



Manuale Utente

Regolatore per motore CC a spazzole

Codice RS: 434544

Codice RS: 206421

1. Designazione del prodotto

I controllori per motori a spazzole RS 434544 e RS 434546 sono dispositivi elettronici progettati per azionare e controllare motori a spazzole CC.

2. Funzioni e possibilità

I controllori sono progettati per controllare la velocità, l'accelerazione, la decelerazione e la direzione di rotazione del motore. Le unità forniscono anche il posizionamento in base ai segnali di un encoder in caso di utilizzo di un motore con sensori Hall.

Il motore è controllato da segnali esterni o da comandi trasmessi tramite RS-485 con il protocollo Modbus. Il controllore può anche funzionare secondo un algoritmo precedentemente registrato nella sua memoria da un utente.

Controllo tramite RS-485 Modbus

Il controllore può essere controllato a distanza tramite la linea di comunicazione fisica RS-485 utilizzando il protocollo industriale Modbus:

- L'impostazione viene eseguita scrivendo o leggendo i parametri corrispondenti nei/dai registri del controllore.
- I protocolli supportati sono RTU e ASCII, le velocità sono 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 57600, 115200, 128000 baud.
- Vengono implementati movimenti verso una posizione preimpostata di destinazione o uno dei quattro punti di riferimento predefiniti.
- Il controllore può funzionare autonomamente sotto il controllo di un programma utente (fino a 1024 comandi), che è pre-registrato in una memoria non volatile. Sono supportati salti condizionali e incondizionali (relativi e assoluti), chiamate a subroutine, cicli e timer.
- Gli ingressi programmabili IN1 e IN2 possono essere utilizzati come segnali START/STOP, REVERSE o per altri scopi a discrezione dell'utente.
- Il controllore può eseguire il posizionamento nell'intervallo da - 2147483647 a + 2147483648 impulsi se utilizza un motore con sensori Hall.
- Il controllore ha contatti RT sul pannello frontale per la connessione di un resistore terminale.

Controllo tramite segnali esterni

Per controllare il motore tramite segnali esterni, sono previsti:

- Contatti per il collegamento dei segnali di ingresso esterni IN1 e IN2, il cui scopo e la cui elaborazione sono determinati dall'utente. Gli ingressi possono essere utilizzati anche come segnali START/STOP (avvio/arresto movimento) e DIR (direzione).
- Contatto HARD STOP per il collegamento di un segnale di allarme - frenatura del motore controllata in caso di interruzione del circuito di emergenza.

Il controllore fornisce una funzione di protezione da sovraccorrente del motore. La corrente massima consentita nella fase del motore è impostata dall'utente.

3. Caratteristiche tecniche

Modello	RS 434544	RS 434546
Tensione di alimentazione, VCC	12 - 24	
Protezione dell'alimentazione, VCC	8 - 30	
Corrente nominale nella fase del motore, A	<20	<40
Protezione da cortocircuito hardware (tempo di intervento 15 μ s), A	30	100
Limitazione della corrente di fase, A	1...20	2...40
Tensione massima nella fase del motore, V	0,98 x U _{sup}	
Minimo diverso da zero nella fase del motore, V	0,01 x U _{sup}	
Parametri dei segnali esterni IN1 e IN2:		
Massima resistenza dei contatti chiusi, kOhm	4.7	
Corrente massima in ingresso, mA	0.5	
Dimensioni (non più), mm	116x100x23	
Interfaccia di comunicazione	RS-485, Modbus – ASCII o RTU	

Le dimensioni generali e di collegamento del controllore sono mostrate in Fig. 1.

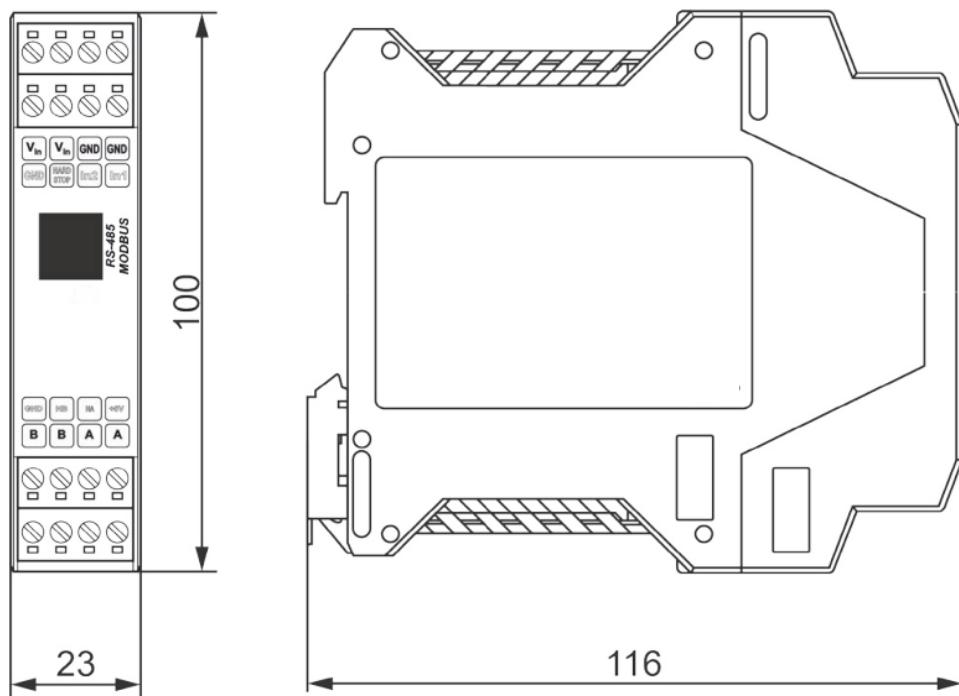


Fig. 1. Dimensioni generali e di collegamento del controllore

Lo schema di collegamento è mostrato in Fig. 2

Condizioni ambientali:

Temperatura ambiente: 0...+50°C

Umidità: 90% RH o inferiore alla condizione di +25°C

Condensazione e congelamento: nessuno

4. Elementi costruttivi e di controllo

Il controllore è progettato come una scheda a circuito stampato con elementi elettronici, coperta da una custodia con un supporto per guida DIN. Sulla parte superiore della custodia sono presenti simboli grafici per i controlli e l'assegnazione dei pin. Oltre ai componenti elettronici, sulla scheda sono presenti elementi di indicazione e controllo, terminali di collegamento e connettori:

- morsetti a vite per il collegamento dell'alimentazione, degli avvolgimenti del motore a spazzole, delle linee dell'encoder e del circuito di controllo;
- terminali IN1 e IN2 per il collegamento dei segnali di ingresso di controllo;
- terminale per il collegamento dei contatti del segnale di arresto di emergenza;
- Indicatore LED del funzionamento del dispositivo;
- Connettore RJ11 (6P6C) per il collegamento delle linee RS-485;
- Contatti per il collegamento della resistenza terminale interna RT;
- Circuito di frenatura integrato per assorbire l'energia generata dal motore (durante la fase di decelerazione, rotazione forzata).

È previsto un ingresso di segnale esterno "HARD STOP" per l'arresto di emergenza del motore. Gli ingressi esterni IN1 e IN2 possono essere utilizzati per avviare e arrestare il motore tramite segnali esterni, nonché per controllare il senso di rotazione del motore.

Tutti i parametri di funzionamento del motore e del controllo del movimento possono essere eseguiti tramite software e comandi trasmessi via RS-485 tramite il protocollo Modbus.

La disposizione e lo scopo dei terminali sono mostrati in Fig. 2



1. GND dell'alimentazione
2. GND dell'alimentazione
3. Alimentazione 12 – 24 VDC
4. Alimentazione 12 – 24 VDC
5. Segnale "IN1" (contatto pulito)
6. Segnale "IN2" (contatto pulito)
7. Segnale di arresto di emergenza "HARD STOP"
8. GND del segnale
9. Uscita +5V per l'alimentazione dei sensori Hall
10. Linea dati A del sensore Hall
11. Linea dati B del sensore Hall
12. GND dei sensori Hall
13. Cavo motore A
14. Cavo motore A
15. Cavo motore B
16. Cavo motore B

RS-485 Modbus - Connettore RJ11 (6P6C) per il collegamento delle linee dati RS-485

RT - contatti per il collegamento della resistenza di terminazione

Fig. 2. Disposizione e scopo dei terminali e degli elementi di controllo

5. Assemblaggio e collegamento

Si prega di leggere attentamente questo manuale prima del collegamento e dell'assemblaggio.

Si prega di effettuare il cablaggio solo quando l'alimentazione è spenta. Non tentare di modificare il cablaggio mentre l'alimentazione è attiva.

Assicurarsi di un contatto affidabile nei terminali di collegamento. Durante il cablaggio, osservare la polarità e la gestione dei cavi. L'inversione di polarità e la sovratensione danneggeranno il controller.

IMPORTANTE: A causa delle elevate correnti, si consiglia di posizionare l'alimentatore in prossimità dell'unità e di utilizzare cavi con una sezione di 3 mm² (AWG-12). L'alimentatore deve fornire il 20% di corrente in più rispetto al massimo possibile consumato durante il funzionamento.

Lunghezza consigliata dei cavi di alimentazione:

- Non più di 100 cm a correnti fino a 10 A.
- Non più di 50 cm a correnti da 10 a 20 A.
- Non più di 25 cm a correnti da 20 a 40 A.

