nei parametri **Accelerazione [0x06]** (registro 7) e **Decelerazione [0x07]** (registro 8) da parte dello Slave con indirizzo 1 è il seguente:

Response PDU (in formato esadecimale)

[01][10][00][06][00][02][A1][C9]

dove:

[01] = indirizzo dello Slave

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

[00][06] = indirizzo iniziale (parametro **Accelerazione [0x06]**, registro 7)

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

[A1][C9] = CRC



NOTA

Per ulteriori esempi si veda anche alla sezione "Esempi di programmazione" a pagina 90.



IMPORTANTE

Per motivi di sicurezza, durante il movimento dell'unità ROTADRIVE 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 **Abilitazione Watch dog** 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 Parametri di programmazione

8.1 Parametri disponibili

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

Nome parametro [Indirizzo registro]

[Numero registro, tipo variabile, attributo]

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

8.1.1 Parametri Dati macchina

I parametri **Dati macchina** sono accessibili sia in lettura che scrittura; per leggere il valore di un parametro utilizzare il codice funzione **03 Read Holding Registers** (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 Registers** (scrittura di più registri); per ogni informazione sui codici funzione implementati riferirsi alla sezione "7.4.1 Codici funzione implementati" a pagina 56.

Distanza_giro [0x00]

[Registro 1, Unsigned16, rw]

Questo parametro definisce 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 **Distanza_giro [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 i parametri **Velocità Jog [0x0C]**, **Velocità di lavoro [0x0D]** e **Valore di preset [0x16-0x17]**. Per maggiori informazioni riferirsi a pagina 47 e ai rispettivi parametri.

Si badi inoltre che i parametri di seguito elencati sono tutti espressi in relazione al parametro **Distanza_giro [0x00]**; di conseguenza la modifica del valore nel

parametro **Distanza_giro [0x00]** comporta necessariamente una ridefinizione dei valori da essi espressi. I parametri sono: **Tolleranza di posizione [0x01]**, **Max errore di inseguimento [0x03]**, **Accelerazione [0x06]**, **Decelerazione [0x07]**, **Delta spazio positivo [0x08-0x09]**, **Delta spazio negativo [0x0A-0x0B]**, **Target position [0x2B-0x2C]**, **Posizione corrente [0x02-0x03]**, **Velocità corrente [0x04]** e **Errore di inseguimento [0x05-0x06]**. Si veda per esempio la relazione che intercorre tra **Distanza_giro [0x00]** e i valori di velocità, illustrata alla pagina 71.



NOTA

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

Tolleranza di posizione [0x01]

[Registro 2, Unsigned16, rw]

Questo parametro definisce la finestra di tolleranza da applicare al valore di **Target position [0x2B-0x2C]**. Se l'asse è all'interno di questa finestra per il tempo impostato nel parametro **Tempo asse in tolleranza [0x02]**, allora è segnalata la condizione attraverso il bit di stato **Asse in posizione**. Il parametro è espresso in impulsi.

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

Tempo asse in tolleranza [0x02]

[Registro 3, Unsigned16, rw]

Rappresenta il tempo di assestamento dell'asse all'interno della finestra di tolleranza definita in **Tolleranza di posizione [0x01]** dopo il quale viene dichiarata la condizione attraverso il bit di stato **Asse in posizione**. Il parametro è espresso in millisecondi.

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

Max errore di inseguimento [0x03]

[Registro 4, Unsigned16, 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 **Errore di inseguimento** e il dispositivo arresta il proprio movimento. Il parametro è espresso in impulsi.

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

Kp anello di posizione [0x04]

[Registro 5, 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 = 400 (min. = 0, max. = 1000)

Ki anello di posizione [0x05]

[Registro 6, 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 = 100 (min. = 0, max. = 1000)

Accelerazione [0x06]

[Registro 7, Unsigned16, rw]

Questo parametro definisce il valore di accelerazione usato dal dispositivo. Il parametro è espresso in [impulsi/sec²].

Default = 5000 per RD1xA-...-T12-... (min. = 100, max. = 10000)

Default = 2500 per RD1xA-...-T24-... (min. = 100, max. = 10000)

Default = 1000 per RD1xA-...-T48-... (min. = 100, max. = 10000)

Default = 500 per RD1xA-...-T92-... (min. = 100, max. = 10000)

Decelerazione [0x07]

[Registro 8, Unsigned16, rw]

Questo parametro definisce il valore di decelerazione usato dal dispositivo. Il parametro è espresso in [impulsi/sec²].

Default = 5000 per RD1xA-...-T12-... (min. = 100, max. = 10000)

Default = 2500 per RD1xA-...-T24-... (min. = 100, max. = 10000)

Default = 1000 per RD1xA-...-T48-... (min. = 100, max. = 10000)

Default = 500 per RD1xA-...-T92-... (min. = 100, max. = 10000)

Delta spazio positivo [0x08-0x09]

[Registri 9-10, Unsigned32, rw]

È il valore utilizzato per calcolare il massimo spostamento in avanti (positivo) rispetto al Preset. Se si raggiunge il massimo spostamento in avanti viene attivata la segnalazione sul bit di stato **Finecorsa SW +**. Il parametro è espresso in impulsi encoder.

Finecorsa SW + = Valore di preset [0x16-0x17] + Delta spazio positivo [0x08-0x09].

Default = 523263 (min. = 0, max. = 523263)



ATTENZIONE

Si badi che il valore massimo di volta in volta ammesso in questo parametro tiene conto dello scaling impostato.



ESEMPIO

Con **Distanza_giro [0x00]** = 1024 e **Valore di preset [0x16-0x17]** = 0, il valore massimo ammesso in **Delta spazio positivo [0x08-0x09]** sarà:

$(1024 \text{ imp./giro} * 511 \text{ giri}) - 1 = 523263$

Con **Distanza_giro [0x00]** = 256 e **Valore di preset [0x16-0x17]** = 0, il valore massimo ammesso in **Delta spazio positivo [0x08-0x09]** sarà:

$(256 \text{ imp./giro} * 511 \text{ giri}) - 1 = 130815$

Si vedano anche gli ulteriori esempi al paragrafo "6.4 Distanza_giro [0x00], Velocità Jog [0x0C], Velocità di lavoro [0x0D], Valore di preset [0x16-0x17], Delta spazio positivo [0x08-0x09] e Delta spazio negativo [0x0A-0x0B]" a pagina 47.



