



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

**Блок управления бесколлекторным
двигателем постоянного тока БУВД-20Modbus**

Паспорт
БУВД20.MODBUS.001.ПС

г. Санкт-Петербург

2024

1. НАЗНАЧЕНИЕ УСТРОЙСТВА

Блок управления бесколлекторным двигателем БУВД-20Modbus представляет собой электронное устройство, предназначенное для четырехквadrантного управления бесколлекторными двигателями.

2. ОПИСАНИЕ И ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

Блоки управления позволяют управлять скоростью, ускорением, торможением и направлением вращения двигателя. Блоки также обеспечивают возможность позиционирования по сигналам от встроенных датчиков Холла и удержания положения.

Управление бесколлекторным 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, предусмотрены логические и математические операции с данными.
- 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.
- 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:

- Клеммы для подключения потенциометра или внешнего сигнала 0-5В для аналогового управления скоростью.
- Контакты для подключения внешних сигналов In1 и In2, назначение и способ обработки которых определяется пользователем. В том числе входы могут использоваться для подачи сигналов START/STOP (старт/остановка) и DIR (направление вращения).
- Контакт HARD STOP для подключения аварийного сигнала, обеспечивающего быстрое управляемое торможение двигателя при разрыве аварийного контура. После остановки происходит переход в режим обесточивания двигателя, снимается момент с вала и предотвращается несанкционированный запуск.

Таблица 1. Технические характеристики блока.

Модель	БУВД-20Modbus
Напряжение питания, В	24 - 48
Допустимый диапазон напряжения питания до срабатывания защиты, В	20 - 51
Номинальный ток, А	<20
Диапазон регулирования скорости, $\omega_{\max}/\omega_{\min}$	400
Входное сопротивление входа SPEED, кОм	20
Диапазон входного напряжения входа SPEED, В	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

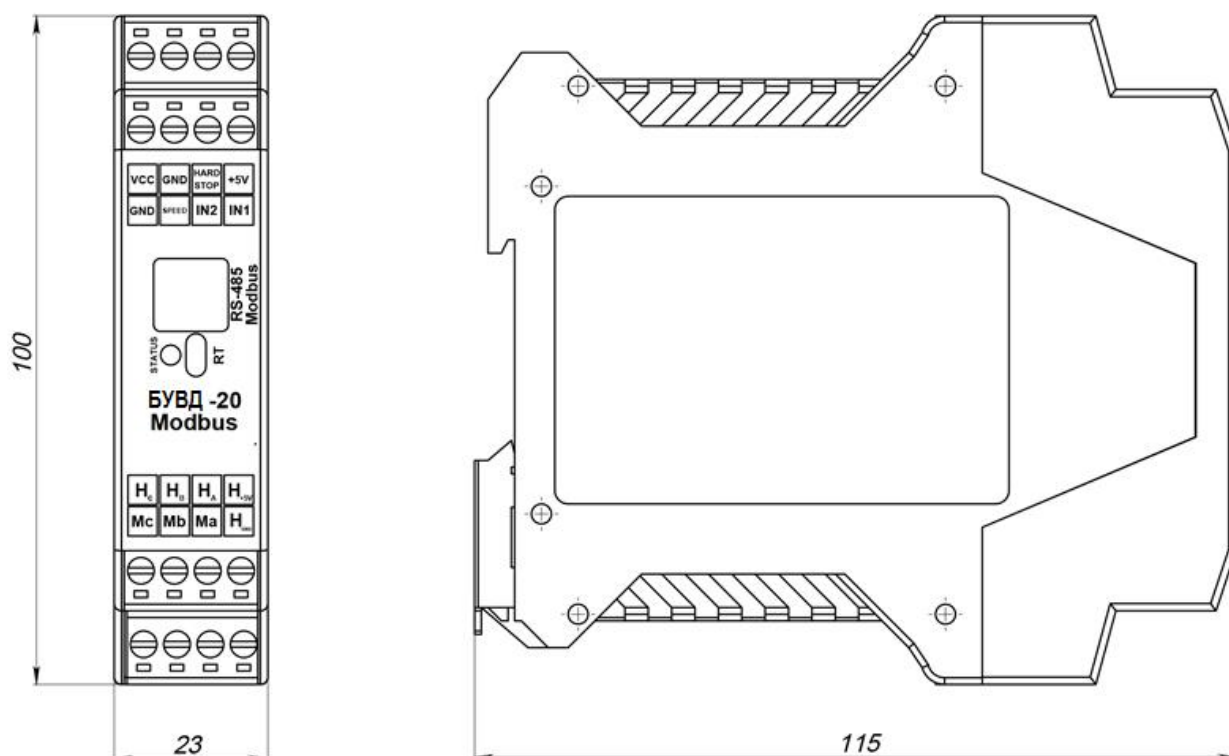


Рис.1. Габаритные и присоединительные размеры блока БУВД-20Modbus

Условия эксплуатации блока:

- температура окружающего воздуха – (0...+40) °С
- относительная влажность воздуха до 90%

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:

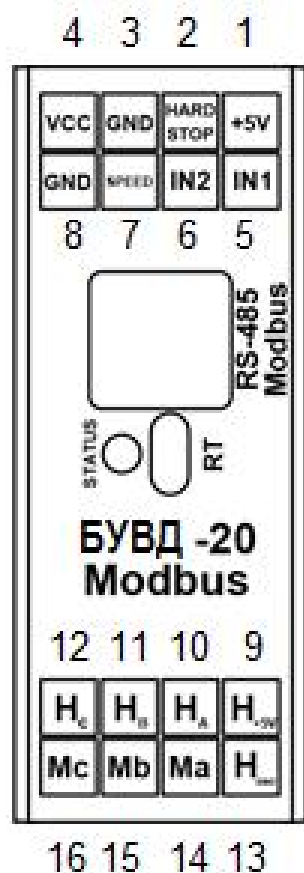
- винтовые клеммы для подключения соединительных проводов линий управления, питания и двигателя;
- клеммы IN1 и IN2 для подключения управляющих входных сигналов;
- клеммы для подключения внешнего потенциометра для регулирования скорости вращения двигателя;
- клемма для подключения контактов сигнала аварийной остановки;

- светодиодный индикатор работы устройства;
- разъем RJ12 (6P6C) для подключения линий RS-485;
- контакты RT, при замыкании которых подключается внутренний терминальный резистор;
- встроенная тормозная схема для поглощения генерируемой двигателем энергии (при выбеге, принудительном вращении);

Для управления скоростью внешним аналоговым сигналом предназначен вход «SPEED». 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. Выход 5 В для внешнего потенциометра
2. Сигнал аварийной остановки «HARD STOP»
3. Общий провод питания
4. Питание 24 – 48 В
5. Сигнал IN1 (сухой контакт)
6. Сигнал IN2 (сухой контакт)
7. Аналоговый вход задания скорости внешним потенциометром
8. Общий провод (сигнальный)
9. Выход питания датчиков Холла
10. Датчик Холла фазы А
11. Датчик Холла фазы В
12. Датчик Холла фазы С
13. Общий провод датчиков Холла
14. Фаза А двигателя
15. Фаза В двигателя
16. Фаза С двигателя

RS-485 Modbus – разъем RJ12 (6P6C) для подключения линий данных RS-485
RT – контакты для подключения терминального резистора (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. Монтаж необходимо осуществлять в следующем порядке:

1. Выполните соединение устройства с двигателем – в соответствии со схемой на рис. 2. Фазы двигателя подключаются к клеммам 14 - 16. Сигналы датчиков Холла двигателя подключаются к клеммам 10 – 12. Земля датчика Холла подключается к клемме 13, питание датчиков Холла к клемме 9.
2. Подключите внешние цепи управления – в соответствии со схемой подключения на

рис. 3:

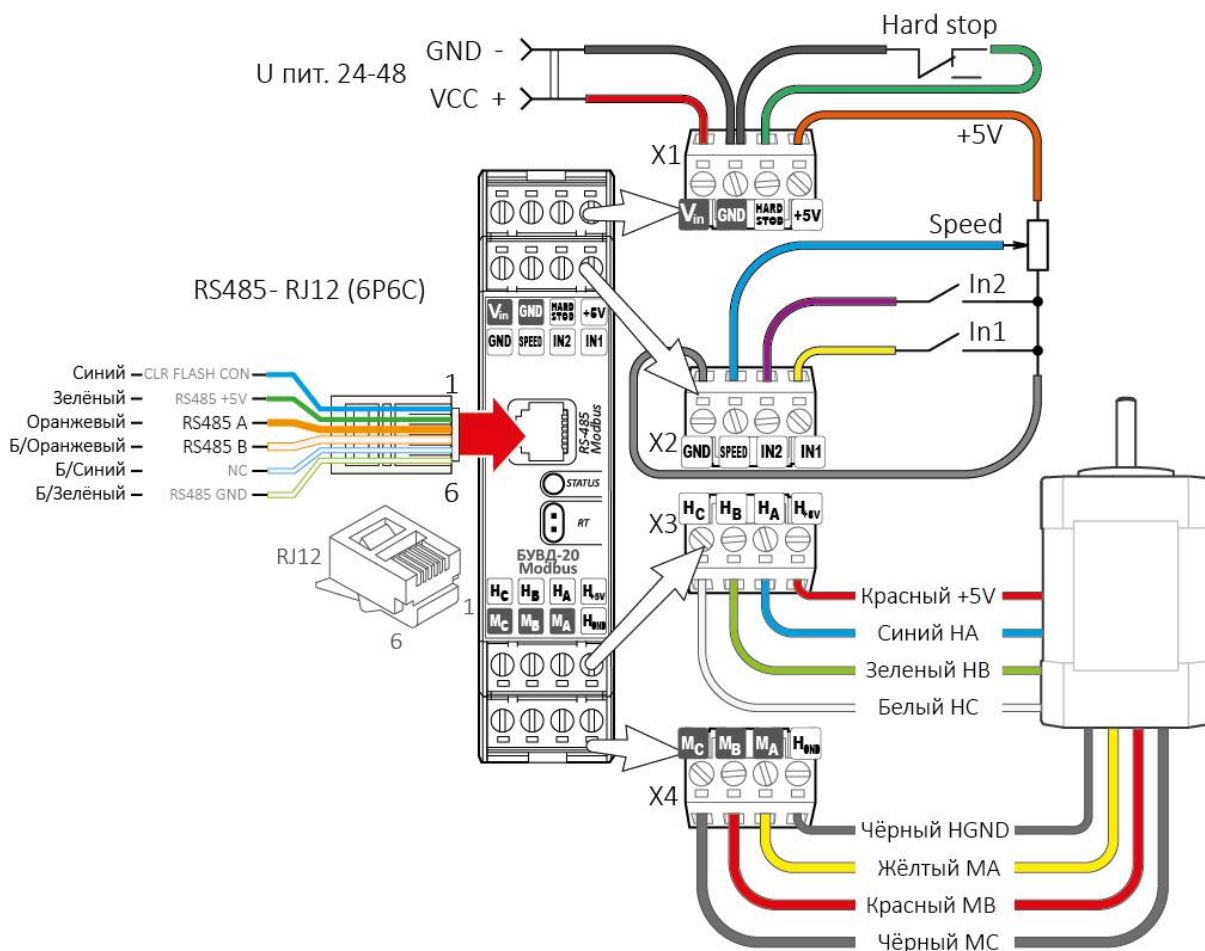


Рис. 3. Подключение питания и внешних цепей управления

3. При необходимости установите перемычку на контакты RT для подключения внутреннего терминального сопротивления.
4. 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 вход 4, «-» источника питания подключите – на вход 3. 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. Для регулировки скорости при помощи внешнего источника сигнала подключите внешний резистор к клеммам 1 «+5 V», 7 «SPEED» и 8 «GND». Минимальное сопротивление соответствует максимальной скорости, при увеличении сопротивления происходит снижение скорости. Для возможности регулирования скорости внешним аналоговым сигналом необходимо установить регистр USE_EXTERN_SPEED = 1 (см. описание регистров в разделе 6).
5. 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.
6. Подключите линии интерфейса RS-485 согласно пункту 4.

7. Включите блок питания. Устройство готово к настройке по протоколу Modbus.
8. Задайте необходимые настройки работы привода по протоколу Modbus - направление вращения, настройка работы внешних входов IN1 и IN2, способ управления (см. раздел 6).
9. При управлении приводом по протоколу Modbus передавайте команды через интерфейс RS-485. Таблица регистров блока, их назначение и возможные команды управления двигателем приведены в разделе 6.
10. При управлении приводом при помощи внешних сигналов для старта и остановки двигателя и для управления направлением вращения используйте сигналы IN1 и IN2. Регулирование скорости осуществляется внешним аналоговым сигналом, подаваемым на вход SPEED, или командами по Modbus.

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 подавлениядребезга контактов (см. рис. 5).

