



SMART MOTOR DEVICES

<https://www.smd.ee>

**Controlador de motor de escobillas de CC
BMSD-20Modbus / BMSD-40Modbus**

**Manual de usuario
BMSD.Modbus.001
2025**



1. Designación del producto

Los controladores de motor de escobillas BMSD-20Modbus y BMSD-40Modbus son dispositivos electrónicos diseñados para operar y controlar motores de escobillas de cc.

2. Funciones y posibilidades

Los controladores están diseñados para controlar la velocidad, la aceleración, la deceleración y el sentido de giro del motor. Las unidades también proporcionan posicionamiento basado en señales de un codificador en caso de utilizar un motor con sensores hall.

El motor se controla mediante señales externas o mediante comandos transmitidos a través de RS-485 mediante el protocolo Modbus. El controlador también puede funcionar de acuerdo con un algoritmo previamente grabado en su memoria 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 admitidos 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 de destino 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 memoria no volátil. Se admiten saltos condicionales e incondicionales (relativos y absolutos), llamadas a subrutinas, bucles y temporizadores.
- Las entradas programables IN1 e IN2 se pueden utilizar como señales START/STOP, REVERSE o para otros fines a discreción del usuario.
- El controlador puede realizar posicionamiento en el rango de -2147483647 a +2147483648 pulsos si utiliza un motor con sensores hall.
- El controlador tiene contactos RT en el panel frontal para una conexión de resistencia de terminación.

Control mediante señales externas

Para controlar el motor mediante señales externas, se proporcionan los siguientes:

- Contactos para conectar las señales externas IN1 e IN2, cuyo propósito y procesamiento son determinados por 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 - frenado controlado del motor en caso de interrupción del circuito de emergencia.

El controlador proporciona una función de protección contra sobrecorriente del motor. La corriente máxima permitida en la fase del motor es establecida por el usuario.



3. Características técnicas

Modelo	BMSD-20Modbus	BMSD-40Modbus
Tensión de alimentación, VCC	12 - 24	
Protección de la fuente de alimentación, VCC	8 - 30	
Corriente nominal en la fase del motor, A	<20	<40
Protección por hardware contra cortocircuitos (tiempo de funcionamiento 15 μ s), A	30	100
Limitación de corriente de fase, A	1...20	2...40
Tensión máxima en la fase del motor, V	0.98 x U _{sup}	
Mínimo no nulo en la fase del motor, V	0.01 x U _{sup}	
Parámetros de las señales externas IN1 e IN2:		
Resistencia máxima de los contactos cerrados, kOhm	4.7	
Corriente de entrada máxima, mA	0.5	
Dimensiones (no más de), 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.

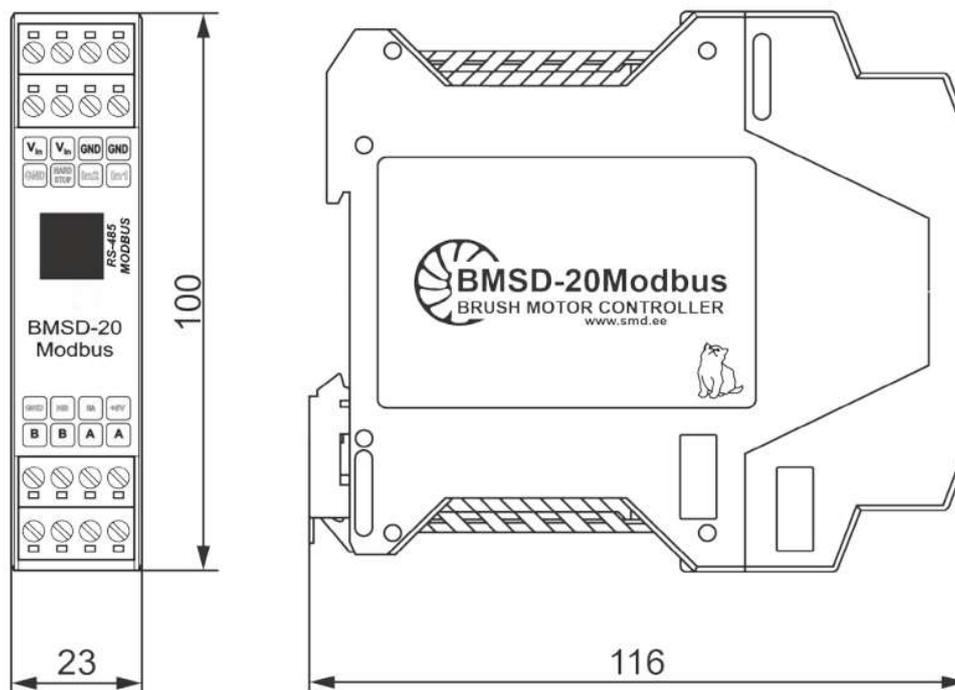


Fig. 1. Las dimensiones generales y de conexión del controlador BMSD-20Modbus.