ATTENZIONE

I valori di finecorsa devono essere verificati ogniqualvolta vengono modificati i parametri **Distanza_giro [0x00]** e **Valore di preset [0x16-0x17]**. Dopo ogni modifica del parametro **Distanza_giro [0x00]** si deve reimpostare anche **Valore di preset [0x16-0x17]** in modo da definire lo zero asse in quanto il sistema di riferimento è variato. Dopo la modifica del parametro **Valore di preset [0x16-0x17]** 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 di **Delta spazio positivo [0x08-0x09]** e **Delta spazio negativo [0x0A-0x0B]** impostati. Per una descrizione dettagliata si veda a pagina 47.

Delta spazio negativo [0x0A-0x0B]

[Registri 11-12, Unsigned32, rw]

E' il valore utilizzato per calcolare il massimo spostamento all'indietro (negativo) rispetto al Preset. Se si raggiunge il massimo spostamento all'indietro viene attivata la segnalazione sul bit di stato **Fincorsa SW -**. Il parametro è espresso in impulsi encoder.

Fincorsa SW - = Valore di preset [0x16-0x17] - Delta spazio negativo [0x0A-0x0B].

Default = 523263 (min. = 0, max. = 523263)



ATTENZIONE

Si badi che il valore massimo di volta in volta ammesso in questo parametro tiene conto dello scaling impostato.



ESEMPIO

Con **Distanza_giro [0x00]** = 1024 e **Valore di preset [0x16-0x17]** = 0, il valore massimo ammesso in **Delta spazio negativo [0x0A-0x0B]** sarà:
 $(1024 \text{ imp./giro} * 511 \text{ giri}) - 1 = 523263$

Con **Distanza_giro [0x00]** = 256 e **Valore di preset [0x16-0x17]** = 0, il valore massimo ammesso in **Delta spazio negativo [0x0A-0x0B]** sarà:
 $(256 \text{ imp./giro} * 511 \text{ giri}) - 1 = 130815$

Si vedano anche gli ulteriori esempi al paragrafo "6.4 Distanza_giro [0x00], Velocità Jog [0x0C], Velocità di lavoro [0x0D], Valore di preset [0x16-0x17], Delta spazio positivo [0x08-0x09] e Delta spazio negativo [0x0A-0x0B]" a pagina 47.



ATTENZIONE

I valori di finecorsa devono essere verificati ogniqualvolta vengono modificati i parametri **Distanza_giro [0x00]** e **Valore di preset [0x16-0x17]**. Dopo ogni modifica del parametro **Distanza_giro [0x00]** si deve reimpostare anche **Valore di preset [0x16-0x17]** in modo da definire lo zero asse in quanto il sistema di riferimento è variato. Dopo la modifica del parametro **Valore di preset [0x16-0x17]** 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 di **Delta spazio positivo [0x08-0x09]** e **Delta spazio negativo [0x0A-0x0B]** impostati. Per una descrizione dettagliata si veda a pagina 47.

Velocità Jog [0x0C]

[Registro 13, Unsigned16, rw]

Questo parametro definisce la velocità massima del dispositivo nell'utilizzo delle funzioni **Jog +** e **Jog -**. Il parametro è espresso in impulsi/secondo.

Default = 4266 per RD1xA-...-T12-... (min. = 1, max. = 4266)

Default = 2133 per RD1xA-...-T24-... (min. = 1, max. = 2133)

Default = 1066 per RD1xA-...-T48-... (min. = 1, max. = 1066)

Default = 556 per RD1xA-...-T92-... (min. = 1, max. = 556)

Velocità di lavoro [0x0D]

[Registro 14, Unsigned16, rw]

Questo parametro definisce la velocità massima del dispositivo utilizzato in modalità automatica (movimenti controllati mediante il bit di comando **Start** per il raggiungimento della quota impostata in **Target position [0x2B-0x2C]**).

Il parametro è espresso in impulsi/secondo.

Default = 4266 per RD1xA-...-T12-... (min. = 1, max. = 4266)

Default = 2133 per RD1xA-...-T24-... (min. = 1, max. = 2133)

Default = 1066 per RD1xA-...-T48-... (min. = 1, max. = 1066)

Default = 556 per RD1xA-...-T92-... (min. = 1, max. = 556)



ATTENZIONE

A ogni modifica del parametro **Distanza_giro [0x00]** si devono poi reimpostare **Velocità Jog [0x0C]** e **Velocità di lavoro [0x0D]** in quanto le velocità sono espresse in impulsi al secondo. Nel calcolo delle velocità si deve sempre rispettare la seguente relazione:

$$\frac{vel_{min} * Distanza / giro}{1024} \leq Velocità \leq \frac{vel_{max} * Distanza / giro}{1024}$$

Per una spiegazione dettagliata si veda a pagina 47.

Durata corrente di stacco [0x0E]

[Registro 15, Unsigned16, rw]

Questo parametro definisce il tempo massimo per il quale il motore è controllato con la corrente di stacco durante la partenza (si veda al parametro **Corrente di stacco [0x13]**). Il parametro è espresso in millisecondi.

Default = 2000 (min. = 0, max. = 3000)

Direzione conteggio [0x0F]

[Registro 16, Unsigned16, rw]

Direzione di rotazione del motore per ottenere un incremento positivo della posizione. La direzione di rotazione è stabilita guardando il dispositivo dall'estremità dell'asse.

0 = rotazione oraria (default)

1 = rotazione antioraria



ATTENZIONE

La modifica di questo parametro influenza la posizione calcolata dal controllore. Si deve quindi reimpostare il parametro **Valore di preset [0x16-0x17]** e verificare i valori nei parametri **Delta spazio positivo [0x08-0x09]** e **Delta spazio negativo [0x0A-0x0B]**.