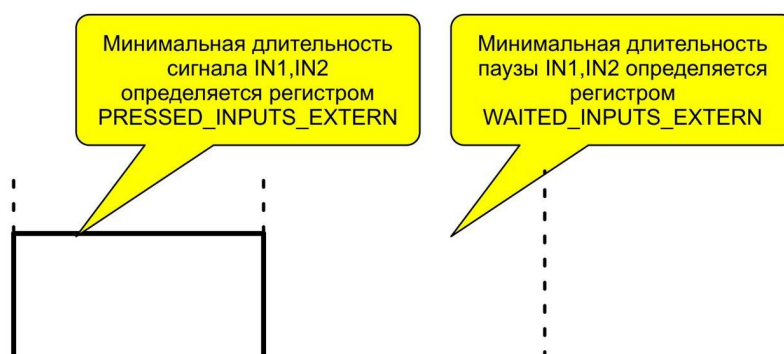


Рис. 5 - Механизм подавлениядребезга контактов.

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.
2004h	Coils	INV_DIR_bit	1-бит	Инвертирование направления вращения мотора
2005h	Coils	INV_CNT_bit	1-бит	Инвертирование направления счета позиции
2006h	Coils	ACC_ON_bit	1-бит	Включение ускорения, задаваемого регистром ACC
2007h	Coils	DEC_ON_bit	1-бит	Включение замедления, задаваемого регистром DEC
2008h	Coils	FQ_REGULATOR_ON_bit	1-бит	Включение четырех квадрантного регулятора скорости
2009h	Coils	AUTO_HOLD_MODE_bit	1-бит	Включение автоматического режима удержания

Данные управляющие сигналы доступны как для чтения, так и для записи. При записи в соответствующий регистр значения 1 активирует ту или иную функцию. Группа регистров 2000h-2003h после исполнения сбрасываются в 0.

2004h – INV_DIR_bit – Инверсия направления вращения. Используется для совместимости подключения двигателя с направлением вращения.

2005h – INV_CNT_bit – Инверсия направления счета позиции. Используется для совместимости подключения двигателя с направлением счета позиции.

Внимание: Запуск двигателя с неверно указанным направлением вращения или

направлением счета позиции может привести к неконтролируемому разгону двигателя до максимальных оборотов. Вращение двигателя останавливается только сигналом HARD_STOP.

2009h – AUTO_HOLD_MODE_bit – Функция запоминает значение тока при остановке мотора для последующего старта с данной величины.

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 тока фазы двигателя.
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
300Ah	Input Register	SPEED_INPUT_VALUE	16-бит	Показания АЦП входа SPEED

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 тока фазы двигателя, значение в мА.

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

3003h - CURRENT_POSITION - текущая позиция. Позиционирование производится 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 $TEMPERATURE_MCU / 10 \text{ } ^\circ\text{C}$.

3006h - TEMPERATURE_MOSFET - температура силовой части. Температура в области установки ключей MOSFET. Температура вычисляется как $TEMPERATURE_MOSFET / 10 \text{ } ^\circ\text{C}$.

3007h - TEMPERATURE_BRAKE - температура тормозной схемы. Температура в области тормозного резистора. Температура вычисляется как $TASK_COUNTER / 10 \text{ } ^\circ\text{C}$.

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
- 5 - программа пользователя завершена без принудительной остановки двигателя (END)
- 6 - программа пользователя завершена с принудительной остановкой двигателя (ENDH)
- 7 - программа пользователя остановлена из-за ошибки

300Ah - SPEED_INPUT_VALUE показания аналогового цифрового преобразователя входа SPEED в условных единицах.

6.4. Регистры настройки передачи данных по RS-485 Modbus

Адрес	Тип	Название	Размерность	Описание
5000h	Holding Register	SLAVE_ADDRESS_MODBUS	16-бит	ID устройства (Адрес устройства). Допустимые значения: 0...247. (0 - широковещательный адрес, без ответного сообщения)
5001h	Holding Register	TYPE_MODBUS	16-бит	Вариант настройки протокола.
5002h	Holding Register	BITRATE_MODBUS	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_ROTATION	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 максимального рабочего тока в обмотке двигателя. Диапазон 1000 мА до 20000 мА
500Ah	Holding Register	HOLD_CURRENT	16-бит	Ток удержания, в режиме остановки. Диапазон от 1000 мА до 20000 мА
500Bh	Holding Register	SPEED	16-бит	Заданная скорость вращения. Диапазон от 35 до 14000 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	N_POLE	16-бит	Количество полюсов двигателя. Диапазон от 1 до 12
5010h	Holding Register	USE_EXTERN_SPEED	16-бит	Выбор способа регулирования скорости 1 – по Modbus 2 – внешним сигналом на входе SPEED
5011h	Holding Register	RESERVED	16-бит	резерв
5012h	Holding Register	OFFSET_COMPENSATION	16-бит	Поправка торможения двигателя
5013h	Holding Register	PRESSED_INPUTS_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	POSITION	32-бит	Позиция, в которую необходимо переместиться. Диапазон значений от -2147483647 до + 2147483648
501Bh	Holding Register	POSITION1	32-бит	Заранее устанавливаемая пользователем позиция №1 Диапазон значений от -2147483647 до + 2147483648
501Dh	Holding Register	POSITION2	32-бит	Заранее устанавливаемая пользователем позиция №2 Диапазон значений от -2147483647 до + 2147483648

