



Manuel d'utilisation

Contrôleur de moteur à courant continu à balais

RS Stock No.: 434544

RS Stock No.: 434546

1. Désignation du produit

Les contrôleurs de moteur à balais RS 434544 et RS 434546 sont des dispositifs électroniques conçus pour faire fonctionner et contrôler les moteurs à courant continu à balais.

2. Fonctions et possibilités

Les contrôleurs sont conçus pour contrôler la vitesse, l'accélération, la décélération et le sens de rotation du moteur. Les unités permettent également le positionnement basé sur les signaux d'un encodeur en cas d'utilisation d'un moteur avec des capteurs à effet Hall.

Le moteur est contrôlé soit par des signaux externes, soit par des commandes transmises via RS-485 par le protocole Modbus. Le contrôleur peut également fonctionner selon un algorithme préalablement enregistré dans sa mémoire par un utilisateur.

Contrôle via RS-485 Modbus

Le contrôleur peut être contrôlé à distance via la ligne de communication physique RS-485 en utilisant le protocole industriel Modbus :

- Le réglage s'effectue en écrivant ou en lisant les paramètres correspondants dans/depuis les registres du contrôleur.
- Les protocoles pris en charge sont RTU et ASCII, les débits sont 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 57600, 115200, 128000 bauds.
- Des déplacements vers une position prédéfinie cible ou l'un des quatre points de référence prédéfinis sont mis en œuvre.
- Le contrôleur peut fonctionner de manière autonome sous le contrôle d'un programme utilisateur (jusqu'à 1024 commandes), qui est préenregistré dans une mémoire non volatile. Les sauts conditionnels et inconditionnels (relatifs et absolus), les appels de sous-programmes, les boucles et les temporisateurs sont pris en charge.
- Les entrées programmables IN1 et IN2 peuvent être utilisées comme signaux START/STOP, REVERSE, ou à d'autres fins à la discrétion de l'utilisateur.
- Le contrôleur peut effectuer un positionnement dans la plage de -2 147 483 647 à +2 147 483 648 impulsions s'il utilise un moteur avec des capteurs à effet Hall.
- Le contrôleur possède des contacts RT sur le panneau avant pour la connexion d'une résistance de terminaison.

Contrôle à l'aide de signaux externes

Pour contrôler le moteur par des signaux externes, les éléments suivants sont fournis :

- Contacts pour la connexion des signaux d'entrée externes IN1 et IN2, dont l'objectif et le traitement sont déterminés par l'utilisateur. Les entrées peuvent également être utilisées comme signaux START/STOP (démarrage/arrêt du mouvement) et DIR (direction).
- Contact HARD STOP pour la connexion d'un signal d'alarme - freinage du moteur contrôlé en cas de coupure du circuit d'urgence.

Le contrôleur offre une fonction de protection contre les surintensités du moteur. Le courant maximal autorisé dans la phase du moteur est défini par l'utilisateur.

3. Caractéristiques techniques

Modèle	RS 434544	RS 434546
Tension d'alimentation, VDC	12 - 24	
Protection de l'alimentation, VDC	8 - 30	
Courant nominal dans la phase moteur, A	<20	<40
Protection matérielle contre les courts-circuits (temps de fonctionnement 15 µs), A	30	100
Limitation du courant de phase, A	1...20	2...40
Tension maximale dans la phase moteur, V	0,98 x U _{sup}	
Minimum non nul dans la phase moteur, V	0,01 x U _{sup}	
Paramètres des signaux externes IN1 et IN2 :		
Résistance maximale des contacts fermés, kOhm	4.7	
Courant d'entrée maximal, mA	0.5	
Dimensions (hors tout), mm	116x100x23	
Interface de communication	RS-485, Modbus – ASCII ou RTU	

Les dimensions hors tout et de raccordement du contrôleur sont indiquées sur la Fig. 1.

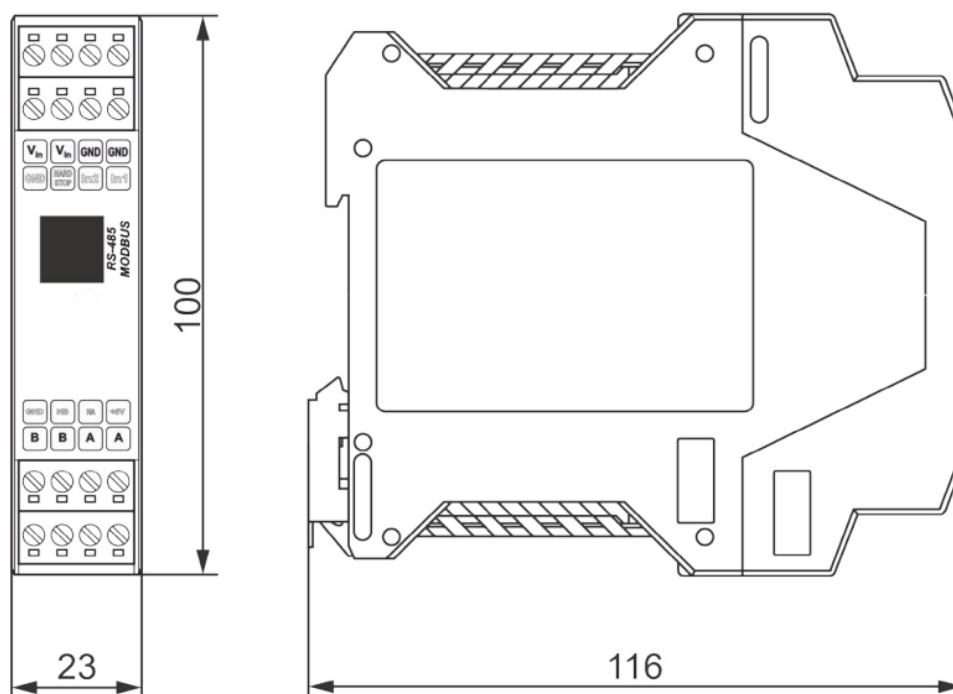


Fig. 1. Dimensions hors tout et de raccordement.

Le schéma de connexion est illustré à la Fig. 2

Conditions environnementales :

Température ambiante : 0...+50 °C

Humidité : 90 % HR ou moins à +25 °C

Condensation et gel : aucun

4. Construction et éléments de commande

Le contrôleur est conçu comme une carte de circuit imprimé avec des éléments électroniques, recouverte d'un boîtier avec un montage sur rail DIN. Sur le dessus du boîtier, il y a des symboles graphiques pour les commandes et l'affectation des broches. Outre les composants électroniques, la carte comporte des éléments d'indication et de commande, des bornes de connexion et des connecteurs :

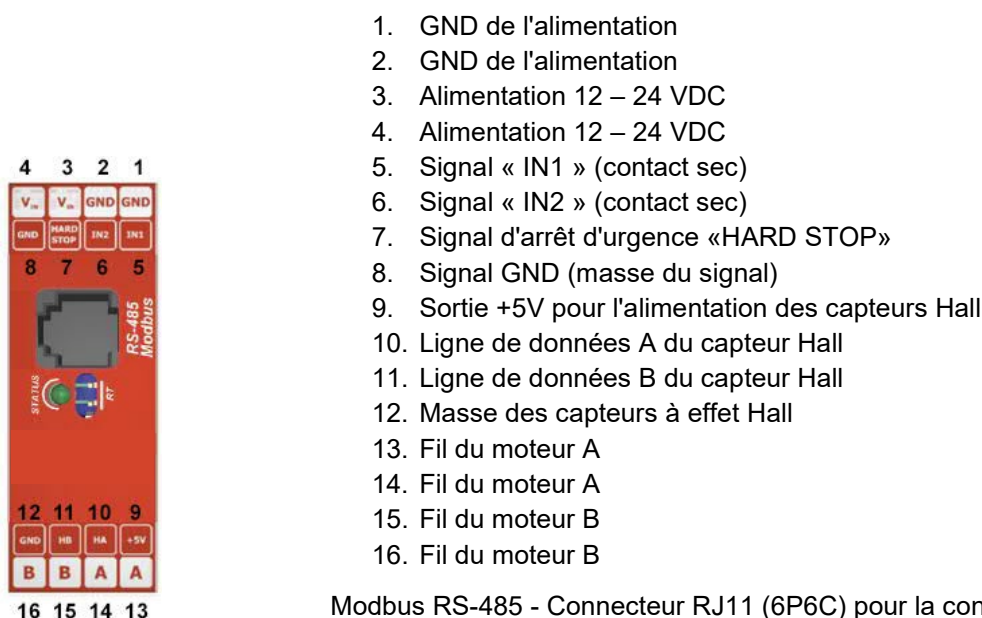
- bornes à vis pour la connexion de l'alimentation, des enroulements du moteur à courant continu à balais, des lignes de l'encodeur et du circuit de commande ;

- bornes IN1 et IN2 pour la connexion des signaux d'entrée de commande ;
- borne pour la connexion des contacts du signal d'arrêt d'urgence ;
- Voyant LED de fonctionnement de l'appareil ;
- Connecteur RJ11 (6P6C) pour la connexion des lignes RS-485 ;
- Contacts pour la connexion de la résistance terminale interne RT ;
- Circuit de freinage intégré pour absorber l'énergie générée par le moteur (pendant la décélération, rotation forcée).