El esquema de conexión se muestra en la Fig. 2

Condiciones ambientales:

Temperatura ambiente: 0...+50°C

Humedad: 90% HR o menos en condición de +25°C

Condensación y congelación: ninguna



4. Construcción y elementos de control

El controlador está diseñado como una placa de circuito con elementos electrónicos, cubierta con una caja con un montaje en carril din. En la parte superior de la caja, hay símbolos gráficos para los controles y 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;
- terminal para conectar contactos de señal de parada de emergencia;
- indicador led de funcionamiento del dispositivo;
- Conector RJ11 (6P6C) para conectar líneas RS-485;
- contactos para conectar la resistencia de terminación interna RT;
- circuito de frenado incorporado para absorber la energía generada por el motor (durante la marcha por inercia, rotación forzada).

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 llevar a cabo mediante software y mediante comandos transmitidos a través de RS-485 a través del protocolo Modbus.

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



1. GND de la fuente de alimentación
2. GND de la fuente de alimentación
3. Fuente de alimentación 12 – 24 VCC
4. Fuente de alimentación 12 – 24 VCC
5. Señal "IN1" (contacto seco)
6. Señal "IN2" (contacto seco)
7. Señal de parada de emergencia "HARD STOP"
8. Señal GND
9. Salida de +5V para la alimentación de sensores hall
10. Línea de datos A del sensor hall
11. Línea de datos B del sensor hall
12. GND de los sensores hall
13. Cable A del motor
14. Cable A del motor
15. Cable B del motor
16. Cable B del motor

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

RT - contactos para conectar la resistencia terminal

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



5. Montaje y conexión

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

Por favor, cablee solo cuando la alimentación esté apagada. No intente cambiar el cableado mientras la alimentación está encendida.

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

IMPORTANTE: Debido a las altas corrientes, se recomienda ubicar la fuente de alimentación cerca de la unidad y utilizar cables con una sección transversal de 3 mm² (AWG-12). La fuente de alimentación debe proporcionar un 20% más de corriente que el máximo posible consumido durante el funcionamiento.

Longitud recomendada de los cables de alimentación:

- No más de 100 cm con corrientes de hasta 10 A.
- No más de 50 cm con corrientes de 10 a 20A.
- No más de 25 cm con corrientes de 20 a 40A.

Con una corriente máxima de hasta 20 A, se permite utilizar terminales de alimentación y de fase en una misma línea. Para una corriente máxima de más de 20 A, se deben utilizar ambas líneas de los terminales de alimentación y de fase.

Siga la siguiente instrucción durante la conexión:

1. Conecte un motor al controlador según la Fig. 2. El motor debe estar conectado a los terminales A y B (13 – 16) del controlador. Las señales del sensor hall deben estar conectadas a los terminales HA y HB (10 – 11). La GND de los sensores hall debe estar conectada al terminal GND (12), la alimentación de las señales del sensor hall debe estar conectada al terminal +5V (9).
2. Conecte los elementos de control externos de acuerdo con los esquemas del diagrama de conexión en la Fig. 3:

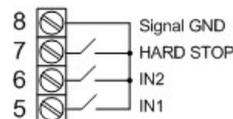


Fig. 3. Conexión de elementos de control externos.

Tipo de señales externas «IN1», «IN2», «HARD STOP» - contacto seco.

3. Conecte las líneas de la interfaz RS-485 al conector RJ11 según la Fig. 4.

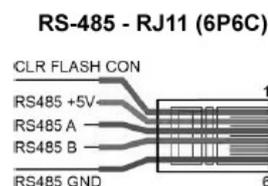


Fig. 4. Asignación de pines del conector RJ11 - RS-485 Modbus.

4. Si es necesario, coloque un puente en los pines RT para conectar la resistencia de terminación interna.
5. Conecte el dispositivo a la unidad de fuente 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 los terminales 3 y 4, conecte el "-" de la fuente de alimentación a los terminales 1 y 2 del controlador. El procedimiento de desmontaje está en orden inverso.



6. Operación

1. Asegúrese de que la fuente de alimentación esté apagada. Por favor, cablee 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. 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 una parada de emergencia del motor. Se permite el funcionamiento con contacto cerrado.
5. Conecte las líneas de la interfaz RS-485 según la sección 5.
6. Encienda la fuente de alimentación. El dispositivo está listo para la configuración a través del protocolo Modbus.
7. Establezca los parámetros de operación necesarios mediante comandos a través del protocolo Modbus: limitación de corriente del motor, dirección de rotación, configuración de la operación de las entradas externas IN1 e IN2, y método de control.
8. Para controlar el variador utilizando el protocolo Modbus, envíe comandos a través de la interfaz RS-485. La tabla de registros del controlador, su propósito y los posibles comandos de control del motor se detallan a continuación.
9. Para controlar el variador con señales externas para arrancar y detener el motor y para controlar el sentido de giro, utilice las señales IN1 e IN2. La regulación de la velocidad se realiza 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 Modbus estándar (ASCII o RTU).