501Fh	Holding Register	POSITION3	32-бит	Заранее устанавливаемая пользователем позиция №3 Диапазон значений от -2147483647 до +2147483648
5021h	Holding Register	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. В режиме управления по Modbus (5004h - MODE_DEVICE = 1) запуск и остановка двигателя производится установкой бита в регистры 2000h и 2001h соответственно. В режиме управления внешними сигналами запуск двигателя производится по фронту сигнала на входе IN1, остановка – по фронту сигнала на входе IN2.
- 3 – перемещение в заданное положение. В режиме управления по Modbus (5004h - MODE_DEVICE = 1) номер позиции задается регистром POSITION_N (от 1 до

4), координаты позиций задаются регистрами POSITION1, POSITION2, POSITION3, POSITION4 соответственно. В режиме управления внешними сигналами используются только две координаты POSITION1 и POSITION2. Выбор координаты и перемещение осуществляется по одноименному входу. Преждевременная остановка двигателя возможна по сигналу от противоположного входа.

5007h – **MODE_EXT_IN** – конфигурация режима работы внешних входов IN1, IN2, 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.

Логика работы входов может быть настроена, если выбран режим управления внешними сигналами (5004h - **MODE_DEVICE** = 2) и непрерывное вращение в сторону (5006h – **MODE_ROTATION** = 1).

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** – установка максимального рабочего тока в обмотке двигателя – от 1000 мА до 20000 мА

500Ah – **HOLD_CURRENT** – ток удержания в режиме остановки. В регистре указывается значение тока фазы в диапазоне от 1000 мА до 20000 мА. Установка нулевого значения отключает функцию удержания.

500Bh – **SPEED** – заданная скорость вращения при управлении по Modbus (от 35 до 14000 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 10 соответствует 100 об/сек², значение 1000 соответствует 5000 об/сек².

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

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

500Fh – **N_POLE** – количество полюсов двигателя.

5010h – **USE_EXTERN_SPEED** – установка режима задания скорости вращения, возможные значения:

- 1 – командами по Modbus
- 2 – внешним аналоговым сигналом на входе SPEED

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 – **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 – **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;
- 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

Сброс ошибки может быть выполнен записью нулевого значения в регистр 5023h – ERROR.

5024h – **FLAG_SAVE_INI** – регистр сохранения пользовательских настроек – при записи значения 0x37FA в данный регистр будут сохранены в энергонезависимую память настройки, определяемые регистрами 2004h...2009h, 5000h..501Fh, 5028h...502Ch.

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.

