Контроллер шагового двигателя

# OSM-88RA

Прошивка OSM MB

Полное описание и руководство по эксплуатации

Версия 15-0313

Санкт-Петербург
2013

Компания «Онитекс» благодарит за внимание к нашей продукции. Пожалуйста, ознакомьтесь с настоящей инструкцией. Данное руководство поможет быстро внедрить устройство в эксплуатацию и избежать возможных проблем в работе устройства. Помните, что нарушение инструкции по эксплуатации является причиной для отказа в гарантийном обслуживании устройства.

*Производитель постоянно работает над совершенствованием выпускаемых изделий и оставляет за собой право на внесение изменений и модификаций, не ухудшающих характеристик устройств, без предварительного уведомления. Данные изменения производитель вносит в новые версии руководств по эксплуатации.*

*Производитель многократно проверил данные, изложенные в данном руководстве, и считает их полноценными и исчерпывающими, однако не несет ответственности за возможные ошибки и неточности, возникшие вследствие типографских ошибок или иных причин.*

*Ни одна часть данного руководства не может быть скопирована без письменного согласия ООО «Онитекс».*

# Оглавление

1. Описание устройства 4
2. Технические характеристики 7
3. Условия эксплуатации 8
4. Подключение устройства 9
4.1 Начало работы 9
4.2 Подключение двигателя 12
4.3 Выставление тока и режима дробления шага двигателя. 13
4.4 Выбор интерфейса 14
4.5 Подключение интерфейса RS-485 16
4.6 Подключение интерфейса RS-232 21
4.7 Использование преобразователей интерфейсов 22
5. Режимы работы устройства 23
5.1 Общее описание режимов 23
5.2 Режим драйвера 23
5.3 Автоматический режим 26
5.4 Режим контроллера 27
5.5 Программный режим 28
5.6 Подключение в режиме контроллера и программном режиме 28
5.7 Использование счетчиков и энкодера 31
6. Управление устройством по протоколу Modbus 32
6.1 Описание протокола 32
6.2 Типы данных Modbus 33
6.3 Функции Modbus 33
6.4 Управление контроллерами OSM 35
6.5 Примеры использования протокола. 41
6.6 Использование программы Modbus Terminal 43
7. Использование SDK 44
8. Обновление встроенного программного обеспечения 45
9. Гарантийные обязательства и техническая поддержка 47

# 1. Описание устройства

Устройство управления шаговыми двигателями OSM-88RA (далее – устройство, блок) является современным, многофункциональным, надежным устройством, предназначенным для управления шаговыми двигателями небольшой и средней мощности. Блок работает с 4-, 6-, 8-ми выводными шаговыми двигателями с током обмоток до 8А.

Данное устройство реализует несколько режимов работы. Это позволяет использовать его для различных целей, начиная от применения в координатных станках с ЧПУ, и заканчивая автономной работой в составе технологических линий. Устройство может управляться стандартными логическими сигналами «ШАГ», «НАПРАВЛЕНИЕ» и «РАЗРЕШЕНИЕ», работать в автоматическом режиме, вращая двигатель со скорость, заданной потенциометром, а так же управляться в режиме реального времени по последовательному порту.

Отличием данной версии блока управления является то, что по умолчанию он работает в режиме контроллера (прошивка OSM MB) с управлением по протоколу Modbus, однако может быть перепрограммирован прошивкой OSM ASCII, изменяющей алгоритм его работы на стандартную функциональность контроллеров OSM с управлением по ASCII-протоколу. В данном руководстве описана работа контроллера в режиме управления по Modbus.

Внешний вид устройства показан на рисунке 1.1.