Con una corrente massima fino a 20 A, è consentito utilizzare sia i terminali di alimentazione che quelli di fase su una linea. Per una corrente massima superiore a 20 A, è necessario utilizzare entrambe le linee dei terminali di alimentazione e di fase.

Seguire le seguenti istruzioni durante il collegamento:

1. Collegare un motore al controller secondo la Fig. 2. Il motore deve essere collegato ai terminali A e B (13 – 16) del controller. I segnali del sensore Hall devono essere collegati ai terminali HA e HB (10 – 11). La massa (GND) dei sensori HALL deve essere collegata al terminale GND (12), l'alimentazione dei segnali del sensore HALL deve essere collegata al terminale +5V (9).
2. Collegare gli elementi di controllo esterni secondo gli schemi di collegamento in Fig. 3:

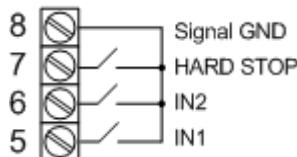


Fig. 3. Collegamento degli elementi di controllo esterni

Tipo di segnali esterni «IN1», «IN2», «HARD STOP» - contatto pulito.

- Collegare le linee dell'interfaccia RS-485 al connettore RJ11 secondo la Fig. 4.

RS-485 - RJ11 (6P6C)

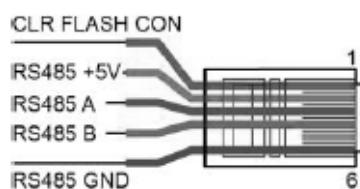


Fig. 4. Assegnazione dei pin del connettore RJ11 - RS-485 Modbus

- Se necessario, posizionare un jumper sui pin RT per collegare la resistenza terminale interna.
- Collegare il dispositivo all'alimentatore, osservando la polarità. L'alimentatore deve essere selezionato con un margine (per evitare cadute di tensione). Lo spessore dei cavi di collegamento deve corrispondere al consumo di corrente del motore. Collegare il "+" dell'alimentatore ai terminali 3 e 4, collegare il "-" dell'alimentatore ai terminali 1 e 2 del controller. La procedura di smontaggio è in ordine inverso.

6. Funzionamento

- Assicurarsi che l'alimentazione sia spenta. Si prega di effettuare il cablaggio solo quando l'alimentazione è spenta.
- Collegare il motore e l'alimentatore al controller secondo la sezione 5.
- Selezionare il metodo di controllo: controllo tramite comandi via Modbus o segnali esterni (registro MODE_DEVICE - vedere sezione 6.5, Fig. 6 e Fig. 7).
- Collegare i segnali di controllo "IN1", "IN2" e il segnale di arresto di emergenza "HARD STOP" secondo la sezione 5. Il segnale "HARD STOP" viene utilizzato per l'arresto di emergenza del motore. Il funzionamento è consentito con il contatto chiuso.
- Collegare le linee dell'interfaccia RS-485 secondo la sezione 5.
- Accendere l'alimentazione. Il dispositivo è pronto per la configurazione tramite protocollo Modbus.
- Impostare i parametri di funzionamento necessari tramite comandi via protocollo Modbus: limitazione della corrente del motore, direzione di rotazione, impostazione del funzionamento degli ingressi esterni IN1 e IN2 e metodo di controllo.
- Per controllare l'azionamento tramite protocollo Modbus, inviare comandi attraverso l'interfaccia RS-485. La tabella dei registri del controller, il loro scopo e i possibili comandi di controllo del motore sono riportati di seguito.
- Per controllare l'azionamento con segnali esterni per avviare e arrestare il motore e per controllare la direzione di rotazione, utilizzare i segnali IN1 e IN2. La regolazione della velocità viene eseguita tramite comandi via protocollo Modbus.

Controllo MODBUS

Per la trasmissione dati tramite l'interfaccia RS-485, viene utilizzato il protocollo di comunicazione standard Modbus (ASCII o RTU).

L'unità di controllo ha le seguenti impostazioni di fabbrica:

- ID = 1
- Velocità: 115200 baud
- Controllo di parità: pari
- Bit di dati: 8
- Bit di stop: 1
- MODBUS RTU

6.1. Registri di controllo degli ingressi

Indirizzo	Tipo	Nome	Dimensione	Descrizione
1000h	Ingresso Discreto	IN1_bit	1-bit	Stato del segnale di ingresso IN1
1001h	Ingresso Discreto	IN2_bit	1-bit	Stato del segnale di ingresso IN2
1002h	Ingresso Discreto	IN_HARD_STOP_bit	1-bit	Stato del segnale di ingresso HARD_STOP
5007h	Registro di Mantenimento	MODE_EXT_IN	16-bit	Configurazione degli ingressi esterni IN1, IN2.
5013h	Registro di Mantenimento	PRESSED_INPUTS_EXTERN	16-bit	Durata minima impulso positivo agli ingressi digitali IN1, IN2 (ms)
5014h	Registro di Mantenimento	WAITED_INPUTS_EXTERN	16-bit	Tempo minimo per la parte negativa dell'impulso agli ingressi digitali IN1, IN2 (ms)

Registri di stato degli ingressi 1000h..1002h sono di sola lettura.

1000h **-IN1_bit-** rappresenta lo stato dell'ingresso fisico IN1.

1001h **-IN2_bit-** rappresenta lo stato dell'ingresso fisico IN2.

1002h **-IN_HARD_STOP_bit-** rappresenta lo stato dell'ingresso fisico HARD_STOP.

Possibili valori del registro 1000h..1002h:

- 1 – ingresso in cortocircuito con l'uscita GND
- 0 – ingresso aperto con l'uscita GND

5007h **-MODE_EXT_IN-** il valore del registro determina lo scopo e il metodo di elaborazione dei segnali IN1 e IN2 (vedere la descrizione nella sezione 6.5).

5013h **-PRESSED_INPUTS_EXTERN** e 5014h **-WAITED_INPUTS_EXTERN-** il tempo minimo (impostato in ms) delle parti positive e negative dell'impulso agli ingressi digitali IN1, IN2 - i registri vengono utilizzati per sopprimere il rimbalzo dei contatti (vedere Fig. 5).

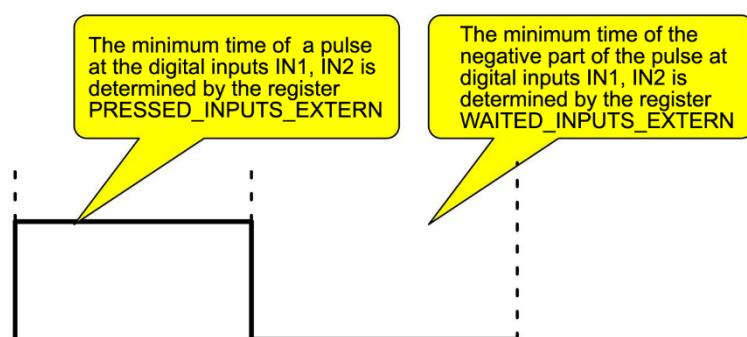


Fig. 5. Soppressione del rimbalzo dei contatti

6.2. Registri di controllo motore

Indirizzo	Tipo	Nome	Dimensione	Descrizione
2000h	Bobine	START_bit	1-bit	Avviare la rotazione del motore con un'accelerazione impostata
2001h	Bobine	STOP_bit	1-bit	Arresto rotazione motore con decelerazione impostata
2002h	Bobine	HARD_STOP_bit	1-bit	Arresto di emergenza brusco della rotazione del motore
2003h	Bobine	CLR_POSITION_bit	1-bit	Azzeroamento del registro della posizione corrente POSITION_VALUE.

Questi registri di controllo sono disponibili sia per la lettura che per la scrittura. Quando si scrive nel registro corrispondente, il valore 1 attiva una o l'altra funzione, dopodiché il registro viene resettato a 0.

6.3. Registri di stato

Indirizzo	Tipo	Nome	Dimensione	Descrizione
3000h	Registro di Input	STATUS	16-bit	Lo stato attuale del controllo motore.
3001h	Registro di Input	CURRENT_VALID	16-bit	Il valore attuale della corrente consumata dal motore.
3002h	Registro di Input	SPEED_VALID	16-bit	Il valore istantaneo della velocità di rotazione.
3003h	Registro di Input	CURRENT_POSITION	32-bit	Posizione attuale. I valori vanno da -2147483647 a +2147483648
3005h	Registro di Input	TEMPERATURE MCU	16-bit	Temperatura CPU
3006h	Registro di Input	TEMPERATURE_MOSFET	16-bit	Temperatura del circuito di potenza
3007h	Registro di Input	TEMPERATURE_BRAKE	16-bit	Temperatura circuito freno
3008h	Registro di Input	TASK_COUNTER	16-bit	Restituisce un valore casuale (per verificare il funzionamento del canale di trasmissione)
3009h	Registro di Input	STATUS_USER_PROGRAM	16-bit	Lo stato attuale dell'esecuzione del programma utente

Questo gruppo di registri è di sola lettura e rappresenta lo stato corrente del controller.

3000h -**STATUS**- stato corrente del controllo motore, possibili valori:

- 0 arresto motore
- 1 rotazione in avanti
- 2 rotazione indietro

3001h -**CURRENT_VALID**- il valore corrente della corrente consumata dal motore. Rappresenta il valore della corrente consumata dal motore dall'alimentatore, il valore in mA.

3002 -**SPEED_VALID**- il valore istantaneo della velocità di rotazione. Le unità di misura sono giri al minuto.