Une entrée de signal externe «HARD STOP» est prévue pour l'arrêt d'urgence du moteur. Les entrées externes IN1 et IN2 peuvent être utilisées pour démarrer et arrêter le moteur à l'aide de signaux externes, ainsi que pour contrôler le sens de rotation du moteur.

Tous les paramètres de fonctionnement du moteur et de contrôle du mouvement peuvent être exécutés par logiciel et par des commandes transmises via RS-485 via le protocole Modbus.

La disposition et la fonction des bornes sont illustrées à la Fig. 2



Modbus RS-485 - Connecteur RJ11 (6P6C) pour la connexion des lignes de données RS-485

RT - contacts pour connecter la résistance de terminaison

Fig. 2. Disposition et fonction des bornes et des éléments de commande.

5. Assemblage et connexion

Veuillez lire attentivement ce manuel avant la connexion et l'assemblage.

Veuillez effectuer le câblage uniquement lorsque l'alimentation est coupée. N'essayez pas de modifier le câblage lorsque l'alimentation est sous tension.

Veuillez assurer un contact fiable dans les bornes de connexion. Pendant le câblage, veuillez respecter la polarité et la gestion des fils. Une inversion de polarité et une surtension endommageront le contrôleur.

IMPORTANT : En raison des courants élevés, il est recommandé de placer l'alimentation à proximité de l'unité et d'utiliser des fils d'une section de 3 mm² (AWG-12). L'alimentation doit fournir 20 % de courant en plus du maximum possible consommé pendant le fonctionnement.

Longueur recommandée des fils d'alimentation :

- Pas plus de 100 cm pour des courants jusqu'à 10 A.
- Pas plus de 50 cm pour des courants de 10 à 20 A.
- Pas plus de 25 cm pour des courants de 20 à 40 A.

Pour un courant maximal jusqu'à 20 A, il est permis d'utiliser les bornes d'alimentation et de phase sur une seule ligne. Pour un courant maximal supérieur à 20 A, les deux lignes des bornes d'alimentation et de phase doivent être utilisées.

Suivez les instructions suivantes lors de la connexion :

1. Connectez un moteur au contrôleur conformément à la Fig. 2. Le moteur doit être connecté aux bornes A et B (13 – 16) du contrôleur. Les signaux du capteur Hall doivent être connectés aux bornes HA et HB (10 – 11). La masse des capteurs Hall doit être connectée à la borne GND (12), l'alimentation des signaux des capteurs Hall doit être connectée à la borne +5V (9).
2. Connectez les éléments de commande externes selon les schémas de connexion de la Fig. 3 :

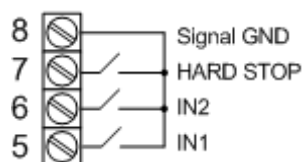


Fig. 3. Connexion des éléments de commande externes.

Type de signaux externes « IN1 », « IN2 », «HARD STOP» - contact sec.

3. Connectez les lignes de l'interface RS-485 au connecteur RJ11 conformément à la Fig. 4.

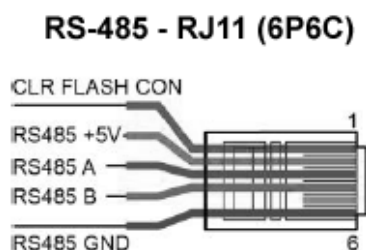


Fig. 4. Attribution des broches du connecteur RJ11 - RS-485 Modbus.

4. Si nécessaire, placez un cavalier sur les broches RT pour connecter la résistance terminale interne.
5. Connectez l'appareil au bloc d'alimentation en respectant la polarité. Le bloc d'alimentation doit être sélectionné avec une marge (pour éviter les chutes de tension). L'épaisseur des fils de connexion doit correspondre à la consommation de courant du moteur. Connectez le « + » de l'alimentation aux bornes 3 et 4, connectez le « - » de l'alimentation aux bornes 1 et 2 du contrôleur. La procédure de démontage est en ordre inverse.

6. Fonctionnement

1. Assurez-vous que l'alimentation est coupée. Veuillez effectuer le câblage uniquement lorsque l'alimentation est coupée.
2. Connectez le moteur et l'alimentation au contrôleur conformément à la section 5.
3. Sélectionnez la méthode de contrôle : contrôle par commandes via Modbus ou signaux externes (registre MODE_DEVICE - voir section 6.5, Fig. 6 et Fig. 7).

4. Connectez les signaux de commande "IN1", "IN2" et le signal d'arrêt d'urgence "HARD STOP" conformément à la section 5. Le signal "HARD STOP" est utilisé pour un arrêt d'urgence du moteur. Le fonctionnement est autorisé avec un contact fermé.
5. Connectez les lignes de l'interface RS-485 conformément à la section 5.
6. Mettez l'alimentation sous tension. L'appareil est prêt pour la configuration via le protocole Modbus.
7. Définissez les paramètres de fonctionnement nécessaires par des commandes via le protocole Modbus - limitation du courant moteur, sens de rotation, réglage du fonctionnement des entrées externes IN1 et IN2, et méthode de contrôle.
8. Pour contrôler le variateur à l'aide du protocole Modbus, envoyez des commandes via l'interface RS-485. Le tableau des registres du contrôleur, leur fonction et les commandes possibles de contrôle du moteur sont indiqués ci-dessous.
9. Pour contrôler le variateur avec des signaux externes pour démarrer et arrêter le moteur et pour contrôler le sens de rotation, utilisez les signaux IN1 et IN2. La régulation de la vitesse est effectuée par des commandes via le protocole Modbus.

Contrôle MODBUS

Pour la transmission de données via l'interface RS-485, le protocole de communication standard Modbus (ASCII ou RTU) est utilisé.

L'unité de contrôle a les paramètres d'usine suivants :

- ID = 1
- Débit : 115200 bauds
- Contrôle de parité : pair
- Bits de données : 8
- Bit d'arrêt : 1
- MODBUS RTU

6.1. Registres de commande d'entrée

Adresse	Type	Nom	Taille	Description
1000h	Entrée discrète	IN1_bit	1-bit	État du signal d'entrée IN1
1001h	Entrée discrète	IN2_bit	1-bit	État du signal d'entrée IN2
1002h	Entrée discrète	IN_HARD_STOP_bit	1-bit	État du signal d'entrée HARD_STOP
5007h	Registre de maintien	MODE_EXT_IN	16 bits	Configuration des entrées externes IN, IN2.
5013h	Registre de maintien	PRESSED_INPUTS_EXTERN	16 bits	Durée minimale d'impulsion positive aux entrées numériques IN1, IN2 (ms)
5014h	Registre de maintien	WAITED_INPUTS_EXTERN	16 bits	Durée minimale de la partie négative de l'impulsion aux entrées numériques IN1, IN2 (ms)

Registres d'état d'entrée 1000h..1002h sont en lecture seule.

1000h –**IN1_bit**– représente l'état de l'entrée physique IN1.

1001h –**IN2_bit**– représente l'état de l'entrée physique IN2.

1002h –**IN_HARD_STOP_bit**– représente l'état de l'entrée physique HARD_STOP.

Valeurs de registre possibles 1000h..1002h :

- 1 – entrée court-circuitée à la sortie GND
- 0 – entrée ouverte avec la sortie GND

5007h **-MODE_EXT_IN-** la valeur du registre détermine la fonction et la méthode de traitement des signaux IN1 et IN2 (voir description dans la section 6.5).