Рисунок 1.1 - Внешний вид контроллера OSM-88RA

Габаритные и присоединительные размеры устройства указаны на рисунке 1.2

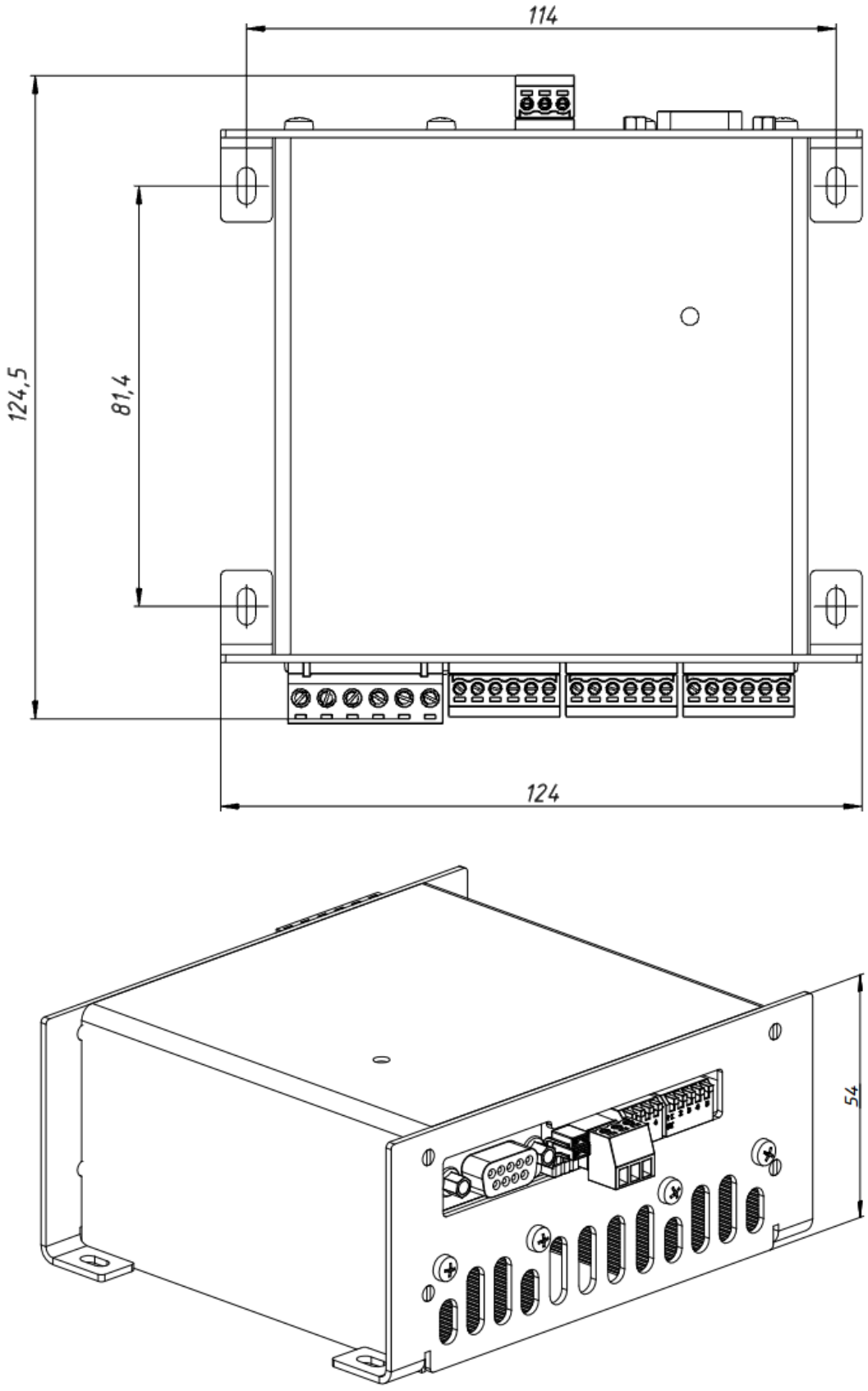



Рисунок 1.2 - Габаритные и присоединительные размеры устройства

Основные отличительные особенности устройства:

- установка скорости, ускорения, режима дробления шага, величины перемещения, направления движения шагового двигателя;
- дробление полного шага до 1/16;
- функция удаленного обновления встроенного программного обеспечения для возможности прошивки новой версии;
- наличие аналогового входа и разъема подключения внешнего потенциометра;
- большое количество пользовательских настроек;
- управление по промышленному протоколу Modbus с возможностью адресации до 32-х устройств;
- пять дополнительных входов для приема сигналов от внешних устройств (датчиков), возможность работы с инкрементальным энкодером;
- один дополнительный выход для подачи сигналов внешним устройствам;
- работа с программой Modbus Terminal;
- возможность управления контроллером с использованием библиотеки OsmModbusDriver, входящей в состав SDK.

  На рисунке 1.3 представлена структурная схема контроллера.

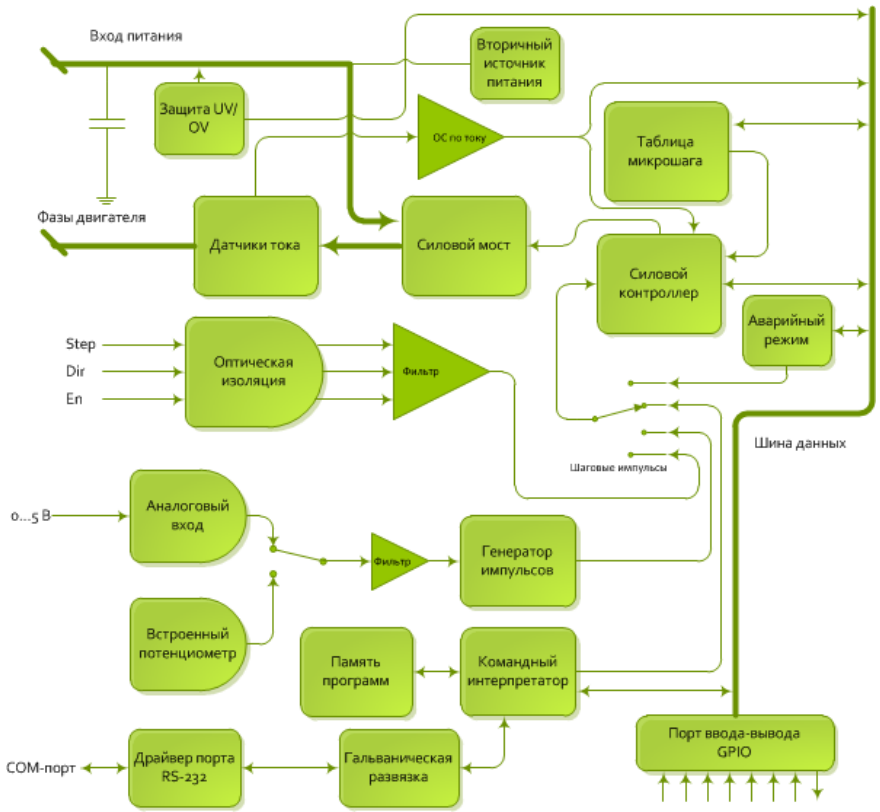



Рисунок 1.3 - Структурная схема контроллера

Данная структурная схема дает общее представление о режимах работы контроллера, которые будут описаны в настоящем руководстве.

# 2. Технические характеристики

Таблица 2.1 - Технические характеристики устройства.

