«НПО Электропривод»
https://www.electroprivod.ru

Блок управления коллекторным двигателем постоянного тока

Паспорт
BMSD.MODBUS.001.ПС

г. Санкт-Петербург

2020

1. НАЗНАЧЕНИЕ УСТРОЙСТВА

Блоки управления коллекторным двигателем постоянного тока BMSD-20Modbus и BMSD-40Modbus представляют собой электронные устройства, предназначенные для управления коллекторными двигателями.

2. ОПИСАНИЕ И ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

Блоки управления позволяют управлять скоростью, ускорением, торможением и направлением вращения двигателя. Блоки также обеспечивают возможность стабилизации скорости по сигналам от датчика Холла при работе с двигателем с энкодером.

Управление коллекторным двигателем осуществляется либо внешними сигналами, либо программно (командами, передаваемыми по протоколу Modbus или по предварительно записанному в память блока алгоритму).

Управление по RS-485 Modbus

Блок имеет возможность удаленного управления по физической линии связи RS-485 с использованием промышленного протокола Modbus:

- Настойка производится путем записи или чтения соответствующих параметров в/из регистров блока.
- Поддерживаются протоколы RTU и ASCII на скоростях: 600,1200, 2400, 4800, 9600, 14400, 19200,38400, 57600, 115200,128000 бод.
- Реализованы перемещения в произвольную или одну из четырех заранее заданных реперных точек.
- Блок имеет возможность автономной работы под управлением пользовательской программы (длиной до 1024 команд), заранее записанной в энергонезависимую память. Поддерживаются условные и безусловные переходы (относительные и абсолютные), вызовы подпрограмм, циклы, таймеры ожидания.
- Программируемые входы IN1 и IN2 можно использовать в качестве сигналов START/STOP, REVERS, либо для иных целей по усмотрению пользователя.
- Блок имеет возможность осуществления позиционирования в диапазоне от – 2147483647 до + 2147483648 переключений датчиков Холла (при работе с двигателем с датчиками Холла).
- Блок имеет контакты RT на лицевой панели для подключения внутреннего терминального резистора, для этого необходимо замкнуть контакты перемычкой.

Управление внешними сигналами

Для управления двигателем внешними сигналами предусмотрены:

- Контакты для подключения внешних сигналов IN1 и IN2, назначение и способ обработки которых определяется пользователем, используются для подачи сигналов START/STOP (старт/остановка) и DIR (направление вращения).

- Контакт HARD STOP для подключения аварийного сигнала, обеспечивающего работу функции Safe Stop 1 (SS1) - быстрое управляемое торможение двигателя при разрыве аварийного контура. После остановки происходит переход в одно из следующих состояний: клеммы двигателя замкнуты или клеммы двигателя разомкнуты (данное состояние определяется регистром 5011h MODE_COIL п.6.5). Так же предотвращается несанкционированный запуск.

Блок обеспечивает функцию защиты двигателя от перегрузки по току. Максимальное разрешенное значение тока двигателя устанавливается пользователем.

Технические характеристики блока

Модель	BMSD-20Modbus	BMSD-40Modbus
Напряжение питания, В	12 – 24	
Допустимый диапазон напряжения питания до срабатывания защиты, В	8 – 30	
Номинальный ток в фазе, А	<20	<40
Аппаратная защита от короткого замыкания (время срабатывания 15 мкс), А	30	100
Ограничение тока фазы (время срабатывания 5 с), А	1…20	2…60
Максимальное напряжение на двигателе, В	0,98 x Uпит	
Минимальное ненулевое напряжение на двигателе, В	0,01 x Uпит	
Собственный ток потребления, мА	≤ 50 при Uпит=24В	
Диапазон регулирования скорости, ωв/ωн	500	
Параметры внешних сигналов «IN1» и «IN2»:		

Максимальное сопротивление замкнутых контактов, кОм	4,7
Максимальный входной ток, мА	0.5
Габаритные размеры не более, мм	116x100x23
Коммуникационный интерфейс	RS-485, Modbus – ASCII или RTU

Габаритные и присоединительные размеры блока приведены на Рис. 1

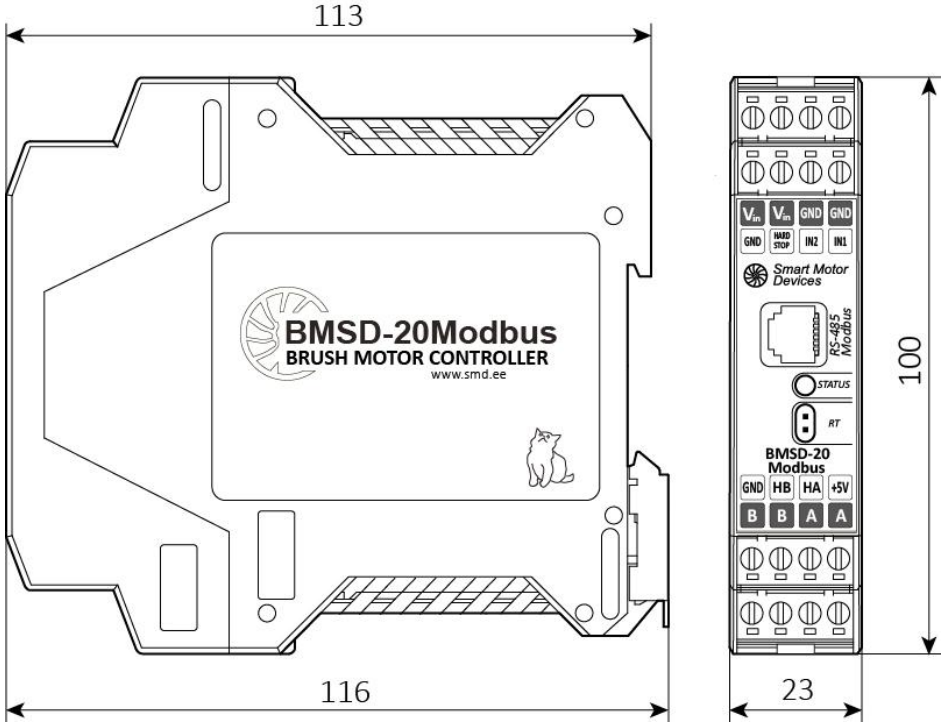

Рис.1. Габаритные и присоединительные размеры блока BMSD-20Modbus и BMSD-40Modbus

Условия эксплуатации блока:

- температура окружающего воздуха – (0…+50)ºС
- относительная влажность воздуха до 90%

Схема подключения приведена на Рис. 2

3. КОНСТРУКЦИЯ

Блок выполнен в виде платы с расположенными на ней электронными компонентами. Пластиковый корпус блока предназначен для установки на DIN рейку. На верхней части корпуса имеются графические обозначения органов управления и назначения выводов.

Кроме электронных компонентов на плате располагаются:

- винтовые клеммы для подключения соединительных проводов линий управления, питания и двигателя;

- клеммы IN1 и IN2 для подключения управляющих входных сигналов;

- клемма для подключения контактов сигнала аварийной остановки;

- светодиодный индикатор работы устройства;

- разъем RJ11 (6P6C) для подключения линий RS-485;

- контакты для подключения внутреннего терминального резистора RT;

- встроенная тормозная схема для поглощения генерируемой двигателем энергии (при выбеге, принудительном вращении);

Для аварийной остановки двигателя предусмотрен вход внешнего сигнала «HARD STOP». Для старта и остановки двигателя внешними сигналами, а также для управления направлением вращения двигателя можно использовать внешние входы IN1 и IN2.

Все параметры работы двигателя и управление движением могут осуществляться программно и командами, передаваемыми по RS-485 по протоколу Modbus.

Схема расположения и назначение клемм представлены на рис. 2.

1. Общий провод питания
2. Общий провод питания
3. Питание 12 – 24 В
4. Питание 12 – 24 В
5. Сигнал IN1 (сухой контакт)
6. Сигнал IN2 (сухой контакт)
7. Сигнал аварийной остановки «HARD STOP»
8. Общий провод (сигнальный)
9. Выход +5В питания датчика Холла
10. Линия данных А датчика Холла
11. Линия данных В датчика Холла
12. Общий провод датчика Холла
13. Вывод двигателя А
14. Вывод двигателя А
15. Вывод двигателя В
16. Вывод двигателя В

RS-485 Modbus – разъем RJ11 (6P6C) для подключения линий данных RS-485
RT – контакты для подключения терминального сопротивления (необходимо устанавливать перемычку)

Рис. 2. Схема расположения и назначение клемм и элементов управления

4. ПОДКЛЮЧЕНИЕ И УПРАВЛЯЮЩИЕ СИГНАЛЫ БЛОКА

Запрещается подключать или отсоединять двигатель от блока при включенном питании. При подключении блока следует соблюдать полярность. Несоблюдение полярности, а также превышение напряжения питания приводит к повреждению блока.