5013h **-PRESSED_INPUTS_EXTERN** et 5014h **-WAITED_INPUTS_EXTERN-** le temps minimum (défini en ms) des parties positive et négative de l'impulsion aux entrées numériques IN1, IN2 - les registres sont utilisés pour supprimer le rebondissement des contacts (voir Fig. 5).

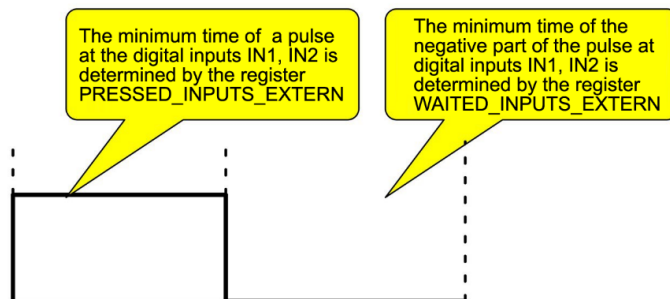


Fig. 5. Suppression du rebondissement des contacts.

6.2. Registres de commande du moteur

Adresse	Type	Nom	Taille	Description
2000h	Bobines	START_bit	1-bit	Démarrer la rotation du moteur avec une accélération définie
2001h	Bobines	STOP_bit	1-bit	Arrêter la rotation du moteur avec une décélération définie
2002h	Bobines	HARD_STOP_bit	1-bit	Arrêt d'urgence brutal de la rotation du moteur
2003h	Bobines	CLR_POSITION_bit	1-bit	Remise à zéro du registre de position actuelle POSITION_VALUE.

Ces registres de contrôle sont disponibles en lecture et en écriture. Lors de l'écriture dans le registre correspondant, la valeur 1 active une fonction ou une autre, après quoi le registre est réinitialisé à 0.

6.3. Registres d'état

Adresse	Type	Nom	Taille	Description
3000h	Registre d'entrée	STATUS	16 bits	L'état actuel du contrôle du moteur.
3001h	Registre d'entrée	CURRENT_VALID	16 bits	La valeur actuelle du courant consommé par le moteur.
3002h	Registre d'entrée	SPEED_VALID	16 bits	La valeur instantanée de la vitesse de rotation.
3003h	Registre d'entrée	CURRENT_POSITION	32 bits	Position actuelle. Les valeurs vont de -2147483647 à +2147483648
3005h	Registre d'entrée	TEMPERATURE_MCU	16 bits	Température du processeur
3006h	Registre d'entrée	TEMPERATURE_MOSFET	16 bits	Température du circuit de puissance
3007h	Registre d'entrée	TEMPERATURE_BRAKE	16 bits	Température du circuit de freinage
3008h	Registre d'entrée	TASK_COUNTER	16 bits	Retourne une valeur aléatoire (pour vérifier le fonctionnement du canal de transmission)
3009h	Registre d'entrée	STATUS_USER_PROGRAM	16 bits	L'état actuel de l'exécution du programme utilisateur

Ce groupe de registres est en lecture seule et représente l'état actuel du contrôleur.

3000h – **STATUS**- état actuel du contrôle du moteur, valeurs possibles :

- 0 arrêt du moteur
- 1 rotation avant
- 2 rotation arrière

3001h – **CURRENT_VALID**- la valeur actuelle du courant consommé par le moteur. Représente la valeur du courant consommé par le moteur de l'alimentation, la valeur en mA.

3002 – **SPEED_VALID**- la valeur instantanée de la vitesse de rotation. Les unités de mesure sont les tours par minute.

3003h – **CURRENT_POSITION**– la position actuelle. Le positionnement est effectué dans la plage de valeurs de -2147483647 à + 2147483648 du nombre total de commutations du capteur Hall, les fronts montants et descendants sont pris en compte.

3005h – **TEMPERATURE_MCU**— température du processeur central. La température est calculée comme $TEMPERATURE_MCU/10$ deg/C.

3006h – **TEMPERATURE_MOSFET**- température du circuit de puissance - la température dans la zone d'installation des commutateurs MOSFET. La température est calculée comme $TEMPERATURE_MOSFET/10$ °C.

3007h – **TEMPERATURE_BRAKE** - température du circuit de freinage – la température au niveau de la résistance de freinage. La température est calculée comme $TEMPERATURE_MOSFET/10$ °C.

3008h – **TASK_COUNTER**- renvoie un nombre aléatoire compris entre 0x0000 et 0xFFFF.

3009h – **STATUS_USER_PROGRAM** - l'état actuel de l'exécution du programme utilisateur, valeurs possibles :

- 1 – le programme utilisateur est arrêté par une commande via Modbus
- 2 – le programme utilisateur est démarré par une commande via Modbus (cet état est de courte durée seulement, avant que l'état 4 ne soit établi)
- 3 – le programme utilisateur est démarré après la mise sous tension (cet état est de courte durée seulement, avant que l'état 4 ne soit établi)
- 4 – le programme utilisateur est en cours d'exécution
- 5 – le programme utilisateur est terminé avec la commande END
- 6 – le programme utilisateur est terminé avec la commande ENDF
- 7 – le programme utilisateur est arrêté en raison d'une erreur

6.4. Registres des paramètres de transmission de données RS-485 Modbus

Adresse	Type	Nom	Taille	Description
5000h	Registre de maintien	SLAVE_ADDRESS_MODBUS	16 bits	ID du contrôleur (adresse de l'appareil). Valeurs valides : 0..247. (0 - adresse de diffusion, pas de message de réponse)
5001h	Registre de maintien	TYPE_MODBUS	16 bits	Paramètres de transmission de données
5002h	Registre de maintien	BITRATE_MODBUS	16 bits	Réglage de la vitesse de transmission
5003h	Registre de maintien	TIMEOUT_BROADCAST_MODBUS	16 bits	Un délai supplémentaire entre le paquet reçu et le message de réponse.

5001h – **TYPE_MODBUS**– paramètres de transmission de données, valeurs valides :

- 1 – ASCII, 7 bits de données, pair, 1 bit d'arrêt,
- 2 – ASCII, 7 bits de données, impair, 1 bit d'arrêt
- 3 – RTU, 8 bits de données, pair, 1 bit d'arrêt

- 4 – RTU, 8 bits de données, impair, 1 bit d'arrêt
- 5 – RTU, 8 bits de données, aucun, 2 bits d'arrêt

5002h -**BITRATE_MODBUS**– débit en bauds Modbus, valeurs possibles :

- 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**- est utilisé si le dispositif de contrôle met du temps à passer du mode émission au mode réception.

Après avoir modifié les paramètres de communication, les nouvelles valeurs doivent être enregistrées à l'aide du registre FLAG_SAVE_INI (voir section 6.5), puis le contrôleur doit être redémarré. Après le redémarrage, la connexion RS-485 sera effectuée en utilisant les nouveaux paramètres.

6.5. Registres pour le réglage du fonctionnement du variateur

Adresse	Type	Nom	Taille	Description
5004h	Registre de maintien	MODE_DEVICE	16 bits	Mode de fonctionnement du contrôleur.
5005h	Registre de maintien	MODE_USER_PROGRAM	16 bits	Commande de contrôle pour le démarrage d'un programme utilisateur.
5006h	Registre de maintien	MODE_ROTATION	16 bits	Mode de rotation.
5007h	Registre de maintien	MODE_EXT_IN	16 bits	Configuration des entrées externes IN, IN2.
5008h	Registre de maintien	POSITION_N	16 bits	Le numéro de position à atteindre. Les valeurs vont de 1 à 4
5009h	Registre de maintien	REF_CURRENT	16 bits	Limitation de la consommation de courant. Plage : 1000 mA à 20000 mA pour RS 434544 2000 mA à 40000 mA pour RS 434546
500Ah	Registre de maintien	RATED_SPEED	16 bits	Vitesse nominale du moteur : Plage de 1000 à 15000 tr/min
500Bh	Registre de maintien	SPEED	16 bits	Régler la vitesse de rotation. Plage de 30 à 15000 tr/min.
500Ch	Registre de maintien	ACC	16 bits	Régler l'accélération. Plage de 10 à 1000.
500Dh	Registre de maintien	DEC	16 bits	Régler la décélération. Plage de 10 à 1000.
500Eh	Registre de maintien	DIRECTION	16 bits	Sens de rotation. 1 – avant 2 – en arrière