3003h -**CURRENT_POSITION**- la posizione corrente. Il posizionamento viene eseguito nell'intervallo di valori da -2147483647 a +2147483648 del numero totale di commutazioni del sensore Hall, vengono presi in considerazione sia i fronti di salita che di discesa.

3005h - **TEMPERATURE_MCU** — temperatura del processore centrale. La temperatura è calcolata come TEMPERATURE_MCU/10 gradi/°C.

3006h - **TEMPERATURE_MOSFET** - temperatura del circuito di potenza - la temperatura nella zona di installazione degli interruttori MOSFET. La temperatura viene calcolata come TEMPERATURE_MOSFET/10 gradi/°C.

3007h - **TEMPERATURE_BRAKE**- temperatura circuito freno – la temperatura nell'area della resistenza di frenatura. La temperatura viene calcolata come TEMPERATURE_MOSFET/10 gradi/°C.

3008h -**TASK_COUNTER**- restituisce un numero casuale nell'intervallo da 0x0000 a 0xFFFF.

3009h - **STATUS_USER_PROGRAM** - lo stato corrente dell'esecuzione del programma utente, possibili valori:

- 1 – programma utente arrestato da un comando tramite Modbus
- 2 – programma utente avviato da un comando tramite Modbus (questo stato è solo di breve durata, prima che venga impostato lo stato 4)
- 3 – programma utente avviato dopo l'accensione della tensione di alimentazione (questo stato è solo di breve durata, prima che venga impostato lo stato 4)
- 4 – programma utente in esecuzione
- 5 – programma utente terminato con il comando END
- 6 – programma utente terminato con il comando ENDF
- 7 – programma utente arrestato a causa di un errore

6.4. Registri delle impostazioni di trasmissione dati Modbus RS-485

Indirizzo	Tipo	Nome	Dimensione	Descrizione
5000h	Registro di Mantenimento	SLAVE_ADDRESS_MODBUS	16-bit	ID del controllore (indirizzo dispositivo). Valori validi: 0...247. (0 - indirizzo broadcast, nessun messaggio di risposta)
5001h	Registro di Mantenimento	TYPE_MODBUS	16-bit	Impostazioni di trasmissione dati
5002h	Registro di Mantenimento	BITRATE_MODBUS	16-bit	Impostazione della velocità di trasmissione
5003h	Registro di Mantenimento	TIMEOUT_BROADCAST_MODBUS	16-bit	Un ulteriore ritardo tra il pacchetto ricevuto e il messaggio di risposta.

5001h -**TYPE_MODBUS**– impostazioni di trasmissione dati, valori validi:

- 1 – ASCII, 7 bit dati, pari, 1 bit di stop,
- 2 – ASCII, 7 bit dati, dispari, 1 bit di stop
- 3 – RTU, 8 bit dati, pari, 1 bit di stop
- 4 – RTU, 8 bit dati, dispari, 1 bit di stop
- 5 – RTU, 8 bit dati, nessuno, 2 bit di stop

5002h -**BITRATE_MODBUS**– velocità di trasmissione Modbus, valori possibili:

- 0 – 600
- 1 – 1200,
- 2 – 2400,
- 3 – 4800,
- 4 – 9600,
- 5 – 14400,
- 6 – 19200,
- 7 – 38400,
- 8 – 57600,
- 9 – 115200,
- 10 – 128000

5003h -**TIMEOUT_BROADCAST_MODBUS**- viene utilizzato se il dispositivo di controllo impiega molto tempo per passare dalla modalità di trasmissione alla modalità di ricezione.

Dopo aver modificato le impostazioni di comunicazione, i nuovi valori devono essere salvati utilizzando il registro FLAG_SAVE_INI (vedere la sezione 6.5), quindi il controller deve essere riavviato. Dopo il riavvio, la connessione RS-485 verrà eseguita utilizzando le nuove impostazioni.

6.5. Registri per l'impostazione del funzionamento dell'azionamento

Indirizzo	Tipo	Nome	Dimensione	Descrizione
5004h	Registro di Mantenimento	MODE_DEVICE	16-bit	Modalità di funzionamento del controller.
5005h	Registro di Mantenimento	MODE_USER_PROGRAM	16-bit	Comando di controllo per l'avvio di un programma utente.
5006h	Registro di Mantenimento	MODE_ROTATION	16-bit	Modalità di rotazione.
5007h	Registro di Mantenimento	MODE_EXT_IN	16-bit	Configurazione degli ingressi esterni IN, IN2.
5008h	Registro di Mantenimento	POSITION_N	16-bit	Il numero di posizione a cui spostarsi. I valori vanno da 1 a 4
5009h	Registro di Mantenimento	REF_CURRENT	16-bit	Limitazione dell'assorbimento di corrente. Intervallo: 1000 mA a 20000 mA per RS 434544 Da 2000 mA a 40000 mA per RS 434546
500Ah	Registro di Mantenimento	RATED_SPEED	16-bit	Velocità nominale del motore: Intervallo da 1000 a 15000 giri/min
500Bh	Registro di Mantenimento	SPEED	16-bit	Imposta la velocità di rotazione. Intervallo da 30 a 15000 giri/min.
500Ch	Registro di Mantenimento	ACC	16-bit	Imposta accelerazione. Intervallo da 10 a 1000.
500Dh	Registro di Mantenimento	DEC	16-bit	Imposta decelerazione. Intervallo da 10 a 1000.
500Eh	Registro di Mantenimento	DIRECTION	16-bit	Senso di rotazione. 1 – avanti 2 – indietro
500Fh	Registro di Mantenimento	PULSES_PER_REVOLUTION	16-bit	Numero di impulsi per giro da un sensore Hall: da 1 a 12
5010h	Registro di Mantenimento	USE_HALL	16-bit	Utilizzo dei sensori Hall: 0 – senza sensori Hall 1 – un sensore Hall 2 – due sensori Hall

5011h	Registro di Mantenimento	MODE_COIL	16-bit	Stato dei terminali del motore a motore fermo: 0 - aperto 1 - chiuso
5012h	Registro di Mantenimento	OFFSET_COMPENSATION	16-bit	Correzione della frenata del motore
5013h	Registro di Mantenimento	PRESSED_INPUTS_EXTERN	16-bit	Durata minima impulso positivo agli ingressi digitali IN1, IN2 (ms)
5014h	Registro di Mantenimento	WAITED_INPUTS_EXTERN	16-bit	Tempo minimo per la parte negativa dell'impulso agli ingressi digitali IN1, IN2 (ms)
5015h	Registro di Mantenimento	OFFSET	32-bit	L'offset da spostare, il valore viene modificato dal controller durante il funzionamento. Intervallo di valori da -2147483647 a + 2147483648
5017h	Registro di Mantenimento	OFFSET_CONST	32-bit	Incremento - l'offset a cui è necessario spostarsi.
5019h	Registro di Mantenimento	TARGET_POSITION	32-bit	La posizione da raggiungere. Intervallo di valori da -2147483647 a + 2147483648
501Bh	Registro di Mantenimento	TARGET_POSITION1	32-bit	Posizione preimpostata utente №1 Intervallo di valori da -2147483647 a + 2147483648
501Dh	Registro di Mantenimento	TARGET_POSITION2	32-bit	Posizione preimpostata utente №2 Intervallo di valori da -2147483647 a + 2147483648
501Fh	Registro di Mantenimento	TARGET_POSITION3	32-bit	Posizione preimpostata utente №3 Intervallo di valori da -2147483647 a + 2147483648
5021h	Registro di Mantenimento	TARGET_POSITION4	32-bit	Posizione preimpostata utente №4 Intervallo di valori da -2147483647 a + 2147483648
5023h	Registro di Mantenimento	ERROR	16-bit	Registro errori

5024h	Registro di Mantenimento	FLAG_SAVE_INI	16-bit	Registro per il salvataggio delle impostazioni utente.(Registri 5000h..501Fh). Valore: 0x37FA
5025h	Registro di Mantenimento	FLAG_SAVE_USER_PROGRAM	16-bit	Registro per la scrittura o la lettura di un programma utente da o verso la memoria non volatile. Valore da scrivere: 0x8426 Valore da leggere: 0x9346
5026h	Registro di Mantenimento	FLAG_RESTART	16-bit	Registro di riavvio. Valore: 0x95AF

5004h -**MODE_DEVICE**- modalità di funzionamento del controller, valori possibili:

- 1 – controllo tramite Modbus
- 2 – controllo tramite segnali esterni

5005h -**MODE_USER_PROGRAM**- comando di controllo per l'avvio di un programma utente, valori possibili:

- 1 – arresta il programma utente
- 2 – avvia il programma utente
- 3 – avvia il programma utente non appena viene accesa l'alimentazione (è necessario il pre-salvataggio delle impostazioni - vedere il registro FLAG_SAVE_INI)

Il programma utente può essere memorizzato nella memoria flash interna del controller. Per avviare l'esecuzione di un programma utente, è necessario scrivere 2 nel registro MODE_USER_PROGRAM. Per interrompere l'esecuzione del programma, è necessario scrivere 1 nel registro MODE_USER_PROGRAM.

5006h -**MODE_ROTATION**- modalità di rotazione, valori possibili:

- 1 – rotazione continua
- 2 – offset della quantità specificata dal registro OFFSET
- 3 – spostamento in una data posizione. Il numero di posizione è specificato dal registro POSITION_N (da 1 a 4), le coordinate delle posizioni sono specificate rispettivamente dai registri POSITION1, POSITION2, POSITION3 e POSITION4.