La unidad de control tiene la siguiente configuración de fábrica:

- ID = 1
- Tasa: 115200 baudios
- Comprobación 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	entrada discreta	IN1_bit	1 bit	Estado de la señal de entrada IN1
1001h	entrada discreta	IN2_bit	1 bit	Estado de la señal de entrada IN2
1002h	entrada discreta	IN_HARD_STOP_bit	1 bit	Estado de la señal de entrada HARD_STOP
5007h	registro de retención	MODE_EXT_IN	16 bits	Configuración de las entradas externas IN, IN2.
5013h	registro de retención	PRESSED_INPUTS_EXTERN	16 bits	Tiempo mínimo de pulso positivo en las entradas digitales IN1, IN2 (ms)
5014h	registro de retención	WAITED_INPUTS_EXTERN	16 bits	Tiempo mínimo para la parte negativa del pulso en las entradas digitales IN1, IN2 (ms)

Registros de estado de la 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.

Valores de registro posibles 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).

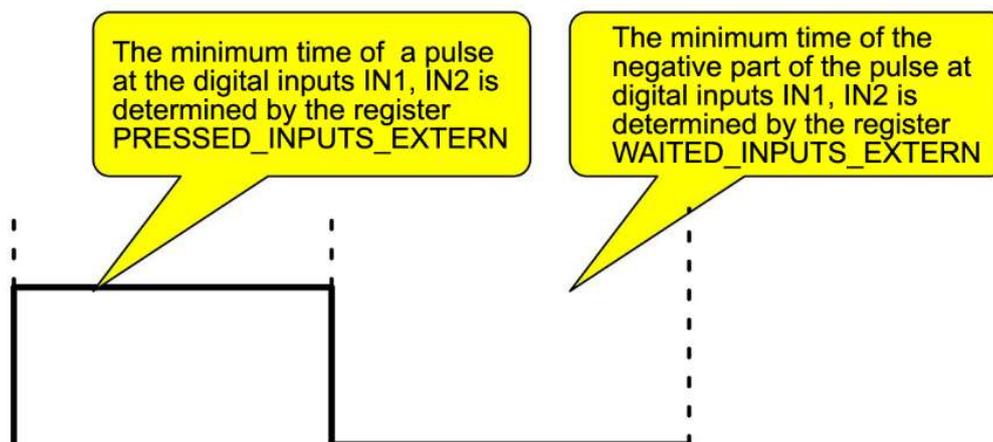


Fig. 5. Supresión del rebote de contacto.



6.2. registros de control del motor

Dirección	Tipo	Nombre	Tamaño	Descripción
2000h	bobinas	START_bit	1 bit	Iniciar la rotación del motor con una aceleración establecida
2001h	bobinas	STOP_bit	1 bit	Detener la rotación del motor con una deceleración establecida
2002h	bobinas	HARD_STOP_bit	1 bit	Parada de emergencia brusca de la rotación del motor
2003h	bobinas	CLR_POSITION_bit	1 bit	Poner a cero el registro de posición actual POSITION_VALUE.

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, después de lo cual el registro se restablece a 0.

6.3. Registros de estado

Dirección	Tipo	Nombre	Tamaño	Descripción
3000h	registro de entrada	STATUS	16 bits	El estado actual del control del motor.
3001h	registro de entrada	CURRENT_VALID	16 bits	El valor actual de la corriente consumida por el motor.
3002h	registro de entrada	SPEED_VALID	16 bits	El valor instantáneo de la velocidad de giro.
3003h	registro de entrada	CURRENT_POSITION	32-bit	Posición actual. Los valores van de -2147483647 a +2147483648
3005h	registro de entrada	TEMPERATURE_MCU	16 bits	temperatura de la CPU
3006h	registro de entrada	TEMPERATURE_MOSFET	16 bits	temperatura del circuito de potencia
3007h	registro de entrada	TEMPERATURE_BRAKE	16 bits	temperatura del circuito de frenado
3008h	registro de entrada	TASK_COUNTER	16 bits	Devuelve un valor aleatorio (para comprobar el funcionamiento del canal de transmisión)
3009h	registro de entrada	STATUS_USER_PROGRAM	16-bit	El estado actual de la ejecución del programa de usuario

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

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

- 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 giro. 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 y bajada.



3005h -**TEMPERATURE_MCU**— temperatura del procesador central. La temperatura se calcula como $TEMPERATURE_MCU/10$ deg/C.

3006h -**TEMPERATURE_MOSFET**- temperatura del circuito de potencia - la temperatura en el área de instalación de interruptores mosfet. La temperatura se calcula como $TEMPERATURE_MOSFET/10$ deg/C.

3007h -**TEMPERATURE_BRAKE**- temperatura del circuito de frenado – la temperatura en el área de la resistencia de frenado. La temperatura se calcula como $TEMPERATURE_MOSFET/10$ deg/C.

3008h -**TASK_COUNTER**- devuelve un número aleatorio en el rango 0x0000 a 0xFFFF.

3009h - **STATUS_USER_PROGRAM** - el estado actual de la ejecución 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 un período corto, antes de que se establezca el estado 4)
- 3 – el programa de usuario se inicia después de que se enciende el voltaje de alimentación (este estado es solo un período corto, antes de que se establezca el estado 4)
- 4 – el programa de usuario está ejecutando
- 5 – el programa de usuario finaliza con el comando END
- 6 – el programa de usuario finaliza con el comando ENDF
- 7 – el programa de usuario se detiene debido a un error

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

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

5001h -**TYPE_MODBUS**– configuración de la 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 mucho tiempo en cambiar del modo de transmisión al modo de recepción.