ВАЖНО: Из-за больших токов рекомендуется располагать источник питания в непосредственной близости от блока и использовать провода с сечением 3 мм^2 (AWG-8). Источник питания должен обеспечивать ток на 20% больше максимально возможного, потребляемого в процессе эксплуатации.

Рекомендованная длина питающих проводов:

- не более 100 см при токах до 10 А.
- не более 50 см при токах от 10 до 20А.
- не более 25 см при токах от 20 до 40А.

При максимальном токе до 20 А допускается использование по одной линии как питающих, так и фазных клемм. При максимальном токе более 20 А необходимо использовать обе линии как питающих, так и фазных клемм.

Монтаж необходимо осуществлять в следующем порядке:

1. Выполните соединение устройства с двигателем – в соответствии со схемой на рис. 2. Двигатель подключается к выходам блока, обозначенным А и В (клеммы 13 – 16). Сигналы датчика Холла подключаются к клеммам 10 – 11 (HA, HB). Земля датчика Холла подключается к клемме 12 (GND), питание датчиков Холла к клемме 9 (+5V).
2. Подключите внешние цепи управления – в соответствии со схемой подключения на рис. 3:

Рис. 3. Подключение внешних цепей управления

3. Подключите линии интерфейса RS-485 к разъему RJ11 в соответствии с рис. 4.

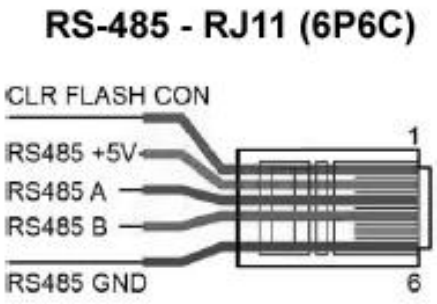

Рис. 4 – назначение контактов разъема RJ11 – RS-485 Modbus.

4. При необходимости установите перемычку на контакты RT для подключения внутреннего терминального сопротивления.
5. Выполните соединение устройства с блоком питания соблюдая полярность. Мощность источника питания должна быть выбрана с запасом (чтобы не было просадок питающего напряжения). Толщина соединительных проводов должна соответствовать потребляемому двигателем току. «+» источника питания подключите – на входы 3 и 4 блока, «-» источника питания подключите – на входы 1 и 2 блока. Демонтаж системы осуществляется в обратном порядке.

5. ПОРЯДОК РАБОТЫ С БЛОКОМ

1. Убедитесь, что питание блока выключено.
2. Выполните подключение блока к двигателю и источнику питания согласно пункту 4.
3. Выберите способ управления: управление командами по Modbus или внешними сигналами (регистр MODE_DEVICE – см. раздел 6.5, рис.6 и рис.7).
4. Выполните подключение сигналов управления «IN1», «IN2» и сигнала аварийной остановки «HARD STOP» согласно пункту 4. Сигнал «HARD STOP» используется для аварийной остановки двигателя. Работа разрешена при замкнутом контакте.
5. Подключите линии интерфейса RS-485 согласно пункту 4.
6. Включите блок питания. Устройство готово к настройке по протоколу Modbus.
7. Задайте необходимые настройки работы привода по протоколу Modbus - ограничение тока в фазе двигателя, направление вращения, настройка работы внешних входов IN1 и IN2, способ управления (см. раздел 6).
8. Передавайте команды управления двигателем через интерфейс RS-485. Таблица регистров блока, их назначение и возможные команды управления двигателем приведены в разделе 6.
9. При управлении приводом при помощи внешних сигналов для старта и остановки двигателя и для управления направлением вращения используйте сигналы IN1 и IN2.

6. УПРАВЛЕНИЕ ПО MODBUS

Для передачи данных через интерфейс RS-485 используется стандартный протокол обмена данными Modbus (ASCII или RTU).

Блок управления имеет следующие заводские настройки:

- ID = 1
- Скорость: 115200 бод
- Проверка четности: чет
- Бит данных: 8
- Стоп бит: 1
- MODBUS RTU

6.1. Регистры управления входными сигналами

Адрес	Тип	Название	Размерность	Описание
1000h	Discrete Input	IN1_bit	1-бит	Состояние входного сигнала IN1
1001h	Discrete Input	IN2_bit	1-бит	Состояние входного сигнала IN2
1002h	Discrete Input	IN_HARD_STOP _bit	1-бит	Состояние входного сигнала HARD_STOP

5007h	Holding Register	MODE_EXT_IN	16-бит	Конфигурация режима работы внешних входов IN1,IN2.
5013h	Holding Register	PRESSED_INPUTS_EXTERN	16-бит	Минимальное время положительного импульса на цифровых входах IN1, IN2 (мс)
5014h	Holding Register	WAITED_INPUTS_EXTERN	16-бит	Минимальное время отрицательной части импульса на цифровых входах IN1, IN2 (мс)

Регистры состояний входных сигналов 1000h..1002h доступны только для чтения.

1000h – IN1_bit – отображает состояние физического входа IN1.

1001h – IN2_bit – отображает состояние физического входа IN2.

1002h – IN_HARD_STOP_bit – отображает состояние физического входа HARD_STOP.

Возможные значения регистров 1000h..1002h:

- 1 – вход замкнут на выход GND
- 0 – вход разомкнут с выходом GND

5007h - MODE_EXT_IN – значение регистра определяет назначение и способ обработки сигналов IN1 и IN2 (см. описание в разделе 6.5).

5013h – PRESSED_INPUTS_EXTERN и 5014h – WAITED_INPUTS_EXTERN – Минимальное время (задается в мс) положительной и отрицательной части импульса на цифровых входах IN1, IN2 - регистры используются для подавления дребезга контактов (см. рис. 5).

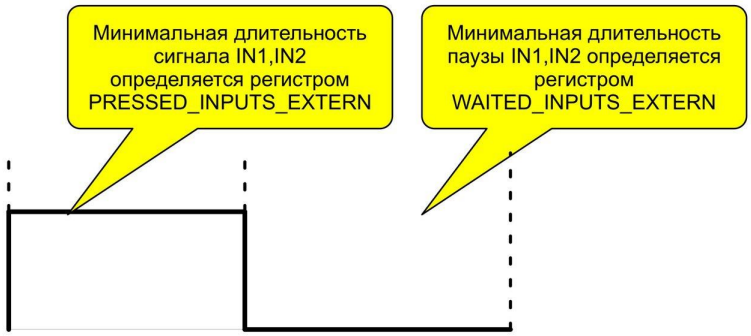

Рис. 5 - Механизм подавления дребезга контактов.

6.2. Регистры управления двигателем

Адрес	Тип	Название	Размерность	Описание
2000h	Coils	START_bit	1-бит	Запуск вращения двигателя с заданным ускорением
2001h	Coils	STOP_bit	1-бит	Остановка вращения двигателя с заданным замедлением
2002h	Coils	HARD_STOP_bit	1-бит	Резкая, экстренная остановка вращения двигателя
2003h	Coils	CLR_POSITION_bit	1-бит	Сброс в нулевое значение регистра текущей позиции POSITION_VALUE.

Данные управляющие сигналы доступны как для чтения, так и для записи. При записи в соответствующий регистр значения 1 активирует ту или иную функцию, после этого регистр сбрасывается в 0.

6.3. Регистры состояния

Адрес	Тип	Название	Размерность	Описание
3000h	Input Register	STATUS	16-бит	Текущее состояние управления двигателем.
3001h	Input Register	CURRENT_VALID	16-бит	Текущее значение потребляемого двигателем тока.
3002h	Input Register	SPEED_VALID	16-бит	Мгновенное значение скорости вращения.
3003h	Input Register	CURRENT_POSITION	32-бит	Текущая позиция. Диапазон значений от -2147483647 до + 2147483648
3005h	Input Register	TEMPERATURE_MCU	16-бит	Температура центрального процессора

3006h	Input Register	TEMPERATURE_ MOSFET	16-бит	Температура силовой части
3007h	Input Register	TEMPERATURE_ BRAKE	16-бит	Температура тормозной схемы
3008h	Input Register	TASK_COUNTER	16-бит	Возвращает случайное значение (для контроля работоспособности канала передачи)
3009h	Input Register	STATUS_USER_ PROGRAM	16-бит	Статус программы пользователя

Данная группа регистров доступна только для чтения и отражает текущее состояние блока управления.