Kp anello di corrente [0x10]

[Registro 17, Unsigned16, rw]

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

Default = 200 (min. = 0, max. = 1000)

Ki anello di corrente [0x11]

[Registro 18, Unsigned16, rw]

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

Default = 30 (min. = 0, max. = 1000)

Corrente massima [0x12]

[Registro 19, Unsigned16, rw]

Questo parametro definisce la corrente massima erogata dall'elettronica di potenza per il controllo del motore. Il parametro è espresso in mA (milliampere). Il valore in questo parametro non può essere superiore al valore impostato in **Corrente di stacco [0x13]**.

Default = 2000 (min. = 10, max. = 2000)

Corrente di stacco [0x13]

[Registro 20, Unsigned16, rw]

Questo parametro definisce la corrente massima erogata solo alla partenza e per un tempo massimo definito alla variabile **Durata corrente di stacco [0x0E]**. Il parametro è espresso in mA (milliampere).

Default = 4000 (min. = 10, max. = 4000)

Valore di offset [0x14-0x15]

[Registri 21-22, Integer32, ro]

Questo parametro definisce la differenza tra la posizione trasmessa dal dispositivo e la posizione reale: posizione reale – preset. Il valore è espresso in impulsi.

Valore di preset [0x16-0x17]

[Registri 23-24, Integer32, rw]

Usare questo parametro per assegnare e impostare un valore di Preset. La funzione di Preset è utilizzata per assegnare un determinato valore a una posizione fisica dell'asse. La posizione fisica prescelta avrà perciò il valore assegnato in questo parametro e tutte le altre posizioni assumeranno un valore conseguente. Il valore di preset sarà assegnato alla posizione dell'asse al momento dell'invio del parametro.

Default = 0 (min. = -1048576, max. = 1048576)



ATTENZIONE

Il parametro **Valore di preset [0x16-0x17]** deve essere reimpostato ogniqualvolta viene modificato il parametro **Distanza_giro [0x00]**. Dopo la modifica del parametro **Valore di preset [0x16-0x17]** 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 di **Delta spazio positivo [0x08-0x09]** e **Delta spazio negativo [0x0A-0x0B]** impostati. Per una descrizione dettagliata si veda a pagina 47.

Rapporto di riduzione [0x18]

[Registro 25, Unsigned16, ro]

E' il rapporto di riduzione degli ingranaggi interni interposti tra il motore e l'asse del dispositivo. E' un valore di sola lettura.

Default = 12 per RD1xA-...-T12-...

Default = 24 per RD1xA-...-T24-...

Default = 48 per RD1xA-...-T48-...

Default = 92 per RD1xA-...-T92-...

Ampiezza passo jog [0x19]

[Registro 26, Unsigned16, rw]

Se è abilitato il controllo del jog a passo (bit 4 **Jog incrementale** 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; quindi lo slave si arresta in attesa di un nuovo comando.

Default = 100 (min. = 1, max. = 10000).

Registro comandi Extra [0x29]

[Registro 42, Unsigned16, rw]

Struttura byte del registro **Registro comandi Extra [0x29]**:

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

Byte 0

Lettura assoluta

bit 0 Questa funzione è riservata all'utilizzo dei soli tecnici di Lika Electronic.

bit 1 ... 7 Non utilizzati.

Byte 1 Non utilizzato.

Control Word [0x2A]

[Registro 43, Unsigned16, rw]

Contiene i comandi da inviare in tempo reale al dispositivo 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 **Jog incrementale** = 0, lo Slave si muove in direzione positiva per tutto il tempo in cui **Jog +** = 1; se invece il bit 4 **Jog incrementale** = 1 (abilitazione jog a passo), lo slave esegue un singolo passo in direzione positiva in corrispondenza del fronte di salita di **Jog +** la cui ampiezza, espressa in impulsi, è impostata al parametro **Ampiezza passo jog [0x19]**; poi si arresta in attesa di un nuovo comando. La velocità, l'accelerazione e la decelerazione sono definiti dai dati macchina **Velocità Jog [0x0C]**, **Accelerazione [0x06]** e **Decelerazione [0x07]**. Per una descrizione più dettagliata del controllo jog si veda a pagina 45.

Jog -

bit 1

Se il bit 4 **Jog incrementale** = 0, lo Slave si muove in direzione negativa per tutto il tempo in cui **Jog -** = 1; se invece il bit 4 **Jog incrementale** = 1 (abilitazione jog a passo), lo slave esegue un singolo passo in direzione negativa in corrispondenza del fronte di salita di **Jog -** la cui ampiezza, espressa in impulsi, è impostata al parametro **Ampiezza passo jog [0x19]**; poi si arresta in attesa di un nuovo comando. La velocità, l'accelerazione e la decelerazione sono definiti dai dati macchina **Velocità Jog [0x0C]**, **Accelerazione [0x06]** e **Decelerazione [0x07]**. Per una descrizione più dettagliata del controllo jog si veda a pagina 45.

Stop

bit 2

Se impostato a "=1" lo Slave è libero di eseguire i comandi di movimento ricevuti. Se durante il movimento questo bit diventa "=0" allora lo Slave si ferma seguendo la decelerazione prevista nel dato macchina **Decelerazione [0x07]**. Per un arresto immediato del movimento, utilizzare

il bit 7 **Emergenza**.

Reset allarmi

bit 3



Questo bit è normalmente impostato a "0". Lo stato normale del dispositivo è ristabilito nel cambio di stato da "0" a "1" di questo bit. Questo comando toglie lo Slave dalla condizione di allarme solo se non sono più presenti le condizioni che hanno causato l'errore.

Si badi che, se l'allarme è relativo a dati macchina non validi (si veda **Dati macchina non validi** e **Elenco DM errati [0x09-0x0A]**), si può tornare allo stato normale solo impostando dati macchina validi. L'allarme **Errore memoria flash** non è ripristinabile.

Jog incrementale

bit 4



Se il bit 4 "= 0", l'attivazione dei bit **Jog +** e **Jog -** procura il movimento manuale dello slave per tutto il tempo in cui **Jog + / Jog - = 1**. Impostando questo bit a 1 si abilita la funzione di jog a passo. 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 al parametro **Ampiezza passo jog [0x19]**; quindi lo slave si arresta in attesa di un nuovo comando.