Después de cambiar la configuración de la comunicación, los nuevos valores deben guardarse utilizando el registro FLAG_SAVE_INI (ver sección 6.5), y luego el controlador debe reiniciarse. Después del reinicio, la conexión RS-485 se realizará utilizando la nueva configuración.

6.5. Registros para la configuración del funcionamiento del accionamiento

Dirección	Tipo	Nombre	Tamaño	Descripción
5004h	registro de retención	MODE_DEVICE	16 bits	Modo de operación del controlador.
5005h	registro de retención	MODE_USER_PROGRAM	16 bits	Comando de control para iniciar un programa de usuario.
5006h	registro de retención	MODE_ROTATION	16 bits	Modo de rotación.
5007h	registro de retención	MODE_EXT_IN	16 bits	Configuración de las entradas externas IN, IN2.
5008h	registro de retención	POSITION_N	16 bits	El número de posición al que moverse. Los valores van de 1 a 4
5009h	registro de retención	REF_CURRENT	16 bits	Limitación del consumo de corriente. Rango: 1000 mA a 20000 mA para BMSD-20Modbus 2000 mA a 40000 mA para BMSD-40Modbus
500Ah	registro de retención	RATED_SPEED	16 bits	Velocidad nominal del motor: Rango de 1000 a 15000 rpm
500Bh	registro de retención	SPEED	16 bits	Velocidad de giro establecida. Rango de 30 a 15000 rpm.
500Ch	registro de retención	ACC	16 bits	Aceleración establecida. Rango de 10 a 1000.
500Dh	registro de retención	DEC	16 bits	Deceleración establecida. Rango de 10 a 1000.
500Eh	registro de retención	DIRECTION	16 bits	Dirección de rotación. 1 – hacia adelante 2 – hacia atrás
500Fh	registro de retención	PULSES_PER_REVOLUTION	16 bits	Número de pulsos por revolución de un sensor hall: Rango de 1 a 12
5010h	registro de retención	USE_HALL	16 bits	Uso de sensores hall: 0 – sin sensores hall 1 – un sensor hall 2 – dos sensores hall
5011h	registro de retención	MODE_COIL	16 bits	Estado de los terminales del motor cuando está parado: 0 - abierto 1 - cerrado



5012h	registro de retención	OFFSET_COMPENSATION	16 bits	Corrección de frenado del motor
5013h	registro de retención	PRESSED_INPUTS_EXTERN	16 bits	Tiempo mínimo de pulso positivo en las entradas digitales IN1, IN2 (ms)
5014h	registro de retención	WAITED_INPUTS_EXTERN	16 bits	Tiempo mínimo para la parte negativa del pulso en las entradas digitales IN1, IN2 (ms)
5015h	registro de retención	OFFSET	32-bit	El desplazamiento a mover, el valor es modificado por el controlador durante la operación. Rango de valores de -2147483647 a + 2147483648
5017h	registro de retención	OFFSET_CONST	32-bit	Incremento - el desplazamiento al que es necesario moverse.
5019h	registro de retención	TARGET_POSITION	32-bit	La posición de destino. Rango de valores de -2147483647 a + 2147483648
501Bh	registro de retención	TARGET_POSITION1	32-bit	posición preestablecida por el usuario №1 Rango de valores de -2147483647 a + 2147483648
501Dh	registro de retención	TARGET_POSITION2	32-bit	posición preestablecida por el usuario №2 Rango de valores de -2147483647 a + 2147483648
501Fh	registro de retención	TARGET_POSITION3	32-bit	posición preestablecida por el usuario №3 Rango de valores de -2147483647 a + 2147483648
5021h	registro de retención	TARGET_POSITION4	32-bit	posición preestablecida por el usuario №4 Rango de valores de -2147483647 a + 2147483648
5023h	registro de retención	ERROR	16 bits	Registro de errores
5024h	registro de retención	FLAG_SAVE_INI	16 bits	registro para guardar la configuración del usuario. (Registros 5000h..501Fh). Valor: 0x37FA
5025h	registro de retención	FLAG_SAVE_USER_PROGRAM	16 bits	Registro para escribir o leer un programa de usuario en o desde la memoria no volátil. Valor para escribir: 0x8426 Valor para leer: 0x9346
5026h	registro de retención	FLAG_RESTART	16 bits	Registro de reinicio. Valor: 0x95AF

5004h -**MODE_DEVICE**- modo de operación del controlador, valores posibles:

- 1 – controla través de Modbus
- 2 – control por 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 enciende la alimentación (se requiere guardar previamente la configuración - ver registro FLAG_SAVE_INI)

El programa de usuario se puede almacenar en la memoria flash interna del controlador. Para iniciar la ejecución de un programa de usuario, es necesario escribir 2 en el registro MODE_USER_PROGRAM. Para detener la ejecución del programa, es necesario escribir 1 en el registro MODE_USER_PROGRAM.

