



**SMART MOTOR DEVICES**

<https://www.smd.ee>

**Controlador de motor CC sin escobillas  
BLSD-20Modbus**

**Manual de Usuario  
BLSD.20.Modbus.001  
2025**



## 1. Designación del producto

Los controladores de motor sin escobillas B LSD-20Modbus son dispositivos electrónicos diseñados para operar y controlar motores CC síncronos trifásicos sin escobillas con codificador Hall.

## 2. Funciones y posibilidades

Los controladores están diseñados para controlar la velocidad, la aceleración, la deceleración y la dirección de rotación del motor. Las unidades también proporcionan posicionamiento basado en las señales de los sensores Hall incorporados. También es posible el mantenimiento de la posición.

El motor sin escobillas se controla mediante señales externas o mediante comandos transmitidos a través de RS-485 por el protocolo Modbus. El controlador también puede funcionar según un algoritmo previamente grabado en la memoria de bloques por un usuario.

### Control a través de RS-485 Modbus

El controlador se puede controlar de forma remota a través de la línea de comunicación física RS-485 utilizando el protocolo industrial Modbus:

- La configuración se realiza escribiendo o leyendo los parámetros correspondientes en/desde los registros del controlador.
- Los protocolos compatibles son RTU y ASCII, las velocidades son 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 57600, 115200, 128000 baudios.
- Se implementan movimientos a una posición preestablecida objetivo o a uno de los cuatro puntos de referencia predefinidos.
- El controlador puede funcionar de forma autónoma bajo el control de un programa de usuario (hasta 1024 comandos), que se graba previamente en la memoria no volátil. Se admiten saltos condicionales e incondicionales (relativos y absolutos), llamadas a subrutinas, bucles, temporizadores, y se proporcionan operaciones lógicas y matemáticas con datos.
- Las entradas programables IN1 e IN2 se pueden utilizar como señales START/STOP, REVERS o para otros fines a discreción del usuario.
- El controlador puede realizar el posicionamiento en el rango de -2147483647 a +2147483648 conmutaciones del sensor Hall.
- El controlador tiene contactos RT en el panel frontal para la conexión de una resistencia terminal.

### Control mediante señales externas

Para controlar el motor mediante señales externas, se proporciona lo siguiente:

- Terminales para conectar un potenciómetro o una señal externa de 0-5 VDC para el control analógico de la velocidad.
- Contactos para conectar las señales externas In1 e In2, cuyo propósito y procesamiento lo determina el usuario. Las entradas también se pueden utilizar como señales START/STOP (inicio/parada de movimiento) y DIR (dirección).
- Contacto HARD STOP para conectar una señal de alarma que asegura el frenado controlado del motor en caso de interrupción del circuito de emergencia. Después de la parada, el motor cambia al modo desenergizado, lo que evita el arranque accidental.



### 3. Características técnicas

Modelo	BLSD-20Modbus
Tensión de alimentación, VDC	24 - 48
Protección de la fuente de alimentación, VDC	20 - 51
Corriente nominal del motor, A	<20
Corriente máxima del motor, A	<30
Resistencia de entrada de la señal de SPEED, kOhm	20
Rango de voltaje de entrada de la entrada SPEED, VDC	0..5
Dimensiones (no más), mm	116x100x23
Interfaz de comunicación	RS-485, Modbus – ASCII o RTU

Las dimensiones generales y de conexión del controlador se muestran en la Fig. 1

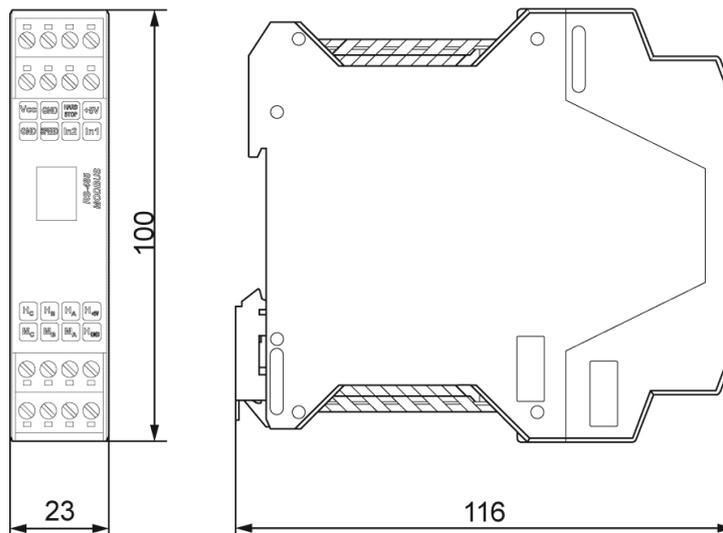


Fig.1. Dimensiones generales y de conexión del controlador BLSD-20Modbus

El esquema de conexión se muestra en la Fig. 2

#### Condiciones ambientales:

Temperatura ambiente: 0...+50°C

Humedad: 90% HR o menos a +25°C

Condensación y congelación: ninguna

### 4. Construcción y elementos de control

BLSD-20Modbus está diseñado como una placa de circuito con elementos electrónicos, cubierta con una carcasa con montaje en riel DIN. En la parte superior de la carcasa hay símbolos gráficos para los controles y las asignaciones de pines. Además de los componentes electrónicos, hay elementos indicadores y de control, terminales de conexión y conectores en la placa:

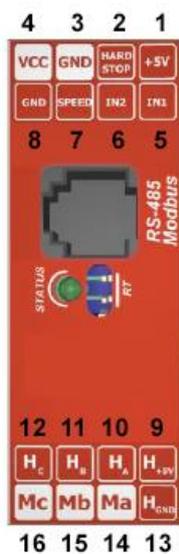
- terminales de tornillo para conectar la fuente de alimentación, los devanados del motor sin escobillas, las líneas del codificador y el circuito de control;
- terminales IN1 e IN2 para conectar señales de entrada de control;
- terminales para conectar un potenciómetro externo para regular la velocidad de rotación del motor;
- terminal para conectar los contactos de la señal de parada de emergencia;
- indicador LED de funcionamiento del dispositivo;
- conector RJ11 (6P6C) para conectar líneas RS-485;
- contactos para conectar la resistencia terminal interna RT;
- circuito de frenado incorporado para absorber la energía generada por el motor (durante la inercia, rotación forzada).



La entrada "SPEED" está diseñada para el control de velocidad mediante una señal analógica externa. Se proporciona una entrada de señal externa "HARD STOP" para la parada de emergencia del motor. Las entradas externas IN1 e IN2 se pueden utilizar para arrancar y detener el motor mediante señales externas, así como para controlar el sentido de giro del motor.

Todos los parámetros del funcionamiento del motor y el control de movimiento se pueden realizar mediante software y mediante comandos transmitidos a través de RS-485 mediante el protocolo Modbus.

La disposición y el propósito de los terminales se muestran en la Fig. 2



1. Salida +5 VDC para potenciómetro externo
2. Señal de parada de emergencia "HARD STOP"
3. GND de la fuente de alimentación
4. Fuente de alimentación 24 – 48 VDC
5. Señal "IN1" (contacto limpio)
6. Señal "IN2" (contacto limpio)
7. Entrada de señal analógica – para la conexión de un potenciómetro externo de regulación de velocidad
8. GND de la señal
9. Salida para alimentación de sensores Hall
10. Sensor Hall – fase A
11. Sensor Hall – fase B
12. Sensor Hall – fase C
13. GND de los sensores Hall
14. Fase A del motor
15. Fase B del motor
16. Fase C del motor

RS-485 Modbus - Conector RJ11 (6P6C) para conectar las líneas de datos RS-485

RT - contactos para conectar la resistencia terminal

Fig. 2. Disposición y propósito de los terminales y elementos de control

## 5. Montaje y conexión

Por favor, lea atentamente este manual antes de la conexión y el montaje.

Por favor, realice el cableado solo cuando la alimentación esté apagada. No intente cambiar el cableado mientras la alimentación esté ON.