Si badi che nell'utilizzo dei pulsanti manuali di jog (si veda "4.4.4 Pulsanti JOG + e JOG - (Figura 5)" a pagina 25) la funzione di jog incrementale è disabilitata; non è cioè possibile eseguire passi jog utilizzando i pulsanti manuali.

bit 5

Non utilizzato.

Start

bit 6

Se impostato a "=1" il dispositivo si muove allo scopo di raggiungere la posizione di target specificata (vedi **Target position [0x2B-0x2C]** a pagina 78). Per la descrizione del controllo di posizione si veda a pagina 44.

Emergenza

bit 7

Questo bit deve essere normalmente alto ("=1") altrimenti il dispositivo bloccherà istantaneamente ogni suo movimento. Per procurare un arresto non immediato, che utilizzi la decelerazione programmata, usare il bit 2 **Stop**.

Byte 1

Abilitazione Watch dog

bit 8 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 (attivazione dell'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.

Salva parametri

bit 9 Il salvataggio nella memoria non volatile dei parametri impostati è eseguito in corrispondenza del fronte di salita di questo bit; in altri termini il salvataggio dei valori impostati è richiesto alla variazione del bit dal livello logico basso (0) al livello logico alto (1).

Carica parametri di default

bit 10 Il caricamento dei parametri di default (parametri impostati durante la messa a punto in azienda del dispositivo che permettono un funzionamento standard e sicuro del dispositivo) è eseguito in corrispondenza del fronte di salita di questo bit; in altri termini il caricamento dei valori di default è richiesto alla variazione del bit dal livello logico basso (0) al livello logico alto (1). A pagina 96 è disponibile l'elenco dei dati macchina e il rispettivo valore di default preimpostato da Lika Electronic.

Esegui preset conteggio

bit 11 Il conteggio assume il valore scritto nella variabile **Valore di preset [0x16-0x17]**. L'operazione è eseguita in corrispondenza del fronte di salita di questo bit, cioè alla variazione del bit dal livello logico basso (0) al livello logico alto (1).

Asse in coppia

bit 12 Mantenimento asse in coppia a fine posizionamento. Funzione disponibile solo per la versione RD1A (versione senza freno), nella versione RD12A (versione con freno) il bit 12 non è gestito.

Se impostato "=0", con asse in posizione il PWM è disattivato.

Se impostato a "=1", con asse in posizione il PWM rimane attivo.

OUT 1

bit 13

Attiva / disattiva l'uscita digitale 1 del dispositivo. Il significato delle uscite è esplicitato nel capitolo "Parametri di programmazione" a pagina 66.

OUT 1 = 0 uscita 1 bassa (non attiva)

OUT 1 = 1 uscita 1 alta (attiva)

Sblocco freno

bit 14

Disponibile solo per la versione RD12A (versione con freno), nella versione RD1A (versione senza freno) il bit 14 non è gestito. Il modello RD12A è provvisto di un freno che, all'arresto del dispositivo, si attiva inibendo la possibilità di qualsiasi movimento dell'asse. Se si imposta questo bit "=1", il freno è disattivato; se si imposta "=0", il freno è gestito in modo automatico dal sistema.



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

bit 15

Non utilizzato.

Target position [0x2B-0x2C]

[Registri 44-45, Integer32, rw]

Questo parametro definisce la posizione di arrivo programmata. Il dispositivo si muove allo scopo di raggiungere questa posizione quando viene inviato il comando **Start**, i bit di **Stop** e **Emergenza** sono "=1" e il dispositivo non è in stato di allarme.



Funzione override di posizione

Durante il posizionamento è possibile modificare la posizione target; per fare ciò è sufficiente inviare nuovamente il comando **Start** con il nuovo valore in **Target position [0x2B-0x2C]**.



NOTA

Non è possibile abilitare le funzioni **Jog +**, **Jog -** e **Start** contemporaneamente. Per esempio: se viene inviato allo Slave il comando **Jog +** durante il movimento verso la posizione target, il comando di jog sarà ignorato; se si inviano i comandi **Jog +** e **Jog -** contemporaneamente il dispositivo non si muove o, se è già in movimento, arresta la sua corsa.

Solo con watch dog attivo (**Abilitazione Watch dog** in **Control Word [0x2A]** impostato a "=1"), se durante un movimento la comunicazione nella rete Modbus dovesse interrompersi (per esempio a causa della rottura del cavo), il dispositivo arresta ogni suo movimento e attiva il bit di allarme **Watch dog** (attivazione dell'allarme al ripristino della comunicazione con la rete Modbus).



NOTA

Per salvare i parametri modificati eseguire **Salva parametri**.
Nel caso di spegnimento del dispositivo i dati non salvati andranno persi!

8.1.2 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 "7.4.1 Codici funzione implementati" a pagina 56.

Registro allarmi [0x00]

[Registro 1, Unsigned16, ro]

Informa sugli allarmi presenti nel dispositivo.

Struttura byte allarmi:

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

Codici registro allarmi previsti:

Byte 0

Dati macchina non validi

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 in **Elenco DM errati [0x09-0x0A]**.

Errore memoria flash

bit 1 Errore interno non ripristinabile.

bit 2 Non utilizzato.

Errore di inseguimento

bit 3 La differenza tra la posizione reale e quella teorica è superiore al valore del parametro **Max errore di inseguimento [0x03]**; si consiglia di ridurre la velocità di lavoro.

Asse non sincronizzato

bit 4 Errore interno non ripristinabile.

Target non valido

bit 5 Posizione comandata oltre i finecorsa.

Emergenza

bit 6 Il bit 7 **Emergenza** in **Control Word [0x2A]** è stato forzato a 0; oppure sono attivi allarmi nel dispositivo.

Sovracorrente

bit 7 Il valore della corrente di alimentazione è superiore al massimo consentito.