| | | |
|---|---|---|
| Общие характеристики: | | |
| Максимальный выходной ток | 8 | А |
| Минимальный выходной ток | 1 | А |
| Габаритные размеры, не более | 125x124x54 | мм |
| Масса нетто, не более | 0.8 | кг |
| Коэффициенты дробления шага | 1, 1/2, 1/4, 1/16 | |
| Максимальная частота входа STEP, не менее | 100 000 | Гц |
| Диапазон частот встроенного генератора импульсов | 1…15 000 | Гц |
| Дискретность установки частоты генератора | 1 | Гц |
| Напряжение питания | 25…90 | Вольт |
| Ошибка установки величины фазных токов, не более | +/-5 | % |
| Оптически изолированные входы управления: | | |
| Входное сопротивление | 1 | кОм |
| Максимальное напряжение на входах | 9 | Вольт |
| Напряжение изоляции входа STEP | 2500 | Вольт |
| Напряжение изоляции входов DIR и EN | 2000 | Вольт |
| Неизолированные сигнальные входы управления с подтяжкой: | | |
| Номинал подтягивающего резистора | 3.6 | кОм |
| Аналоговый вход: | | |
| Разрешение АЦП | 8 | бит |
| Входное сопротивление, не менее | 500 | кОм |
| Диапазон входных напряжений | 0…5 | Вольт |
| Сигнальный изолированный выход: | | |
| Тип | NPN | |
| Максимальный ток, не более | 10 | мА |
| Максимальное напряжение коллектор-эмиттер | 50 | Вольт |
| Напряжение изоляции максимальное, не менее | 3700 | |
| Интерфейс RS232 и RS-485: | | |
| Напряжение гальванической изоляции, не менее | 1000 | Вольт |
| Скорость обмена | 9600…115200 | бит/с |
| Количество бит | 8 | |
| Четность | нет | |
| Стоповые биты | 1 | |
| Управление потоком | аппаратное | |
| Дальность связи по RS-232, не более | 15 | м |
| Дальность связи по RS-485, не более | 1000 | м |
| Число устройств на линии RS-232 | 1 | |
| Число устройств на линии RS-485 | 32 | |
| Параметры климатического исполнения (ГОСТ 15150-69): | | |
| Диапазон температур, | -25… +40 | °С |
| Относительная влажность (при 25°С 6 мес. в году) | до 90% | |
| Атмосферное давление | 650…800 | мм. рт. ст. |
| Степень защиты оболочки по ГОСТ 14254-96 | IP31 | |

# 3. Условия эксплуатации

- Внимание! Во избежание поражения электрическим током необходимо соблюдать изложенные ниже правила.
- При работе устройства в блоке питания имеется опасное для жизни напряжение 220В, соблюдайте правила электробезопасности.
- Все подключения и монтажные работы производить только при выключенном из розетке источнике питания, на *обесточенной установке*.
- Все подключения проводить в строгом соответствии с данным руководством и руководством к источнику питания.
- Необходимо обеспечить заземление источника питания, персонального компьютера и других устройств, входящих в систему управления, в соответствии с ПУЭ.
- Запрещается соединять минусовой провод питания, идущий к контроллеру, с защитным занулением или заземлением.
- Подключение или отключение двигателя во время работы устройства запрещено, так как может вывести его из строя

Устройство предназначено для работы от стабилизированного или нестабилизированного источника питания с выходным напряжением 25-90 Вольт. Максимально допустимое напряжение питания устройства составляет 95 Вольт, включая флуктуации питающего напряжения и обратную ЭДС двигателя, таким образом, источник питания не должен выдавать больше 70 Вольт. Не допускается эксплуатация устройства при температурах, выходящих за пределы указанных в таблице 2.1.

# 4. Подключение устройства

## 4.1 Начало работы

Все подключения и монтажные работы проводить только в обесточенном состоянии.

На рисунке 4.1.1 изображены все управляющие входы и элементы контроллера.