Por favor, proporcione un contacto fiable en los terminales de conexión. Durante el cableado, por favor, observe la polaridad y la gestión de los cables. La polaridad inversa y la sobretensión dañarán el controlador.

Siga las siguientes instrucciones durante la conexión:

1. Conecte un motor al controlador según la fig. 2. Las fases del motor deben conectarse a los terminales 14 - 16. Las señales de los sensores Hall deben conectarse a los terminales 10 - 12. La GND de los sensores HALL debe conectarse al terminal 13, la alimentación de las señales de los sensores HALL debe conectarse al terminal 9.
2. Conecte los elementos de control externos de acuerdo con los esquemas del diagrama de conexión en la fig. 3:

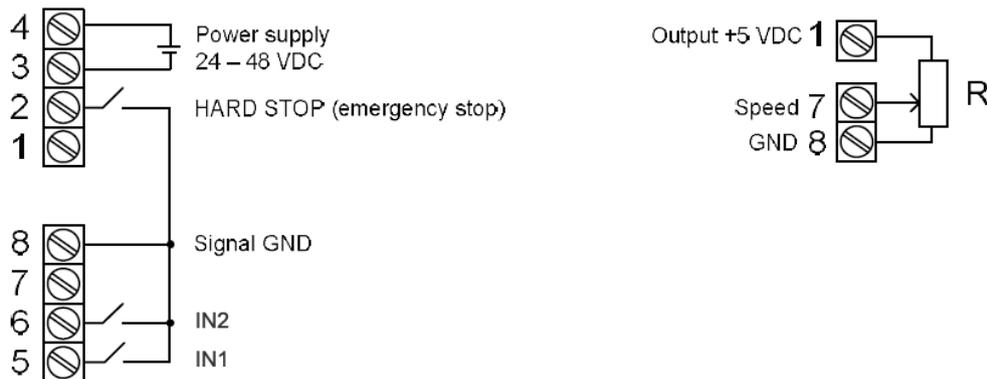


Fig. 3. Conexión de la fuente de alimentación y elementos de control externos

- tipo de señales externas «IN1», «IN2», «HARD STOP» - contacto limpio;
  - resistencia total del potenciómetro externo para el control de velocidad - aproximadamente 4..5KOhm.
3. Conecte las líneas de la interfaz RS-485 al conector RJ11 según la fig. 4.

#### RS-485 - RJ11 (6P6C)

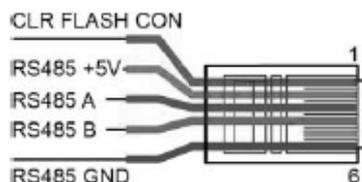


Fig.4 - asignación de pines del conector RJ11 - RS-485 Modbus.

4. Si es necesario, coloque un puente en los pines RT para conectar la resistencia terminal interna.
5. Conecte el dispositivo a la unidad de alimentación observando la polaridad. La unidad de fuente de alimentación debe seleccionarse con un margen (para evitar caídas de tensión). El grosor de los cables de conexión debe corresponder al consumo de corriente del motor. Conecte el "+" de la fuente de alimentación - a la entrada 4, conecte el "-" de la fuente de alimentación - a la entrada 3. El procedimiento de desmontaje es en orden inverso.

## 6. Funcionamiento

1. Asegúrese de que la fuente de alimentación esté apagada. Por favor, realice el cableado solo cuando la alimentación esté apagada.
2. Conecte el motor y la fuente de alimentación al controlador según la sección 5.
3. Seleccione el método de control: control por comandos a través de Modbus o señales externas (registro MODE\_DEVICE - ver sección 6.5, fig. 6 y fig. 7).
4. Para ajustar la velocidad utilizando una fuente de señal externa, conecte una resistencia externa a los terminales 1 "+5 V", 7 "SPEED" y 8 "GND". La resistencia mínima corresponde a la velocidad máxima; a medida que aumenta la resistencia, la velocidad disminuye. Para poder controlar la velocidad mediante una señal analógica externa, configure el registro USE\_EXTERN\_SPEED = 1 (consulte la descripción de los registros a continuación).
6. Conecte las señales de control "IN1", "IN2" y la señal de parada de emergencia "HARD STOP" según la sección 5. La señal "HARD STOP" se utiliza para la parada de emergencia del motor. El funcionamiento está permitido con el contacto cerrado.
7. Conecte las líneas de la interfaz RS-485 según la sección 5.
8. Encienda la fuente de alimentación. El dispositivo está listo para la configuración a través del protocolo Modbus.



9. Configure los parámetros de operación necesarios mediante comandos a través del protocolo Modbus: limitación de corriente del motor, dirección de rotación, configuración del funcionamiento de las entradas externas IN1 e IN2, método de control.
10. Para controlar el variador mediante el protocolo Modbus, envíe comandos a través de la interfaz RS-485. A continuación se muestra la tabla de registros del controlador, su propósito y los posibles comandos de control del motor.
11. Para controlar el variador con señales externas para arrancar y detener el motor y para controlar la dirección de rotación, utilice las señales IN1 e IN2. La regulación de la velocidad se realiza mediante una señal analógica externa aplicada a la entrada SPEED, o mediante comandos a través del protocolo Modbus.

### Control MODBUS

Para la transmisión de datos a través de la interfaz RS-485, se utiliza el protocolo de comunicación estándar Modbus (ASCII o RTU).

La unidad de control tiene la siguiente configuración de fábrica:

- ID = 1
- Velocidad: 115200 baudios
- Control de paridad: par
- Bits de datos: 8
- Bit de parada: 1
- MODBUS RTU

### 6.1. Registros de control de entrada

Dirección	Tipo	Nombre	Tamaño	Descripción
1000h	Discrete Input	IN1_bit	1-bit	Estado de la señal de entrada IN1
1001h	Discrete Input	IN2_bit	1-bit	Estado de la señal de entrada IN2
1002h	Discrete Input	IN_HARD_STOP_bit	1-bit	Estado de la señal de entrada HARD_STOP
5007h	Holding Register	MODE_EXT_IN	16 bits	Configuración de las entradas externas IN, IN2.
5013h	Holding Register	PRESSED_INPUTS_EXTERN	16 bits	Tiempo mínimo de pulso positivo en las entradas digitales IN1, IN2 (ms)
5014h	Holding Register	WAITED_INPUTS_EXTERN	16 bits	Tiempo mínimo para la parte negativa del pulso en las entradas digitales IN1, IN2 (ms)

Registros de estado de entrada 1000h..1002h son de solo lectura.

1000h -**IN1\_bit**- representa el estado de la entrada física IN1.

1001h -**IN2\_bit**- representa el estado de la entrada física IN2.

1002h -**IN\_HARD\_STOP\_bit**- representa el estado de la entrada física HARD\_STOP.

Posibles valores de registro 1000h..1002h:

- 1 – entrada en cortocircuito con la salida GND
- 0 – entrada abierta con la salida GND

5007h -**MODE\_EXT\_IN**- el valor del registro determina el propósito y el método de procesamiento de las señales IN1 e IN2 (ver descripción en la sección 6.5).

5013h -**PRESSED\_INPUTS\_EXTERN** y 5014h -**WAITED\_INPUTS\_EXTERN**- el tiempo mínimo (establecido en ms) de las partes positiva y negativa del pulso en las entradas digitales IN1, IN2 - los registros se utilizan para suprimir el rebote de contacto (ver Fig. 5).