3000h – STATUS - текущее состояние управления двигателем, возможные значения:

- 0 - двигатель остановлен
- 1 - вращение в основном направлении
- 2 - вращение в обратном направлении

3001h - CURRENT_VALID - текущее значение потребляемого двигателем тока. Отражает значение тока потребляемого двигателем от блока питания, значение в мА.

3002 - SPEED_VALID - мгновенное значение скорости вращения. Единицы измерения обороты в минуту.

3003h - CURRENT_POSITION - текущая позиция. Позиционирование производиться в диапазоне значений от -2147483647 до + 2147483648 суммарного количества переключений датчиков Холла, учитываются и передние и задние фронты.

3005h - TEMPERATURE_MCU - температура центрального процессора. Температура вычисляется как TEMPERATURE_MCU / 10 гр/С.

3006h - TEMPERATURE_MOSFET - температура силовой части. Температура в области установки ключей MOSFET. Температура вычисляется как TEMPERATURE_MOSFET / 10 гр/С.

3007h - TEMPERATURE_BRAKE - температура тормозной схемы. Температура в области тормозного резистора. Температура вычисляется как TASK_COUNTER / 10 гр/С.

3008h - TASK_COUNTER - Возвращает псевдослучайное число в диапазоне от 0x0000 до 0xFFFF.

3009h - STATUS_USER_PROGRAM - Статус программы пользователя, возможные значения:

- 1 - программа пользователя была принудительно остановлена по Modbus протоколу
- 2 - программа пользователя запускается по Modbus протоколу (статус активен непродолжительное время до установки статуса 4)
- 3 - программа пользователя запускается при подаче питания (статус активен непродолжительное время до установки статуса 4)
- 4 - выполнение программы пользователя.

6.4. Регистры настройки передачи данных по RS-485 Modbus

Адрес	Тип	Название	Размерность	Описание
5000h	Holding Register	SLAVE_ADDR ESS_MODBUS	16-бит	ID устройства (Адрес устройства). Допустимые значения: 0...247. (0 - широковещательный адрес, без ответного сообщения)
5001h	Holding Register	TYPE_MODBU S	16-бит	Вариант настройки протокола.
5002h	Holding Register	BITRATE_MOD BUS	16-бит	Настройка скорости передачи по протоколу ModBUS.
5003h	Holding Register	TIMEOUT_ BROADCAST_ MODBUS	16-бит	Дополнительная задержка между принятым пакетом и ответным сообщением.

5001h - TYPE_MODBUS – настройки передачи данных, возможные значения:

- 1 – ASCII, 7 бит данных, чет, 1 стоп бит,
- 2 – ASCII, 7 бит данных, нечет, 1 стоп бит
- 3 – RTU, 8 бит данных, чет, 1 стоп бит
- 4 – RTU, 8 бит данных, нечет, 1 стоп бит
- 5 – RTU, 8 бит данных, нет, 2 стоп бит

5002h - BITRATE_MODBUS – скорость передачи данных по Modbus, возможные значения:

- 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 - используется, Если управляющее устройство медленное требуется время для переключения из режима передачи в режим приёма данных.

После изменения настроек передачи данных необходимо сохранить новые значения при помощи регистра FLAG_SAVE_INI (см. раздел 6.5), а затем перезагрузить привод. После перезагрузки подключение по RS-485 будет выполняться по новым настройкам.

6.5. Регистры настройки работы привода

Адрес	Тип	Название	Размерность	Описание
5004h	Holding Register	MODE_DEVICE	16-бит	Режим работы устройства
5005h	Holding Register	MODE_USER_ PROGRAM	16-бит	Команда управления запуском пользовательской программы.
5006h	Holding Register	MODE_ROTATI ON	16-бит	Режим вращения.
5007h	Holding Register	MODE_EXT_IN	16-бит	Конфигурация режима работы внешних входов IN1, IN2.
5008h	Holding Register	POSITION_N	16-бит	Выбор номера позиции, для перемещения. Диапазон значений от 1 до 4
5009h	Holding Register	REF_CURRENT	16-бит	Ограничение потребляемого контроллером тока. Диапазон: 1000 – 20000 мА для BMSD-20Modbus 2000 – 40000 мА для BMSD-40Modbus
500Ah	Holding Register	RATED_SPEED	16-бит	Номинальная скорость вращения двигателя. Диапазон от 1000 до 15000 об/мин.
500Bh	Holding Register	SPEED	16-бит	Заданная скорость вращения. Диапазон от 30 до 15000 об/мин.
500Ch	Holding Register	ACC	16-бит	Заданное ускорение. Диапазон от 10 до 1000
500Dh	Holding Register	DEC	16-бит	Заданное замедление. Диапазон от 10 до 1000
500Eh	Holding Register	DIRECTION	16-бит	Направление вращения. 1 – вращение в основном направлении 2 – вращение в обратном направлении.
500Fh	Holding Register	PULSES-PER_REVOLUTI ON	16-бит	Количество импульсов на оборот, по одному входу датчика Холла. Диапазон от 1 до 12
5010h	Holding Register	USE_HALL	16-бит	Установка режима работы с датчиками Холла или без. 0 – без датчиков Холла 1 – с 1 датчиком Холла 2 – с 2 датчиком Холла
5011h	Holding Register	MODE_COIL	16-бит	Состояние клемм двигателя при остановке: 0 – разомкнуты 1 – замкнуты
5012h	Holding Register	OFFSET_ COMPENSATIO N	16-бит	Поправка торможения двигателя
5013h	Holding Register	PRESSED_INP UTS_EXTERN	16-бит	Минимальное время положительного импульса на цифровых входах IN1, IN2

5014h	Holding Register	WAITED_INPUTS_EXTERN	16-бит	Минимальное время отрицательной части импульса на цифровых входах IN1, IN2
5015h	Holding Register	OFFSET	32-бит	Смещение, на которое необходимо переместиться. Диапазон значений от -2147483647 до + 2147483648
5017h	Holding Register	OFFSET_CONST	32-бит	Приращение - позиция, на которую необходимо переместиться, значение изменяется контроллером в процессе работы.
5019h	Holding Register	TARGET_POSITION	32-бит	Позиция, в которую необходимо переместиться. Диапазон значений от -2147483647 до + 2147483648
501Bh	Holding Register	TARGET_POSITION1	32-бит	Заранее устанавливаемая пользователем позиция №1 Диапазон значений от -2147483647 до + 2147483648
501Dh	Holding Register	TARGET_POSITION2	32-бит	Заранее устанавливаемая пользователем позиция №2 Диапазон значений от -2147483647 до + 2147483648
501Fh	Holding Register	TARGET_POSITION3	32-бит	Заранее устанавливаемая пользователем позиция №3 Диапазон значений от -2147483647 до + 2147483648
5021h	Holding Register	TARGET_POSITION4	32-бит	Заранее устанавливаемая пользователем позиция №4 Диапазон значений от -2147483647 до + 2147483648
5023h	Holding Register	ERROR	16-бит	Регистр ошибок.
5024h	Holding Register	FLAG_SAVE_INI	16-бит	Регистр сохранения пользовательских настроек. (Диапазон 5000h..501Fh). Значение: 0x37FA
5025h	Holding Register	FLAG_SAVE_USER_PROGRAM	16-бит	Регистр записи или чтения пользовательской программы в/из энергонезависимой памяти. Значение для записи: 0x8426 Значение для чтения: 0x9346
5026h	Holding Register	FLAG_RESTART	16-бит	Регистр перезагрузки. Значение: 0x95AF

5004h - **MODE_DEVICE** – режим работы устройства, возможные значения:

- 1 – управление по Modbus
- 2 – управление внешними сигналами

5005h – **MODE_USER_PROGRAM** – команда управления запуском пользовательской программы, возможные значения:

- 1 – остановка работы пользовательской программы
- 2 – старт работы пользовательской программы
- 3 – старт пользовательской программы сразу после подачи питания на блок (требуется предварительное сохранение настроек – см. регистр FLAG_SAVE_INI).

5006h – **MODE_ROTATION** – режим вращения, возможные значения:

- 1 – непрерывное вращение в сторону
- 2 – смещение на величину, заданную регистром OFFSET
- 3 – перемещение в заданное положение. Номер позиции задается регистром POSITION_N (от 1 до 4), координаты позиций задаются регистрами POSITION1, POSITION2, POSITION3, POSITION4 соответственно.

5007h – **MODE_EXT_IN** – конфигурация режима работы внешних входов IN1, IN2 в режиме управления внешними сигналами, возможные значения:

- 1 – Вход IN1 используется как сигнал старта/остановки привода, обрабатывается по

спадающему фронту импульса; вход IN2 используется как сигнал реверса, обрабатывается по спадающему фронту импульса.