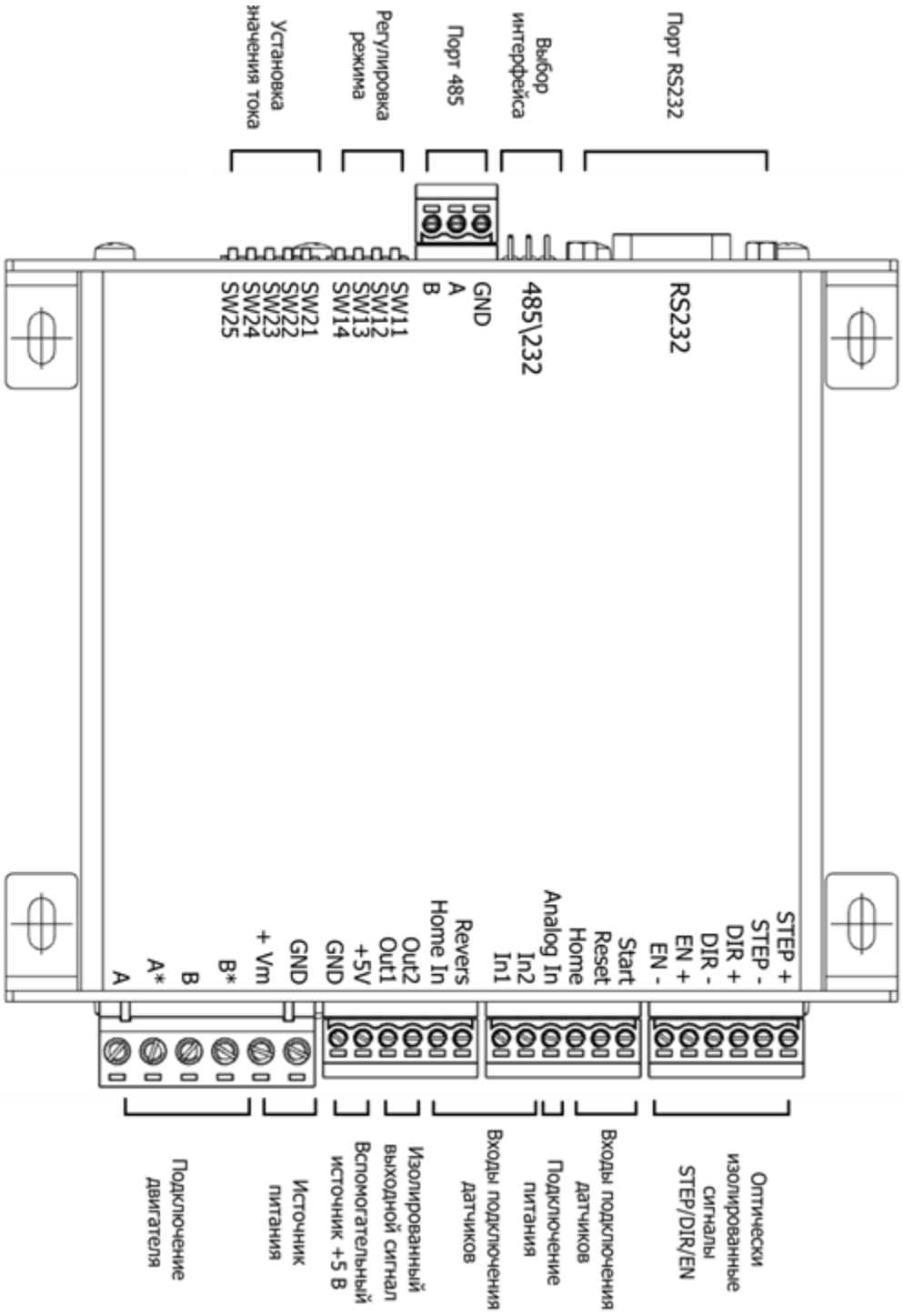



Рисунок 4.1.1 - Расположение управляющих входов и элементов управления

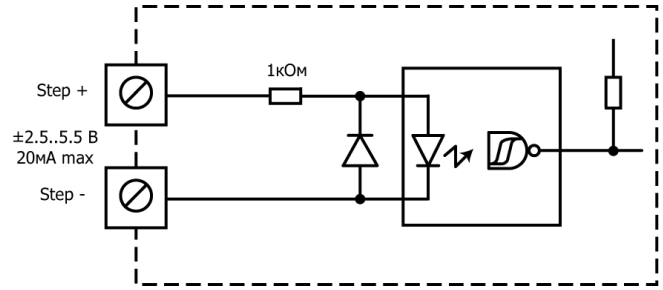
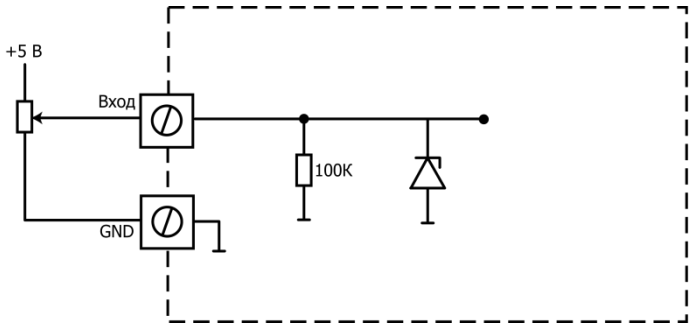
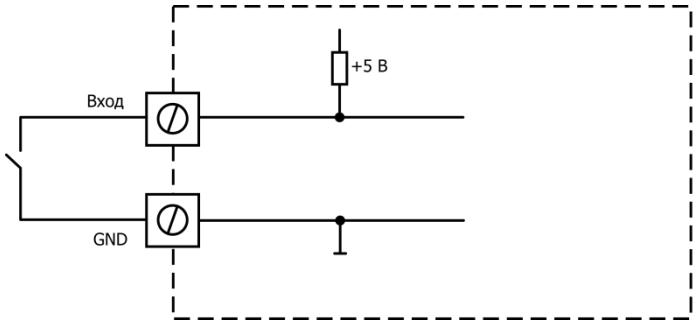
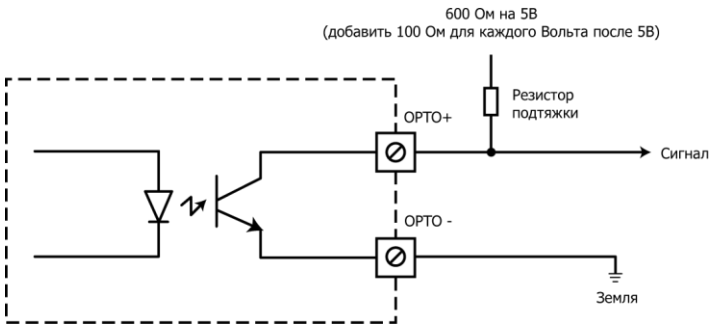
Функциональное назначение управляющих входов и выходов указано в таблице 4.1.1.

Таблица 4.1.1 - Функциональное назначение управляющих входов и выходов

| Вход/Выход | Контроллер (Modbus) | Автоматический | Режим драйвера |
|---|---|---|---|
| GND | Общий источника питания | | |
| +V | Плюс источника питания | | |
| A, A* | Фаза А шагового двигателя | | |
| B, B* | Фаза В шагового двигателя | | |
| GND/+5V | Вспомогательный источник +5В | | |
| **Out1/Out2** | Выход типа «общий коллектор» для управления внешними устройствами | *Не используется* | |
| Home in | Вход датчика начального положения | *Не используется* | |
| Revers | Реверс/<br>Вход счетчика INT_counter/<br>Вход канала А энкодера | Реверс | *Не используется* |
| In1 | Дискретный вход 1 | *Не используется* | |
| In2 | Дискретный вход 1 | *Не используется* | |
| Analog in | *Не используется* | Аналоговый вход | *Не используется* |
| Home | Старт поиска начального положения | *Не используется* | |
| Reset | Сброс контроллера | *Не используется* | |
| Start | *Не используется* | *Не используется* | |
| EN+/EN- | Вход счетчика EN_counter | Включение напряжения на обмотках | |
| DIR+/DIR- | Дополнительный оптоизолированный вход DIR | Направление вращения | |
| STEP+/STEP- | Дополнительный оптоизолированный вход STEP/<br>Вход канала В энкодера | *Не используется* | Вход шаговых импульсов |
| RS-232 | Порт RS-232 | *Не используется* | |
| RS-485 | Порт RS-485 | *Не используется* | |

В устройстве применяются различные типы управляющих входов и портов ввода-вывода. Описания структурных схем управляющих входов приведены в таблице 4.1.2.