500Fh	Registre de maintien	PULSES_PER_REVOLUTION	16 bits	Nombre d'impulsions par tour depuis un capteur Hall : Plage de 1 à 12
5010h	Registre de maintien	USE_HALL	16 bits	Utilisation des capteurs à effet Hall : 0 – sans capteurs à effet Hall 1 – un capteur à effet Hall 2 – deux capteurs à effet Hall
5011h	Registre de maintien	MODE_COIL	16 bits	État des bornes du moteur à l'arrêt : 0 - ouvert 1 - fermé
5012h	Registre de maintien	OFFSET_COMPENSATION	16 bits	Correction du freinage du moteur
5013h	Registre de maintien	PRESSED_INPUTS_EXTERN	16 bits	Durée minimale d'impulsion positive aux entrées numériques IN1, IN2 (ms)
5014h	Registre de maintien	WAITED_INPUTS_EXTERN	16 bits	Durée minimale de la partie négative de l'impulsion aux entrées numériques IN1, IN2 (ms)
5015h	Registre de maintien	OFFSET	32 bits	Le décalage à déplacer, la valeur est modifiée par le contrôleur pendant le fonctionnement. Plage de valeurs de -2147483647 à +2147483648
5017h	Registre de maintien	OFFSET_CONST	32 bits	Incrément - le décalage auquel il est nécessaire de se déplacer.
5019h	Registre de maintien	TARGET_POSITION	32 bits	La position à atteindre. Plage de valeurs de -2147483647 à +2147483648
501Bh	Registre de maintien	TARGET_POSITION1	32 bits	Position prééglée par l'utilisateur n°1 Plage de valeurs de -2147483647 à +2147483648
501Dh	Registre de maintien	TARGET_POSITION2	32 bits	Position prééglée par l'utilisateur n°2 Plage de valeurs de -2147483647 à +2147483648
501Fh	Registre de maintien	TARGET_POSITION3	32 bits	Position prééglée par l'utilisateur n°3 Plage de valeurs de -2147483647 à +2147483648
5021h	Registre de maintien	TARGET_POSITION4	32 bits	Position prééglée par l'utilisateur n°4 Plage de valeurs de -2147483647 à +2147483648
5023h	Registre de maintien	ERROR	16 bits	Registre des erreurs
5024h	Registre de maintien	FLAG_SAVE_INI	16 bits	Registre pour la sauvegarde des paramètres utilisateur. (Registres 5000h..501Fh). Valeur : 0x37FA
5025h	Registre de maintien	FLAG_SAVE_USER_PROGRAM	16 bits	Registre pour écrire ou lire un programme utilisateur vers ou depuis la mémoire non volatile.

				Valeur à écrire : 0x8426 Valeur à lire : 0x9346
5026h	Registre de maintien	FLAG_RESTART	16 bits	Registre de redémarrage. Valeur : 0x95AF

5004h –**MODE_DEVICE**– mode de fonctionnement du contrôleur, valeurs possibles :

- 1 – contrôle via Modbus
- 2 – contrôle par signaux externes

5005h –**MODE_USER_PROGRAM**– commande de contrôle pour démarrer un programme utilisateur, valeurs possibles :

- 1 – arrêter le programme utilisateur
- 2 – démarrer le programme utilisateur
- 3 – démarrer le programme utilisateur dès la mise sous tension (pré-enregistrement des paramètres requis - voir registre FLAG_SAVE_INI)

Le programme utilisateur peut être stocké dans la mémoire flash interne du contrôleur. Pour lancer l'exécution d'un programme utilisateur, il est nécessaire d'écrire 2 dans le registre **MODE_USER_PROGRAM**. Pour arrêter l'exécution du programme, il est nécessaire d'écrire 1 dans le registre **MODE_USER_PROGRAM**.

5006h –**MODE_ROTATION**– mode de rotation, valeurs possibles :

- 1 – rotation continue
- 2 – décalage de la quantité spécifiée par le registre **OFFSET**
- 3 – déplacement vers une position donnée. Le numéro de position est spécifié par le registre **POSITION_N** (de 1 à 4), les coordonnées des positions sont spécifiées par les registres **POSITION1**, **POSITION2**, **POSITION3** et **POSITION4** en conséquence.

5007h –**MODE_EXT_IN**– configuration du mode de fonctionnement des entrées externes IN1, IN2 dans le mode de contrôle des signaux externes, valeurs possibles :

- 1 – L'entrée IN1 est utilisée comme signal de démarrage/arrêt du variateur, elle est traitée sur le front descendant de l'impulsion ; l'entrée IN2 est utilisée comme signal d'inversion, elle est traitée sur le front descendant de l'impulsion.
- 2 - L'entrée IN1 est utilisée comme signal de démarrage/arrêt du variateur, elle est traitée sur le front descendant de l'impulsion ; l'entrée IN2 est utilisée comme signal de référence de direction, traité en fonction du niveau du signal.
- 3 - L'entrée IN1 est utilisée comme signal de démarrage/arrêt du variateur, elle est traitée en fonction du niveau du signal : présence d'un signal - activer la rotation du moteur, absence de signal - arrêt ; l'entrée IN2 est utilisée comme signal d'inversion, elle est traitée sur le front descendant de l'impulsion.
- 4 - L'entrée IN1 est utilisée comme signal de démarrage/arrêt du variateur, elle est traitée en fonction du niveau du signal : présence d'un signal - activer la rotation du moteur, absence de signal - arrêt ; l'entrée IN2 est utilisée comme signal de référence de direction, traitée en fonction du niveau du signal.
- 5 – l'entrée IN1 est utilisée comme signal pour démarrer et arrêter le variateur dans le sens avant, IN2 - comme signal pour démarrer et arrêter le variateur dans le sens inverse ; les deux signaux sont traités par niveau.

5008h –**POSITION_N**– sélection du numéro de position pour le déplacement (de 1 à 4) - utilisé dans le mode de déplacement vers une position donnée (**MODE_ROTATION** = 3) en conjonction avec les registres **POSITION1**, **POSITION2**, **POSITION3**, **POSITION4**.

5009h –**REF_CURRENT**– réglage de la limite de consommation de courant. La plage de valeurs admissibles pour

RS 434544 : de 1000 mA à 20000 mA ; pour le RS 434546 : de 2000 mA à 40000 mA.

500Ah –**RATED_SPEED**– vitesse nominale du moteur : de 1000 à 15000 tr/min. Il est impossible de connaître la vitesse réelle du moteur si celui-ci est utilisé sans capteurs à effet Hall. Pour cette raison, la vitesse maximale du moteur (qui correspond à l'alimentation maximale du moteur) est utilisée pour calculer les valeurs de consigne de la vitesse. Si la **RATED_SPEED** n'est pas correctement définie, la vitesse réelle du moteur diffère sensiblement de la valeur cible, qui est définie dans le registre **SPEED**.

500Bh –**SPEED**– vitesse de rotation cible (de 30 à 15000 tr/min). La valeur de la vitesse cible doit être inférieure à la valeur **MAX_SPEED**.

500Ch –**ACC**– accélération donnée (de 10 à 1000)

500Dh –**DEC**– décélération donnée (de 10 à 1000)

Les valeurs d'accélération et de décélération dans les registres **ACC** et **DEC** sont des valeurs linéaires conventionnelles qui déterminent le taux d'accélération/décélération. Une valeur de 10 correspond à 100 tr/s², une valeur de 1000 correspond à 5000 tr/s².

500Eh –**DIRECTION**– sens de rotation, valeurs possibles :

- 1 – rotation avant
- 2 – rotation arrière.

500Fh –**PULSES-PER_REVOLUTION**– nombre d'impulsions par tour à partir d'un capteur à effet Hall (de 1 à 12).

5010h –**USE_HALL**– utilisation des capteurs à effet Hall :

- 0 – un moteur sans capteurs à effet Hall
- 1 – un capteur à effet Hall
- 2 – deux capteurs à effet Hall.

5011h –**MODE_COIL**– État des bornes du moteur à l'arrêt :

- 0 - ouvert
- 1 - fermé

5012h –**OFFSET_COMPENSATION**– Correction de freinage du moteur calculée expérimentalement pour les réglages actuels **ACC**, **DEC**, **SPEED**. Si l'arrêt se produit au-delà du point calculé, la valeur de compensation est égale à l'erreur de positionnement avec un signe négatif, si le moteur s'arrête avant le point d'arrêt, la valeur de compensation est positive. Par exemple, l'erreur d'atteinte de la position spécifiée est de 15 incréments, ce qui signifie **OFFSET_COMPENSATION** = -15, et cette correction n'est valable que pour les réglages actuels d'accélération, de décélération et de vitesse.