Byte 1

Sovratemperatura

bit 8 Superamento della temperatura interna accettabile rilevata per mezzo di una sonda (si veda il parametro **Temperatura [0x08]**).

bit 9 Non utilizzato.

Sottotensione

bit 10 Il valore della tensione di alimentazione è inferiore al minimo consentito.

Watch dog

bit 11 Con Watch dog attivo (**Abilitazione Watch dog** in **Control Word [0x2A]** impostato a "=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 è segnalato 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.

bit 12 ... 15 Non utilizzati.

Per ripristinare una condizione di errore utilizzare il bit **Reset allarmi**. Questo bit è normalmente impostato a "0". Lo stato normale del dispositivo è ristabilito nel cambio di stato da "0" a "1" di questo bit. Questo comando toglie lo Slave dalla condizione di allarme solo se non sono più presenti le condizioni che hanno causato l'errore.



Si badi che, se l'allarme è relativo a dati macchina non validi (si veda **Dati macchina non validi** e **Elenco DM errati [0x09-0x0A]**), si può tornare allo

stato normale solo impostando dati macchina validi. L'allarme **Errore memoria flash** non è ripristinabile.

Status word [0x01]

[Registro 2, Unsigned16, ro]

Questo registro contiene le informazioni relative allo stato del dispositivo.

Struttura byte del registro **Status word [0x01]**:

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

Byte 0

Asse in posizione

bit 0

Se "=1" il dispositivo ha raggiunto la posizione programmata per il tempo definito al parametro **Tempo asse in tolleranza [0x02]**. Rimane attivo fino a quando l'errore di posizione è inferiore al valore impostato in **Tolleranza di posizione [0x01]**.

bit 1:

Non utilizzato.

Asse abilitato

bit 2

Riporta lo stato di abilitazione del motore. Il bit è "=1" quando il motore è abilitato, cioè il PWM è attivo e l'asse 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.

Finecorsa SW +

bit 3

Se "=1" il dispositivo ha raggiunto la posizione programmata come finecorsa positivo. Si veda il dato macchina **Delta spazio positivo [0x08-0x09]**.

Finecorsa SW -

bit 4

Se "=1" il dispositivo ha raggiunto la posizione programmata come finecorsa negativo. Si veda il dato macchina **Delta spazio negativo [0x0A-0x0B]**.

Allarme

bit 5

Se "=1" si è verificato un allarme, si vedano i dettagli in **Registro allarmi [0x00]** a pagina 80.

Asse in movimento

bit 6

Se "=0" il dispositivo è fermo.

Se "=1" il dispositivo è in movimento.

Comando in corso

bit 7

Se "=0" il controllore non sta eseguendo nessun comando.

Se "=1" il controllore sta eseguendo un comando.

Byte 1

Target raggiunto

bit 8

Se "=1" il dispositivo ha raggiunto la posizione programmata in **Target position [0x2B-0x2C]**. Rimane attivo fino ai successivi comandi **Target position [0x2B-0x2C]** o **Reset allarmi**.

Pulsante 1 Jog +

bit 9

L'unità RD1xA prevede l'installazione di tre pulsanti alloggiati all'interno del corpo del dispositivo e accessibili mediante la rimozione di un coperchio PG. Quando si preme il pulsante 1 JOG + il bit 9 è forzato alto "=1"; se il pulsante 1 non è attivo il bit è "=0". Per maggiori informazioni si veda la sezione "4.4 Selettori e pulsanti (Figura 5)" a pagina 22.

Pulsante 2 Jog -

bit 10

L'unità RD1xA prevede l'installazione di tre pulsanti alloggiati all'interno del corpo del dispositivo e accessibili mediante la rimozione di un coperchio PG. Quando si preme il pulsante 2 JOG - il bit 10 è forzato alto "=1"; se il pulsante 2 non è attivo il bit è "=0". Per maggiori informazioni si veda la sezione "4.4 Selettori e pulsanti (Figura 5)" a pagina 22.

Pulsante 3 Preset

bit 11

L'unità RD1xA prevede l'installazione di tre pulsanti alloggiati all'interno del corpo del dispositivo e accessibili mediante la rimozione di un coperchio PG. Quando si preme il pulsante 3 PRESET il bit 11 è forzato alto "=1"; se il pulsante 3 non è attivo il bit è "=0". Per maggiori informazioni si veda la sezione "4.4 Selettori e pulsanti (Figura 5)" a pagina 22.

Saturazione DAC

bit 12

La corrente erogata dall'elettronica di potenza per il controllo del motore ha raggiunto il livello massimo e non può essere ulteriormente aumentata.

IN 1

bit 13

Stato dell'ingresso digitale 1 del dispositivo. Il significato degli ingressi è esplicitato nel capitolo "Parametri di programmazione" a pagina 66.

IN 1 = 0 ingresso 1 basso (non attivo)

IN 1 = 1 ingresso 1 alto (attivo)

IN 2

bit 14

Stato dell'ingresso digitale 2 del dispositivo. Il significato degli ingressi è esplicitato nel capitolo "Parametri di programmazione" a pagina 66.

IN 2 = 0 ingresso 2 basso (non attivo)

IN 2 = 1 ingresso 2 alto (attivo)

IN 3

bit 15

Stato dell'ingresso digitale 3 del dispositivo. Il significato degli ingressi è esplicitato nel capitolo "Parametri di programmazione" a pagina 66.

IN 3 = 0 ingresso 3 basso (non attivo)

IN 3 = 1 ingresso 3 alto (attivo)

Posizione corrente [0x02-0x03]

[Registri 3-4, Integer32, ro]

Posizione del dispositivo al momento dell'invio della richiesta. Il valore è espresso in impulsi.

Velocità corrente [0x04]

[Registro 5, Integer16, ro]

Velocità del dispositivo espressa in impulsi/secondo [imp/s], aggiornata ogni secondo.

Errore di inseguimento [0x05-0x06]

[Registri 6-7, Integer32, ro]

Questa variabile contiene la differenza tra la posizione richiesta e la posizione attuale istante per istante. Se questo valore supera il dato macchina **Max errore**

di inseguimento [0x03] il dispositivo genera l'allarme **Errore di inseguimento** e il dispositivo interrompe il proprio movimento. Il valore è espresso in impulsi.