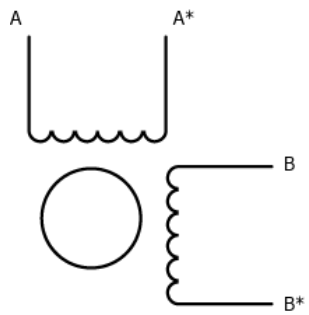
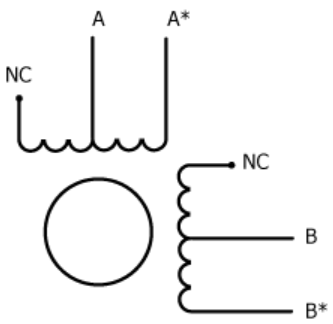
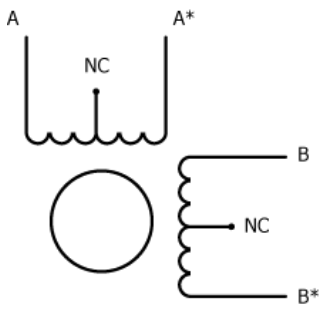
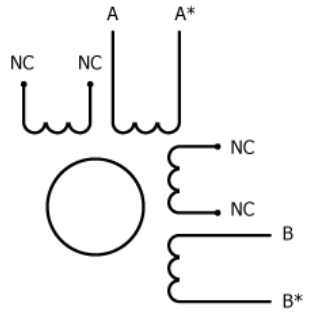
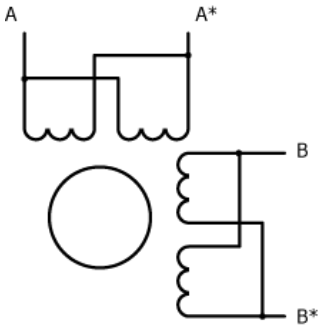
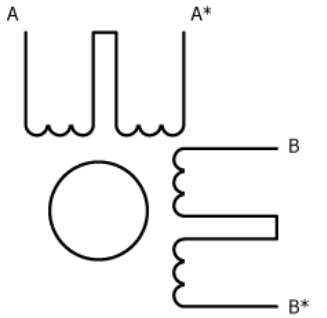
Таблица 4.1.2 – Структурные схемы управляющих входов

| Входы | Назначение | Структурная схема |
|---|---|---|
| STEP<br>DIR<br>EN | Входы дифференциальные оптически-изолированные |  |
| Analog in | Вход аналоговый неизолированный |  |
| Home in<br>Revers<br>In1<br>In2<br>Analog in<br>Home<br>Reset<br>Start | Вход сигнальный с подтяжкой неизолированный |  |
| Out | Выход сигнальный оптически изолированный |  |

# 4.2 Подключение двигателя

Устройство работает с 4-, 6-, 8-ми выводными шаговыми двигателями. В таблице 5.3 представлены возможные способы включения фаз двигателя. Выводы фаз шагового двигателя подключаются к выходам устройства A, A*, B, B* в соответствии с таблицей 4.2.1.

Таблица 4.2.1 – Способы включения фаз двигателя

| | | |
|---|---|---|
|  |  |  |
| 1. Биполярный ШД. | 2. Биполярный с отводом от середины обмотки | 3. Биполярный с отводом от середины обмотки |
|  |  |  |
| 4. Униполярный четырехфазный ШД, подключение двух обмоток | 5. Четырехфазный униполярный ШД, параллельное подключение. | 6. Четырехфазный униполярный ШД, последовательное подключение. |

1. Подключение однозначно определено.
2. Момент и ток равны паспортным.
3. Момент выше в 1.4 раза.
4. Момент и ток равны паспортным.
5. Момент выше в 1.4 раза, рекомендуется для высоких частот.
6. Момент выше в 1.4 раза, рекомендуется для низких частот.

# 4.3 Выставление тока и режима дробления шага двигателя.

Установка тока двигателя в режиме драйвера и автоматическом режиме производится переключателями SW22 – SW25 (таблица 4.3.1). В режиме управления по Modbus ток задается записью значения в регистр Modbus. Переключатели SW22 – SW25 в режиме контроллера задают адрес устройства.

Ток контроллера не должен превышать номинальный ток, указанный в паспорте на двигатель, иначе двигатель будет перегреваться и может выйти из строя.

Таблица 4.3.1 – Выставление тока двигателя

| | SW22 | SW23 | SW24 | SW25 |
|---|---|---|---|---|
| 1 | OFF | OFF | OFF | OFF |
| 1.5 | OFF | OFF | OFF | ON |
| 2 | OFF | OFF | ON | OFF |
| 2.4 | OFF | OFF | ON | ON |
| 3 | OFF | ON | OFF | OFF |
| 3.3 | OFF | ON | OFF | ON |
| 3.8 | OFF | ON | ON | OFF |
| 4.2 | OFF | ON | ON | ON |
| 4.7 | ON | OFF | OFF | OFF |
| 5 | ON | OFF | OFF | ON |
| 5.5 | ON | OFF | ON | OFF |
| 6 | ON | OFF | ON | ON |
| 6.4 | ON | ON | OFF | OFF |
| 7 | ON | ON | OFF | ON |
| 7.5 | ON | ON | ON | OFF |
| 8 | ON | ON | ON | ON |

Устройство имеет возможность управления током покоя. При включенном режиме SLEEP устройство уменьшает ток двигателя в два раза через одну секунду после прекращения вращения. Это позволяет уменьшить нагрев двигателя и контроллера, а так же уменьшить энергопотребление (рисунок 4.3.1).

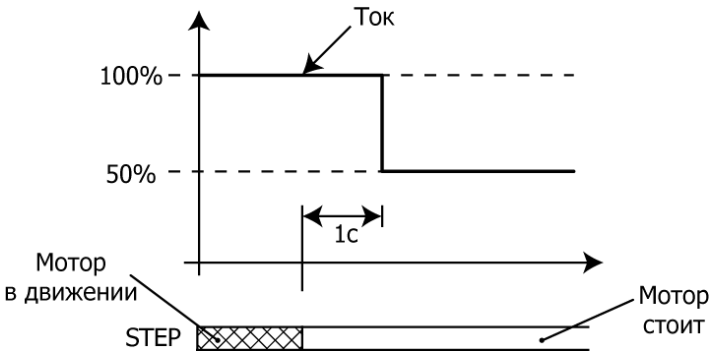



Рисунок 4.3.1 – Режим SLEEP

Включение режима SLEEP происходит переключателем SW21.

- Ток при простое 100%  SW21 = OFF
- Ток при простое 50%  SW21 = ON

В режиме контроллера имеется возможность задавать длительность и величину тока покоя записью соответствующих значений в регистры Modbus.