- 2 – Вход IN1 используется как сигнал старта/остановки привода, обрабатывается по спадающему фронту импульса; вход IN2 используется как сигнал задания направления, обрабатывается по уровню сигнала.
- 3 – Вход IN1 используется как сигнал старта и остановки привода, обрабатывается по уровню сигнала: наличие сигнала – разрешение вращения двигателя, отсутствие сигнала - остановка; вход IN2 используется как сигнал реверса, обрабатывается по спадающему фронту импульса.
- 4 – Вход IN1 используется как сигнал старта и остановки привода, обрабатывается по уровню сигнала: наличие сигнала – разрешение вращения двигателя, отсутствие сигнала - остановка; вход IN2 используется как сигнал задания направления, обрабатывается по уровню сигнала.
- 5 – вход IN1 используется как сигнал старта и остановки привода в прямом направлении, IN2 – как сигнал старта и остановки привода в обратном направлении; оба сигнала обрабатываются по уровню.

5008h – **POSITION_N** – выбор номера позиции для перемещения (от 1 до 4) – используется в режиме перемещения в заданную координату (MODE_ROTATION = 3) совместно с регистрами POSITION1, POSITION2, POSITION3, POSITION4.

5009h – **REF_CURRENT** – Ограничение потребляемого контроллером тока. (в мА). Диапазон допустимых значений для BMSD-20Modbus – от 1000 мА до 20000 мА, для BMSD-40Modbus – 2000 мА до 40000 мА

500Ah – **RATED_SPEED** - Номинальная скорость вращения двигателя (паспортное значение) от 1000 до 15000 об/мин. Данный регистр используется при USE_HALL[5010h] = 0 (двигатель без датчиков Холла).

500Bh – **SPEED** – заданная скорость вращения при управлении по Modbus (от 30 до 15000 об/мин).

500Ch – **ACC** – заданное ускорение (от 10 до 1000)

500Dh – **DEC** – заданное замедление (от 10 до 1000)

Величины ускорения и замедления в регистрах ACC и DEC являются условными линейными величинами, определяющими интенсивность ускорения/замедления. Значение 0 соответствует 100 об/сек^2, значение 1000 соответствует 5000 об/сек^2.

500Eh – **DIRECTION** – направление вращения, возможные значения:

- 1 – вращение в основном направлении
- 2 – вращение в обратном направлении.

500Fh – **PULSES-PER_REVOLUTION** – количество пар полюсов датчика Холла (от 1 до 12).

5010h – **USE_HALL** – установка режима работы с датчиками Холла или без них, возможные значения:

- 0 - без датчиков Холла (Скорость двигателя Vдвиг. = (SPEED[500Bh] * Vпит.)/15000),
- 1 - один датчик Холла, контроллер выполняет поддержание установленной скорости вращения,
- 2 - два датчика Холла, контроллер выполняет управление скоростью вращения, доступны механизмы позиционирования

5011h – MODE_COIL – состояние клемм двигателя при остановке, возможные значения:

- 0 - При остановке клеммы двигателя разомкнуты;
- 1 - При остановке клеммы двигателя замкнуты.

5012h - **OFFSET_COMPENSATION** – Экспериментально вычисляемая поправка торможения двигателя, при текущих настройках ACC, DEC, SPEED. Если остановка происходит за расчётной точкой, то значение компенсации равно погрешности позиционирования с отрицательным знаком, если мотор останавливается перед точкой останова, то значение компенсации положительное. Например, ошибка выхода в заданное положение 15 инкрементов, значит OFFSET_COMPENSATION = -15, и данная поправка действительна только для текущих настроек ускорения, замедления, скорости.

5013h и 5014h – **PRESSED_INPUTS_EXTERN** и **WAITED_INPUTS_EXTERN** – Минимальное время в мс положительной и отрицательной части импульса на цифровых входах IN1, IN2 – используется для подавления дребезга контактов (см. рис. 5).

5015h – **OFFSET** – смещение, на которое необходимо переместиться, используется при MODE_ROTATION = 2, допустимые значения от -2147483647 до + 2147483648.

Перед началом перемещения необходимо задать требуемую величину смещения в регистре OFFSET, который обрабатывается контроллером как счетчик оставшегося перемещения. В процессе отработки заданного перемещения, значение OFFSET уменьшается.

5017h – **OFFSET_CONST** – Приращение - позиция, на которую необходимо переместиться, значение изменяется контроллером в процессе работы.

5019h – **TARGET_POSITION** – Позиция, в которую необходимо переместиться, допустимые значения от -2147483647 до + 2147483648. В режиме перемещения в заданную координату, перед перемещением в этот регистр копируется координата из POSITION1-POSITION4, при этом они сами значений не меняют.

501Bh - 5021h – **TARGET_POSITION1..4** – Заранее устанавливаемая пользователем позиция №1..4, используется при MODE_ROTATION = 3, номер позиции определяется регистром POSITION_N, допустимые значения от -2147483647 до + 2147483648.

5023h – **ERROR** – ошибки, возникающие при работе блока - каждый бит регистра сигнализирует о конкретной ошибке:

- 0 бит – выход за диапазон питающего напряжения;
- 1 бит – короткое замыкание обмоток двигателя;
- 2 бит – перегрев тормозной схемы;
- 3 бит – перегрев силовой части;
- 4 бит – ошибка подключения датчиков Холла;
- 5 бит – экстренная остановка;
- 6 бит – перегрев процессора;
- 7 бит – тестовая управляющая программа;
- 8 бит – ошибка выполнения пользовательской программы;
- 9 бит – ошибка чтения или записи настроек;
- 10 бит - ошибка в работе выходных транзисторных ключей;
- 11 бит - превышено ограничение потребляемого тока;
- 12 бит - предупреждение о невозможности расчёта точки останова;
- 13 бит - предупреждение о попытке записи в регистр значения, выходящего за допустимый диапазон;
- 14 бит - предупреждение об ошибке чётности в принятом кадре Modbus;
- 15 бит - предупреждение об использовании режима позиционирования с количеством датчиков меньше двух.

5024h – **FLAG_SAVE_INI** – регистр сохранения пользовательских настроек – при записи значения 0x37FA в данный регистр будут сохранены в энергонезависимую память настройки, определяемые регистрами 5000h..501Fh.

5025h – **FLAG_SAVE_USER_PROGRAM** – запись значения 0x8426 в данный регистр запускает процедуру сохранения пользовательской программы из временного буфера в энергонезависимую память блока. Запись значения 0x9346 запускает процедуру чтения пользовательской программы из энергонезависимой памяти блока во временный буфер (см. раздел 6.6.).

5026h – **FLAG_RESTART** – запись значения 0x95AF в данный регистр запускает перезагрузку блока.