5007h -**MODE_EXT_IN**- configurazione della modalità di funzionamento degli ingressi esterni IN1, IN2 nella modalità di controllo dei segnali esterni, valori possibili:

- 1 – L'ingresso IN1 viene utilizzato come segnale di avvio/arresto dell'azionamento, viene elaborato sul fronte di discesa dell'impulso; l'ingresso IN2 viene utilizzato come segnale di inversione, viene elaborato sul fronte di discesa dell'impulso.
- 2 - L'ingresso IN1 viene utilizzato come segnale di avvio/arresto dell'azionamento, viene elaborato sul fronte di discesa dell'impulso; l'ingresso IN2 viene utilizzato come segnale di riferimento di direzione, elaborato in base al livello del segnale.
- 3 - L'ingresso IN1 è utilizzato come segnale di avvio/arresto dell'azionamento, viene elaborato in base al livello del segnale: presenza di un segnale - abilita la rotazione del motore, nessun segnale - arresto; l'ingresso IN2 è utilizzato come segnale di inversione, viene elaborato sul fronte di discesa dell'impulso.
- 4 - L'ingresso IN1 è utilizzato come segnale di avvio/arresto dell'azionamento, viene elaborato in base al livello del segnale: presenza di un segnale - abilita la rotazione del motore, nessun segnale - arresto; l'ingresso IN2 è utilizzato come segnale di riferimento di direzione, elaborato in base al livello del segnale.
- 5 – l'ingresso IN1 è utilizzato come segnale per avviare e arrestare l'azionamento in avanti, IN2 - come segnale per avviare e arrestare l'azionamento nella direzione opposta; entrambi i segnali sono elaborati per livello.

5008h -**POSITION_N**- selezione del numero di posizione per il movimento (da 1 a 4) - utilizzato nella modalità di movimento verso una data posizione (MODE_ROTATION = 3) in combinazione con i registri POSITION1, POSITION2, POSITION3, POSITION4.

5009h –**REF_CURRENT**– impostazione del limite di consumo di corrente. L'intervallo di valori consentiti per RS 434544: da 1000 mA a 20000 mA; per RS 434546: da 2000 mA a 40000 mA.

500Ah –**RATED_SPEED**– velocità nominale del motore: da 1000 a 15000 giri/min. È impossibile conoscere la velocità effettiva del motore se il motore viene utilizzato senza sensori Hall. Per questo motivo, la velocità massima del motore (che corrisponde alla massima alimentazione sul motore) viene utilizzata per calcolare i valori impostati della velocità. Se RATED_SPEED non è impostato correttamente, la velocità effettiva del motore differisce sostanzialmente dal valore target, che è impostato nel registro SPEED.

500Bh –**SPEED**– velocità di rotazione target (da 30 a 15000 giri/min). Il valore della velocità target deve essere inferiore al valore MAX_SPEED.

500Ch –**ACC**– accelerazione data (da 10 a 1000)

500Dh –**DEC**– decelerazione data (da 10 a 1000)

I valori di accelerazione e decelerazione nei registri ACC e DEC sono valori lineari convenzionali che determinano la velocità di accelerazione/decelerazione. Un valore di 10 corrisponde a 100 rps², un valore di 1000 corrisponde a 5000 rps².

500Eh –**DIRECTION**– direzione di rotazione, possibili valori:

- 1 – rotazione in avanti
- 2 – rotazione indietro.

500Fh –**PULSES_PER_REVOLUTION**– numero di impulsi per giro da un sensore Hall(da 1 a 12).

5010h –**USE_HALL**– utilizzo dei sensori Hall:

- 0 – un motore senza sensori Hall
- 1 – un sensore Hall
- 2 – due sensori Hall.

5011h –**MODE_COIL**– Stato dei terminali del motore a motore fermo:

- 0 - aperto
- 1 - chiuso

5012h -**OFFSET_COMPENSATION**– Correzione di frenatura del motore calcolata sperimentalmente per le impostazioni correnti ACC, DEC, SPEED. Se l'arresto si verifica oltre il punto calcolato, il valore di compensazione è uguale all'errore di posizionamento con segno negativo, se il motore si ferma prima del punto di arresto, il valore di compensazione è positivo. Ad esempio, l'errore di raggiungimento della posizione specificata è di 15 incrementi, il che significa OFFSET_COMPENSATION = -15, e questa correzione è valida solo per le impostazioni correnti di accelerazione, decelerazione e velocità.

5013h e 5014h –**PRESSED_INPUTS_EXTERN** e **WAITED_INPUTS_EXTERN**– tempo minimo di una parte positiva e di una parte negativa di un impulso agli ingressi digitali IN1, IN2 – utilizzato per sopprimere il rimbalzo dei contatti (vedi fig. 5).

5015h –**OFFSET**– l'offset da spostare viene utilizzato quando MODE_ROTATION = 2, valori validi da -2147483647 a + 2147483648. Prima di iniziare il movimento, è necessario impostare il valore di offset richiesto nel registro OFFSET o OFFSET_CONST, che viene elaborato dal controller come contatore del movimento rimanente. Durante l'esecuzione del movimento specificato, il valore OFFSET diminuisce.

5017h –**OFFSET_CONST**-Incremento - l'offset a cui è necessario spostarsi. Se il valore del registro OFFSET_CONST ≠ 0, viene copiato nel registro OFFSET ad ogni avvio del motore.

5019h –**TARGET_POSITION**– posizione verso cui spostarsi, valori consentiti da -2147483647 a + 2147483648. Nella modalità di spostamento verso una data posizione, la coordinata da POSITION1-POSITION4 viene copiata in questo registro prima dello spostamento, mentre i registri POSITION1-POSITION4 non cambiano i loro valori.

501Bh - 5021h –**TARGET_POSITION1..4**– Posizione preimpostata dall'utente №1..4, utilizzata quando MODE_ROTATION = 3, il numero di posizione è determinato dal registro POSITION_N, valori consentiti da - 2147483647 a + 2147483648.

5023h –**ERROR**– errori che si verificano durante il funzionamento del controller - ogni bit del registro segnala un errore specifico:

- bit 0 - fuori dall'intervallo di tensione di alimentazione;
- bit 1 - cortocircuito degli avvolgimenti del motore;
- bit 2 - surriscaldamento del circuito del freno;
- bit 3 - surriscaldamento del circuito di potenza;
- bit 4 - errore di collegamento dei sensori Hall;
- bit 5 - arresto di emergenza;
- bit 6 - surriscaldamento della MCU;
- bit 7 - programma di controllo di test;
- bit 8 - errore di esecuzione del programma utente;
- bit 9 - errore di lettura o scrittura delle impostazioni;
- bit 10 - errore nel funzionamento degli interruttori a transistor di uscita;
- bit 11 - superamento del limite di corrente assorbita;
- bit 12 - avviso sull'impossibilità di calcolare il punto di interruzione;
- bit 13 - avviso su un tentativo di scrivere nel registro un valore fuori intervallo;
- bit 14 - errore di parità di trasmissione RS-485;
- bit 15 - utilizzo della modalità di posizionamento con meno di due sensori Hall.

5024h –**FLAG_SAVE_INI**– registro per il salvataggio delle impostazioni utente - quando si scrive il valore 0x37FA in questo registro, le impostazioni definite dai registri 5000h..501Fh verranno salvate nella memoria non volatile.

5025h –**FLAG_SAVE_USER_PROGRAM**– scrivendo il valore 0x8426 in questo registro si avvia la procedura per il salvataggio del programma utente dal buffer temporaneo alla memoria non volatile del controller. Scrivendo il valore 0x9346 si avvia la procedura per la lettura del programma utente dalla memoria non volatile del controller in un buffer temporaneo (vedere la sezione 6.6.).

5026h –**FLAG_RESTART**– scrivendo il valore 0x95AF in questo registro si provoca il riavvio del controller.

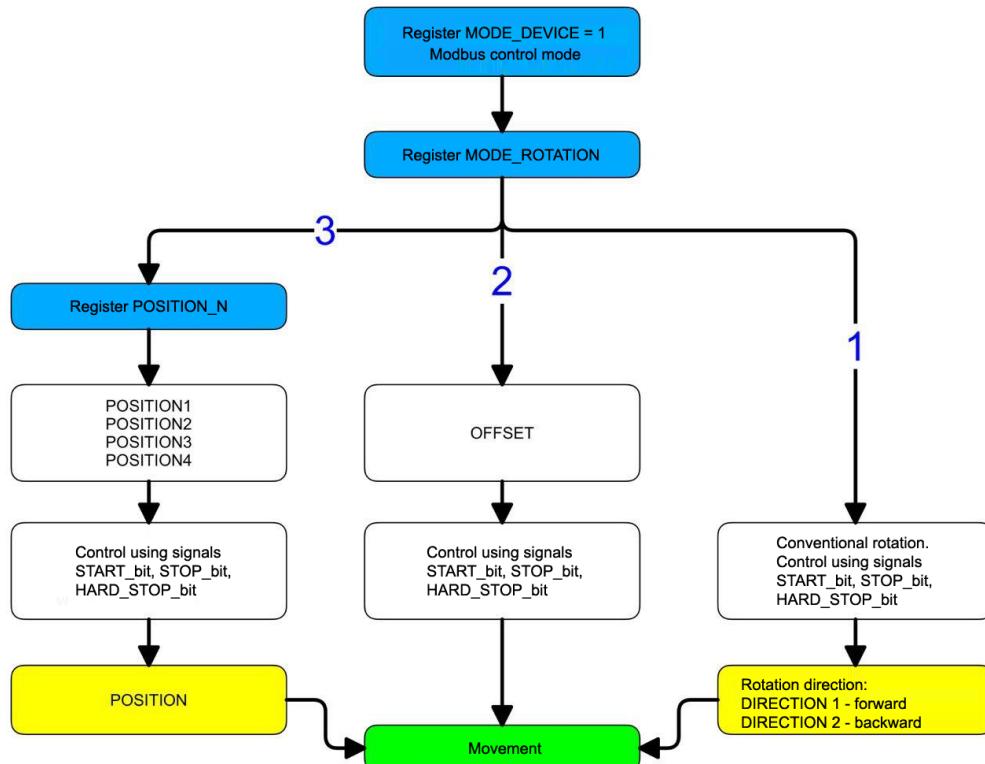


Fig. 6. Diagramma di flusso per la selezione della modalità operativa durante il controllo dell'azionamento tramite Modbus