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 y 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 retroceso, 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 de acuerdo con el nivel de señal.
- 3 - La entrada IN1 se utiliza como una señal de arranque/parada del variador, se procesa de acuerdo con el nivel de señal: presencia de una señal - habilitar la rotación del motor, sin señal - detener; la entrada IN2 se utiliza como una señal de retroceso, 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 señal: presencia de una señal - habilita la rotación del motor, sin señal - se detiene; la entrada IN2 se utiliza como señal de referencia de dirección, procesada según el nivel de señal.
- 5 – la entrada IN1 se utiliza como señal para arrancar y detener el variador en la dirección de avance, IN2 - como señal para arrancar y detener 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) - utilizado en el modo de movimiento a una posición dada (MODE_ROTATION = 3) en conjunto con los registros POSITION1, POSITION2, POSITION3, POSITION4.

5009h –**REF_CURRENT**– ajuste del límite de consumo de corriente. El rango de valores permitidos para BMSD-20Modbus: de 1000 mA a 20000 mA; para BMSD-40Modbus: de 2000 mA a 40000 mA.

500Ah –**RATED_SPEED**– velocidad nominal del motor: de 1000 a 15000 rpm. Es imposible conocer la velocidad real del motor si el motor se utiliza sin sensores hall. Por esta razón, la velocidad máxima del motor (que corresponde al suministro máximo al motor) se utiliza para calcular los valores establecidos de la velocidad. Si la RATED_SPEED no se establece correctamente, la velocidad real del motor difiere sustancialmente del valor objetivo, que se establece en el registro SPEED.

500Bh –**SPEED**– velocidad de rotación objetivo (de 30 a 15000 rpm). El valor de la velocidad objetivo debe ser inferior al valor MAX_SPEED.

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/desaceleración. Un valor de 10 corresponde a 100 rps², un valor de 1000 corresponde a 5000 rps².

500Eh –**DIRECTION**– sentido de giro, valores posibles:

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

500Fh –**PULSES-PER_REVOLUTION**– número de pulsos por revolución de un sensor hall (de 1 a 12).

5010h –**USE_HALL**– uso de sensores hall:

- 0 – un motor sin sensores hall
- 1 – un sensor hall
- 2 – dos sensores hall.

5011h –**MODE_COIL**– Estado de los terminales del motor cuando está parado:

- 0 - abierto
- 1 - cerrado

5012h –**OFFSET_COMPENSATION**– Corrección de frenado del motor calculada experimentalmente para la configuración actual 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 **OFFSET_COMPENSATION** = -15, y esta corrección es válida solo para la configuración actual de aceleración, deceleración y velocidad.

5013h y 5014h –**PRESSED_INPUTS_EXTERN** y **WAITED_INPUTS_EXTERN**– tiempo mínimo de una parte positiva y una negativa de un pulso en las entradas digitales IN1, IN2 – utilizado para suprimir el rebote de contacto (ver fig. 5).

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

5017h –**OFFSET_CONST**– Incremento - el offset al que es necesario moverse. Si el valor del registro **OFFSET_CONST** ≠ 0, se copia al registro **OFFSET** en cada arranque del motor.

5019h –**TARGET_POSITION**– posición a la que moverse, valores permitidos desde -2147483647 hasta + 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 desde - 2147483647 hasta + 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 transistor de salida;
- bit 11 - se excede el límite de corriente de consumo;
- 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 en la transmisión RS-485;
- bit 15 - uso del modo de posicionamiento con menos de dos sensores Hall.

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**– escribir el valor 0x8426 en este registro inicia el procedimiento para guardar el programa de usuario desde el búfer temporal a la memoria no volátil del controlador. Escribir el valor 0x9346 inicia el procedimiento para leer el programa de 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.