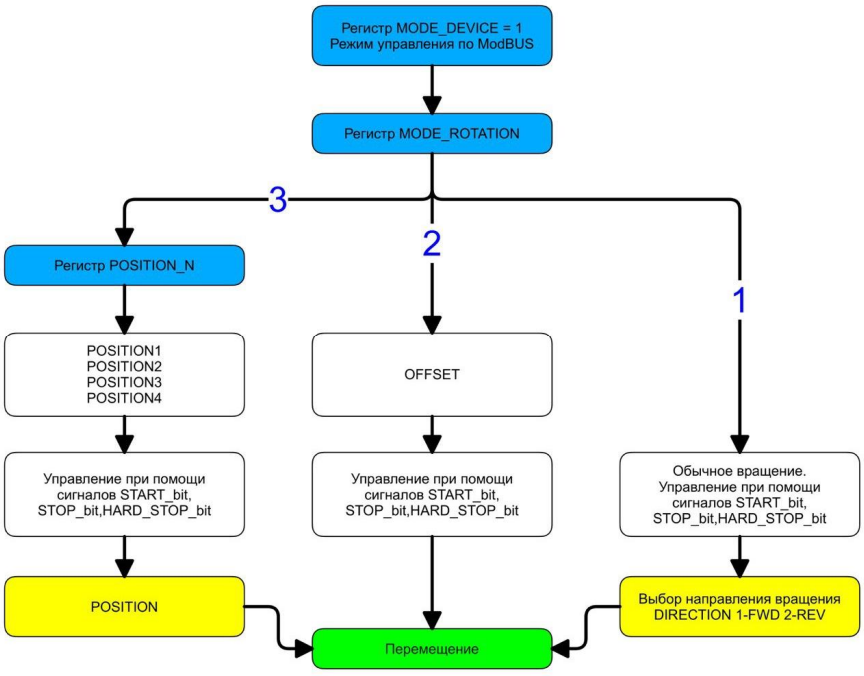

Рис.6. Блок-схема выбора режима работы при управлении приводом по Modbus

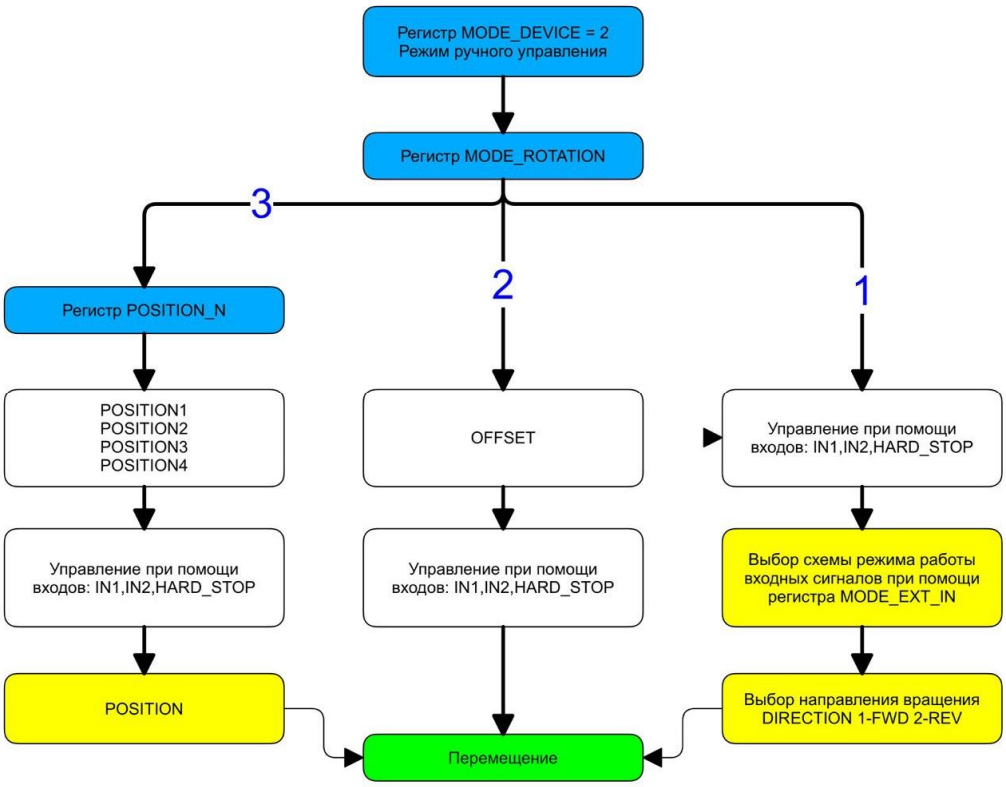

Рис.7. Блок-схема выбора режима работы при управлении приводом внешними сигналами

6.6. Запись и чтение пользовательской программы

Для чтения и записи пользовательской программы используется временный буфер оперативной памяти 1024 команд. Для сохранения программы из временного буфера в энергонезависимую память и для чтения программы из памяти блока во временный буфер используется специальный регистр FLAG_SAVE_INI.

Адрес	Тип	Название	Размерность	Описание
5025h	Holding Register	FLAG_SAVE_USER_PROGRAM	16-бит	Регистр записи или чтения пользовательской программы в/из энергонезависимой памяти. Значение для записи: 0x8426 Значение для чтения: 0x9346
6000h	Holding Register	WRITE_CMD	16-бит	Регистр адреса. При записи в данный регистр значения адреса происходит перенос команды из поля CMD_W в указанный адрес временного буфера пользовательской программы.
6001h	Holding Register	CMD_W	32-бит	Пользовательская команда
6003h	Holding Register	READ_CMD	16-бит	Регистр адреса. При записи в данный регистр значения адреса происходит перенос команды из указанного адреса временного буфера пользовательской программы в поле CMD_R.
6004h	Holding Register	CMD_R	32-бит	Пользовательская команда

Составление и запись пользовательской программы в память контроллера

Каждая инструкция пользовательской программы состоит из двух слов памяти (32 бита) – команда (16 бит) и данные команды (16 бит). При составлении пользовательской программы инструкции сначала заносятся во временный буфер контроллера. Для записи во временный буфер необходимо записать инструкцию в регистр CMD_W, а затем записать в регистр WRITE_CMD адрес этой инструкции во внутреннем буфере. При записи адреса в регистр WRITE_CMD инструкция переносится из регистра CMD_W во внутренний буфер. После составления пользовательской программы во временном буфере необходимо записать значение 0x8426 в регистр FLAG_SAVE_USER_PROGRAM – программа будет перенесена из временного буфера во FLASH память контроллера.

Чтение пользовательской программы из памяти контроллера

Для чтения программы из FLASH памяти необходимо сначала перенести ее во временный буфер контроллера. Для этого необходимо записать значение 0x9346 в регистр FLAG_SAVE_USER_PROGRAM – программа будет перенесена из FLASH памяти контроллера во временный буфер. Затем для чтения инструкции из временного буфера необходимо записать адрес инструкции в регистр READ_CMD – при записи адреса в данный регистр инструкция будет скопирована из временного буфера в регистр CMD_R.

Схема процедур чтения и записи пользовательской программы условно представлена на рис. 8.

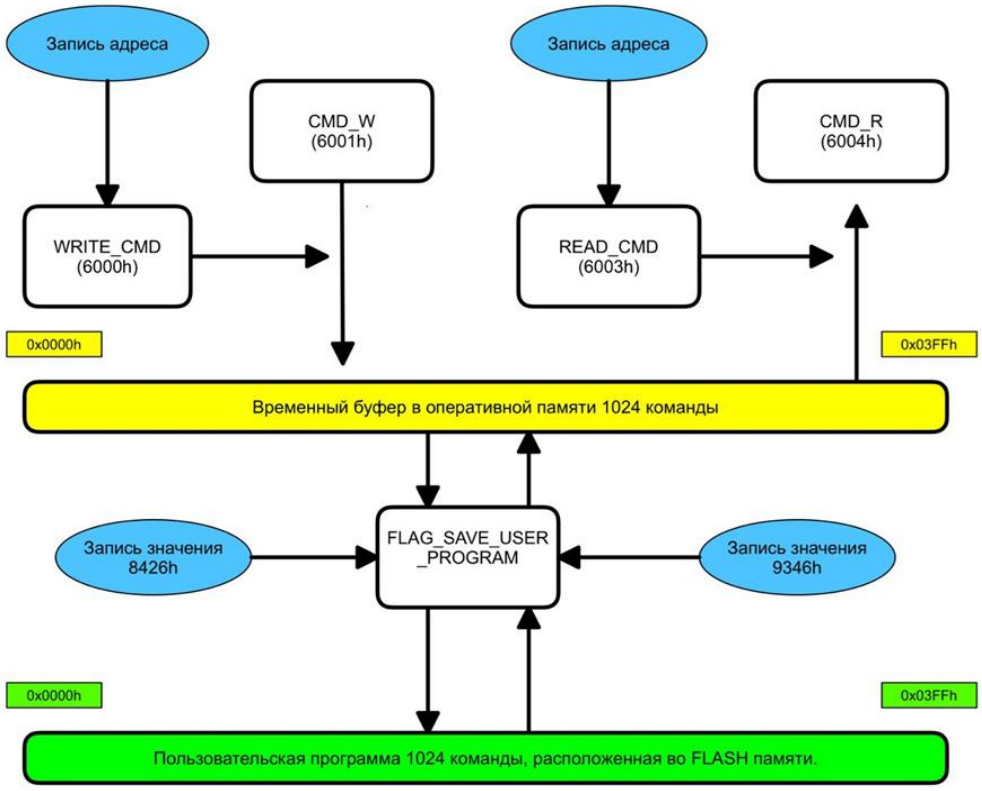

Рис.8. Блок-схема процедур чтения и записи пользовательской программы.

6.7. Системные регистры

Адрес	Тип	Название	Размерность	Описание
7000h	Holding Register	AX_REG	16-бит	Регистр хранения данных для пользовательской программы
7001h	Holding Register	BX_REG	16-бит	Регистр хранения данных для пользовательской программы
7002h	Holding Register	CX_REG	16-бит	Регистр хранения данных для пользовательской программы
7003h	Holding Register	DX_REG	16-бит	Регистр хранения данных для пользовательской программы
7004h	Holding Register	EX_REG	16-бит	Регистр хранения данных для пользовательской программы
7005h	Holding Register	FX_REG	16-бит	Регистр хранения данных для пользовательской программы
7006h	Holding Register	PC_REG	16-бит	Регистр-указатель на текущую исполняемую пользовательскую команду.
7007h	Holding Register	GX_REG	16-бит	Регистр хранения данных для пользовательской программы
7008h	Holding Register	HX_REG	16-бит	Регистр хранения данных для пользовательской программы
7009h	Holding Register	IX_REG	16-бит	Регистр хранения данных для пользовательской программы
700Ah	Holding Register	JX_REG	16-бит	Регистр хранения данных для пользовательской программы

Системные регистры AX_REG..FX_REG, GX_REG..JX_REG (диапазон значений 0...65535) предназначены для временного хранения данных во время выполнения пользовательской программы. PC_REG – указатель на текущую выполняемую инструкцию пользовательской программы. Более подробное описание в разделе 6.9.