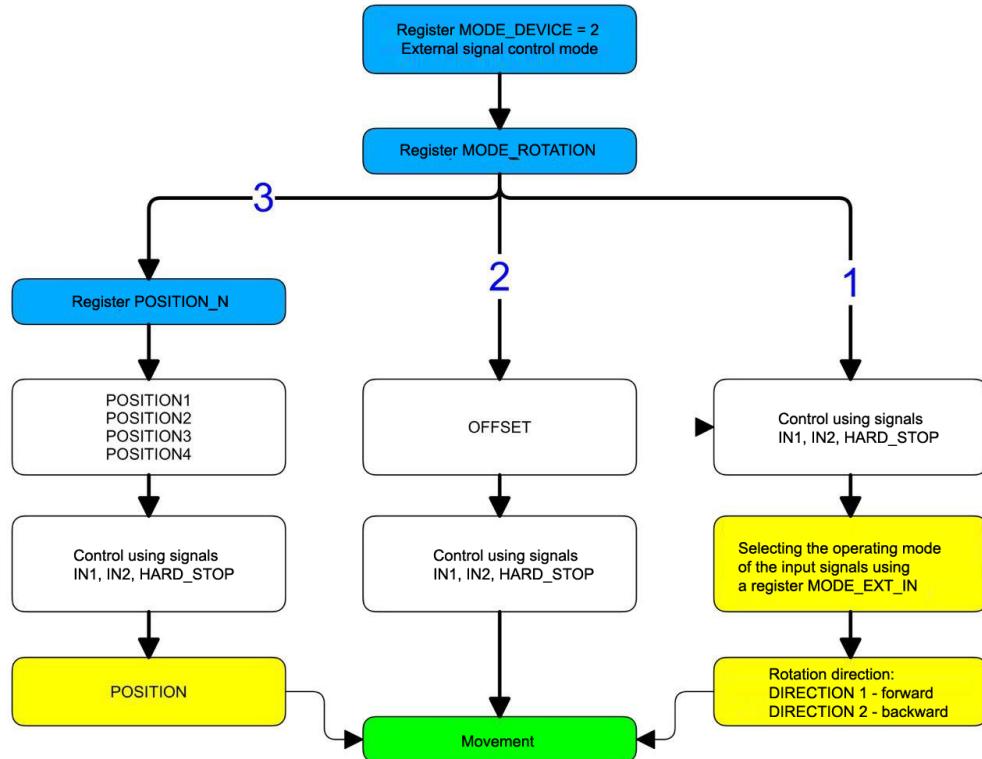


Fig. 7. Diagramma di flusso per la selezione della modalità operativa durante il controllo dell'azionamento tramite segnali esterni

6.6. Lettura e scrittura di un programma utente

Un buffer temporaneo di 1024 comandi viene utilizzato per leggere e scrivere un programma utente. Il registro speciale FLAG_SAVE_INI viene utilizzato per salvare il programma dal buffer temporaneo alla memoria non volatile e per leggere il programma dalla memoria del controller al buffer temporaneo.

Indirizzo	Tipo	Nome	Dimensione	Descrizione
5025h	Registro Holding	FLAG_SAVE_USER_PROGRAM	16-bit	Registro per la scrittura o la lettura di un programma utente da o verso la memoria non volatile del controller. Valore di scrittura: 0x8426 Valore di lettura: 0x9346
6000h	Registro Holding	WRITE_CMD	16-bit	Registro di indirizzi. Quando il valore dell'indirizzo viene scritto in questo registro, il comando viene trasferito dal campo CMD_W all'indirizzo specificato del buffer temporaneo del programma utente.
6001h	Registro Holding	CMD_W	32-bit	Istruzioni del programma utente
6003h	Registro Holding	READ_CMD	16-bit	Registro di indirizzi. Quando il valore dell'indirizzo viene scritto in questo registro, il comando viene trasferito dall'indirizzo specificato del buffer temporaneo del programma utente al campo CMD_R.
6004h	Registro Holding	CMD_R	32-bit	Istruzioni del programma utente

Assemblaggio e scrittura di un programma utente nella memoria del controller

Ogni istruzione del programma utente è composta da due parole di memoria (32 bit) - comando (16 bit) e dati del comando (16 bit). Durante l'assemblaggio di un programma utente, le istruzioni vengono prima scritte in un buffer temporaneo nel controller. Per scrivere nel buffer temporaneo, è necessario scrivere un'istruzione nel registro CMD_W e quindi scrivere l'indirizzo di questa istruzione nel buffer interno nel registro WRITE_CMD. Quando l'indirizzo viene scritto nel registro WRITE_CMD, l'istruzione viene trasferita dal registro CMD_W al buffer interno. Dopo aver composto un programma utente in un buffer temporaneo, è necessario scrivere il valore 0x8426 nel registro FLAG_SAVE_USER_PROGRAM - il programma verrà trasferito dal buffer temporaneo alla memoria FLASH del controller.

Lettura di un programma utente dalla memoria del controller

Per leggere il programma utente dalla memoria FLASH è necessario trasferirlo nel buffer temporaneo del controller. A tale scopo, scrivere il valore 0x9346 nel registro FLAG_SAVE_USER_PROGRAM: il programma verrà trasferito dalla memoria FLASH del controller al buffer temporaneo. Quindi, per leggere un'istruzione dal buffer temporaneo, è necessario scrivere l'indirizzo dell'istruzione nel registro READ_CMD; quando l'indirizzo viene scritto in questo registro, l'istruzione verrà copiata dal buffer temporaneo al registro CMD_R.

Un diagramma delle procedure di lettura e scrittura di un programma utente è mostrato in Fig. 8.

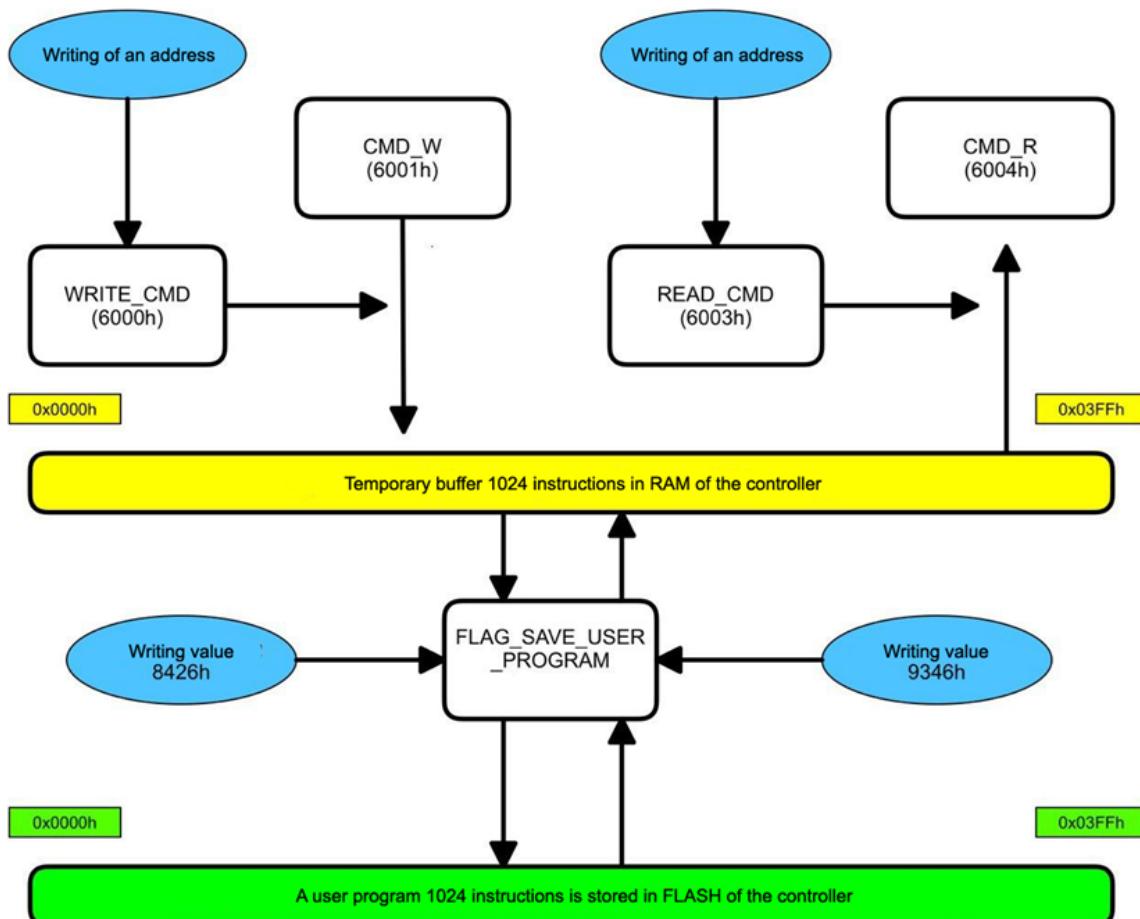


Fig. 8. Diagramma di flusso delle procedure di lettura e scrittura di un programma utente