5013h et 5014h –**PRESSED_INPUTS_EXTERN** et **WAITED_INPUTS_EXTERN**– durée minimale d'une partie positive et d'une partie négative d'une impulsion aux entrées numériques **IN1**, **IN2** – utilisée pour supprimer le rebondissement des contacts (voir fig. 5).

5015h –**OFFSET**– le décalage à déplacer est utilisé lorsque **MODE_ROTATION** = 2, valeurs valides de -2147483647 à +2147483648. Avant de commencer le mouvement, il est nécessaire de définir la valeur de décalage requise dans le registre **OFFSET** ou **OFFSET_CONST**, qui est traitée par le contrôleur comme un compteur du mouvement restant. Pendant l'exécution du mouvement spécifié, la valeur **OFFSET** diminue.

5017h –**OFFSET_CONST**-Incrément - le décalage qu'il est nécessaire d'appliquer. Si la valeur du registre **OFFSET_CONST** ≠ 0, elle est copiée dans le registre **OFFSET** à chaque démarrage du moteur.

5019h –**TARGET_POSITION**– position à atteindre, valeurs autorisées de -2147483647 à +2147483648. En mode déplacement vers une position donnée, la coordonnée de **POSITION1-POSITION4** est copiée dans ce registre avant le déplacement, tandis que les registres **POSITION1-POSITION4** ne changent pas leurs valeurs.

501Bh - 5021h –**TARGET_POSITION1..4**– Position préréglée par l'utilisateur №1..4, utilisée lorsque **MODE_ROTATION** = 3, le numéro de position est déterminé par le registre **POSITION_N**, valeurs autorisées de -2147483647 à +2147483648.

5023h –**ERROR**– erreurs survenant pendant le fonctionnement du contrôleur - chaque bit du registre signale une erreur spécifique :

- bit 0 - hors de la plage de tension d'alimentation ;
- bit 1 - court-circuit des enroulements du moteur ;
- bit 2 - surchauffe du circuit de freinage ;
- bit 3 - surchauffe du circuit de puissance ;
- bit 4 - erreur de connexion des capteurs Hall ;
- bit 5 - arrêt d'urgence ;
- bit 6 - surchauffe du MCU ;
- bit 7 - programme de contrôle de test ;
- bit 8 - erreur d'exécution du programme utilisateur ;
- bit 9 - erreur de lecture ou d'écriture des paramètres ;
- bit 10 - erreur de fonctionnement des commutateurs à transistors de sortie ;
- bit 11 - la limite de courant de consommation est dépassée ;
- bit 12 - avertissement concernant l'impossibilité de calculer le point d'arrêt ;
- bit 13 - avertissement concernant une tentative d'écriture dans le registre d'une valeur hors plage ;
- bit 14 – erreur de parité de transmission RS-485 ;
- bit 15 - utilisation du mode de positionnement avec moins de deux capteurs Hall.

5024h –**FLAG_SAVE_INI**– registre pour la sauvegarde des paramètres utilisateur - lors de l'écriture de la valeur 0x37FA dans ce registre, les paramètres définis par les registres 5000h..501Fh seront enregistrés dans la mémoire non volatile.

5025h –**FLAG_SAVE_USER_PROGRAM**– l'écriture de la valeur 0x8426 dans ce registre lance la procédure d'enregistrement du programme utilisateur du tampon temporaire vers la mémoire non volatile du contrôleur. L'écriture de la valeur 0x9346 lance la procédure de lecture du programme utilisateur de la mémoire non volatile du contrôleur dans un tampon temporaire (voir section 6.6.).

5026h –**FLAG_RESTART**– l'écriture de la valeur 0x95AF dans ce registre provoque le redémarrage du contrôleur.

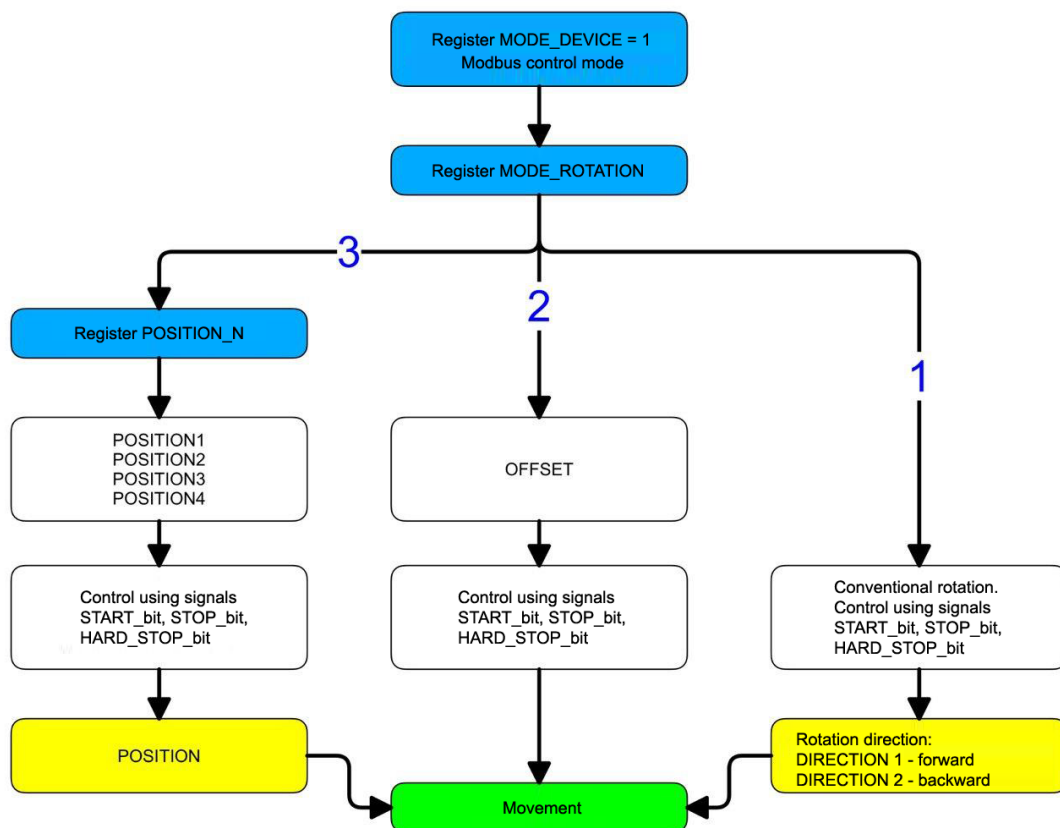


Fig. 6. Organigramme de sélection du mode de fonctionnement lors du contrôle du variateur via Modbus.

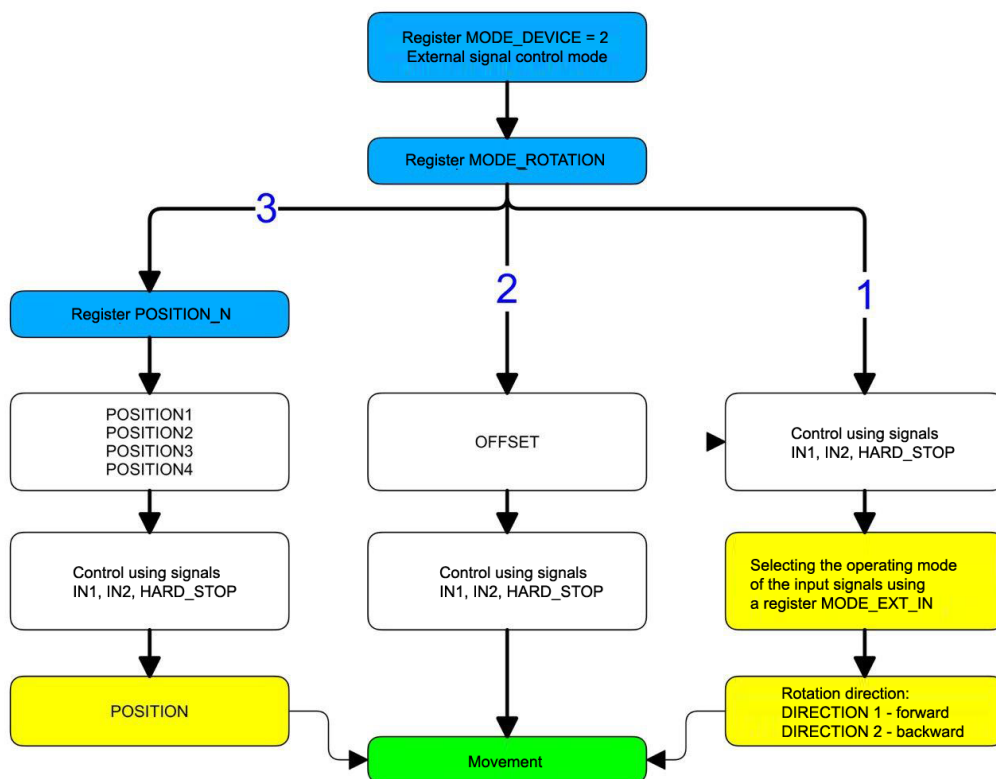


Fig. 7. Organigramme de sélection du mode de fonctionnement lors du contrôle du variateur à l'aide de signaux externes.