6.8. Регистры идентификации

Адрес	Тип	Название	Размерность	Описание
8001h	Input Register	HW_MAJOR	16-бит	Тип драйвера
8002h	Input Register	HW_MINOR	16-бит	Версия аппаратной части
8003h	Input Register	FW_MAJOR	16-бит	Идентификатор программного обеспечения
8004h	Input Register	FW_MINOR	16-бит	Версия программного обеспечения

Регистры необходимые для определения по сети функционального назначения блока управления, его характеристики и версии программного обеспечения.

Для блока BMSD-20Modbus значения:

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

Для блока BMSD-40Modbus значения:

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

6.9. Инструкции пользовательской программы

Структура 32-битных инструкций пользовательской программы представлена на схеме на рис. 9. Данная инструкция занимает одну командную строку в памяти пользовательской программы.

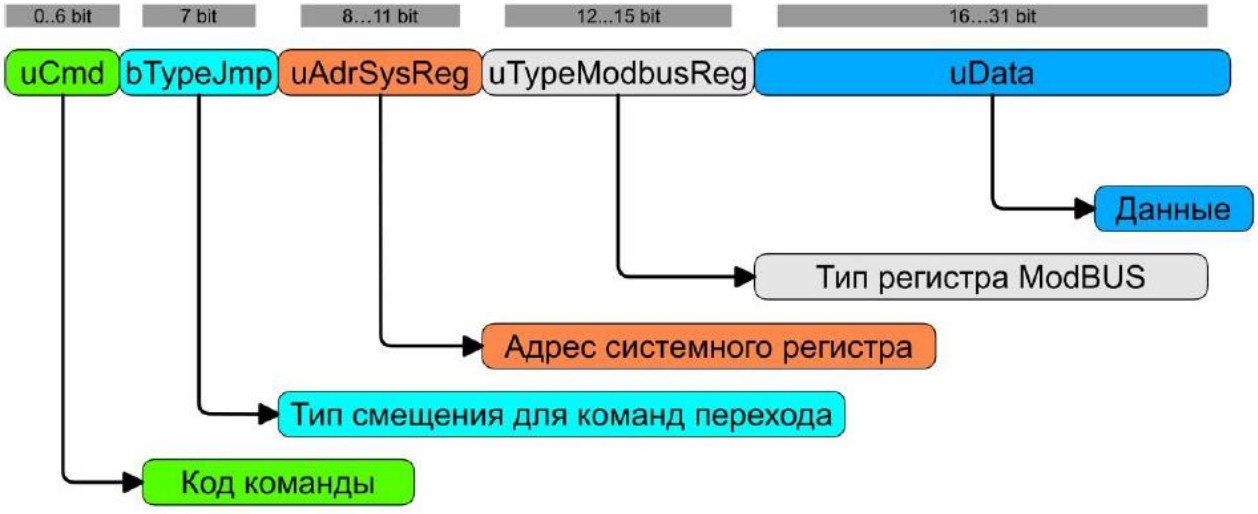

Рис.9 – структура инструкции пользовательской программы

Инструкция пользовательской программы имеет размер 32 бита и содержит следующие поля:

uCmd – 7 бит – код команды

bTypeJmp – 1 бит - тип смещения для команд перехода:

- 0 – абсолютное значение
- 1 – относительное значение

uAdrSysReg – 4 бита - адрес системного регистра AX_REG…FX_REG с номерами 0…5

uTypeModbusReg – 4 бита – тип регистра Modbus

- 0 – Discrete Inputs
- 1 – Coils
- 2 – Inputs
- 3 – Holding registers

uData – 16 бит – данные.

64-битные инструкции состроят из двух командных строк по 32-бита. Первая командная строка данной инструкции представляет собой команду аналогичную 32-битной инструкции. Вторая командная строка содержит 16 или 32-бита данных **uCMD_DATA**.

Команды с префиксом D, например, DMOV оперируют 32-битными данными, расположенными в двух последовательных 16-битных регистрах. В поле адреса данной команды указывается младший регистр. При чтении значения команда автоматически читает два последовательных 16-битных регистра начиная с указанного адреса и формирует единую 32-битную величину. При записи значения команда разбивает 32-битное число на две 16-битные части и записывает их в два последовательных регистра начиная с указанного адреса.

В следующей таблице приведены команды с вариантами заполнения полей. Если поле не указанно, значит в данной команде оно не используется.

uCmd		-	-	-
0x00	CMD_STOP_PROGRAM	-	-	-
0x0D	CMD_FULL_ STOP_PROGRAM			
uCmd		**bTypeJmp**	**uData**	-
0x05	CMD_JMP	0 – абсолютное значение 1 – относительное значение	Адрес перехода 0..1024 или смещение +- 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	Время паузы в мс 0..65535	-	-
0x0A	CMD_CALL	Адрес подпрограммы 0..1024		
0x0B	CMD_RETURN	-		
0x0C	CMD_FOR	Длина цикла и количество циклов		
uCmd		**uAdrSysReg**	-	-
0x15	CMD_NOT_SYSREG	Адрес системного регистра (0..9)	-	-
0x17	CMD_DNOT_SYSREG			
uCmd		**uTypeModbusReg**	**uData**	-
0x16	CMD_NOT_MODBUS	1 – Coils 3 – Holding registers	Адрес регистра ModBUS 0..65535	-
0x18	CMD_DNOT_MODBUS			
uCmd		**uAdrSysReg**	**uData**	-
0x01	CMD_MOV_SYSREG_CONST	Адрес системного регистра (0..9)	Константа 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	Адрес системного регистра (0..9)	0 – Discrete Inputs 1 – Coils 2 – Inputs 3 – Holding registers	Адрес регистра 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 – Coils 3 – Holding registers	Адрес регистра ModBUS 0..65535	Константа 0..65535 (в отдельной командной строке)
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	Адрес системного регистра (0..9)	0 – Discrete Inputs 1 – Coils 2 – Inputs 3 – Holding registers	Адрес регистра 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	Адрес системного регистра (0..9)	Адрес системного регистра (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	Адрес системного регистра (0..9)	Константа 0…4294967295 (в отдельной командной строке)	-
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	Адрес системного регистра (0..9)	0 – Discrete Inputs 1 – Coils 2 – Inputs 3 – Holding	Адрес регистра ModBUS 0..65535
0x46	CMD_DADD_SYSREG_MODBUS			
0x47	CMD_DSUB_SYSREG_MODBUS			
0x48	CMD_DDIV_SYSREG_MODBUS			

0x49	CMD_DMUL_SYSREG_MODBUS		registers	
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 – Coils 3 – Holding registers	Адрес регистра ModBUS 0..65535	Константа 0…4294967295 (в отдельной командной строке)
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	Адрес системного регистра (0..9)	0 – Discrete Inputs 1 – Coils 2 – Inputs 3 – Holding registers	Адрес регистра 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	Адрес системного регистра (0..9)	Адрес системного регистра (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	Адрес системного регистра (0..9)	0 – сдвиг влево 1 – сдвиг вправо	Величина сдвига
0x45	CMD_DSH_SYSREG_CONST			
uCmd		**uTypeModbusReg**	**uData**	**bTypeJmp**
0x2F	CMD_SH_MODBUS_CONST	1 – Coils 3 – Holding registers	Адрес регистра ModBUS 0..65535	0 – сдвиг влево 1 – сдвиг вправо
0x54	CMD_DSH_MODBUS_CONST			

CMD_STOP_PROGRAM – (код команды 0x00) – остановка выполнения пользовательской программы, без выхода из режима выполнения пользовательской программы. При завершении выполнения программы все регистры и состояние двигателя остаются такими, как были до выполнения команды (двигатель продолжает вращение, если вращался перед выполнением команды). Перед следующим запуском программы необходимо отправить команду CMD_FULL_STOP_PROGRAM.

CMD_FULL_STOP_PROGRAM – (код команды 0x0D) – остановка выполнения пользовательской программы и выход из режима работы по программе. При завершении выполнения программы все регистры и состояние двигателя возвращаются к исходным значениям, двигатель останавливается.