6.7. Registri di sistema

Indirizzo	Tipo	Nome	Dimensione	Descrizione
7000h	Registro di Mantenimento	AX_REG	16-bit	Registro di memorizzazione dati per il programma utente
7001h	Registro di Mantenimento	BX_REG	16-bit	Registro di memorizzazione dati per il programma utente
7002h	Registro di Mantenimento	CX_REG	16-bit	Registro di memorizzazione dati per il programma utente
7003h	Registro di Mantenimento	DX_REG	16-bit	Registro di memorizzazione dati per il programma utente
7004h	Registro di Mantenimento	EX_REG	16-bit	Registro di memorizzazione dati per il programma utente
7005h	Registro di Mantenimento	FX_REG	16-bit	Registro di memorizzazione dati per il programma utente
7006h	Registro di Mantenimento	PC_REG	16-bit	Puntatore di registro al comando utente attualmente in esecuzione
7007h	Registro di Mantenimento	GX_REG	16-bit	Registro di memorizzazione dati per il programma utente
7008h	Registro di Mantenimento	HX_REG	16-bit	Registro di memorizzazione dati per il programma utente
7009h	Registro di Mantenimento	IX_REG	16-bit	Registro di memorizzazione dati per il programma utente
700Ah	Registro di Mantenimento	JX_REG	16-bit	Registro di memorizzazione dati per il programma utente

Registri di sistema AX_REG...FX_REG (intervallo di valori 0..65535) sono destinati alla memorizzazione temporanea dei dati durante l'esecuzione di un programma utente. PC_REG è un puntatore all'istruzione del programma utente attualmente in esecuzione. Descrizione più dettagliata nella sezione 6.9.

6.8. Registri di identificazione

Indirizzo	Tipo	Nome	Dimensione	Descrizione
8001h	Registro di Input	HW_MAJOR	16-bit	Tipo di driver
8002h	Registro di Input	HW_MINOR	16-bit	Versione hardware
8003h	Registro di Input	FW_MAJOR	16-bit	Identificativo del software
8004h	Registro di Input	FW_MINOR	16-bit	Versione del software

I registri sono necessari per determinare lo scopo funzionale dell'unità di controllo, le sue caratteristiche e la versione del software tramite la rete.

Per RS 434544, i valori sono:

- HW_MAJOR 1001
- HW_MINOR x
- FW_MAJOR x
- FW_MINOR x

Per RS 434546, i valori sono:

- HW_MAJOR 1002
- HW_MINOR x

- FW_MAJOR x
- FW_MINOR x

6.9. Istruzioni del programma utente

La struttura delle istruzioni del programma utente è mostrata nel diagramma in Fig. 9.

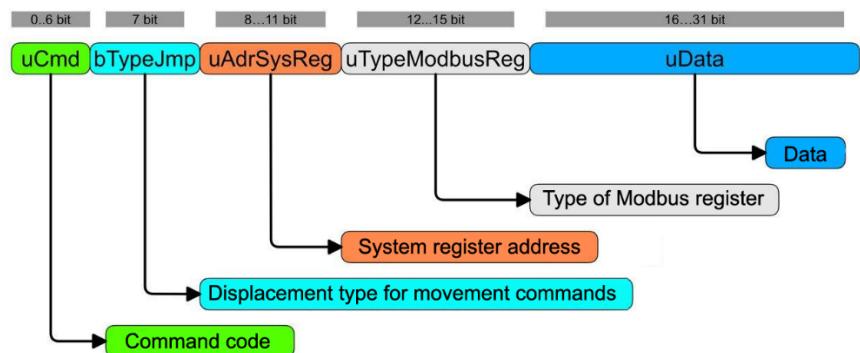


Fig. 9. Struttura di un'istruzione del programma utente

L'istruzione del programma utente ha una dimensione di 32 bit e contiene i seguenti campi:

uCmd – 7 bit – codice comando

bTypeJmp – 1 bit – tipo di spostamento per i comandi di movimento:

- 0 – valore assoluto
- 1 – valore relativo

uAdrSysReg – 4 bit - indirizzo del registro di sistema AX_REG ... FX_REG con numeri da 0 a 5

uTypeModbusReg – 4 bit – tipo di registro Modbus:

- 0 – Ingressi Discreti
- 1 – Bobine
- 2 – Ingressi
- 3 – Registri di mantenimento

uData – 16 bit - dati.

Le istruzioni a 64 bit sono costituite da due righe di comando a 32 bit. La prima riga di comando di questa istruzione è un comando simile a un'istruzione a 32 bit. La seconda riga di comando contiene dati **uCMD_DATA** a 16 o 32 bit.

Le istruzioni con prefisso D, come DMOV, operano su dati a 32 bit situati in due registri a 16 bit consecutivi. Nel campo indirizzo di tali istruzioni è indicato il registro inferiore. Durante la lettura di un valore, il comando legge automaticamente due registri a 16 bit consecutivi a partire dall'indirizzo specificato e forma un singolo valore a 32 bit. Durante la scrittura di un valore, l'istruzione divide un numero a 32 bit in due parti a 16 bit e le scrive in due registri consecutivi a partire dall'indirizzo specificato.

La tabella seguente elenca i comandi con le opzioni per la compilazione dei campi. Se il campo non è specificato, non viene utilizzato in questo comando.

uCmd		-	-	-
0x00	CMD_STOP_PROGRAM			
0x0D	CMD_FULL_STOP_PROGRAM	-	-	-
uCmd		bTypeJmp	uData	-
0x05	CMD_JMP			
0x06	CMD_JMP_AX_PARI_BX			
0x07	CMD_JMP_AX_NOPARI_BX			
0x08	CMD_JMP_AX_MORE_BX			
0x09	CMD_JMP_AX_LESS_BX			
0x63	CMD_DJMP_GX_PARI_IX	0 – valore assoluto 1 – valore relativo	Indirizzo di spostamento 0..1024 o offset +-1024	-
0x64	CMD_DJMP_GX_NOPARI_IX			
0x65	CMD_DJMP_GX_MORE_IX			
0x66	CMD_DJMP_GX_LESS_IX			

uCmd		uData	-	-
0x04	CMD_DELAY	Tempo di ritardo 0..65535 ms	-	-
0x0A	CMD_CALL	Indirizzo subroutine 0..1024		
0x0B	CMD_RETURN	-		
0x0C	CMD_FOR	Durata del ciclo e numero di cicli		
uCmd		uAdrRegSys	-	-
0x15	CMD_NOT_SYSREG	Indirizzo del registro di sistema (0..9)	-	-
0x17	CMD_DNOT_SYSREG			
uCmd		uTypeModbusReg	uData	-
0x16	CMD_NOT_MODBUS	1 – Bobine 3 – Registri di mantenimento	Indirizzo del registro Modbus 0..65535	-
0x18	CMD_DNOT_MODBUS			
uCmd		uAdrRegSys	uData	-
0x01	CMD_MOV_SYSREG_CONST	Indirizzo del registro di sistema (0..9)	Costante 0..65535	-
0x19	CMD_ADD_SYSREG_CONST			
0x1A	CMD_SUB_SYSREG_CONST			
0x1B	CMD_DIV_SYSREG_CONST			
0x1C	CMD_MUL_SYSREG_CONST			
0x1D	CMD_AND_SYSREG_CONST			
0x1E	CMD_OR_SYSREG_CONST			
0x1F	CMD_XOR_SYSREG_CONST			
uCmd		uAdrRegSys	uTypeModbusReg	uData
0x03	CMD_MOV_SYSREG_MODBUS	Indirizzo del registro di sistema (0..9)	0 – Ingressi Discreti 1 – Bobine 2 – Ingressi 3 – Registri di mantenimento	Indirizzo del registro Modbus 0..65535
0x21	CMD_ADD_SYSREG_MODBUS			
0x22	CMD_SUB_SYSREG_MODBUS			
0x23	CMD_DIV_SYSREG_MODBUS			
0x24	CMD_MUL_SYSREG_MODBUS			
0x25	CMD_AND_SYSREG_MODBUS			
0x26	CMD_OR_SYSREG_MODBUS			
0x27	CMD_XOR_SYSREG_MODBUS			
uCmd		uTypeModbusReg	uData	uCMD_DATA
0x0E	CMD_MOV_MODBUS_CONST	1 – Bobine 3 – Registri di mantenimento	Indirizzo del registro Modbus 0..65535	Costante 0..65535 (comando separato linea)
0x28	CMD_ADD_MODBUS_CONST			
0x29	CMD_SUB_MODBUS_CONST			
0x2A	CMD_DIV_MODBUS_CONST			
0x2B	CMD_MUL_MODBUS_CONST			
0x2C	CMD_AND_MODBUS_CONST			
0x2D	CMD_OR_MODBUS_CONST			
0x2E	CMD_XOR_MODBUS_CONST			
uCmd		uAdrRegSys	uTypeModbusReg	uData
0x02	CMD_MOV_MODBUS_SYSREG	Indirizzo del registro di sistema (0..9)	0 – Ingressi Discreti 1 – Bobine 2 – Ingressi 3 – Registri di mantenimento	Indirizzo del registro Modbus 0..65535
0x30	CMD_ADD_MODBUS_SYSREG			
0x31	CMD_SUB_MODBUS_SYSREG			
0x32	CMD_DIV_MODBUS_SYSREG			
0x33	CMD_MUL_MODBUS_SYSREG			
0x34	CMD_AND_MODBUS_SYSREG			
0x35	CMD_OR_MODBUS_SYSREG			