6.6. Lecture et écriture d'un programme utilisateur

Un tampon temporaire de 1024 commandes est utilisé pour lire et écrire un programme utilisateur. Le registre spécial FLAG_SAVE_INI est utilisé pour enregistrer le programme du tampon temporaire vers la mémoire non volatile et pour lire le programme de la mémoire du contrôleur vers le tampon temporaire.

Adresse	Type	Nom	Taille	Description
5025h	Registre de maintien	FLAG_SAVE_USER_PROGRAM	16 bits	Registre pour écrire ou lire un programme utilisateur vers ou depuis la mémoire non volatile du contrôleur. Valeur d'écriture : 0x8426 Valeur de lecture : 0x9346
6000h	Registre de maintien	WRITE_CMD	16 bits	Registre d'adresse. Lorsque la valeur d'adresse est écrite dans ce registre, la commande est transférée du champ CMD_W vers l'adresse spécifiée du tampon temporaire du programme utilisateur.
6001h	Registre de maintien	CMD_W	32 bits	Instructions du programme utilisateur
6003h	Registre de maintien	READ_CMD	16 bits	Registre d'adresse. Lorsque la valeur d'adresse est écrite dans ce registre, la commande est transférée de l'adresse spécifiée du tampon temporaire du programme utilisateur vers le champ CMD_R.
6004h	Registre de maintien	CMD_R	32 bits	Instructions du programme utilisateur

Assemblage et écriture d'un programme utilisateur dans la mémoire du contrôleur

Chaque instruction du programme utilisateur est constituée de deux mots mémoire (32 bits) - commande (16 bits) et données de commande (16 bits). Lors de l'assemblage d'un programme utilisateur, les instructions sont d'abord écrites dans un tampon temporaire du contrôleur. Pour écrire dans le tampon temporaire, il est nécessaire d'écrire une instruction dans le registre CMD_W, puis d'écrire l'adresse de cette instruction dans le tampon interne dans le registre WRITE_CMD. Lorsque l'adresse est écrite dans le registre WRITE_CMD, l'instruction est transférée du registre CMD_W vers le tampon interne. Après avoir composé un programme utilisateur dans un tampon temporaire, il est nécessaire d'écrire la valeur 0x8426 dans le registre FLAG_SAVE_USER_PROGRAM - le programme sera transféré du tampon temporaire vers la mémoire FLASH du contrôleur.

Lecture d'un programme utilisateur depuis la mémoire du contrôleur

Il est nécessaire de transférer un programme utilisateur vers la mémoire tampon temporaire du contrôleur pour le lire depuis la mémoire FLASH. Pour ce faire, écrivez la valeur 0x9346 dans le registre FLAG_SAVE_USER_PROGRAM - le programme sera transféré de la mémoire FLASH du contrôleur vers la mémoire tampon temporaire. Ensuite, pour lire une instruction depuis la mémoire tampon temporaire, il est nécessaire d'écrire l'adresse de l'instruction dans le registre READ_CMD - lorsque l'adresse est écrite dans ce registre, l'instruction sera copiée de la mémoire tampon temporaire vers le registre CMD_R.

Un diagramme des procédures de lecture et d'écriture d'un programme utilisateur est présenté à la Fig. 8.

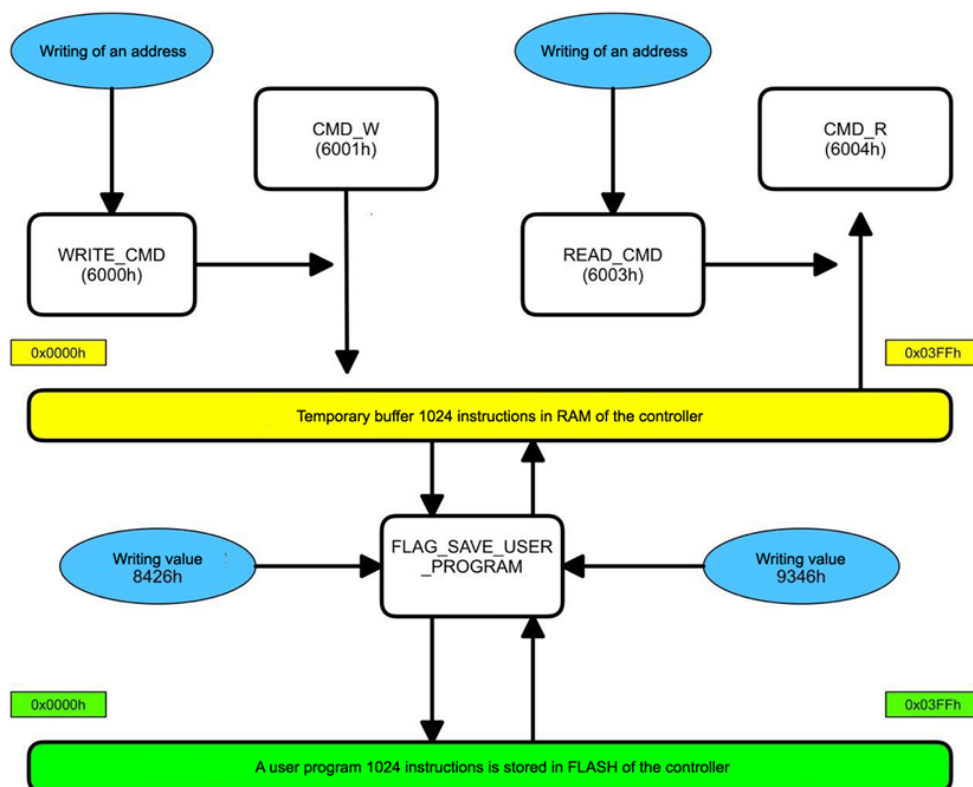


Fig. 8. Organigramme des procédures de lecture et d'écriture d'un programme utilisateur.

6.7. Registres système

Adresse	Type	Nom	Taille	Description
7000h	Registre de maintien	AX_REG	16 bits	Registre de stockage de données pour le programme utilisateur
7001h	Registre de maintien	BX_REG	16 bits	Registre de stockage de données pour le programme utilisateur
7002h	Registre de maintien	CX_REG	16 bits	Registre de stockage de données pour le programme utilisateur
7003h	Registre de maintien	DX_REG	16 bits	Registre de stockage de données pour le programme utilisateur
7004h	Registre de maintien	REG_EX	16 bits	Registre de stockage de données pour le programme utilisateur
7005h	Registre de maintien	FX_REG	16 bits	Registre de stockage de données pour le programme utilisateur
7006h	Registre de maintien	PC_REG	16 bits	Pointeur de registre vers la commande utilisateur en cours d'exécution
7007h	Registre de maintien	GX_REG	16 bits	Registre de stockage de données pour le programme utilisateur
7008h	Registre de maintien	HX_REG	16 bits	Registre de stockage de données pour le programme utilisateur
7009h	Registre de maintien	IX_REG	16 bits	Registre de stockage de données pour le programme utilisateur

700Ah	Registre de maintien	JX_REG	16 bits	Registre de stockage de données pour le programme utilisateur
-------	----------------------	--------	---------	---

Registres système AX_REG..FX_REG (plage de valeurs 0..65535) sont destinés au stockage temporaire des données pendant l'exécution d'un programme utilisateur. PC_REG est un pointeur vers l'instruction du programme utilisateur en cours d'exécution. Description plus détaillée dans la section 6.9.