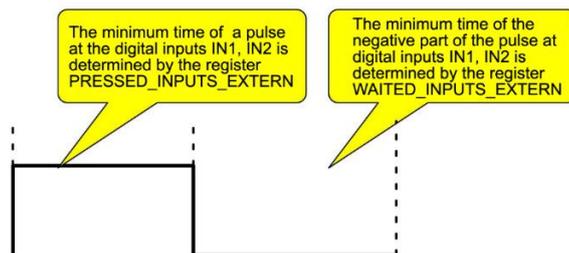


Fig. 5 - Supresión del rebote de contacto

## 6.2. Registros de control del motor

Dirección	Tipo	Nombre	Tamaño	Descripción
2000h	Coils	START_bit	1-bit	Arrancar la rotación del motor con una aceleración establecida
2001h	Coils	STOP_bit	1-bit	Detener la rotación del motor con una deceleración establecida
2002h	Coils	HARD_STOP_bit	1-bit	Parada de emergencia abrupta de la rotación del motor
2003h	Coils	CLR_POSITION_bit	1-bit	Restablecer a cero el registro de posición actual POSITION_VALUE.
2004h	Coils	INV_DIR_bit	1-bit	Invirtiendo el sentido de rotación del motor
2005h	Coils	INV_CNT_bit	1-bit	Invertir el sentido de conteo de posición
2006h	Coils	ACC_ON_bit	1-bit	Habilitación de la aceleración especificada por el registro ACC
2007h	Coils	DEC_ON_bit	1-bit	Activación de la deceleración especificada por el registro DEC
2008h	Coils	FQ_REGULATOR_ON_bit	1-bit	Activación de la regulación de velocidad de cuatro cuadrantes

Estos registros de control están disponibles tanto para lectura como para escritura. Al escribir en el registro correspondiente, el valor 1 activa una u otra función. Los registros 2000h...2003h se restablecen a 0 después de aplicar la función.

## 6.3. Registros de estado

Dirección	Tipo	Nombre	Tamaño	Descripción
3000h	Input Register	STATUS	16 bits	El estado actual del control del motor.
3001h	Input Register	CURRENT_VALID	16 bits	El valor actual de la corriente consumida por el motor.
3002h	Input Register	SPEED_VALID	16 bits	El valor instantáneo de la velocidad de rotación.
3003h	Input Register	CURRENT_POSITION	32-bit	Posición actual. Los valores van desde -2147483647 hasta +2147483648
3005h	Input Register	TEMPERATURE_MCU	16 bits	Temperatura de la CPU
3006h	Input Register	TEMPERATURE_MOSFET	16 bits	Temperatura del circuito de potencia
3007h	Input Register	TEMPERATURE_BRAKE	16 bits	Temperatura del circuito de freno
3008h	Input Register	TASK_COUNTER	16 bits	Devuelve un valor aleatorio (para comprobar el funcionamiento del canal de transmisión)
3009h	Input Register	STATUS_USER_PROGRAM	16 bits	El estado actual del programa de usuario
300Ah	Input Register	SPEED_INPUT_VALUE	16 bits	Lecturas ADC de la entrada de SPEED



Este grupo de registros es de solo lectura y representa el estado actual del controlador.

3000h –**STATUS**- estado actual del control del motor, posibles valores:

- 0 parada del motor
- 1 rotación hacia adelante
- 2 rotación hacia atrás

3001h –**CURRENT\_VALID**- el valor actual de la corriente consumida por el motor. Representa el valor de la corriente consumida por el motor de la fuente de alimentación, el valor en mA.

3002 –**SPEED\_VALID**- el valor instantáneo de la velocidad de rotación. Las unidades de medida son revoluciones por minuto.

3003h –**CURRENT\_POSITION**– la posición actual. El posicionamiento se realiza en el rango de valores de - 2147483647 a + 2147483648 del número total de conmutaciones del sensor Hall, se tienen en cuenta tanto los flancos de subida como los de bajada.

3005h –**TEMPERATURE\_MCU**– temperatura del procesador central. La temperatura se calcula como  $TEMPERATURE\_MCU/10$  deg/C.

3006h –**TEMPERATURA\_MOSFET**- temperatura del circuito de potencia - la temperatura en el área de instalación de los interruptores MOSFET. La temperatura se calcula como  $TEMPERATURE\_MOSFET/10$  grados/°C.

3007h –**TEMPERATURA\_FRENO**- temperatura del circuito de freno – la temperatura en el área de la resistencia de frenado. La temperatura se calcula como  $TEMPERATURE\_MOSFET/10$  grados/°C.

3008h –**TASK\_COUNTER**- devuelve un número aleatorio en el rango de 0x0000 a 0xFFFF.

3009h –**STATUS\_USER\_PROGRAM** - El estado actual del programa de usuario, valores posibles:

- 1 – el programa de usuario se detiene mediante un comando a través de Modbus
- 2 – el programa de usuario se inicia mediante un comando a través de Modbus (este estado es solo por un período corto, antes de que se establezca el estado 4)
- 3 – el programa de usuario se inicia después de que se enciende la tensión de alimentación (este estado es solo por un período corto, antes de que se establezca el estado 4)
- 4 – el programa de usuario se está ejecutando
- 5 – el programa de usuario ha finalizado con el comando END
- 6 – el programa de usuario ha finalizado con el comando ENDF
- 7 – el programa de usuario se ha detenido debido a un error

300Ah - **SPEED\_INPUT\_VALUE** - lecturas del convertidor analógico-digital de la entrada SPEED (medición en unidades relativas).

#### 6.4. Registros de configuración de transmisión de datos Modbus RS-485

Dirección	Tipo	Nombre	Tamaño	Descripción
5000h	Holding Register	SLAVE_ADDRESS_MODBUS	16 bits	ID del controlador (dirección del dispositivo). Valores válidos: 0..247. ( 0 - dirección de difusión, sin mensaje de respuesta)
5001h	Holding Register	TYPE_MODBUS	16 bits	Ajustes de transmisión de datos
5002h	Holding Register	BITRATE_MODBUS	16 bits	Configuración de la velocidad en baudios
5003h	Holding Register	TIMEOUT_BROADCAST_MODBUS	16 bits	Retardo adicional entre el paquete recibido y el mensaje de respuesta.

5001h –**TYPE\_MODBUS**– ajustes de transmisión de datos, valores válidos:

- 1 – ASCII, 7 bit de datos, par, 1 bit de parada,
- 2 – ASCII, 7 bit de datos, impar, 1 bit de parada
- 3 – RTU, 8 bit de datos, par, 1 bit de parada
- 4 – RTU, 8 bit de datos, impar, 1 bit de parada
- 5 – RTU, 8 bit de datos, ninguno, 2 bit de parada

5002h –**BITRATE\_MODBUS**– velocidad en baudios de Modbus, valores posibles:

- 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**- se utiliza si el dispositivo de control tarda un tiempo lento en cambiar del modo de transmisión al modo de recepción.

Después de cambiar la configuración de comunicación, los nuevos valores deben guardarse utilizando el registro FLAG\_SAVE\_INI (consulte la sección 6.5) y luego se debe reiniciar el controlador. Después del reinicio, la conexión RS-485 se realizará utilizando la nueva configuración.

### 6.5. Registros para la configuración del funcionamiento del variador

Dirección	Tipo	Nombre	Tamaño	Descripción
5004h	Holding Register	MODE_DEVICE	16 bits	Modo de operación del controlador
5005h	Holding Register	MODE_USER_PROGRAM	16 bits	Comando de control para iniciar un programa de usuario
5006h	Holding Register	MODE_ROTATION	16 bits	Modo de rotación.
5007h	Holding Register	MODE_EXT_IN	16 bits	Configuración de las entradas externas IN, IN2.
5008h	Holding Register	POSITION_N	16 bits	El número de posición al que moverse. Los valores van de 1 a 4
5009h	Holding Register	REF_CURRENT	16 bits	Limitación de la corriente máxima de funcionamiento en el devanado del motor. Rango de 1000 mA a 20000 mA
500Ah	Holding Register	HOLD_CURRENT	16 bits	Corriente de mantenimiento en modo parada. Rango de 0 a 1000 mA
500Bh	Holding Register	SPEED	16 bits	Ajustar la velocidad de rotación. Rango de 35 a 14000 rpm.
500Ch	Holding Register	ACC	16 bits	Ajustar aceleración. Rango de 10 a 1000.
500Dh	Holding Register	DEC	16 bits	Ajustar la deceleración. Rango de 10 a 1000.
500Eh	Holding Register	DIRECTION	16 bits	Sentido de rotación. 1 – adelante 2 – hacia atrás
500Fh	Holding Register	N_POLE	16 bits	Número de polos del motor. Rango de 1 a 12
5010h	Holding Register	USE_EXTERN_SPEED	16 bits	Método de control de velocidad: 1 – comandos vía Modbus 2 – señal externa en la entrada SPEED
5011h	Holding Register	RESERVED	16 bits	reservado
5012h	Holding Register	OFFSET_COMPENSATION	16 bits	Corrección de frenado del motor