CMD_MOV_SYSREG_CONST – (код команды 0x01) – запись в системный регистр с адресом uAdrSysReg, значения из поля данных uData

CMD_MOV_MODBUS_SYSREG – (код команды 0x02) – запись содержимого из системного регистра uAdrSysReg в пространство регистров ModBUS определяемый полем TypeModbusReg и его адресом в поле uData.

CMD_MOV_SYSREG_MODBUS – (код команды 0x03) – чтение содержимого из пространства регистров ModBUS определяемого полем TypeModbusReg и его адресом в поле uData в один из системных регистров uAdrSysReg

CMD_MOV_MODBUS_CONST – (код команды 0x0E) – запись константы содержащейся в следующей командной строке uCMD_DATA в пространство регистров ModBUS определяемый полем TypeModbusReg и его адресом в поле uData.

CMD_MOV_SYSREG_SYSREG – (код команды 0x0F) – запись содержимого из системного регистра uAdrSysReg в другой системный регистр с адресом uData.

CMD_DELAY – (код команды 0x04) – пауза, мс.

CMD_JMP – (код команды 0x05) – переход по заданному в поле uData адресу.

CMD_JMP_AX_PARI_BX – (код команды 0x06) – переход по заданному в поле uData адресу, если значение системного регистра AX_REG равно значению BX_REG

CMD_JMP_AX_NOPARI_BX – (код команды 0x07) – переход по заданному в поле uData адресу, если значение системного регистра AX_REG не равно значению BX_REG

CMD_JMP_AX_MORE_BX – (код команды 0x08) – переход по заданному в поле uData адресу, если значение системного регистра AX_REG больше значения BX_REG

CMD_JMP_AX_LESS_BX – (код команды 0x09) – переход по заданному в поле uData адресу, если значение системного регистра AX_REG меньше значения BX_REG

CMD_CALL – (код команды 0x0A) – вызова подпрограммы, начинающейся с адрес, заданного в поле uData.

CMD_RETURN – (код команды 0x0B) – возврата из подпрограммы.

CMD_FOR – (код команды 0x0C) – циклическое выполнение последовательности команд. Старший байт поля uData содержит количество команд, расположенных после команды CMD_FOR, которые будут повторяться в цикле. Младший байт поля uData содержит количество повторений. Например: uData = 0x1705 - 0x17 = 23 команды исполняемых в цикле, 0x05 = 5 – количество повторений.

CMD_DJMP_GX_PARI_IX – (код команды 0x63) – переход по заданному в поле uData адресу, если содержимое пары системных регистров GX_REG и HX_REG равно содержимому IX_REG и JX_REG.

CMD_DJMP_GX_NOPARI_IX – (код команды 0x64) – переход по заданному в поле uData адресу, если содержимое пары системных регистров GX_REG и HX_REG не равно содержимому IX_REG и JX_REG.

CMD_DJMP_GX_MORE_IX – (код команды 0x65) – переход по заданному в поле uData адресу, если содержимое пары системных регистров GX_REG и HX_REG больше содержимого IX_REG и JX_REG.

CMD_DJMP_GX_LESS_IX – (код команды 0x66) – переход по заданному в поле uData адресу, если содержимое пары системных регистров GX_REG и HX_REG меньше содержимого IX_REG и JX_REG.

Группа команд, производящая математическую операцию между содержимым системного регистра по адресу uAdrSysReg и константой, находящейся в поле данных uData. Результат помещается в указанный системный регистр.

CMD_ADD_SYSREG_CONST – (код команды 0x19) – сложение.

CMD_SUB_SYSREG_CONST – (код команды 0x1A) – вычитание.

CMD_DIV_SYSREG_CONST – (код команды 0x1B) – деление (остаток отбрасывается).

CMD_MUL_SYSREG_CONST – (код команды 0x1C) – умножение.

CMD_AND_SYSREG_CONST – (код команды 0x1D) – поразрядное логическое И.

CMD_OR_SYSREG_CONST – (код команды 0x1E) – поразрядное логическое ИЛИ.

CMD_XOR_SYSREG_CONST – (код команды 0x1F) – поразрядное логическое исключающее ИЛИ.

Группа команд, производящая математическую операцию между содержимым системного регистра по адресу uAdrSysReg и значением из регистра ModBUS определяемого полем TypeModbusReg по адресу в поле uData. Результат помещается в указанный системный регистр.

CMD_ADD_SYSREG_MODBUS – (код команды 0x21) – сложение.

CMD_SUB_SYSREG_MODBUS – (код команды 0x22) – вычитание.

CMD_DIV_SYSREG_MODBUS – (код команды 0x23) – деление (остаток отбрасывается).

CMD_MUL_SYSREG_MODBUS – (код команды 0x24) – умножение.

CMD_AND_SYSREG_MODBUS – (код команды 0x25) – поразрядное логическое И.

CMD_OR_SYSREG_MODBUS – (код команды 0x26) – поразрядное логическое ИЛИ.

CMD_XOR_SYSREG_MODBUS – (код команды 0x27) – поразрядное логическое исключающее ИЛИ.

Группа команд, производящая математическую операцию между содержимым регистра ModBUS

определяемого полем TypeModbusReg по адресу в поле uData и константой, находящейся в следующей командной строке uCMD_DATA. Результат помещается в указанный регистр ModBUS.

CMD_ADD_MODBUS_CONST – (код команды 0x28) – сложение.

CMD_SUB_MODBUS_CONST – (код команды 0x29) – вычитание.

CMD_DIV_MODBUS_CONST – (код команды 0x2A) – деление (остаток отбрасывается).

CMD_MUL_MODBUS_CONST – (код команды 0x2B) – умножение.

CMD_AND_MODBUS_CONST – (код команды 0x2C) – поразрядное логическое И.

CMD_OR_MODBUS_CONST – (код команды 0x2D) – поразрядное логическое ИЛИ.

CMD_XOR_MODBUS_CONST – (код команды 0x2E) – поразрядное логическое исключающее ИЛИ.

Группа команд, производящая математическую операцию между содержимым регистра ModBUS определяемого полем TypeModbusReg по адресу в поле uData и содержимым системного регистра по адресу uAdrSysReg. Результат помещается в указанный регистр ModBUS.

CMD_ADD_MODBUS_SYSREG – (код команды 0x30) – сложение.

CMD_SUB_MODBUS_SYSREG – (код команды 0x31) – вычитание.

CMD_DIV_MODBUS_SYSREG – (код команды 0x32) – деление (остаток отбрасывается).

CMD_MUL_MODBUS_SYSREG – (код команды 0x33) – умножение.

CMD_AND_MODBUS_SYSREG – (код команды 0x34) – поразрядное логическое И.

CMD_OR_MODBUS_SYSREG – (код команды 0x35) – поразрядное логическое ИЛИ.

CMD_XOR_MODBUS_SYSREG – (код команды 0x36) – поразрядное логическое исключающее ИЛИ.

Группа команд, производящая математическую операцию между содержимым системного регистра по адресу uAdrSysReg и содержимым системного регистра по адресу uCMD_DATA расположенному в следующей командной строке. Результат помещается в системный регистр по адресу uAdrSysReg.

CMD_ADD_SYSREG_SYSREG – (код команды 0x37) – сложение.

CMD_SUB_SYSREG_SYSREG – (код команды 0x38) – вычитание.

CMD_DIV_SYSREG_SYSREG – (код команды 0x39) – деление (остаток отбрасывается).

CMD_MUL_SYSREG_SYSREG – (код команды 0x3A) – умножение.

CMD_AND_SYSREG_SYSREG – (код команды 0x3B) – поразрядное логическое И.

CMD_OR_SYSREG_SYSREG – (код команды 0x3C) – поразрядное логическое ИЛИ.

CMD_XOR_SYSREG_SYSREG – (код команды 0x3D) – поразрядное логическое исключающее ИЛИ.

Группы команд, оперирующие 32-разрядными данными расположенными в двух последовательных 16-разрядных регистрах. При обращении к регистру в команде указывается адрес младшего регистра. Адрес старшего регистра получается путем увеличения адреса младшего регистра не единицу.

Группа команд, производящая математическую операцию между содержимым двух последовательных системных регистров по адресу uAdrSysReg и константой, находящейся в следующей командной строке uCMD_DATA. Результат помещается в пару последовательных системных регистров по адресу uAdrSysReg.

CMD_DADD_SYSREG_CONST – (код команды 0x3E) – сложение.

CMD_DSUB_SYSREG_CONST – (код команды 0x3F) – вычитание.

CMD_DDIV_SYSREG_CONST – (код команды 0x40) – деление (остаток отбрасывается).