6.8. Registres d'identification

Adresse	Type	Nom	Taille	Description
8001h	Registre d'entrée	HW_MAJOR	16 bits	Type de pilote
8002h	Registre d'entrée	HW_MINOR	16 bits	Version matérielle
8003h	Registre d'entrée	FW_MAJOR	16 bits	Identifiant du logiciel
8004h	Registre d'entrée	FW_MINOR	16 bits	Version du logiciel

Les registres sont nécessaires pour déterminer la fonction de l'unité de contrôle, ses caractéristiques et la version du logiciel via le réseau.

Pour RS 434544, les valeurs sont :

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

Pour RS 434546, les valeurs sont :

- HW_MAJOR 1002
- HW_MINOR x
- FW_MAJOR x
- FW_MINOR x

6.9. Instructions du programme utilisateur

La structure des instructions du programme utilisateur est illustrée dans le diagramme de la Fig. 9.

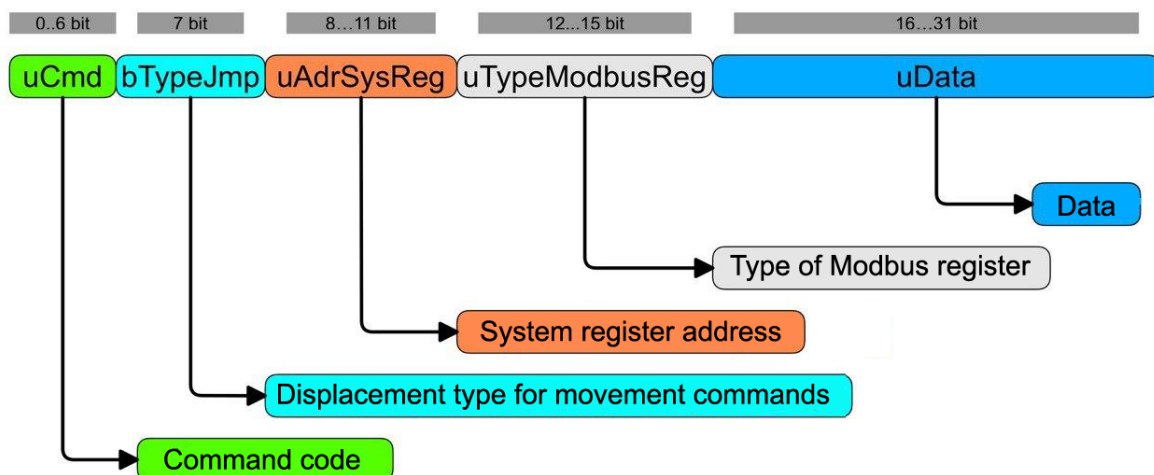


Fig. 9. Structure d'une instruction de programme utilisateur.

L'instruction du programme utilisateur a une taille de 32 bits et contient les champs suivants :

uCmd – 7 bits – code de commande

bTypeJmp – 1 bit – type de déplacement pour les commandes de mouvement :

- 0 – valeur absolue
- 1 – valeur relative

uAdrSysReg – 4 bits - adresse du registre système AX_REG ...FX_REG avec les numéros 0 ... 5

uTypeModbusReg – 4 bits – type de registre Modbus :

- 0 – Entrées discrètes
- 1 – Bobines
- 2 – Entrées
- 3 – Registres de maintien

uData – 16 bits - données.

Les instructions 64 bits sont composées de deux lignes de commande 32 bits. La première ligne de commande de cette instruction est une commande similaire à une instruction 32 bits. La deuxième ligne de commande contient des données **uCMD_DATA** de 16 ou 32 bits.

Les instructions avec un préfixe D, telles que DMOV, opèrent sur des données 32 bits situées dans deux registres 16 bits consécutifs. Dans le champ d'adresse de ces instructions, le registre inférieur est indiqué. Lors de la lecture d'une valeur, la commande lit automatiquement deux registres 16 bits consécutifs à partir de l'adresse spécifiée et forme une seule valeur 32 bits. Lors de l'écriture d'une valeur, l'instruction divise un nombre 32 bits en deux parties 16 bits et les écrit dans deux registres consécutifs à partir de l'adresse spécifiée.

Le tableau suivant répertorie les commandes avec les options de remplissage des champs. Si le champ n'est pas spécifié, il n'est pas utilisé dans cette commande.

uCmd		-	-	-
0x00	CMD_STOP_PROGRAM	-	-	-
0x0D	CMD_FULL_STOP_PROGRAM			
uCmd		bTypeJump	uData	-
0x05	CMD_JMP	0 – valeur absolue 1 – valeur relative	Adresse de déplacement 0..1024 ou décalage +-1024	-
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			
0x64	CMD_DJMP_GX_NOPARI_IX			
0x65	CMD_DJMP_GX_MORE_IX			
0x66	CMD_DJMP_GX_LESS_IX			
uCmd		uData	-	-
0x04	CMD_DELAY	Temps de délai 0..65535 ms	-	-
0x0A	CMD_CALL	Adresse de sous-programme 0..1024		
0x0B	CMD_RETURN	-		
0x0C	CMD_FOR	Durée du cycle et nombre de cycles		
uCmd		uAdrSysReg	-	-
0x15	CMD_NOT_SYSREG	Adresse du registre système (0..9)	-	-
0x17	CMD_DNOT_SYSREG			
uCmd		uTypeModbusReg	uData	-
0x16	CMD_NOT_MODBUS	1 – Bobines 3 – Registres de maintien	Adresse du registre Modbus 0..65535	-
0x18	CMD_DNOT_MODBUS			
uCmd		uAdrSysReg	uData	-
0x01	CMD_MOV_SYSREG_CONST	Adresse du registre système (0..9)	Constante 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		uAdrSysReg	uTypeModbusReg	uData
0x03	CMD_MOV_SYSREG_MODBUS	Adresse du registre système (0..9)	0 – Entrées discrètes 1 – Bobines 2 – Entrées 3 – Registres de maintien	Adresse du registre 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 – Bobines 3 – Registres de maintien	Adresse du registre Modbus 0..65535	Constante 0..65535 (commande séparée ligne)
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		uAdrSysReg	uTypeModbusReg	uData
0x02	CMD_MOV_MODBUS_SYSREG	Adresse du registre système (0..9)	0 – Entrées discrètes 1 – Bobines 2 – Entrées 3 – Registres de maintien	Adresse du registre 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		uAdrSysReg	uData	-
0x0F	CMD_MOV_SYSREG_SYSREG	Adresse du registre système (0..9)	Adresse du registre système (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		uAdrSysReg	uCMD_DATA	-
0x10	CMD_DMOV_SYSREG_CONST	Adresse du registre système (0..9)	Constante 0...4294967295 (commande séparée ligne)	-
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		uAdrSysReg	uTypeModbusReg	uData
0x11	CMD_DMOV_SYSREG_MODBUS	Adresse du registre système (0..9)	0 – Entrées discrètes 1 – Bobines 2 – Entrées 3 – Registres de maintien	Adresse du registre 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 – Bobines 3 – Registres de maintien	Adresse du registre Modbus 0..65535	Constante 0...4294967295 (commande séparée ligne)
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		uAdrSysReg	uTypeModbusReg	uData
0x13	CMD_DMOV_MODBUS_SYSREG	Adresse du registre système (0..9)	0 – Entrées discrètes 1 – Bobines 2 – Entrées 3 – Registres de maintien	Adresse du registre 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		uAdrSysReg	uData	-
0x14	CMD_DMOV_SYSREG_SYSREG	Adresse du registre système (0..9)	Adresse du registre système (0..9)	-
0x5C	CMD_DADD_SYSREG_SYSREG			
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		uAdrSysReg	bTypeJmp	uData
0x20	CMD_SH_SYSREG_CONST	Adresse du registre système (0..9)	0 – décalage à gauche	Valeur de décalage
0x45	CMD_DSH_SYSREG_CONST		1 – décalage à droite	
uCmd		uTypeModbusReg	uData	bTypeJmp
0x2F	CMD_SH_MODBUS_CONST	1 – Bobines 3 – Registres de maintien	Adresse du registre Modbus 0..65535	0 – décalage à gauche
0x54	CMD_DSH_MODBUS_CONST			1 – décalage à droite

CMD_STOP_PROGRAM – (code de commande 0x00) – arrête l'exécution d'un programme utilisateur, sans quitter le mode d'exécution du programme utilisateur. À la fin de l'exécution du programme, tous les registres et états du moteur restent tels qu'ils étaient avant l'exécution de la commande (le moteur continue)

de tourner s'il tournait avant l'exécution de la commande). Avant le prochain démarrage du programme, vous devez envoyer la commande **CMD_FULL_STOP_PROGRAM**.