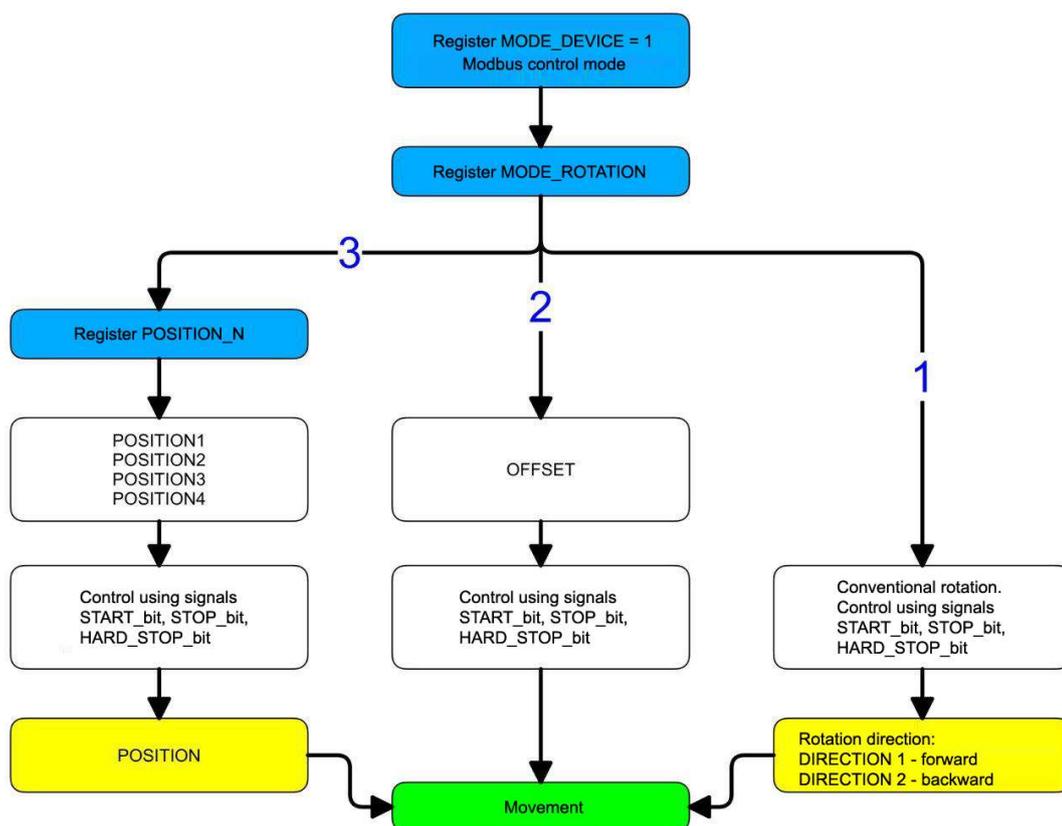


Fig. 6. Diagrama de flujo para seleccionar el modo de funcionamiento al controlar el accionamiento a través de Modbus.

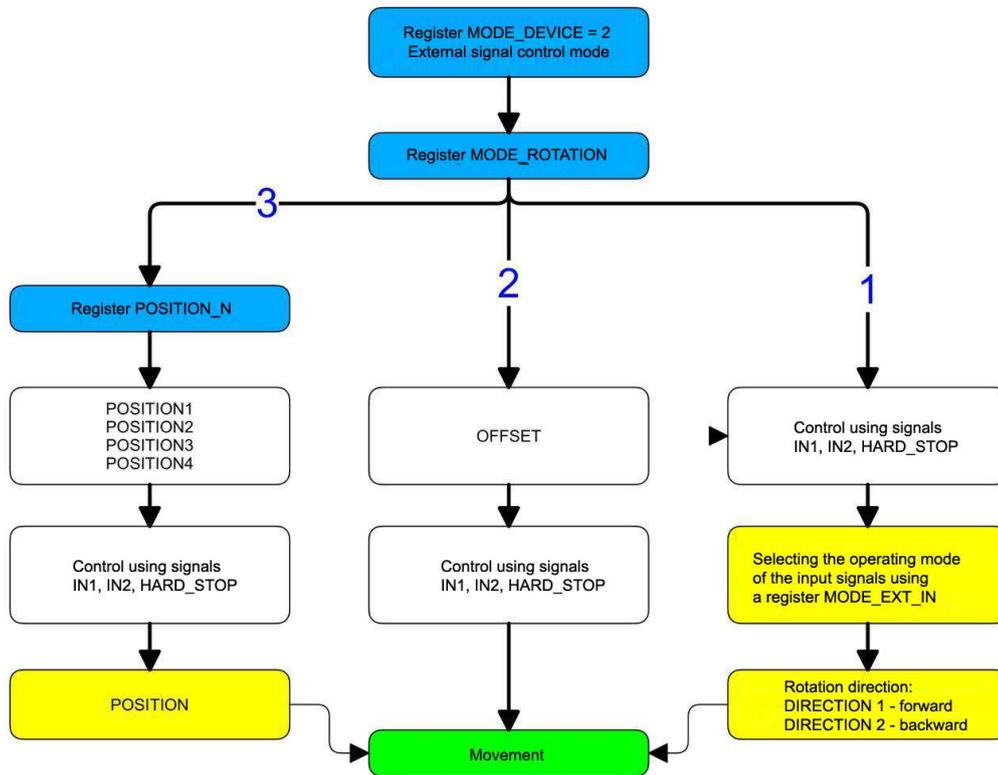


Fig. 7. Diagrama de flujo para seleccionar el modo de funcionamiento al controlar el accionamiento utilizando señales externas.

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	registro de retención	FLAG_SAVE_USER_PROGRAM	16 bits	Registro para escribir o leer un programa de usuario en o desde la memoria no volátil del controlador. Valor para escribir: 0x8426 Valor para leer: 0x9346
6000h	registro de retención	WRITE_CMD	16 bits	Registro de dirección. Cuando el valor de la dirección se escribe en este registro, el comando se transfiere desde el campo CMD_W a la dirección especificada del búfer temporal del programa de usuario.
6001h	registro de retención	CMD_W	32-bit	Instrucciones del programa de usuario
6003h	registro de retención	READ_CMD	16 bits	Registro de dirección. Cuando el valor de la dirección se escribe en este registro, el comando se transfiere desde la dirección especificada del búfer temporal del programa de usuario al campo CMD_R.
6004h	registro de retención	CMD_R	32-bit	Instrucciones 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 la dirección se escribe 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 hacer esto, 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 desde el búfer temporal, es necesario escribir la dirección de la instrucción en el registro READ_CMD; cuando la dirección se escribe en este registro, la instrucción se copiará desde el 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.