CMD_DMUL_SYSREG_CONST – (код команды 0x41) – умножение.

CMD_DAND_SYSREG_CONST – (код команды 0x42) – поразрядное логическое И.

CMD_DOR_SYSREG_CONST – (код команды 0x43) – поразрядное логическое ИЛИ.

CMD_DXOR_SYSREG_CONST – (код команды 0x44) – поразрядное логическое исключающее ИЛИ.

Группа команд, производящая математическую операцию между содержимым двух последовательных системных регистров по адресу uAdrSysReg и значением из двух последовательных регистров ModBUS определяемых полем TypeModbusReg по адресу в поле uData. Результат помещается в пару последовательных системных регистров по адресу uAdrSysReg.

CMD_DADD_SYSREG_MODBUS – (код команды 0x46) – сложение.

CMD_DSUB_SYSREG_MODBUS – (код команды 0x47) – вычитание.

CMD_DDIV_SYSREG_MODBUS – (код команды 0x48) – деление (остаток отбрасывается).

CMD_DMUL_SYSREG_MODBUS – (код команды 0x49) – умножение.

CMD_DAND_SYSREG_MODBUS – (код команды 0x4A) – поразрядное логическое И.

CMD_DOR_SYSREG_MODBUS – (код команды 0x4B) – поразрядное логическое ИЛИ.

CMD_DXOR_SYSREG_MODBUS – (код команды 0x4C) – поразрядное логическое исключающее ИЛИ.

Группа команд, производящая математическую операцию между содержимым двух последовательных регистров ModBUS определяемых полем TypeModbusReg по адресу в поле uData и константой, находящейся в следующей командной строке uCMD_DATA. Результат помещается в последовательную пару указанных регистров ModBUS.

CMD_DADD_MODBUS_CONST – (код команды 0x4D) – сложение.

CMD_DSUB_MODBUS_CONST – (код команды 0x4E) – вычитание.

CMD_DDIV_MODBUS_CONST – (код команды 0x4F) – деление (остаток отбрасывается).

CMD_DMUL_MODBUS_CONST – (код команды 0x50) – умножение.

CMD_DAND_MODBUS_CONST – (код команды 0x51) – поразрядное логическое И.

CMD_DOR_MODBUS_CONST – (код команды 0x52) – поразрядное логическое ИЛИ.

CMD_DXOR_MODBUS_CONST – (код команды 0x53) – поразрядное логическое исключающее ИЛИ.

Группа команд, производящая математическую операцию между содержимым двух последовательных регистров ModBUS определяемых полем TypeModbusReg по адресу в поле uData и содержимым последовательной пары системных регистров по адресу uAdrSysReg. Результат помещается в последовательную пару указанных регистров ModBUS.

CMD_DADD_MODBUS_SYSREG – (код команды 0x55) – сложение.

CMD_DSUB_MODBUS_SYSREG – (код команды 0x56) – вычитание.

CMD_DDIV_MODBUS_SYSREG – (код команды 0x57) – деление (остаток отбрасывается).

CMD_DMUL_MODBUS_SYSREG – (код команды 0x58) – умножение.

CMD_DAND_MODBUS_SYSREG – (код команды 0x59) – поразрядное логическое И.

CMD_DOR_MODBUS_SYSREG – (код команды 0x5A) – поразрядное логическое ИЛИ.

CMD_DXOR_MODBUS_SYSREG – (код команды 0x5B) – поразрядное логическое исключающее ИЛИ.

Группа команд, производящая математическую операцию между содержимым двух последовательных системных регистров по адресу uAdrSysReg и содержимым двух последовательных системных регистров по адресу uCMD_DATA расположенному в следующей командной строке. Результат помещается в последовательную пару системных регистров по адресу uAdrSysReg.

CMD_DADD_SYSREG_SYSREG – (код команды 0x5C) – сложение.

CMD_DSUB_SYSREG_SYSREG – (код команды 0x5D) – вычитание.

CMD_DDIV_SYSREG_SYSREG – (код команды 0x5E) – деление (остаток отбрасывается).

CMD_DMUL_SYSREG_SYSREG – (код команды 0x5F) – умножение.

CMD_DAND_SYSREG_SYSREG – (код команды 0x60) – поразрядное логическое И.

CMD_DOR_SYSREG_SYSREG – (код команды 0x61) – поразрядное логическое ИЛИ.

CMD_DXOR_SYSREG_SYSREG – (код команды 0x62) – поразрядное логическое исключающее ИЛИ.

Группа команд, осуществляющая операцию сдвига данных.

CMD_SH_SYSREG_CONST - (код команды 0x20) – команда осуществляет сдвиг содержимого системного регистра по адресу uAdrSysReg в направлении заданном в поле bTypeJmp (0 – сдвиг влево, 1 – сдвиг вправо). Величина сдвига указана в поле **uData.**

CMD_SH_MODBUS_CONST - (код команды 0x2F) – команда осуществляет сдвиг содержимого регистра ModBUS определяемого полем TypeModbusReg по адресу в поле uData в направлении заданном в поле bTypeJmp (0 – сдвиг влево, 1 – сдвиг вправо). Величина сдвига указана в поле **uCMD_DATA.**

CMD_DSH_SYSREG_CONST - (код команды 0x45) – команда осуществляет сдвиг содержимого двух последовательных системных регистров по адресу uAdrSysReg в направлении заданном в поле bTypeJmp (0 – сдвиг влево, 1 – сдвиг вправо). Величина сдвига указана в поле **uData.**

CMD_DSH_MODBUS_CONST - (код команды 0x54) – команда осуществляет сдвиг содержимого двух последовательных регистров ModBUS определяемого полем TypeModbusReg по адресу в поле uData в направлении заданном в поле bTypeJmp (0 – сдвиг влево, 1 – сдвиг вправо). Величина сдвига указана в поле **uCMD_DATA.**

Важно: при остановке выполнения пользовательской программы командой CMD_STOP_PROGRAM состояние двигателя и всех регистров остаются такими, как были заданы в процессе работы программы. По этой причине, если на момент выполнения команды

CMD_STOP_PROGRAM двигатель находился в состоянии движения, он продолжит выполнение последнего задания, т.е. движение будет продолжаться по завершении работы программы. Вращение привода после завершения программы может быть остановлено записью регистра блока по Modbus (Coils 2001h или Coils 2002h). Для предотвращения неконтролируемого вращения перед командой завершения программы можно установить команду остановки двигателя.

7. СБРОС К ЗАВОДСКИМ НАСТРОЙКАМ

В случае необходимости параметры блока можно сбросить к заводским значениям. Для этого необходимо перед включением питания блока замкнуть первый контакт разъема RJ11 - CLR_FLASH_CON (разъем подключения по RS-485, см. рис. 4) с землей GND источника питания. На блоке начнут поочередно загораться красный и зеленый светодиоды. Удерживайте замкнутыми контакт CLR_FLASH_CON и GND в течение 5 с. После этого параметры блок будут сброшены к заводским значениям.

8. КОМПЛЕКТНОСТЬ

Блок управления бесколлекторным двигателем BMSD-20Modbus или BMSD-40Modbus	1 шт.
Паспорт BMSD20.MODBUS.001.ПС	1 шт.

9. ГАРАНТИЙНЫЕ ОБЯЗАТЕЛЬСТВА.

Изготовитель гарантирует безотказную работу блока в течение 12 месяцев со дня продажи, при соблюдении условий эксплуатации.

Адрес предприятия-изготовителя ООО «НПО Электропривод», 195197, Россия, Санкт-Петербург, Полюстровский пр.43, А. Тел./факс (812) 703-09-81, 493-27-26

10. ПЕРЕЧЕНЬ ИЗМЕНЕНИЙ В ПРОГРАММНОМ ОБЕСПЕЧЕНИИ.

Версия 2.0 (10.09.2021)

1. Увеличен набор команд для пользовательской программы контроллера:

- Расширена группа команд копирования данных:
  - Запись константы в регистр Modbus;
  - Копирование данных из одного системного регистра в другой.
- Добавлены следующие группы команд:
  - Математические операции, работающие с 16-битными данными.
  - Логические операции, работающие с 16-битными данными.
  - Операции сдвига, работающие с 16-битными данными.
  - Математические операции, работающие с 32-битными данными.
  - Логические операции, работающие с 32-битными данными.
  - Операции сдвига, работающие с 32-битными данными.
  - Операции сравнения, работающие с 32-битными данными.

  Описание команд приведено в разделе 6.9.

2. Увеличено количество системных регистров с 6 до 10. Описание системных регистров приведено в разделе 6.7.

Заводской номер:

Дата продажи:

Редакция от 14.02.2022