0x36	CMD_XOR_MODBUS_SYSREG			
	uCmd	uAdrRegSys	uData	-
0x0F	CMD_MOV_SYSREG_SYSREG	Indirizzo del registro di sistema (0..9)	Indirizzo del registro di sistema (0..9)	-
0x37	CMD_ADD_SYSREG_SYSREG			
0x38	CMD_SUB_SYSREG_SYSREG			
0x39	CMD_DIV_SYSREG_SYSREG			
0x3A	CMD_MUL_SYSREG_SYSREG			
0x3B	CMD_AND_SYSREG_SYSREG			
0x3C	CMD_OR_SYSREG_SYSREG			
0x3D	CMD_XOR_SYSREG_SYSREG			
	uCmd	uAdrRegSys	uCMD_DATA	-
0x10	CMD_DMOV_SYSREG_CONST	Indirizzo del registro di sistema (0..9)	Costante 0...4294967295 (comando separato linea)	-
0x3E	CMD_DADD_SYSREG_CONST			
0x3F	CMD_DSUB_SYSREG_CONST			
0x40	CMD_DDIV_SYSREG_CONST			
0x41	CMD_DMUL_SYSREG_CONST			
0x42	CMD_DAND_SYSREG_CONST			
0x43	CMD_DOR_SYSREG_CONST			
0x44	CMD_DXOR_SYSREG_CONST			
	uCmd	uAdrRegSys	uTypeModbusReg	uData
0x11	CMD_DMOV_SYSREG_MODBUS	Indirizzo del registro di sistema (0..9)	0 – Ingressi Discreti 1 – Bobine 2 – Ingressi 3 – Registri di mantenimento	Indirizzo del registro Modbus 0..65535
0x46	CMD_DADD_SYSREG_MODBUS			
0x47	CMD_DSUB_SYSREG_MODBUS			
0x48	CMD_DDIV_SYSREG_MODBUS			
0x49	CMD_DMUL_SYSREG_MODBUS			
0x4A	CMD_DAND_SYSREG_MODBUS			
0x4B	CMD_DOR_SYSREG_MODBUS			
0x4C	CMD_DXOR_SYSREG_MODBUS			
	uCmd	uTypeModbusReg	uData	uCMD_DATA
0x12	CMD_DMOV_MODBUS_CONST	1 – Bobine 3 – Registri di mantenimento	Indirizzo del registro Modbus 0..65535	Costante 0...4294967295 (comando separato linea)
0x4D	CMD_DADD_MODBUS_CONST			
0x4E	CMD_DSUB_MODBUS_CONST			
0x4F	CMD_DDIV_MODBUS_CONST			
0x50	CMD_DMUL_MODBUS_CONST			
0x51	CMD_DAND_MODBUS_CONST			
0x52	CMD_DOR_MODBUS_CONST			
0x53	CMD_DXOR_MODBUS_CONST			
	uCmd	uAdrRegSys	uTypeModbusReg	uData
0x13	CMD_DMOV_MODBUS_SYSREG	Indirizzo del registro di sistema (0..9)	0 – Ingressi Discreti 1 – Bobine 2 – Ingressi 3 – Registri di mantenimento	Indirizzo del registro Modbus 0..65535
0x55	CMD_DADD_MODBUS_SYSREG			
0x56	CMD_DSUB_MODBUS_SYSREG			
0x57	CMD_DDIV_MODBUS_SYSREG			
0x58	CMD_DMUL_MODBUS_SYSREG			
0x59	CMD_DAND_MODBUS_SYSREG			
0x5A	CMD_DOR_MODBUS_SYSREG			
0x5B	CMD_DXOR_MODBUS_SYSREG			
	uCmd	uAdrRegSys	uData	-
0x14	CMD_DMOV_SYSREG_SYSREG			-

0x5C	CMD_DADD_SYSREG_SYSREG	Indirizzo del registro di sistema (0..9)	Indirizzo del registro di sistema (0..9)	
0x5D	CMD_DSUB_SYSREG_SYSREG			
0x5E	CMD_DDIV_SYSREG_SYSREG			
0x5F	CMD_DMUL_SYSREG_SYSREG			
0x60	CMD_DAND_SYSREG_SYSREG			
0x61	CMD_DOR_SYSREG_SYSREG			
0x62	CMD_DXOR_SYSREG_SYSREG			
uCmd		uAdrRegSys	bTypeJmp	uData
0x20	CMD_SH_SYSREG_CONST	Indirizzo del registro di sistema (0..9)	0 – spostamento a sinistra 1 – scorrimento a destra	Valore di spostamento
0x45	CMD_DSH_SYSREG_CONST			
uCmd		uTypeModbusReg	uData	bTypeJmp
0x2F	CMD_SH_MODBUS_CONST	1 – Bobine 3 – Registri di mantenimento	Indirizzo del registro Modbus 0..65535	0 – spostamento a sinistra 1 – scorrimento a destra
0x54	CMD_DSH_MODBUS_CONST			

CMD_STOP_PROGRAM – (codice comando 0x00) – interrompe l'esecuzione di un programma utente, senza uscire dalla modalità di esecuzione del programma utente. Al termine dell'esecuzione del programma, tutti i registri e gli stati del motore rimangono come prima dell'esecuzione del comando (il motore continua a ruotare se stava ruotando prima dell'esecuzione del comando). Prima del successivo avvio del programma, è necessario inviare il comando **CMD_FULL_STOP_PROGRAM**.

CMD_FULL_STOP_PROGRAM – (codice comando 0x0D) - interrompe l'esecuzione del programma utente e esce dalla modalità di funzionamento del programma. Al termine dell'esecuzione del programma, tutti i registri e lo stato del motore tornano ai valori originali e il motore si ferma.

CMD_MOV_SYSREG_CONST – (codice comando 0x01) – scrittura nel registro di sistema con indirizzo uAdrSysReg, valori dal campo dati uData.

CMD_MOV_MODBUS_SYSREG – (codice comando 0x02) – scrittura del contenuto dal registro di sistema uAdrSysReg nello spazio del registro ModBUS definito dal campo TypeModbusReg e il suo indirizzo nel campo uData.

CMD_MOV_MODBUS_SYSREG – (codice comando 0x03) – lettura del contenuto dallo spazio del registro ModBUS determinato dal campo TypeModbusReg e il suo indirizzo nel campo uData in uno dei registri di sistema uAdrSysReg.

CMD_MOV_MODBUS_CONST – (codice comando 0x0E) – scrittura della costante contenuta nella seguente riga di comando uCMD_DATA nello spazio del registro ModBUS definito dal campo TypeModbusReg e il suo indirizzo nel campo uData.

CMD_MOV_SYSREG_SYSREG – (codice comando 0x0F) – scrittura del contenuto dal registro di sistema uAdrSysReg in un altro registro di sistema con indirizzo uData.

CMD_DELAY – (codice comando 0x04) – pausa, ms.

CMD JMP – (codice comando 0x05) – salto all'indirizzo specificato nel campo uData.

CMD JMP_AX_PARI_BX – (codice comando 0x06) – salto all'indirizzo specificato nel campo uData, se il valore nel registro di sistema AX_REG è uguale al valore in BX_REG.

CMD JMP_AX_NOPARI_BX – (codice comando 0x07) – salto all'indirizzo specificato nel campo uData se il valore nel registro di sistema AX_REG non è uguale al valore in BX_REG.

CMD JMP_AX_MORE_BX – (codice comando 0x08) – salto all'indirizzo specificato nel campo uData, se il valore nel registro di sistema AX_REG è maggiore del valore in BX_REG.

CMD JMP_AX_LESS_BX – (codice comando 0x09) – salto all'indirizzo specificato nel campo uData, se il valore nel registro di sistema AX_REG è minore del valore in BX_REG.

CMD_CALL – (codice comando 0x0A) – chiamata a una subroutine che inizia all'indirizzo specificato nel campo uData.

CMD_RETURN – (codice comando 0x0B) – ritorno dalla subroutine.

CMD_FOR – (codice comando 0x0C) – esecuzione ciclica di una sequenza di comandi. Il byte alto del campo uData contiene il numero di comandi che si trovano dopo il comando **CMD_FOR** che verranno ripetuti in un ciclo. Il byte meno

significativo del campo uData contiene il numero di ripetizioni. Ad esempio: uData = 0x1705 - 0x17 = 23 comandi eseguiti in un ciclo, 0x05 = 5 - il numero di ripetizioni.

CMD_DJMP_GX_PARI_IX – (codice comando 0x63) – salto all'indirizzo specificato nel campo uData, se il contenuto di una coppia di registri di sistema GX_REG e HX_REG è uguale al contenuto di IX_REG e JX_REG.

CMD_DJMP_GX_NOPARI_IX – (codice comando 0x64) – salto all'indirizzo specificato nel campo uData, se il contenuto di una coppia di registri di sistema GX_REG e HX_REG non è uguale al contenuto di IX_REG e JX_REG.