5013h	Holding Register	PRESSED_INPUTS_EXTERN	16 bits	Tiempo mínimo de pulso positivo en las entradas digitales IN1, IN2 (ms)
5014h	Holding Register	WAITED_INPUTS_EXTERN	16 bits	Tiempo mínimo para la parte negativa del pulso en las entradas digitales IN1, IN2 (ms)
5015h	Holding Register	OFFSET	32-bit	El desplazamiento a mover. Rango de valores de -2147483647 a + 2147483648
5017h	Holding Register	OFFSET_CONST	32-bit	Incremento: el desplazamiento al que es necesario mover, el valor es cambiado por el controlador durante el funcionamiento.
5019h	Holding Register	TARGET_POSITION	32-bit	La posición a la que desplazarse. Rango de valores de -2147483647 a + 2147483648
501Bh	Holding Register	TARGET_POSITION1	32-bit	Posición preestablecida por el usuario №1 Rango de valores de -2147483647 a + 2147483648
501Dh	Holding Register	TARGET_POSITION2	32-bit	Posición preestablecida por el usuario №2 Rango de valores de -2147483647 a + 2147483648
501Fh	Holding Register	TARGET_POSITION3	32-bit	Posición preestablecida por el usuario №3 Rango de valores de -2147483647 a + 2147483648
5021h	Holding Register	TARGET_POSITION4	32-bit	Posición preestablecida por el usuario №4 Rango de valores de -2147483647 a +2147483648
5023h	Holding Register	ERROR	16 bits	Registro de errores
5024h	Holding Register	FLAG_SAVE_INI	16 bits	Registro para guardar la configuración del usuario.(Registros 5000h..501Fh). Valor: 0x37FA
5025h	Holding Register	FLAG_SAVE_USER_PROGRAM	16 bits	Registro para escribir o leer un programa de usuario hacia o desde la memoria no volátil. Valor a escribir: 0x8426 Valor a leer: 0x9346
5026h	Holding Register	FLAG_RESTART	16 bits	Registro de reinicio. Valor: 0x95AF

5004h -**MODE\_DEVICE**-modo de operación del controlador, valores posibles:

- 1 – control a través de Modbus
- 2 – control mediante señales externas

5005h -**MODE\_USER\_PROGRAM**- comando de control para iniciar un programa de usuario, valores posibles:

- 1 – detener el programa de usuario
- 2 – iniciar el programa de usuario
- 3 – iniciar el programa de usuario tan pronto como se encienda la alimentación (se requiere el pre-guardado de la configuración - ver registro FLAG\_SAVE\_INI).



5006h –**MODE\_ROTATION**– modo de rotación, valores posibles:

- 1 – rotación continua
- 2 – desplazamiento por la cantidad especificada por el registro OFFSET
- 3 – movimiento a una posición dada. El número de posición se especifica mediante el registro POSITION\_N (de 1 a 4), las coordenadas de las posiciones se especifican mediante los registros POSITION1, POSITION2, POSITION3, POSITION4 respectivamente.

5007h –**MODE\_EXT\_IN**– configuración del modo de operación de las entradas externas IN1, IN2 en el modo de control de señales externas, valores posibles:

- 1 – La entrada IN1 se utiliza como una señal de arranque/parada del variador, se procesa en el flanco descendente del pulso; la entrada IN2 se utiliza como una señal de inversión, se procesa en el flanco descendente del pulso.
- 2 - La entrada IN1 se utiliza como una señal de arranque/parada del variador, se procesa en el flanco descendente del pulso; la entrada IN2 se utiliza como una señal de referencia de dirección, procesada según el nivel de la señal.
- 3 - La entrada IN1 se utiliza como una señal de arranque/parada del variador, se procesa según el nivel de la señal: , presencia de una señal - habilitar la rotación del motor, sin señal - parada; la entrada IN2 se utiliza como una señal de inversión, se procesa en el flanco descendente del pulso.
- 4 - La entrada IN1 se utiliza como señal de arranque/parada del variador, se procesa según el nivel de la señal: , presencia de una señal - habilitar la rotación del motor, sin señal - parada; la entrada IN2 se utiliza como señal de referencia de dirección, procesada según el nivel de la señal.
- 5 – la entrada IN1 se utiliza como señal para arrancar y parar el variador en la dirección de avance, IN2 - como señal para arrancar y parar el variador en la dirección opuesta; ambas señales se procesan por nivel.

5008h –**POSITION\_N**– selección del número de posición para el movimiento (de 1 a 4) - se utiliza en el modo de movimiento a una posición dada (MODE\_ROTATION = 3) junto con los registros POSITION1, POSITION2, POSITION3, POSITION4.

5009h –**REF\_CURRENT**– ajuste de la corriente máxima de funcionamiento en el devanado del motor – de 1000 mA a 20000 mA

500Ah –**HOLD\_CURRENT**– corriente de mantenimiento en modo de parada (de 0 a 1000 mA)

500Bh –**SPEED**– velocidad de rotación objetivo para el control Modbus (de 35 a 14000 rpm).

500Ch –**ACC**– aceleración dada (de 10 a 1000)

500Dh –**DEC**– deceleración dada (de 10 a 1000)

Los valores de aceleración y deceleración en los registros ACC y DEC son valores lineales convencionales que determinan la tasa de aceleración/deceleración. Un valor de 10 corresponde a 100 rps<sup>2</sup>, un valor de 1000 corresponde a 5000 rps<sup>2</sup>.

500Eh –**DIRECTION**– dirección de rotación, valores posibles:

- 1 – rotación hacia adelante
- 2 – rotación hacia atrás.

500Fh –**N\_POLE**– número de polos del motor (de 1 a 12).

5010h –**USE\_EXTERN\_SPEED**– configuración del modo de ajuste de la velocidad de rotación, valores posibles:

- 1 – comandos vía Modbus
- 2 – señal analógica externa en la entrada SPEED

5012h –**OFFSET\_COMPENSATION**– Corrección de frenado del motor calculada experimentalmente para los ajustes actuales de ACC, DEC, SPEED. Si la parada se produce más allá del punto calculado, entonces el valor de compensación es igual al error de posicionamiento con signo negativo, si el motor se detiene antes del punto de parada, entonces el valor de compensación es positivo. Por ejemplo, el error de alcanzar la posición especificada es de 15 incrementos, lo que significa que OFFSET\_COMPENSATION = -15, y esta corrección es válida solo para los ajustes actuales de aceleración, deceleración y velocidad.

5013h y 5014h –**PRESSED\_INPUTS\_EXTERN** y **WAITED\_INPUTS\_EXTERN**– tiempo mínimo de las partes positiva y negativa de un pulso en las entradas digitales IN1, IN2 – se utiliza para suprimir el rebote de los contactos (ver fig. 5).



5015h –**OFFSET**– la desviación que se debe mover se utiliza cuando `MODE_ROTATION = 2`, valores válidos de -2147483647 a +2147483648. Antes de iniciar el movimiento, es necesario establecer el valor de desviación requerido en el registro `OFFSET`, que el controlador procesa como un contador del movimiento restante. Durante la ejecución del movimiento especificado, el valor de `OFFSET` disminuye.

5017h –**OFFSET\_CONST**- Incremento - la desviación a la que es necesario mover, el valor es modificado por el controlador durante el funcionamiento.

5019h –**TARGET\_POSITION**– posición a la que desplazarse, valores permitidos de -2147483647 a +2147483648. En el modo de movimiento a una posición dada, la coordenada de `POSITION1-POSITION4` se copia a este registro antes de moverse, mientras que los registros `POSITION1-POSITION4` no cambian sus valores.