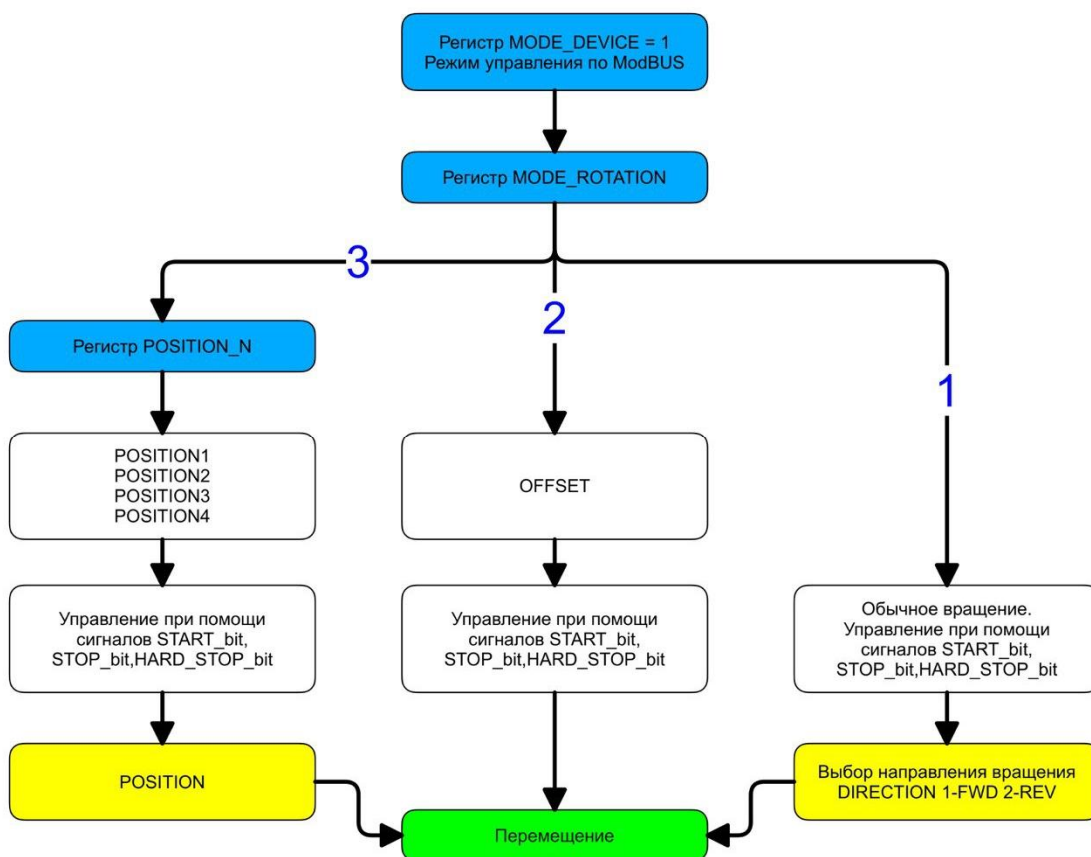


Рис.6. Блок-схема выбора режима работы при управлении приводом по Modbus

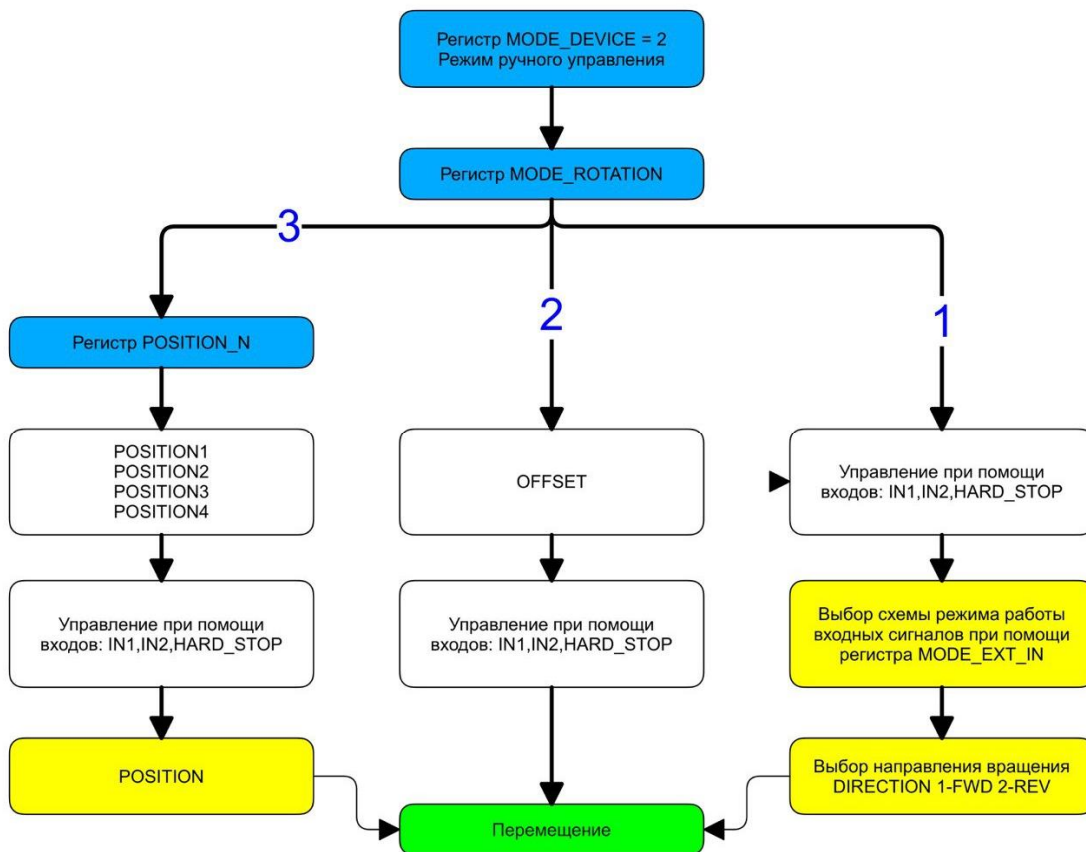


Рис.7. Блок-схема выбора режима работы при управлении приводом внешними сигналами

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.