Выбор режима дробления шага осуществляется джамперами SW13 и SW14. Соответствия между режимом и состоянием DIP-переключателя описано в таблице 4.3.2. Состояние «ON» соответствует опущенному переключателю. В режиме контроллера дробление шага выбирается записью соответствующих значений в регистр Modbus.

Таблица 4.3.2 – Выбор режима дробления шага

| Дробление шага | 1 | 1/2 | 1/4 | 1/16 |
|---|---|---|---|---|
| SW13 | ON | OFF | ON | OFF |
| SW14 | ON | ON | OFF | OFF |

## 4.4 Выбор интерфейса

В режиме контроллера возможно управление по одному из двух типов интерфейса – RS-485 или RS-232.

RS-485 наилучшим образом подходит для управления несколькими устройствами на одной линии или при необходимости управления на большом расстоянии (до 1000м). Интерфейс RS-232 удобно использовать для отладки, либо управления одним устройством на небольшом расстоянии. Кроме того, через RS-232 производится обновление прошивки контроллера.

Выбор интерфейса производится одноименными джамперами (см. схему 4.1.1).

На рисунке 4.4.1 показано положение, соответствующее выбранному порту RS-232. Джамперы при этом соединяют средние контакты с контактами, ближними к разъему порта RS-232.

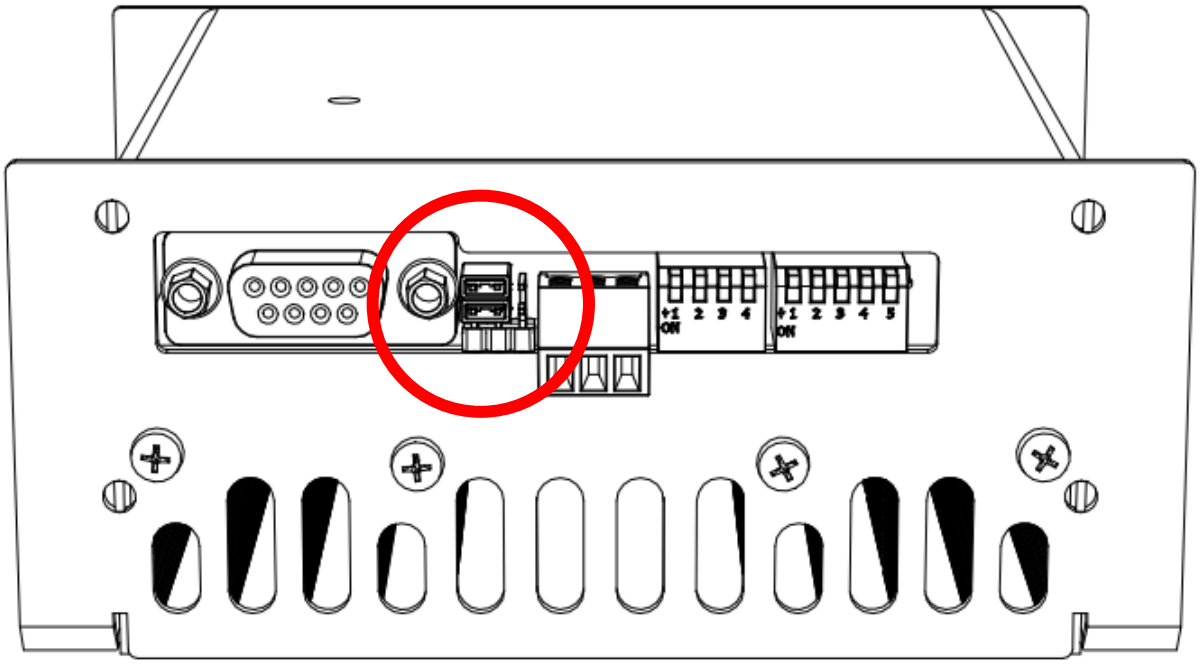


Рисунок 4.4.1 – Позиция переключателя при выборе RS-232

На рисунке 4.4.2 показано положение, соответствующее выбору RS-485. Джамперы соединяют средние контакты с контактами, ближними к RS-485.