501Bh - 5021h –**TARGET\_POSITION1..4**– Posición preestablecida por el usuario №1..4, utilizada cuando `MODE_ROTATION = 3`, el número de posición está determinado por el registro `POSITION_N`, valores permitidos de -2147483647 a +2147483648.

5023h –**ERROR**– errores que ocurren durante el funcionamiento del controlador - cada bit del registro señala un error específico:

- bit 0 - fuera del rango de voltaje de suministro;
- bit 1 - cortocircuito de los devanados del motor;
- bit 2 - sobrecalentamiento del circuito de freno;
- bit 3 - sobrecalentamiento del circuito de potencia;
- bit 4 - error de conexión de los sensores Hall;
- bit 5 - parada de emergencia;
- bit 6 - sobrecalentamiento de la MCU;
- bit 7 - programa de control de prueba;
- bit 8 - error de ejecución del programa de usuario;
- bit 9 - error al leer o escribir la configuración;
- bit 10 - error en el funcionamiento de los interruptores de transistores de salida;
- bit 12 - advertencia sobre la imposibilidad de calcular el punto de interrupción;
- bit 13 - advertencia sobre un intento de escribir en el registro un valor que está fuera de rango;
- bit 14 – error de paridad de transmisión RS-485

5024h –**FLAG\_SAVE\_INI**– registro para guardar la configuración del usuario - al escribir el valor 0x37FA en este registro, la configuración definida por los registros 5000h..501Fh se guardará en la memoria no volátil.

5025h –**FLAG\_SAVE\_USER\_PROGRAM**– al escribir el valor 0x8426 en este registro se inicia el procedimiento para guardar el programa del usuario desde el búfer temporal a la memoria no volátil del controlador. Escribir el valor 0x9346 inicia el procedimiento para leer el programa del usuario desde la memoria no volátil del controlador a un búfer temporal (ver sección 6.6.).

5026h –**FLAG\_RESTART**– escribir el valor 0x95AF en este registro provoca el reinicio del controlador.

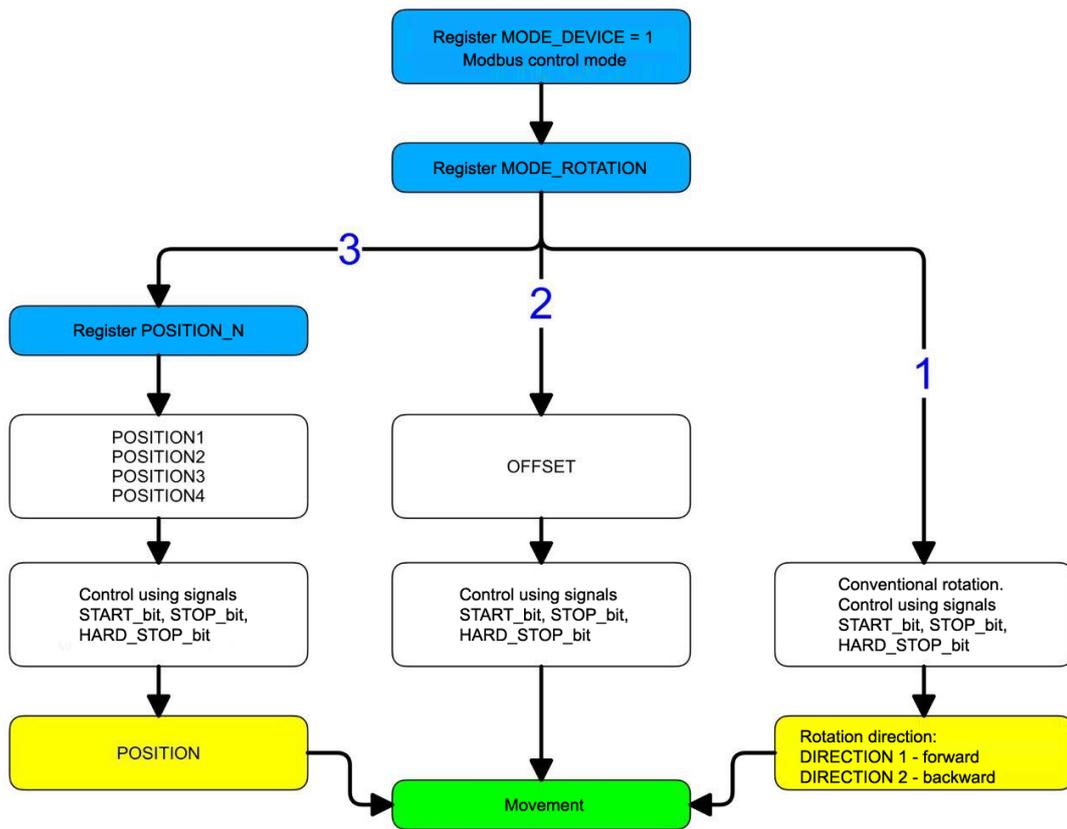


Fig.6. Diagrama de flujo para seleccionar el modo de funcionamiento al controlar el variador mediante Modbus

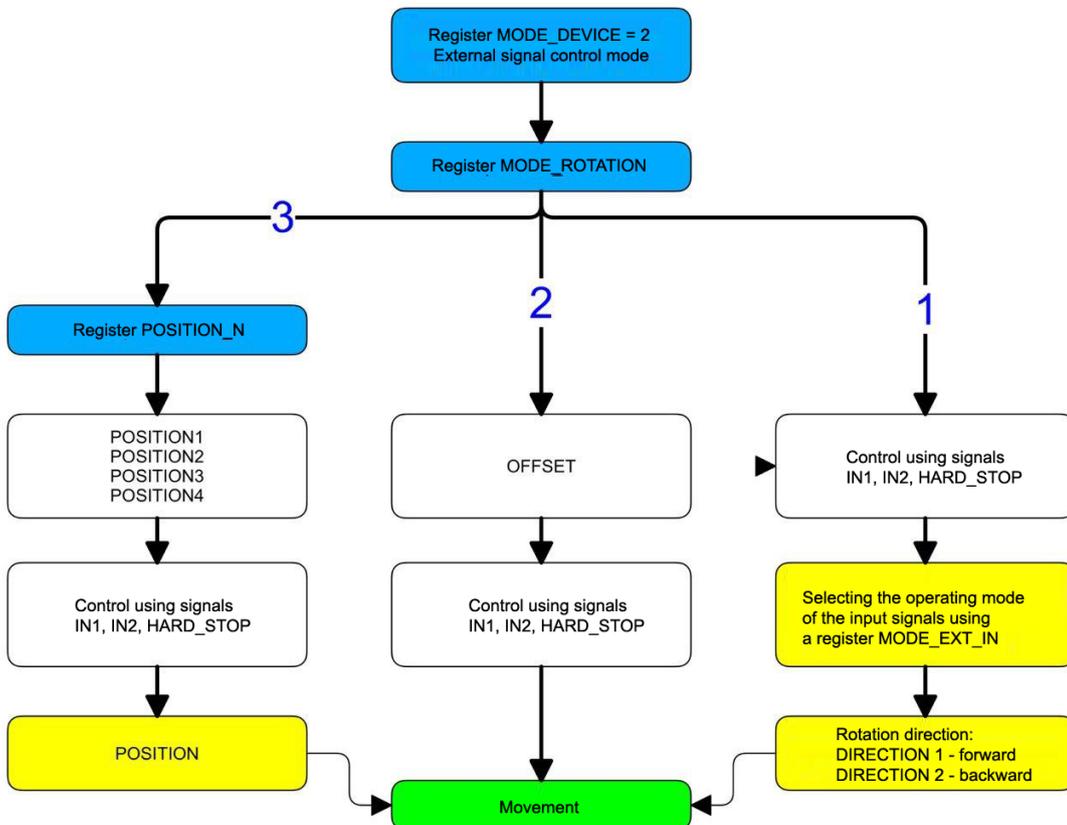


Fig.7. Diagrama de flujo para seleccionar el modo de funcionamiento al controlar el variador mediante señales externas



### 6.6. Lectura y escritura de un programa de usuario

Se utiliza un búfer temporal de 1024 comandos para leer y escribir un programa de usuario. El registro especial FLAG\_SAVE\_INI se utiliza para guardar el programa desde el búfer temporal a la memoria no volátil y para leer el programa desde la memoria del controlador al búfer temporal.