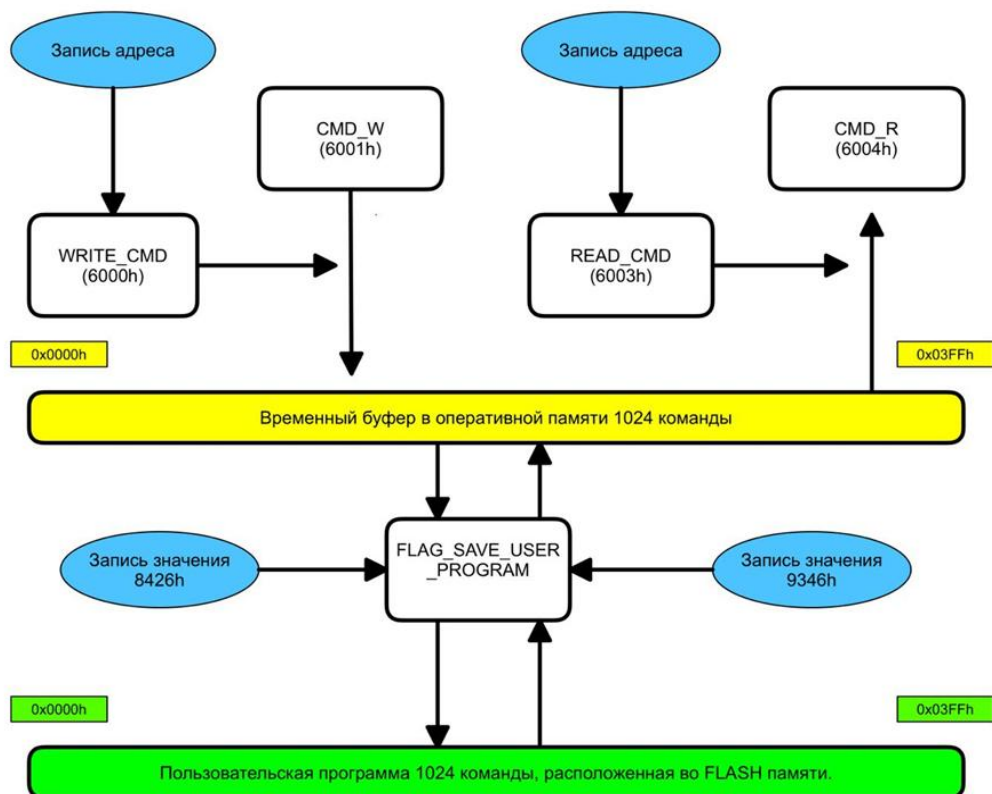


Рис.8. Блок-схема процедур чтения и записи пользовательской программы.

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 коэффициентов ПИД-регулятора

Адрес	Тип	Название	Размерность	Описание
5028h	Holding Register	PROPORTIONAL	32-бит	Пропорциональный коэффициент ПИД регулятора.
502Ah	Holding Register	INTEGRAL	32-бит	Интегральный коэффициент ПИД регулятора.
502Ch	Holding Register	DIFFERENCIAL	32-бит	Дифференциальный коэффициент ПИД регулятора

Изменение коэффициентов ПИД регулятора допускается только при полном понимании выполняемых действий. Не рекомендуется резкое изменение значений коэффициентов.

6.9. 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 БУВД-20Modbus значения:

- HW_MAJOR 1000
- HW_MINOR x
- FW_MAJOR x
- FW_MINOR x

6.10. Инструкции пользовательской программы

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

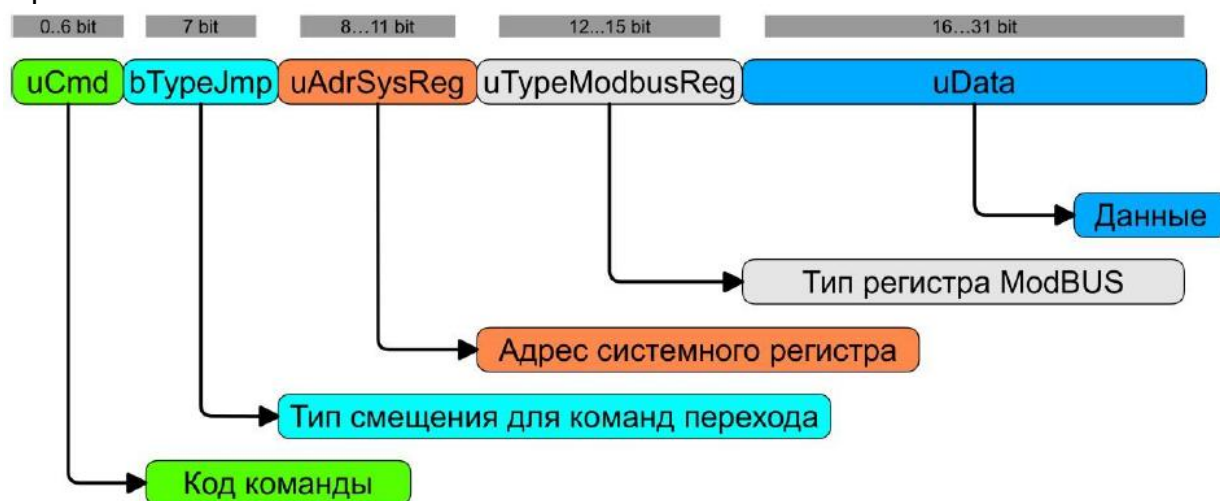


Рис.9 – структура инструкции пользовательской программы

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

uCmd – 7 бит – код команды

bTypeJump – 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 состоят 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 указано, 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	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		registers	
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 registers	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			
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	Адрес регистра	0 – сдвиг влево

0x54	CMD_DSH_MODBUS_CONST	3 – Holding registers	ModBUS 0..65535	1 – сдвиг вправо
------	----------------------	-----------------------	--------------------	---------------------

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.

Математические операции возможны между системными регистрами, регистрами Modbus и константами. Операнды инструкций, оперирующих 32-разрядными данными, располагаются в двух последовательных 16-разрядных регистрах. При обращении к регистру в инструкции указывается адрес младшего регистра. Адрес старшего регистра получается путем увеличения адреса младшего регистра на единицу.

В таблице ниже приведены математические инструкции, коды команд для работы с 16-битными и 32-битными данными и расположение данных при составлении строки инструкции.

Инструкция			Операнд 1	Операнд 2	Результат
Имя	Код				
	16- битные данные	32- битные данные	S1	S2	D
Сложение (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
Вычитание (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
Деление, остаток отбрасывается (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
Умножение (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
Поразрядное логическое И (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
Поразрядное логическое ИЛИ (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
Поразрядное логическое исключающее ИЛИ ($S1 \wedge S2 = D$)					
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
Сдвиг данных ($S1 \gg S2 = D$ или $S1 \ll S2 = D$ – направление сдвига указывается в поле bTypeImp: 0 –влево, 1 –вправо)					
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 - Константа в поле данных uData

CONST_2 - Константа, находящаяся в следующей командной строке uCMD_DATA

SYS_REG_1 – значение, содержащееся в системном регистре (или двух последовательных системных регистрах для 64-битных инструкций) по адресу uAdrSysReg. Адреса системных регистров 0..9.

SYS_REG_2 – значение, содержащееся в системном регистре (или двух последовательных системных регистрах для 64-битных инструкций), адрес которого определяется в следующей командной строке uCMD_DATA

Modbus_REG – значение, содержащееся в регистре Modbus: тип регистра в поле TypeModbusReg, адрес регистра в поле uData

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 RJ12 - 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 БУВД-20Modbus
Паспорт БУВД20.MODBUS.001.ПС

1 шт.
1 шт.

9. ГАРАНТИЙНЫЕ ОБЯЗАТЕЛЬСТВА

Изготовитель гарантирует безотказную работу блока в течение 12 месяцев со дня продажи, при соблюдении условий эксплуатации. Изготовитель оставляет за собой право вносить изменения в конструкцию, комплектацию или технологию изготовления, не ухудшающие характеристик изделия, без отражения изменений в настоящем паспорте и без уведомления потребителей.

Адрес предприятия-изготовителя:

ООО «НПО Электропривод»,

195197, Россия, Санкт-Петербург, Полюстровский пр.43, А.

Тел.: +7 (812) 467-39-58

e-mail: sale@electroprivod.ru

Заводской номер:

Дата продажи:

Редакция от 21.04.2026