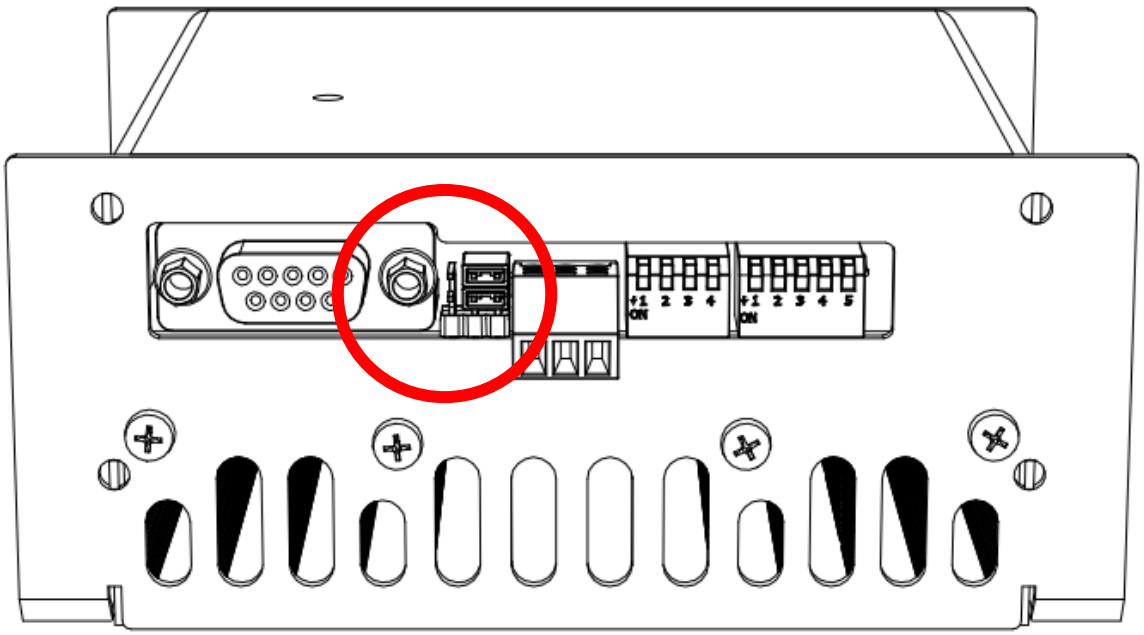


Рисунок 4.4.2 – Позиция переключателя при выборе RS-485

# 4.5 Подключение интерфейса RS-485

RS-485 — стандарт передачи данных по полудуплексному многоточечному последовательному интерфейсу. Стандарт получил широкое распространение в промышленных системах благодаря удобству применения и хорошей помехоустойчивости в силу применения дифференциальной линии передачи данных. Стандарт является основой для многих промышленных сетей, таких, как Modbus. Использование RS-485 позволяет не только увеличить дальность управления шаговым приводом, повысить помехозащищенность линии, но и организовать сеть из нескольких устройств, что существенно расширяет диапазон применения шаговых приводов в составе различных технологических линий и механизмов.

Передача сигнала осуществляется по дифференциальной линии связи. Дифференциальная передача сигнала заключается в том, что в одном проводнике (A) идет прямой сигнал, а во втором (B) – инверсный. В итоге между проводами всегда есть разность потенциалов. Это обеспечивает высокую устойчивость к синфазным помехам. Кроме того, для улучшения защищенности, особенно при большой протяженности линии, применяется экранирование витой пары. Все это позволяет обеспечивать достаточно высокую скорость и дальность передачи, которая может достигать 120м при скорости 10Мбит, либо 1000м при небольших скоростях передачи (порядка 100 кбит).

Важным фактором, влияющим на работоспособность сети RS-485, является наличие терминальных резисторов на концах кабеля. Номинал данных резисторов должен быть равен волновому сопротивлению кабеля. Стандартным значением считается величина 100-120 Ом. Витые пары типа UTP-5 имеют сопротивление 100 Ом, кабели, выпускаемые для сети RS-485 – 120 Ом. Необходимо убедиться в том, что значение терминального резистора равно значению для выбранного кабеля. Кроме того, в топологии сети не должно быть ответвлений. Иными словами, кабель должен входить в устройство в том же месте, откуда он идет в следующее устройство, длина ответвления должна быть минимальной. Все эти меры направлены на то, чтобы уменьшить отражения и стоячие волны в линии.

Технические характеристики RS-485 (по стандарту):

- Допустимое число приёмопередатчиков: 32
- Максимальная длина линии связи: 1200 м
- Максимальная скорость передачи: 10 Мбит/с
- Напряжение выходного сигнала, не менее: ±1,5 В
- Напряжение выходного сигнала, не более: ±5 В
- Максимальный ток короткого замыкания драйвера: 250 мА
- Выходное сопротивление драйвера, не более: 54 Ом
- Входное сопротивление драйвера, не менее: 12 кОм
- Допустимое суммарное входное сопротивление на линии: 375 Ом
- Диапазон нечувствительности к сигналу: ±200 мВ
- Уровень логической единицы: более +200 мВ
- Уровень логического нуля: менее -200 мВ

Контроллеры необходимо подключать проводом типа «витая пара», при этом контакты A и B должны быть соединены поименно во всех устройствах сети, включая ведущее. Следует уделить внимание правильной прокладке кабеля. Кабель не должен идти вдоль силовых линий, при необходимости пересечения кабеля с силовым или высоковольтным проводом необходимо делать это под углом, максимально близком к прямому. Следует так же избегать прокладки кабеля вблизи источников электромагнитных помех, особенно при большой длине провода. Дополнительную устойчивость линии к влиянию электромагнитных помех обеспечит применение экранированной витой пары. Экран необходимо заземлять только с одной стороны для предотвращения протекания тока помехи по слою экрана.

На рисунке 4.5.1 приведена схема подключения одного контроллера без установленного терминального резистора.