Corrente attuale [0x07]

[Registro 8, Integer16, ro]

Questa variabile rappresenta il valore della corrente assorbita dal motore. Il parametro è espresso in mA (milliampere).

Temperatura [0x08]

[Registro 9, Integer16, ro]

Questo registro contiene il valore della temperatura interna al dispositivo rilevata per mezzo di una sonda. Il parametro è espresso in °C (gradi Celsius). La temperatura minima rilevabile è di -20°C.

Elenco DM errati [0x09-0x0A]

[Registri 10-11, Unsigned32, ro]

L'operatore ha impostato valori non consoni e il sistema ha visualizzato il messaggio di allarme **Dati macchina non validi**. Questa variabile indica 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 | Nome parametro |
|-----|-----------------------------------|
| 1 | Distanza_giro [0x00] |
| 7 | Accelerazione [0x06] |
| 8 | Decelerazione [0x07] |
| 9 | Delta spazio positivo [0x08-0x09] |
| 10 | Delta spazio negativo [0x0A-0x0B] |
| 11 | Velocità Jog [0x0C] |
| 12 | Velocità di lavoro [0x0D] |
| 13 | Durata corrente di stacco [0x0E] |
| 14 | Direzione conteggio [0x0F] |
| 17 | Corrente massima [0x12] |
| 18 | Corrente di stacco [0x13] |

| | |
|----|------------------------------|
| 19 | Rapporto di riduzione [0x18] |
| 20 | Ampiezza passo jog [0x19] |
| 26 | Valore di preset [0x16-0x17] |

I2t [0x0B]

[Registro 12, Unsigned16, ro]

Immagine termica o segnale proporzionale alla corrente (solo per monitoraggio).

Dip-switch baud rate [0x0C]

[Registro 13, Unsigned16, ro]

Visualizza il valore della velocità di trasmissione dei dati (baud rate) della porta seriale del dispositivo impostato mediante il dip-switch previsto. Per ogni informazione sull'impostazione del baud rate riferirsi alla sezione "4.4.2 Velocità di trasmissione dei dati: Baud rate e bit di parità (Figura 5)" a pagina 24.

Dip-switch indirizzo [0x0D]

[Registro 14, Unsigned16, ro]

Visualizza l'indirizzo del nodo del dispositivo impostato mediante il dip-switch previsto. Per ogni informazione sull'impostazione dell'indirizzo del nodo riferirsi alla sezione "4.4.1 Indirizzo nodo: Node ID (Figura 5)" a pagina 23.

Versione software [0x0E]

[Registro 15, Unsigned16, ro]

Visualizza la versione software del dispositivo.

Il significato dei 16 bit che compongono il registro è il seguente:

| | | | | | | | | | | | | | | | |
|--------------|----|----|----|----|----|----|----|--------------|----|----|----|----|----|----|----|
| 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
| Ms bit | | | | | | | | Ls bit | | | | | | | |
| Numero major | | | | | | | | Numero minor | | | | | | | |

Il valore 01 02 hex, che corrisponde alla rappresentazione binaria 00000001 00000010, deve perciò essere interpretato come: versione 1.2.

Versione hardware [0x0F]

[Registro 16, Unsigned16, ro]

Visualizza la versione hardware del dispositivo.

Il significato dei 16 bit che compongono il registro è il seguente:

| | | | | | | | | | | | | | | | |
|-------------------|----|----|----|-------------|----|----|----|-------|----|----|----|-------------------|----|----|----|
| 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
| Modello ROTADRIVE | | | | Interfaccia | | | | Freno | | | | Versione hardware | | | |

dove:

| | |
|-----------|--|
| 00 ... 03 | = versione hardware |
| 04 ... 06 | = bit non utilizzati |
| 07 | = freno (0 = senza freno; 1 = con freno) |
| 08 ... 11 | = interfaccia (00 = Modbus; 01 = Profibus; 02 = CANopen; 03 ... 0F = non utilizzati) |
| 12 ... 15 | = modello ROTADRIVE (00 = RD4; 01 = RD1xA; 02 = RD5; 03 ... 0F = non utilizzati) |

Il valore 11 81 hex, che corrisponde alla rappresentazione binaria 00010001 10000001, deve perciò essere interpretato come: versione hardware 1 (bit 0 = 1); dispositivo con freno (bit 7 = 1); interfaccia Profibus (bit 8 = 1); modello RDx1A (bit 12 = 1).



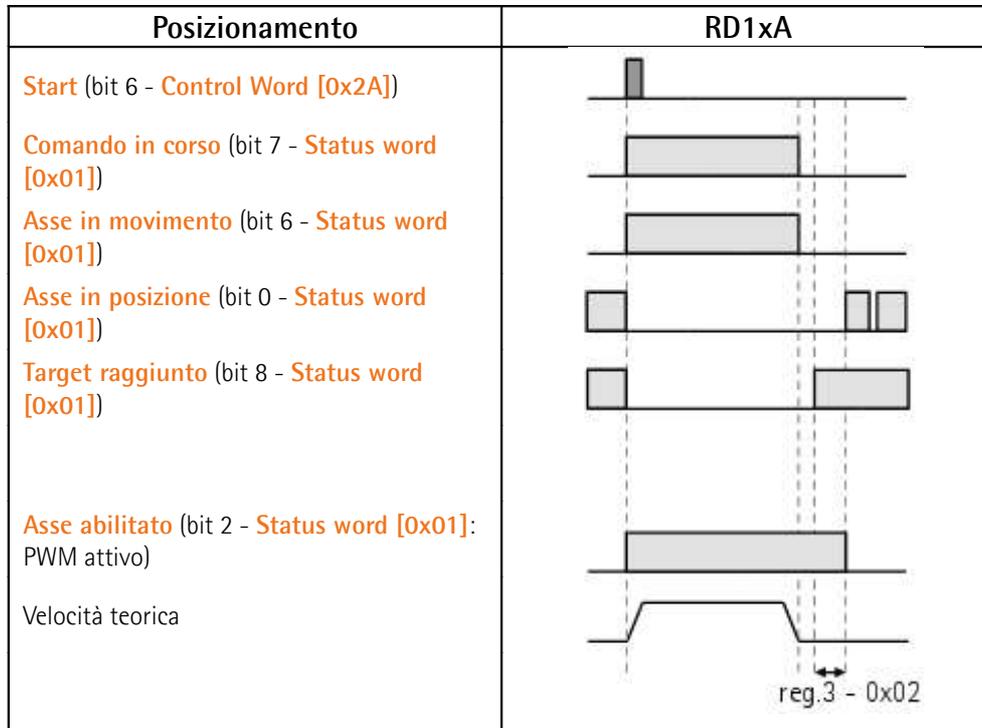
NOTA

Per salvare i parametri modificati eseguire **Salva parametri**.