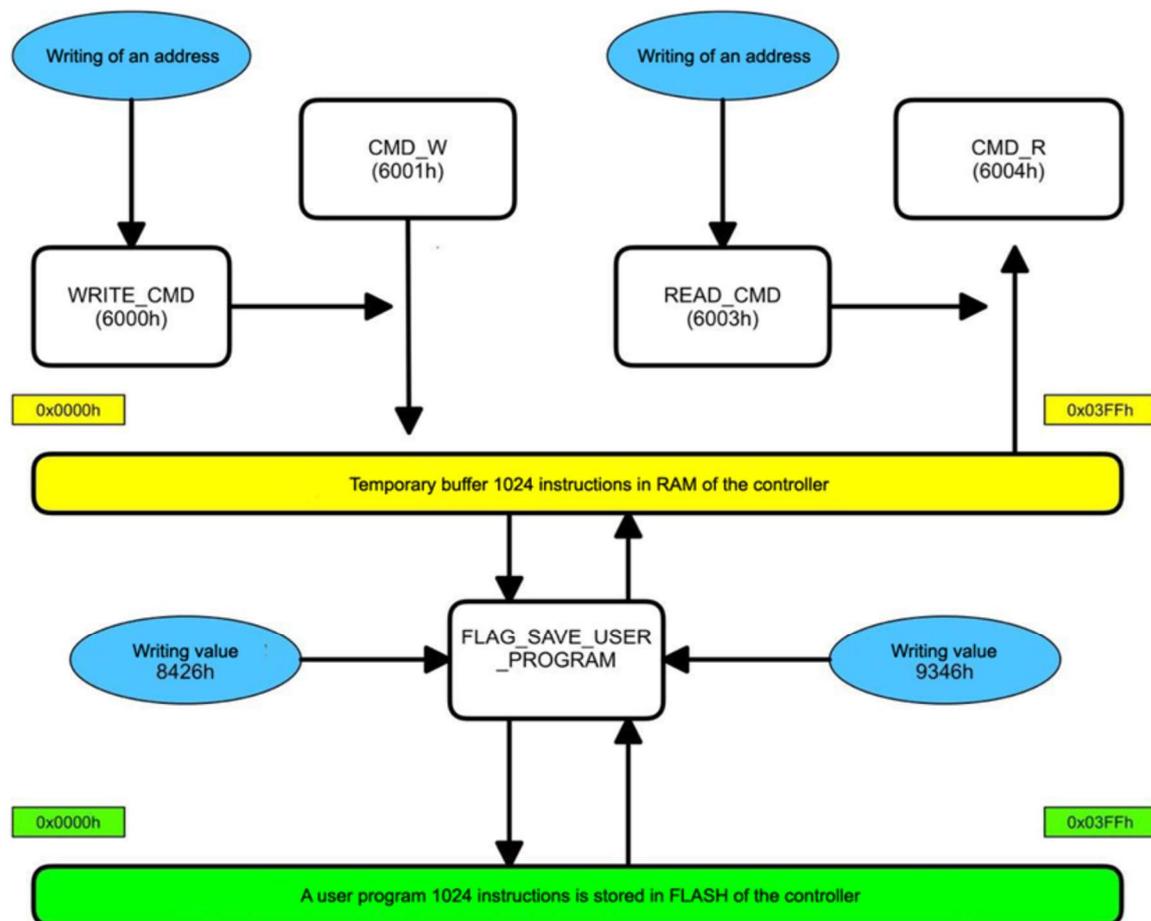


Fig. 8. Diagrama de flujo de los procedimientos para leer y escribir un programa de usuario.



6.7. Registros del sistema

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

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	registro de entrada	HW_MAJOR	16 bits	Tipo de controlador
8002h	registro de entrada	HW_MINOR	16 bits	Versión de hardware
8003h	registro de entrada	FW_MAJOR	16 bits	Identificador de software
8004h	registro de entrada	FW_MINOR	16 bits	Versión de software

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

Para BMSD-20Modbus, los valores son:

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



Para BMSD-40Modbus, los valores son:

- HW_MAJOR 1002
- 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 en la Fig. 9.

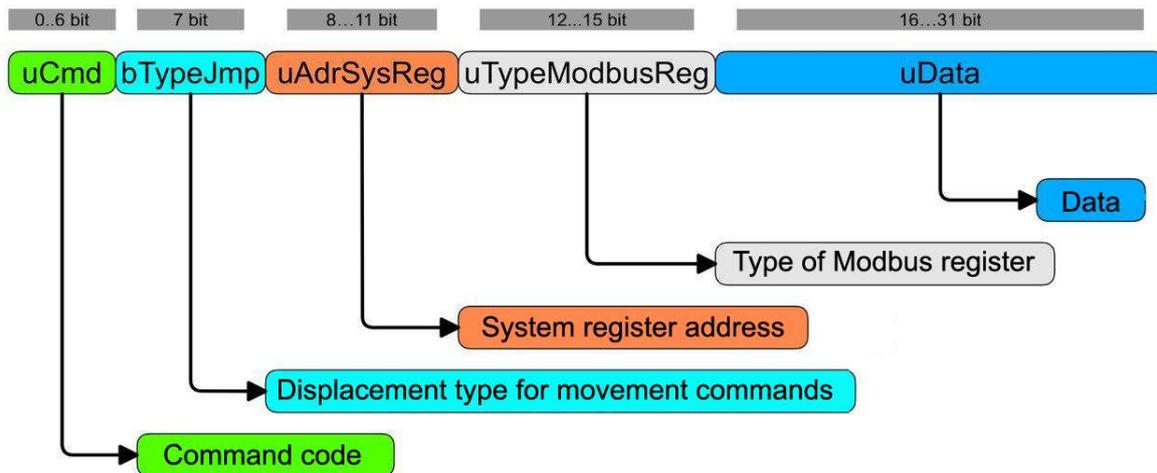


Fig. 9. Estructura de una instrucción de programa de usuario.