Dirección	Tipo	Nombre	Tamaño	Descripción
5025h	Holding Register	FLAG_SAVE_USER_PROGRAM	16 bits	Registro para escribir o leer un programa de usuario hacia o desde la memoria no volátil del controlador. Valor para escribir: 0x8426 Valor de lectura: 0x9346
6000h	Holding Register	WRITE_CMD	16 bits	Registro de dirección. Cuando se escribe el valor de la dirección en este registro, el comando se transfiere desde el campo CMD_W a la dirección especificada del búfer temporal del programa de usuario.
6001h	Holding Register	CMD_W	32-bit	Instrucción del programa de usuario
6003h	Holding Register	READ_CMD	16 bits	Registro de dirección. Cuando se escribe el valor de la dirección en este registro, el comando se transfiere desde la dirección especificada del búfer temporal del programa del usuario al campo CMD_R.
6004h	Holding Register	CMD_R	32-bit	Instrucción del programa de usuario

#### Ensamblaje y escritura de un programa de usuario en la memoria del controlador

Cada instrucción del programa de usuario consta de dos palabras de memoria (32 bits) - comando (16 bits) y datos del comando (16 bits). Al ensamblar un programa de usuario, las instrucciones se escriben primero en un búfer temporal en el controlador. Para escribir en el búfer temporal, es necesario escribir una instrucción en el registro CMD\_W y luego escribir la dirección de esta instrucción en el búfer interno en el registro WRITE\_CMD. Cuando se escribe la dirección en el registro WRITE\_CMD, la instrucción se transfiere desde el registro CMD\_W al búfer interno. Después de componer un programa de usuario en un búfer temporal, es necesario escribir el valor 0x8426 en el registro FLAG\_SAVE\_USER\_PROGRAM - el programa se transferirá desde el búfer temporal a la memoria FLASH del controlador.

#### Lectura de un programa de usuario desde la memoria del controlador

Es necesario transferir un programa de usuario al búfer temporal del controlador para leerlo desde la memoria FLASH. Para ello, escriba el valor 0x9346 en el registro FLAG\_SAVE\_USER\_PROGRAM - el programa se transferirá desde la memoria FLASH del controlador al búfer temporal. Luego, para leer una instrucción del búfer temporal, es necesario escribir la dirección de la instrucción en el registro READ\_CMD - cuando se escribe la dirección en este registro, la instrucción se copiará del búfer temporal al registro CMD\_R.

Un diagrama de los procedimientos para leer y escribir un programa de usuario se muestra en la fig. 8.

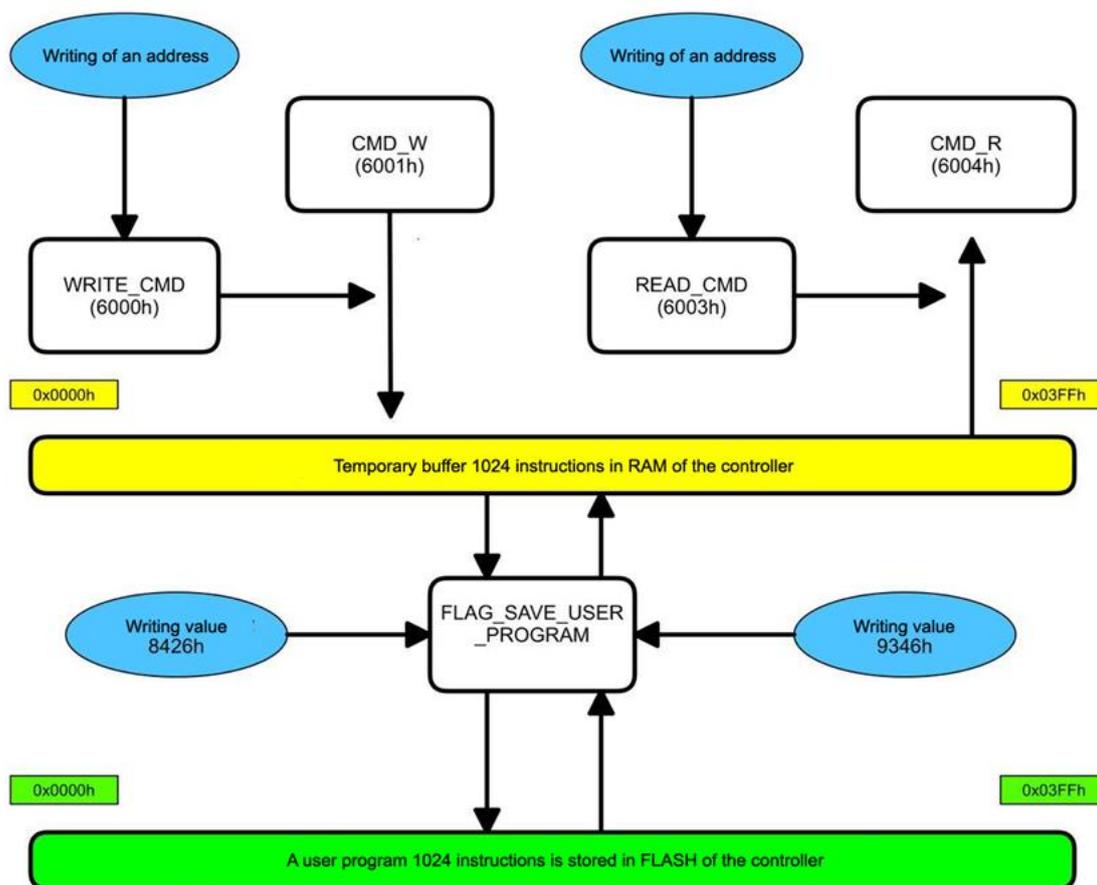


Fig.8. Diagrama de flujo de los procedimientos para leer y escribir un programa de usuario.

6.7. Registros del sistema

Dirección	Tipo	Nombre	Tamaño	Descripción
7000h	Holding Register	AX_REG	16 bits	Registro de almacenamiento de datos para programa de usuario
7001h	Holding Register	BX_REG	16 bits	Registro de almacenamiento de datos para programa de usuario
7002h	Holding Register	CX_REG	16 bits	Registro de almacenamiento de datos para programa de usuario
7003h	Holding Register	DX_REG	16 bits	Registro de almacenamiento de datos para programa de usuario
7004h	Holding Register	EX_REG	16 bits	Registro de almacenamiento de datos para programa de usuario
7005h	Holding Register	FX_REG	16 bits	Registro de almacenamiento de datos para programa de usuario
7006h	Holding Register	PC_REG	16 bits	Puntero de registro al comando de usuario en ejecución
7007h	Holding Register	GX_REG	16 bits	Registro de almacenamiento de datos para programa de usuario
7008h	Holding Register	HX_REG	16 bits	Registro de almacenamiento de datos para programa de usuario
7009h	Holding Register	IX_REG	16 bits	Registro de almacenamiento de datos para programa de usuario
700Ah	Holding Register	JX_REG	16 bits	Registro de almacenamiento de datos para programa de usuario



Los registros del sistema AX\_REG..FX\_REG (rango de valores 0..65535) están destinados al almacenamiento temporal de datos durante la ejecución de un programa de usuario. PC\_REG es un puntero a la instrucción del programa de usuario que se está ejecutando actualmente. Descripción más detallada en la sección 6.9.

### 6.8. Registros de identificación

Dirección	Tipo	Nombre	Tamaño	Descripción
8001h	Input Register	HW_MAJOR	16 bits	Tipo de controlador
8002h	Input Register	HW_MINOR	16 bits	Versión de hardware
8003h	Input Register	FW_MAJOR	16 bits	Identificador de software
8004h	Input Register	FW_MINOR	16 bits	Versión del software

Los registros son necesarios para determinar el propósito funcional de la unidad de control, sus características y la versión del software a través de la red.