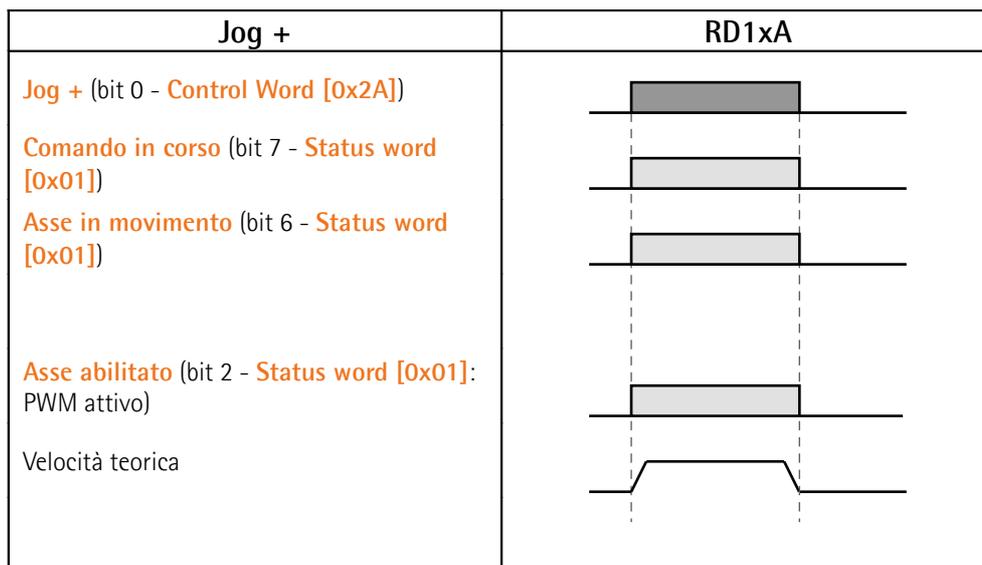
Nel caso di spegnimento del dispositivo i dati non salvati andranno persi!



Esempio 1



Esempio 2



8.2 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.

Per ogni informazione sul significato dei codici eccezione (exception code) fare riferimento alla sezione "MODBUS Exception Responses" alla pagina 48 del documento "MODBUS Application Protocol Specification V1.1b".

9 Esempi di programmazione

Di seguito sono riportati alcuni esempi di lettura e impostazione dei parametri. Tutti i valori sono espressi in notazione esadecimale.

9.1 Utilizzo del codice funzione 03 Read Holding Registers



Esempio 1

Richiesta di lettura dei parametri **Accelerazione [0x06]** (registro 7) e **Decelerazione [0x07]** (registro 8) allo Slave con indirizzo 1.

Request PDU

[01][03][00][06][00][02][24][0A]

dove:

[01] = indirizzo dello Slave

[03] = codice funzione **03 Read Holding Registers**

[00][06] = indirizzo iniziale (parametro **Accelerazione [0x06]**, registro 7)

[00][02] = numero di registri richiesti

[24][0A] = CRC

Response PDU

[01][03][04][03][E8][05][DC][78][8A]

dove:

[01] = indirizzo dello Slave

[03] = codice funzione **03 Read Holding Registers**

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

[03][E8] = valore del registro 7 **Accelerazione [0x06]**, 03 E8 hex = 1000 dec

[05][DC] = valore del registro 8 **Decelerazione [0x07]**, 05 DC hex = 1500 dec

[78][8A] = CRC

Il parametro **Accelerazione [0x06]** (registro 7) ha valore 03 E8 hex, cioè 1000 in notazione decimale; il parametro **Decelerazione [0x07]** (registro 8) ha valore 05 DC hex, cioè 1500 in notazione decimale.

9.2 Utilizzo del codice funzione 04 Read Input Register



Esempio 1

Richiesta di lettura del parametro **Posizione corrente [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 **Posizione corrente [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 **Posizione corrente [0x02-0x03]**, 00 00 hex = 0 dec

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

[E7][F0] = CRC

Il parametro **Posizione corrente [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 **Registro allarmi [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 **Registro allarmi [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 **Registro allarmi [0x00]**, 00 81 hex = 0000 0000
1000 0001 bin

[79][50] = CRC

Ne consegue che nella variabile **Registro allarmi [0x00]** (registro 1) sono attivi i bit 0 e 7, vale a dire (vedi a pagina 80): **Dati macchina non validi e Emergenza.**

9.3 Utilizzo del codice funzione **06 Write Single Register**



Esempio 1

Richiesta di scrittura del valore 05 DC hex (= 1500 dec) nel parametro **Accelerazione [0x06]** (registro 7) dello Slave con indirizzo 1.

Request PDU

[01][06][00][06][05][DC][6B][02]

dove:

[01] = indirizzo dello Slave

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

[00][06] = indirizzo del registro (parametro **Accelerazione [0x06]**, registro 7)

[05][DC] = valore da impostare nel registro

[6B][02] = CRC

Response PDU

[01][06][00][06][05][DC][6B][02]

dove:

[01] = indirizzo dello Slave

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

[00][06] = indirizzo del registro (parametro **Accelerazione [0x06]**, registro 7)

[05][DC] = valore impostato nel registro

[6B][02] = CRC

Nel parametro **Accelerazione [0x06]** (registro 7) viene impostato il valore 05 DC hex, cioè 1500 in notazione decimale.