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 los 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 un 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 un prefijo D, como DMOV, operan con datos de 32 bits ubicados en dos registros de 16 bits consecutivos. En el campo de dirección de tales instrucciones, se indica el registro inferior. Al leer un valor, el comando lee automáticamente dos registros de 16 bits consecutivos 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 comenzando en la dirección especificada.

La siguiente tabla enumera los comandos con opciones para completar los campos. Si el campo no está especificado, 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 salto 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	longitud de 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 retención	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			
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		Constante 0..65535 (línea de comando)
0x28	CMD_ADD_MODBUS_CONST			
0x29	CMD_SUB_MODBUS_CONST			



0x2A	CMD_DIV_MODBUS_CONST			separada)
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 (línea de comando separada)	-
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 (línea de comando separada)
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) – detiene la ejecución del programa de usuario, sin salir del modo de ejecución del programa de usuario. Al final de la ejecución del programa, todos los registros y estados 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 próximo inicio del programa, debe enviar el comando CMD_FULL_STOP_PROGRAM.

CMD_FULL_STOP_PROGRAM – (código de comando 0x0D) - detiene la ejecución del programa de usuario y sale 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, y el motor se detiene.

CMD_MOV_SYSREG_CONST – (código de comando 0x01) – escritura en el registro del sistema con la dirección uAdrSysReg, valores del campo de datos uData

CMD_MOV_MODBUS_SYSREG – (código de comando 0x02) – escritura del contenido del registro del sistema uAdrSysReg al espacio de registros modbus definido por el campo TypeModbusReg y su dirección en el campo uData.

CMD_MOV_MODBUS_SYSREG – (código de comando 0x03) – lectura del contenido del espacio de registros 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) – escritura de la constante contenida en la siguiente línea de comando uCMD_DATA al espacio de registros modbus definido por el campo TypeModbusReg y su dirección en el campo uData.

CMD_MOV_SYSREG_SYSREG – (código de comando 0x0F) – escritura del contenido del registro del sistema uAdrSysReg a otro registro del sistema con la dirección uData.

CMD_DELAY– (código de comando 0x04) – pausa, ms.

CMD_JMP – (código de comando 0x05) – salto a la dirección especificada en el campo uData.

CMD_JMP_AX_PARI_BX – (código de comando 0x06) – salto 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) – salto 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) – salto 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) – salto 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) – llamada a una subrutina que comienza en la dirección especificada en el campo uData.

CMD_RETURN – (código de comando 0x0B) – retorno 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) – salto 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) – salto 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) – salto 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.

Las operaciones matemáticas son posibles 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 de 16 bits consecutivos. Al acceder a un registro, la instrucción especifica la dirección del registro inferior. La dirección del registro superior se obtiene incrementando la dirección del registro inferior 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 las ubicaciones de los operandos:

Comando			Operando 1	Operando 2	Resultado
Nombre	Código				
	Datos de 16 bits	Datos de 32 bits	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, residuo 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 a nivel de bits (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 a nivel de bits (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 a nivel de bits ($S1 \wedge S2 = D$)					
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 de desplazamiento se indica en el campo bTypeImp: 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 del sistema consecutivos para instrucciones de 64 bits) en uAdrSysReg. Direcciones de registros del sistema 0..9

SYS_REG_2 el valor está contenido en el registro del sistema (o dos registros del 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, continuaría ejecutando la última tarea, es decir, el movimiento continuaría cuando el programa hubiera terminado. Una vez finalizado el programa, la rotación del accionamiento se puede detener escribiendo un registro a través de Modbus (Bobinas 2001h o Bobinas 2002h). Para evitar la rotación no controlada, se puede establecer un comando de parada del motor antes del comando de fin de programa, o utilizar el comando CMD_END_STOP_PROGRAM.

7. Restablecer los valores de fábrica

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

8. Entrega en juegos completos

Controlador de motor de escobillas de cc BMSD-20Modbus o BMSD-40Modbus

1 pza

9. Información del fabricante

Smart Motor Devices 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 la realiza el fabricante o una empresa autorizada.

El fabricante garantiza el funcionamiento sin fallos del controlador durante 12 meses a partir de la fecha de venta, siempre que se cumplan las condiciones de funcionamiento.

Dirección del departamento de ventas del fabricante:

Smart Motor Devices OÜ

Akadeemia tee 21/6, 12618, Tallinn, Estonia

Teléfono: + 372 6559914

Correo electrónico: mail@smd.ee

URL: <https://smd.ee>

Última modificación: 07.2025