Para BLSD-20Modbus los valores son:

- HW\_MAJOR 1000
- HW\_MINOR x
- FW\_MAJOR x
- FW\_MINOR x

### 6.9. Instrucciones del programa de usuario

La estructura de las instrucciones del programa de usuario se muestra en el diagrama de la fig. 9.

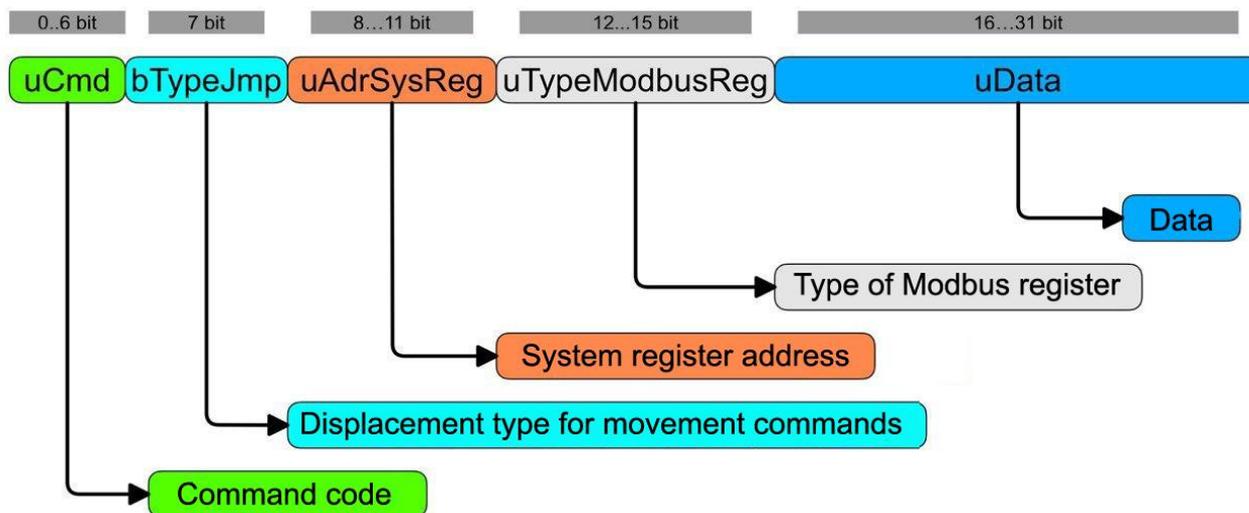


Fig. 9. Estructura de una instrucción de programa de usuario

La instrucción del programa de usuario tiene un tamaño de 32 bits y contiene los siguientes campos:

**uCmd** – 7 bits – código de comando

**bTypeJmp** – 1 bit – tipo de desplazamiento para comandos de movimiento:

- 0 – valor absoluto
- 1 – valor relativo

**uAdrSysReg** – 4 bits - dirección del registro del sistema AX\_REG ... FX\_REG con números 0 ... 5

**uTypeModbusReg** – 4 bits – tipo de registro Modbus:

- 0 – Entradas Discretas
- 1 – Bobinas
- 2 – Entradas
- 3 – Registros de retención

**uData** – 16 bits - datos.



Las instrucciones de 64 bits constan de dos líneas de comando de 32 bits. La primera línea de comando de esta instrucción es un comando similar a una instrucción de 32 bits. La segunda línea de comando contiene datos **uCMD\_DATA** de 16 o 32 bits.

Las instrucciones con el prefijo D, como DMOV, operan con datos de 32 bits ubicados en dos registros consecutivos de 16 bits. En el campo de dirección de dichas instrucciones, se indica el registro inferior. Al leer un valor, el comando lee automáticamente dos registros consecutivos de 16 bits a partir de la dirección especificada y forma un único valor de 32 bits. Al escribir un valor, la instrucción divide un número de 32 bits en dos partes de 16 bits y las escribe en dos registros consecutivos a partir de la dirección especificada.

La siguiente tabla enumera los comandos con opciones para rellenar los campos. Si el campo no se especifica, entonces no se utiliza en este comando.

uCmd		-	-	-
0x00	CMD_STOP_PROGRAM			
0x0D	CMD_FULL_STOP_PROGRAM			
uCmd		bTypeJmp	uData	-
0x05	CMD_JMP	0 – valor absoluto 1 – valor relativo	Dirección de movimiento 0..1024 o desplazamiento +- 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	Tiempo de retardo 0..65535 ms	-	-
0x0A	CMD_CALL	Dirección de subrutina 0..1024		
0x0B	CMD_RETURN	-		
0x0C	CMD_FOR	Duración del ciclo y número de ciclos		
uCmd		uAdrSysReg	-	-
0x15	CMD_NOT_SYSREG	Dirección de registro del sistema (0..9)	-	-
0x17	CMD_DNOT_SYSREG			
uCmd		uTypeModbusReg	uData	-
0x16	CMD_NOT_MODBUS	1 – Bobinas 3 – Registros de retención	Dirección de registro Modbus 0..65535	-
0x18	CMD_DNOT_MODBUS			
uCmd		uAdrSysReg	uData	-
0x01	CMD_MOV_SYSREG_CONST	Dirección de registro del sistema (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	Dirección de registro del sistema (0..9)	0 – Entradas Discretas 1 – Bobinas 2 – Entradas 3 – Registros de	Dirección de 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		retención	
0x26	CMD_OR_SYSREG_MODBUS			
0x27	CMD_XOR_SYSREG_MODBUS			
uCmd		uTypeModbusReg	uData	uCMD_DATA
0x0E	CMD_MOV_MODBUS_CONST	1 – Bobinas 3 – Registros de retención	Dirección de registro Modbus 0..65535	Constante 0..65535 (comando separado línea)
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	Dirección de registro del sistema (0..9)	0 – Entradas Discretas 1 – Bobinas 2 – Entradas 3 – Registros de retención	Dirección de 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		uAdrSysReg	uData	-
0x0F	CMD_MOV_SYSREG_SYSREG	Dirección de registro del sistema (0..9)	Dirección de registro del 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		uAdrSysReg	uCMD_DATA	-
0x10	CMD_DMOV_SYSREG_CONST	Dirección de registro del sistema (0..9)	Constante 0...4294967295 (comando separado línea)	-
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	Dirección de registro del sistema (0..9)	0 – Entradas Discretas 1 – Bobinas 2 – Entradas 3 – Registros de retención	Dirección de 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 – Bobinas 3 – Registros de retención	Dirección de registro Modbus 0..65535	Constante 0...4294967295 (comando separado línea)
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	Dirección de registro del sistema (0..9)	0 – Entradas Discretas 1 – Bobinas 2 – Entradas 3 – Registros de retención	Dirección de 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		uAdrSysReg	uData	-
0x14	CMD_DMOV_SYSREG_SYSREG	Dirección de registro del sistema (0..9)	Dirección de registro del sistema (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	Dirección de registro del sistema (0..9)	0 – desplazamiento a la izquierda 1 – desplazamiento a la derecha	Valor de desplazamiento
0x45	CMD_DSH_SYSREG_CONST			
uCmd		uTypeModbusReg	uData	bTypeJmp
0x2F	CMD_SH_MODBUS_CONST	1 – Bobinas 3 – Registros de retención	Dirección de registro Modbus 0..65535	0 – desplazamiento a la izquierda 1 – desplazamiento a la derecha
0x54	CMD_DSH_MODBUS_CONST			

**CMD\_STOP\_PROGRAM** – (código de comando 0x00) – detener la ejecución de un programa de usuario, sin salir del modo de ejecución del programa de usuario. Al final de la ejecución del programa, todos los registros y el estado del motor permanecen como estaban antes de que se ejecutara el comando (el motor continúa girando si estaba girando antes de que se ejecutara el comando). Antes del siguiente inicio del programa, debe enviar el comando CMD\_FULL\_STOP\_PROGRAM.