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

Nella variabile **Control Word [0x2A]** (registro 43) viene impostato il valore 00 84 hex = 0000 0000 1000 0100 bin. In altri termini, sono forzati a 1 il bit di **Stop** (bit 2 = 1) e il bit di **Emergenza** (bit 7 = 1): il dispositivo è pronto 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

Nella variabile **Control Word [0x2A]** (registro 43) viene impostato il valore 0A 80 hex = 0000 0010 1000 0000 bin. In altri termini, il dispositivo è forzato in stop (bit di **Stop** 2 = 0), ma non in emergenza (bit di **Emergenza** 7 = 1) ed è richiesto il salvataggio dei parametri (bit **Salva parametri** 9 = 1).

9.4 Utilizzo del codice funzione **16 Write Multiple Registers**



Esempio 1

Richiesta di scrittura dei valori 1500 e 1000 nei parametri **Accelerazione [0x06]** (registro 7) e **Decelerazione [0x07]** (registro 8) dello Slave con indirizzo 1.

Request PDU

[01][10][00][06][00][02][04][05][DC][03][E8][B2][0D]

dove:

[01] = indirizzo dello Slave

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

[00][06] = indirizzo iniziale (parametro **Accelerazione [0x06]**, registro 7)

[00][02] = numero di registri richiesti

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

[05][DC] = valore da impostare nel registro 7 **Accelerazione [0x06]**, 05 DC hex = 1000 dec

[03][E8] = valore da impostare nel registro 8 **Decelerazione [0x07]**, 03 E8 hex = 1500 dec

[B2][0D] = CRC

Response PDU

[01][10][00][06][00][02][A1][C9]

dove:

[01] = indirizzo dello Slave

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

[00][06] = indirizzo iniziale (parametro **Accelerazione [0x06]**, registro 7)

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

[A1][C9] = CRC

Ne consegue che nel parametro **Accelerazione [0x06]** (registro 7) si imposta il valore 05 DC hex, cioè 1500 in notazione decimale; nel parametro **Decelerazione [0x07]** (registro 8) si imposta il valore 03 E8 hex, cioè 1000 in notazione decimale.

10 Tabella parametri di default

| Lista parametri | Valore di default | | |
|---|---|--|--|
| Distanza_giro [0x00] imp/giro | 1024 | | |
| Tolleranza di posizione [0x01] imp | 0 | | |
| Tempo asse in tolleranza [0x02] ms | 0 | | |
| Max errore di inseguimento [0x03] imp | 1024 | | |
| Kp anello di posizione [0x04] | 400 | | |
| Ki anello di posizione [0x05] | 100 | | |
| Accelerazione [0x06] imp/s ² | 5000 (RD1xA-...T12-...) 2500 (RD1xA-...T24-...) 1000 (RD1xA-...T48-...) 500 (RD1xA-...T92-...) | | |
| Decelerazione [0x07] imp/s ² | 5000 (RD1xA-...T12-...) 2500 (RD1xA-...T24-...) 1000 (RD1xA-...T48-...) 500 (RD1xA-...T92-...) | | |
| Delta spazio positivo [0x08-0x09] imp | 523263 | | |
| Delta spazio negativo [0x0A-0x0B] imp | 523263 | | |
| Velocità Jog [0x0C] imp/s | 4266 (RD1xA-...T12-...) 2133 (RD1xA-...T24-...) 1066 (RD1xA-...T48-...) 556 (RD1xA-...T92-...) | | |
| Velocità di lavoro [0x0D] imp/s | 4266 (RD1xA-...T12-...) 2133 (RD1xA-...T24-...) 1066 (RD1xA-...T48-...) 556 (RD1xA-...T92-...) | | |
| Durata corrente di stacco [0x0E] ms | 2000 | | |
| Direzione conteggio [0x0F] | 0 | | |
| Kp anello di corrente [0x10] | 200 | | |
| Ki anello di corrente [0x11] | 30 | | |
| Corrente massima [0x12] mA | 2000 | | |
| Corrente di stacco [0x13] mA | 4000 | | |
| Valore di preset [0x16-0x17] imp | 0 | | |
| Rapporto di riduzione [0x18] | 12 (RD1xA-...T12-...) 24 (RD1xA-...T24-...) 48 (RD1xA-...T48-...) 92 (RD1xA-...T92-...) | | |
| Ampiezza passo jog [0x19] P | 100 | | |

Pagina lasciata intenzionalmente bianca

Pagina lasciata intenzionalmente bianca

Pagina lasciata intenzionalmente bianca



| Versione HW-SW | Versione documento | Descrizione |
|--------------------------|--------------------|---|
| 1-1 | 1.0 | Prima stampa |
| 1-2 1-3 1-4 1-5 | 1.1 | Aggiunto Jog incrementale in Control Word [0x2A] e parametro Ampiezza passo jog [0x19]. Aggiornate definizioni jog. Aggiornata descrizione pulsanti jog. Aggiornata definizione Asse abilitato. Aggiornamento file Modbus.exe (V2.2/V2.3/V2.4). |
| 2-6 | 1.2 | Aggiornamento informazioni freno, aggiornamento file Modbus.exe (V2.5) |
| 3-0 | 1.3 | Aggiornamento sezione "Connessioni elettriche". Aggiornamento Kp anello di corrente [0x10]. |
| 3-1 3-2 3-3 | 1.4 | Introduzione modello con rapporto di riduzione -T92, aggiornamento file Modbus.exe (V2.7) |
| 3-3 | 1.5 | Aggiornamento informazione LED |
| 3-3 | 1.6 | Avvertenza per tensione controlettromotrice, aggiornamento registri Tolleranza di posizione [0x01], Max errore di inseguimento [0x03], Ampiezza passo jog [0x19] |



Smaltire separatamente

LIKA Electronic

Via S. Lorenzo, 25
36010 Carré (VI) • Italy
Tel. +39 0445 806600
Fax +39 0445 806699



Italy: eMail info@lika.it - www.lika.it
World: eMail info@lika.biz - www.lika.biz