CMD_FULL_STOP_PROGRAM – (code de commande 0x0D) – arrêt de l'exécution du programme utilisateur et sortie du mode de fonctionnement du programme. À la fin de l'exécution du programme, tous les registres et l'état du moteur reviennent à leurs valeurs d'origine et le moteur s'arrête.

CMD_MOV_SYSREG_CONST – (code de commande 0x01) – écriture dans le registre système à l'adresse uAdrSysReg, des valeurs du champ de données uData.

CMD_MOV_MODBUS_SYSREG – (code de commande 0x02) – écriture du contenu du registre système uAdrSysReg dans l'espace registre ModBUS défini par le champ TypeModbusReg et son adresse dans le champ uData.

CMD_MOV_MODBUS_SYSREG – (code de commande 0x03) – lecture du contenu de l'espace registre ModBUS déterminé par le champ TypeModbusReg et son adresse dans le champ uData dans l'un des registres système uAdrSysReg.

CMD_MOV_MODBUS_CONST – (code de commande 0x0E) – écriture de la constante contenue dans la ligne de commande suivante uCMD_DATA dans l'espace registre ModBUS défini par le champ TypeModbusReg et son adresse dans le champ uData.

CMD_MOV_SYSREG_SYSREG – (code de commande 0x0F) – écriture du contenu du registre système uAdrSysReg dans un autre registre système à l'adresse uData.

CMD_DELAY – (code de commande 0x04) – pause, ms.

CMD_JMP – (code de commande 0x05) – saut à l'adresse spécifiée dans le champ uData.

CMD_JMP_AX_PARI_BX – (code de commande 0x06) – saut à l'adresse spécifiée dans le champ uData, si la valeur du registre système AX_REG est égale à la valeur de BX_REG.

CMD_JMP_AX_NOPARI_BX – (code de commande 0x07) – saut à l'adresse spécifiée dans le champ uData si la valeur du registre système AX_REG n'est pas égale à la valeur de BX_REG.

CMD_JMP_AX_MORE_BX – (code de commande 0x08) – saut à l'adresse spécifiée dans le champ uData, si la valeur du registre système AX_REG est supérieure à la valeur de BX_REG.

CMD_JMP_AX_LESS_BX – (code de commande 0x09) – saut à l'adresse spécifiée dans le champ uData, si la valeur du registre système AX_REG est inférieure à la valeur de BX_REG.

CMD_CALL – (code de commande 0x0A) – appel d'une sous-routine commençant à l'adresse spécifiée dans le champ uData.

CMD_RETURN – (code de commande 0x0B) – retour de la sous-routine.

CMD_FOR – (code de commande 0x0C) – exécution cyclique d'une séquence de commandes. L'octet de poids fort du champ uData contient le nombre de commandes situées après la commande CMD_FOR qui seront répétées dans un cycle. L'octet de poids faible du champ uData contient le nombre de répétitions. Par exemple : uData = 0x1705 - 0x17 = 23 commandes exécutées en boucle, 0x05 = 5 - le nombre de répétitions.

CMD_DJMP_GX_PARI_IX – (code de commande 0x63) – saut à l'adresse spécifiée dans le champ uData, si le contenu d'une paire de registres système GX_REG et HX_REG est égal au contenu de IX_REG et JX_REG.

CMD_DJMP_GX_NOPARI_IX – (code de commande 0x64) – saut à l'adresse spécifiée dans le champ uData, si le contenu d'une paire de registres système GX_REG et HX_REG n'est pas égal au contenu de IX_REG et JX_REG.

CMD_DJMP_GX_MORE_IX – (code de commande 0x65) – saut à l'adresse spécifiée dans le champ uData, si le contenu d'une paire de registres système GX_REG et HX_REG est supérieur au contenu de IX_REG et JX_REG.

CMD_DJMP_GX_LESS_IX – (code de commande 0x66) – saute à l'adresse spécifiée dans le champ uData, si le contenu d'une paire de registres système GX_REG et HX_REG est inférieur au contenu de IX_REG et JX_REG.

Des opérations mathématiques sont possibles entre les registres système, les registres Modbus et les constantes. Les opérandes des instructions qui opèrent sur des données 32 bits sont situés dans deux

registres 16 bits consécutifs. Lors de l'accès à un registre, l'instruction spécifie l'adresse du registre inférieur. L'adresse du registre supérieur est obtenue en incrémentant l'adresse du registre inférieur de un.

Le tableau ci-dessous montre les instructions mathématiques, les codes d'instruction pour les opérations avec des données 16 bits et 32 bits, et les emplacements des opérandes :

Commande			Opérande 1	Opérande 2	Résultat
Nom	Code				
	Données 16 bits	Données 32 bits	S1	S2	D
Addition (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
Soustraction (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
Division, reste non pris en compte (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
Multiplication (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
ET logique bit à 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_AND_MODBUS_CONST	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

OU logique bit à bit ($S1 \mid S2 = D$)					
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
OU exclusif bit à bit ($S1 \wedge S2 = D$)					
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
Décalage de données ($S1 \gg S2 = D$ ou $S1 \ll S2 = D$ – le sens du décalage est indiqué dans le champ bTypeJmp : 0 –gauche, 1 - droite)					
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 - Constante dans le champ de données uData

CONST_2 - Une constante sur la ligne de commande uCMD_DATA suivante

SYS_REG_1 est la valeur contenue dans le registre système (ou deux registres système consécutifs pour les instructions 64 bits) à uAdrSysReg. Adresses des registres système 0..9

SYS_REG_2 la valeur est contenue dans le registre système (ou deux registres système consécutifs pour les données 32 bits) dont l'adresse est spécifiée dans la ligne de commande uCMD_DATA suivante

Modbus_REG – la valeur contenue dans le registre Modbus : type de registre dans le champ TypeModbusReg, adresse du registre dans le champ uData

Important : lorsque l'exécution d'un programme utilisateur est arrêtée par la commande CMD_STOP_PROGRAM, l'état du moteur et de tous les registres reste le même que celui défini pendant le programme. Pour cette raison, si le moteur tournait au moment où la commande CMD_STOP_PROGRAM a été exécutée, il continuerait à exécuter la dernière tâche, c'est-à-dire que le mouvement continuerait une fois le programme terminé. Une fois le programme terminé, la rotation du variateur peut être arrêtée en écrivant un registre via Modbus (Coils 2001h ou Coils 2002h). Pour éviter une rotation incontrôlée, une commande d'arrêt du moteur peut être définie avant la commande de fin de programme, ou utiliser la commande CMD_END_STOP_PROGRAM.

7. Réinitialisation aux paramètres d'usine

Si nécessaire, les paramètres du contrôleur peuvent être réinitialisés aux valeurs d'usine. Pour ce faire, avant de mettre l'appareil sous tension, fermez le premier contact du connecteur RJ11 - CLR_FLASH_CON (connecteur RS-485, voir Fig. 4) avec la terre GND de l'alimentation. Les voyants rouge et vert de l'appareil s'allumeront alternativement. Gardez CLR_FLASH_CON et GND fermés pendant 5 s. Après cela, les paramètres du contrôleur seront réinitialisés aux valeurs d'usine.

8. Livraison en jeux complets

Contrôleur de moteur à courant continu à balais RS 434544 ou RS 434546

1 pièce

9. Informations sur le fabricant

RS Components adhère à la ligne de développement continu et se réserve le droit d'apporter des modifications et des améliorations à la conception et au logiciel du produit sans préavis.

Les informations contenues dans ce manuel sont sujettes à modification à tout moment et sans préavis.

10. Garantie

Toutes les réparations ou modifications sont effectuées par le fabricant ou une société agréée.

Le fabricant garantit le fonctionnement sans défaillance du contrôleur pendant 12 mois à compter de la date de vente lorsque les conditions de fonctionnement sont remplies.

Adresse du service commercial du fabricant :



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

Dernière modification : 07.2025