**CMD\_FULL\_STOP\_PROGRAM** – (código de comando 0x0D) - detener la ejecución del programa de usuario y salir del modo de operación del programa. Al final de la ejecución del programa, todos los registros y el estado del motor vuelven a sus valores originales, el motor se detiene.

**CMD\_MOV\_SYSREG\_CONST**– (código de comando 0x01) – escribir en el registro del sistema con la dirección uAdrSysReg, valores del campo de datos uData

**CMD\_MOV\_MODBUS\_SYSREG**– (código de comando 0x02) – escribir el contenido del registro del sistema uAdrSysReg en el espacio de registro ModBUS definido por el campo TypeModbusReg y su dirección en el campo uData.

**CMD\_MOV\_MODBUS\_SYSREG** – (código de comando 0x03) – leer el contenido del espacio de registro ModBUS determinado por el campo TypeModbusReg y su dirección en el campo uData en uno de los registros del sistema uAdrSysReg

**CMD\_MOV\_MODBUS\_CONST** – (código de comando 0x0E) – escribir la constante contenida en la siguiente línea de comando uCMD\_DATA en el espacio de registro ModBUS definido por el campo TypeModbusReg y su dirección en el campo uData.



**CMD\_MOV\_SYSREG\_SYSREG** – (código de comando 0x0F) – escribe el contenido del registro del sistema uAdrSysReg en otro registro del sistema con la dirección uData.

**CMD\_DELAY** – (código de comando 0x04) – pausa, ms.

**CMD\_JMP** – (código de comando 0x05) – salta a la dirección especificada en el campo uData.

**CMD\_JMP\_AX\_PARI\_BX** – (código de comando 0x06) – salta a la dirección especificada en el campo uData, si el valor en el registro del sistema AX\_REG es igual al valor en BX\_REG.

**CMD\_JMP\_AX\_NOPARI\_BX** – (código de comando 0x07) – salta a la dirección especificada en el campo uData si el valor en el registro del sistema AX\_REG no es igual al valor en BX\_REG.

**CMD\_JMP\_AX\_MORE\_BX** – (código de comando 0x08) – salta a la dirección especificada en el campo uData, si el valor en el registro del sistema AX\_REG es mayor que el valor en BX\_REG.

**CMD\_JMP\_AX\_LESS\_BX** – (código de comando 0x09) – salta a la dirección especificada en el campo uData, si el valor en el registro del sistema AX\_REG es menor que el valor en BX\_REG.

**CMD\_CALL** – (código de comando 0x0A) – llama a una subrutina que comienza en la dirección especificada en el campo uData.

**CMD\_RETURN** – (código de comando 0x0B) – regresa de la subrutina.

**CMD\_FOR** – (código de comando 0x0C) – ejecución cíclica de una secuencia de comandos. El byte alto del campo uData contiene el número de comandos ubicados después del comando CMD\_FOR que se repetirán en un ciclo. El byte menos significativo del campo uData contiene el número de repeticiones. Por ejemplo: uData = 0x1705 - 0x17 = 23 comandos ejecutados en un bucle, 0x05 = 5 - el número de repeticiones.

**CMD\_DJMP\_GX\_PARI\_IX** – (código de comando 0x63) – salta a la dirección especificada en el campo uData, si el contenido de un par de registros del sistema GX\_REG y HX\_REG es igual al contenido de IX\_REG y JX\_REG.

**CMD\_DJMP\_GX\_NOPARI\_IX** – (código de comando 0x64) – salta a la dirección especificada en el campo uData, si el contenido de un par de registros del sistema GX\_REG y HX\_REG no es igual al contenido de IX\_REG y JX\_REG.

**CMD\_DJMP\_GX\_MORE\_IX** – (código de comando 0x65) – salta a la dirección especificada en el campo uData, si el contenido de un par de registros del sistema GX\_REG y HX\_REG es mayor que el contenido de IX\_REG y JX\_REG.

**CMD\_DJMP\_GX\_LESS\_IX** – (código de comando 0x66) – salta a la dirección especificada en el campo uData, si el contenido de un par de registros del sistema GX\_REG y HX\_REG es menor que el contenido de IX\_REG y JX\_REG.

Son posibles las operaciones matemáticas entre registros del sistema, registros Modbus y constantes. Los operandos de las instrucciones que operan con datos de 32 bits se encuentran en dos registros consecutivos de 16 bits. Al acceder a un registro, la instrucción especifica la dirección del registro inferior. La dirección del registro alto se obtiene incrementando la dirección del registro bajo en uno.

La siguiente tabla muestra las instrucciones matemáticas, los códigos de instrucción para la operación con datos de 16 bits y 32 bits, y la ubicación de los operandos:

Comando		Código	Operand 1	Operand 2	Resultado
Nombre					
	16-bit data	32-bit data	S1	S2	D
Suma (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



Resta ( $S1 - S2 = D$ )					
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
División, resto descartado ( $S1 / S2 = D$ )					
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
Multiplicación ( $S1 * S2 = D$ )					
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 lógico bit a 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
OR lógico bit a 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 lógico bit a 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
Desplazamiento de datos ( $S1 \gg S2 = D$ o $S1 \ll S2 = D$ – la dirección del desplazamiento se indica en el campo bTypeJmp: 0 –izquierda, 1 - derecha)					



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 en el campo de datos uData

CONST\_2 - Una constante en la siguiente línea de comando uCMD\_DATA

SYS\_REG\_1 es el valor contenido en el registro del sistema (o dos registros de sistema consecutivos para instrucciones de 64 bits) en uAdrSysReg. Direcciones de los registros del sistema 0..9.

SYS\_REG\_2 el valor está contenido en el registro del sistema (o dos registros de sistema consecutivos para datos de 32 bits) cuya dirección se especifica en la siguiente línea de comando uCMD\_DATA

Modbus\_REG – el valor contenido en el registro Modbus: tipo de registro en el campo TypeModbusReg, dirección de registro en el campo uData

**Importante:** cuando la ejecución de un programa de usuario se detiene mediante el comando CMD\_STOP\_PROGRAM, el estado del motor y todos los registros permanecen igual que como se establecieron durante el programa. Por esta razón, si el motor estaba girando en el momento en que se ejecutó el comando CMD\_STOP\_PROGRAM, continuará ejecutando la última tarea, es decir, el movimiento continuará cuando el programa haya finalizado. Después de que el programa haya finalizado, la rotación del variador se puede detener escribiendo un registro a través de Modbus (Coils 2001h o Coils 2002h). Para evitar la rotación incontrolada, se puede establecer un comando de parada del motor antes del comando de fin del programa.

## 7. Restablecer a los valores predeterminados de fábrica

Si es necesario, los parámetros del controlador se pueden restablecer a los valores de fábrica. Para ello, antes de encender la unidad, cierre el primer contacto del conector RJ11 - CLR\_FLASH\_CON (conector RS-485, véase la Fig. 4) con la tierra GND de la fuente de alimentación. Los LED rojo y verde de la unidad se encenderán alternativamente. Mantenga CLR\_FLASH\_CON y GND cerrados durante 5 s. Después de eso, los parámetros del controlador se restablecerán a los valores de fábrica.

## 8. Entrega en juegos completos

Controlador de motor sin escobillas B LSD-20Modbus

1 ud

## 9. Información del fabricante

Smart Motor Devices OÜ se adhiere a la línea de desarrollo continuo y se reserva el derecho de realizar cambios y mejoras en el diseño y el software del producto sin previo aviso.

La información contenida en este manual está sujeta a cambios en cualquier momento y sin previo aviso.

## 10. Garantía

**Cualquier reparación o modificación es realizada por el fabricante o una empresa autorizada.**

El fabricante garantiza el funcionamiento sin fallos del controlador durante 12 meses desde la fecha de venta cuando se cumplen las condiciones de funcionamiento.

La dirección del departamento de ventas del fabricante:

Smart Motor Devices OÜ

Akadeemia tee 21/6, 12618, Tallinn, Estonia

Phone: + 372 6559914

Email: mail@smd.ee

URL: <https://smd.ee>

Última modificación: 04.2025