CMD_DJMP_GX_MORE_IX – (codice comando 0x65) – salto all'indirizzo specificato nel campo uData, se il contenuto di una coppia di registri di sistema GX_REG e HX_REG è maggiore del contenuto di IX_REG e JX_REG.

CMD_DJMP_GX_LESS_IX – (codice comando 0x66) – salta all'indirizzo specificato nel campo uData, se il contenuto di una coppia di registri di sistema GX_REG e HX_REG è inferiore al contenuto di IX_REG e JX_REG.

Sono possibili operazioni matematiche tra registri di sistema, registri Modbus e costanti. Gli operandi delle istruzioni che operano su dati a 32 bit si trovano in due registri a 16 bit consecutivi. Quando si accede a un registro, l'istruzione specifica l'indirizzo del registro inferiore. L'indirizzo del registro superiore si ottiene incrementando di uno l'indirizzo del registro inferiore.

La tabella seguente mostra le istruzioni matematiche, i codici istruzione per le operazioni con dati a 16 bit e 32 bit e le posizioni degli operandi:

Comando			Operando 1	Operando 2	Risultato
Nome	Codice				
	Dati a 16 bit	Dati a 32 bit	S1	S2	D
Addizione (S1 + S2 = D)					
CMD_ADD_SYSREG_CONST	0x19	0x3E	SYS_REG_1	CONST_1	SYS_REG_1
CMD_ADD_SYSREG_MODBUS	0x21	0x46	SYS_REG_1	Modbus_REG	SYS_REG_1
CMD_ADD_MODBUS_CONST	0x28	0x4D	Modbus_REG	CONST_2	Modbus_REG
CMD_ADD_MODBUS_SYSREG	0x30	0x55	Modbus_REG	SYS_REG_1	Modbus_REG
CMD_ADD_SYSREG_SYSREG	0x37	0x5C	SYS_REG_1	SYS_REG_2	SYS_REG_1
Sottrazione (S1 - S2 = D)					
CMD_SUB_SYSREG_CONST	0x1A	0x3F	SYS_REG_1	CONST_1	SYS_REG_1
CMD_SUB_SYSREG_MODBUS	0x22	0x47	SYS_REG_1	Modbus_REG	SYS_REG_1
CMD_SUB_MODBUS_CONST	0x29	0x4E	Modbus_REG	CONST_2	Modbus_REG
CMD_SUB_MODBUS_SYSREG	0x31	0x56	Modbus_REG	SYS_REG_1	Modbus_REG
CMD_SUB_SYSREG_SYSREG	0x38	0x5D	SYS_REG_1	SYS_REG_2	SYS_REG_1
Divisione, resto scartato (S1 / S2 = D)					
CMD_DIV_SYSREG_CONST	0x1B	0x40	SYS_REG_1	CONST_1	SYS_REG_1
CMD_DIV_SYSREG_MODBUS	0x23	0x48	SYS_REG_1	Modbus_REG	SYS_REG_1
CMD_DIV_MODBUS_CONST	0x2A	0x4F	Modbus_REG	CONST_2	Modbus_REG
CMD_DIV_MODBUS_SYSREG	0x32	0x57	Modbus_REG	SYS_REG_1	Modbus_REG
CMD_DIV_SYSREG_SYSREG	0x39	0x5E	SYS_REG_1	SYS_REG_2	SYS_REG_1
Moltiplicazione (S1 * S2 = D)					
CMD_MUL_SYSREG_CONST	0x1C	0x41	SYS_REG_1	CONST_1	SYS_REG_1
CMD_MUL_SYSREG_MODBUS	0x24	0x49	SYS_REG_1	Modbus_REG	SYS_REG_1

CMD_MUL_MODBUS_CONST	0x2B	0x50	Modbus_REG	CONST_2	Modbus_REG
CMD_MUL_MODBUS_SYSREG	0x33	0x58	Modbus_REG	SYS_REG_1	Modbus_REG
CMD_MUL_SYSREG_SYSREG	0x3A	0x5F	SYS_REG_1	SYS_REG_2	SYS_REG_1
AND logico bit per bit (S1 & S2 = D)					
CMD_AND_SYSREG_CONST	0x1D	0x42	SYS_REG_1	CONST_1	SYS_REG_1
CMD_AND_SYSREG_MODBUS	0x25	0x4A	SYS_REG_1	Modbus_REG	SYS_REG_1
CMD_E_COSTANTI_MODBUS	0x2C	0x51	Modbus_REG	CONST_2	Modbus_REG
CMD_AND_MODBUS_SYSREG	0x34	0x59	Modbus_REG	SYS_REG_1	Modbus_REG
CMD_AND_SYSREG_SYSREG	0x3B	0x60	SYS_REG_1	SYS_REG_2	SYS_REG_1
OR logico bit per bit (S1 S2 = D)					
CMD_OR_SYSREG_CONST	0x1E	0x43	SYS_REG_1	CONST_1	SYS_REG_1
CMD_OR_SYSREG_MODBUS	0x26	0x4B	SYS_REG_1	Modbus_REG	SYS_REG_1
CMD_OR_MODBUS_CONST	0x2D	0x52	Modbus_REG	CONST_2	Modbus_REG
CMD_OR_MODBUS_SYSREG	0x35	0x5A	Modbus_REG	SYS_REG_1	Modbus_REG
CMD_OR_SYSREG_SYSREG	0x3C	0x61	SYS_REG_1	SYS_REG_2	SYS_REG_1
XOR logico bit per bit (S1 ^ S2 = D)					
CMD_XOR_SYSREG_CONST	0x1F	0x44	SYS_REG_1	CONST_1	SYS_REG_1
CMD_XOR_SYSREG_MODBUS	0x27	0x4C	SYS_REG_1	Modbus_REG	SYS_REG_1
CMD_XOR_MODBUS_CONST	0x2E	0x53	Modbus_REG	CONST_2	Modbus_REG
CMD_XOR_MODBUS_SYSREG	0x36	0x5B	Modbus_REG	SYS_REG_1	Modbus_REG
CMD_XOR_SYSREG_SYSREG	0x3D	0x62	SYS_REG_1	SYS_REG_2	SYS_REG_1
Spostamento dati (S1 >> S2 = D o S1 << S2 = D – la direzione dello spostamento è indicata nel campo bTypeJmp: 0 –sinistra, 1 - destra)					
CMD_SH_SYSREG_CONST	0x20	0x45	SYS_REG_1	CONST_1	SYS_REG_1
CMD_SH_MODBUS_CONST	0x2F	0x54	Modbus_REG	CONST_2	Modbus_REG

CONST_1 - Costante nel campo dati uData

CONST_2 - Una costante nella seguente riga di comando uCMD_DATA

SYS_REG_1 è il valore contenuto nel registro di sistema (o in due registri di sistema consecutivi per le istruzioni a 64 bit) in uAdrSysReg. Indirizzi dei registri di sistema 0..9

SYS_REG_2 il valore è contenuto nel registro di sistema (o in due registri di sistema consecutivi per dati a 32 bit) il cui indirizzo è specificato nella seguente riga di comando uCMD_DATA

Modbus_REG – il valore contenuto nel registro Modbus: tipo di registro nel campo TypeModbusReg, indirizzo del registro nel campo uData

Importante: quando l'esecuzione di un programma utente viene interrotta dal comando CMD_STOP_PROGRAM, lo stato del motore e di tutti i registri rimane lo stesso impostato durante il programma. Per questo motivo, se il motore era in rotazione al momento dell'esecuzione del comando CMD_STOP_PROGRAM, continuerebbe a eseguire l'ultima attività, ovvero il movimento continuerebbe al termine del programma. Al termine del programma, la rotazione dell'azionamento può essere arrestata scrivendo un registro tramite Modbus (Coils 2001h o Coils 2002h). Per evitare rotazioni incontrollate, è possibile impostare un comando di arresto del motore prima del comando di fine programma o utilizzare il comando CMD_END_STOP_PROGRAM.

7. Ripristino delle impostazioni di fabbrica

Se necessario, i parametri del controller possono essere ripristinati ai valori di fabbrica. Per fare ciò, prima di accendere l'unità, chiudere il primo contatto del connettore RJ11 - CLR_FLASH_CON (connettore RS-485, vedere Fig. 4) con la massa GND dell'alimentatore. I LED rosso e verde sull'unità si accenderanno alternativamente. Tenere CLR_FLASH_CON e GND chiusi per 5 secondi. Dopodiché, i parametri del controller verranno ripristinati ai valori di fabbrica.

8. Fornitura completa

Regolatore per motore CC a spazzole RS 434544 / RS 434546

1 pz

9. Informazioni sul produttore

RS Components aderisce alla linea di sviluppo continuo e si riserva il diritto di apportare modifiche e miglioramenti al design e al software del prodotto senza preavviso.

Le informazioni contenute in questo manuale sono soggette a modifiche in qualsiasi momento e senza preavviso.

10. Garanzia

Qualsiasi riparazione o modifica viene eseguita dal produttore o da un'azienda autorizzata.

Il produttore garantisce il funzionamento senza guasti del controller per 12 mesi dalla data di vendita, a condizione che le condizioni di funzionamento siano soddisfatte.

Indirizzo del reparto vendite del produttore:



RS Components Ltd, Birchington Rd, Corby, NN17 9RS, United Kingdom, rs-online.com

RS Components GmbH, Mainzer Landstrasse 180, 60327 Frankfurt/Main, Germany, rs-online.com