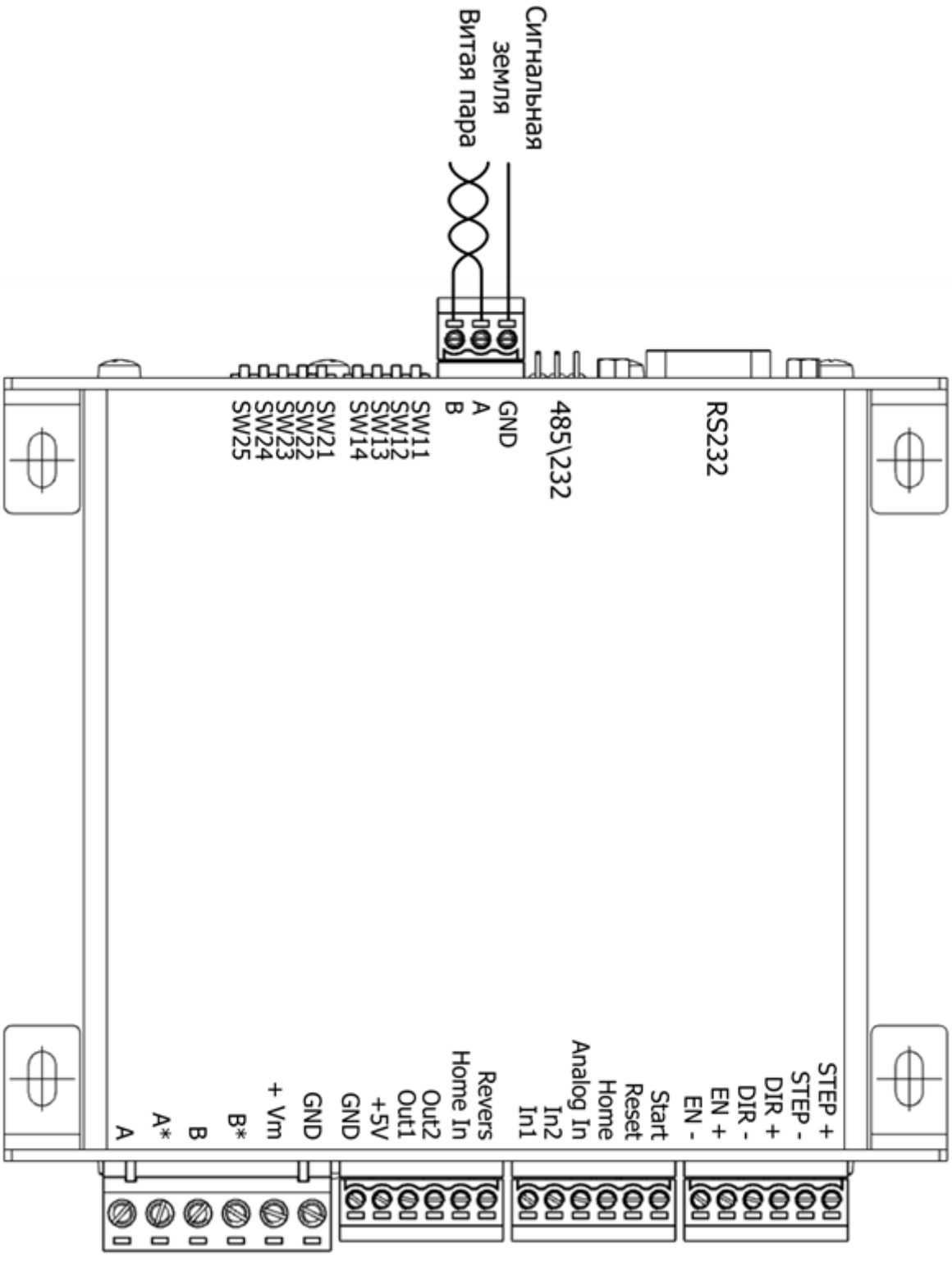



Рисунок 4.5.1 – Схема подключения витой пары к контроллеру

Стандарт RS-485 требует установку терминального резистора на конце линии. Это особенно актуально на высоких скоростях, больших расстояниях или при использовании нескольких устройств на линии. Схема подключения терминального резистора показана на рисунке 4.5.2.

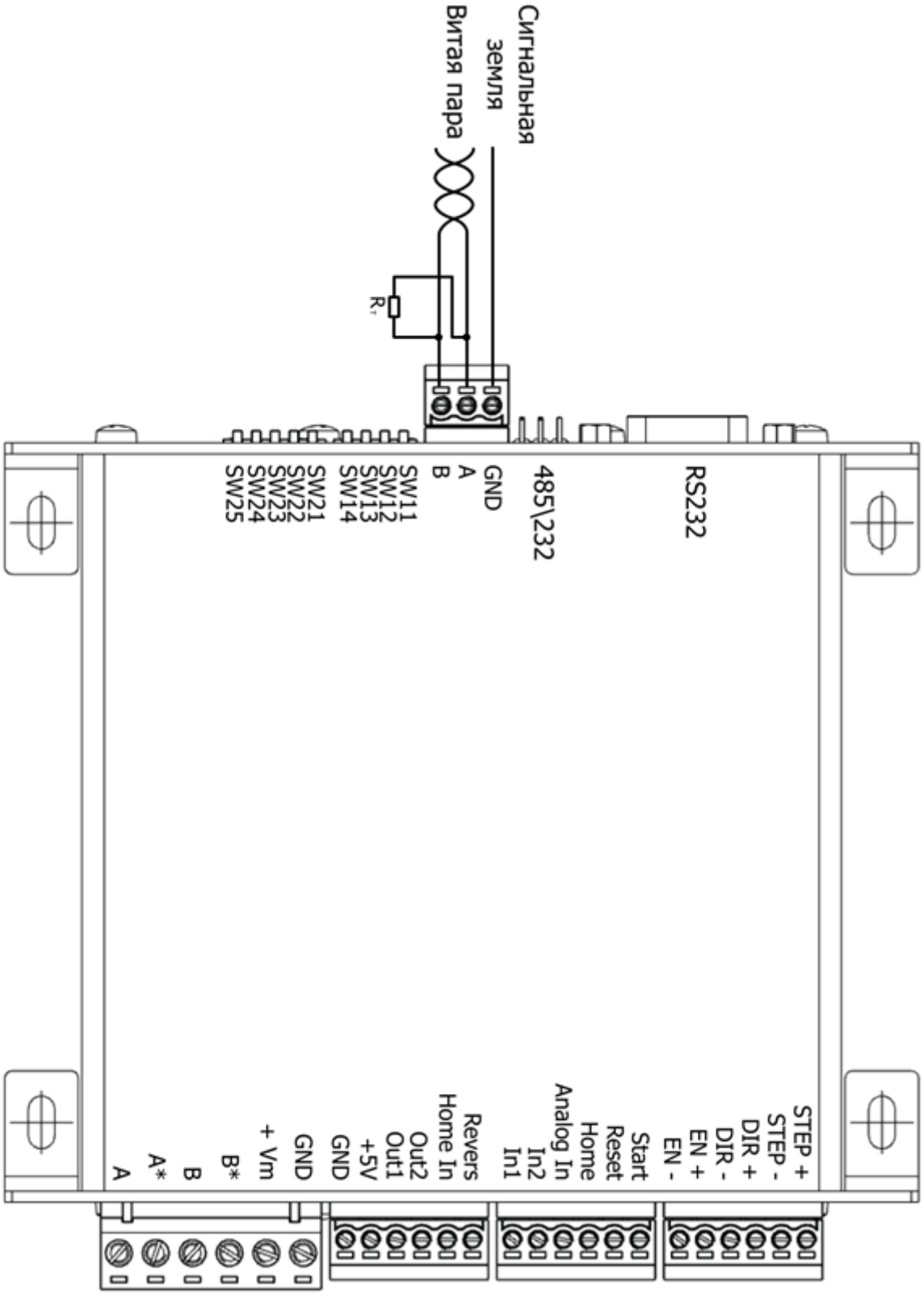



Рисунок 4.5.2 – Схема подключения терминального резистора

При необходимости подключения нескольких устройств на одну линию следует придерживаться правила минимальной длины отвода от линии. Пример подключения показан на рисунке 4.5.3. Терминальный резистор подключается только на передатчике и последнем устройстве в сети.

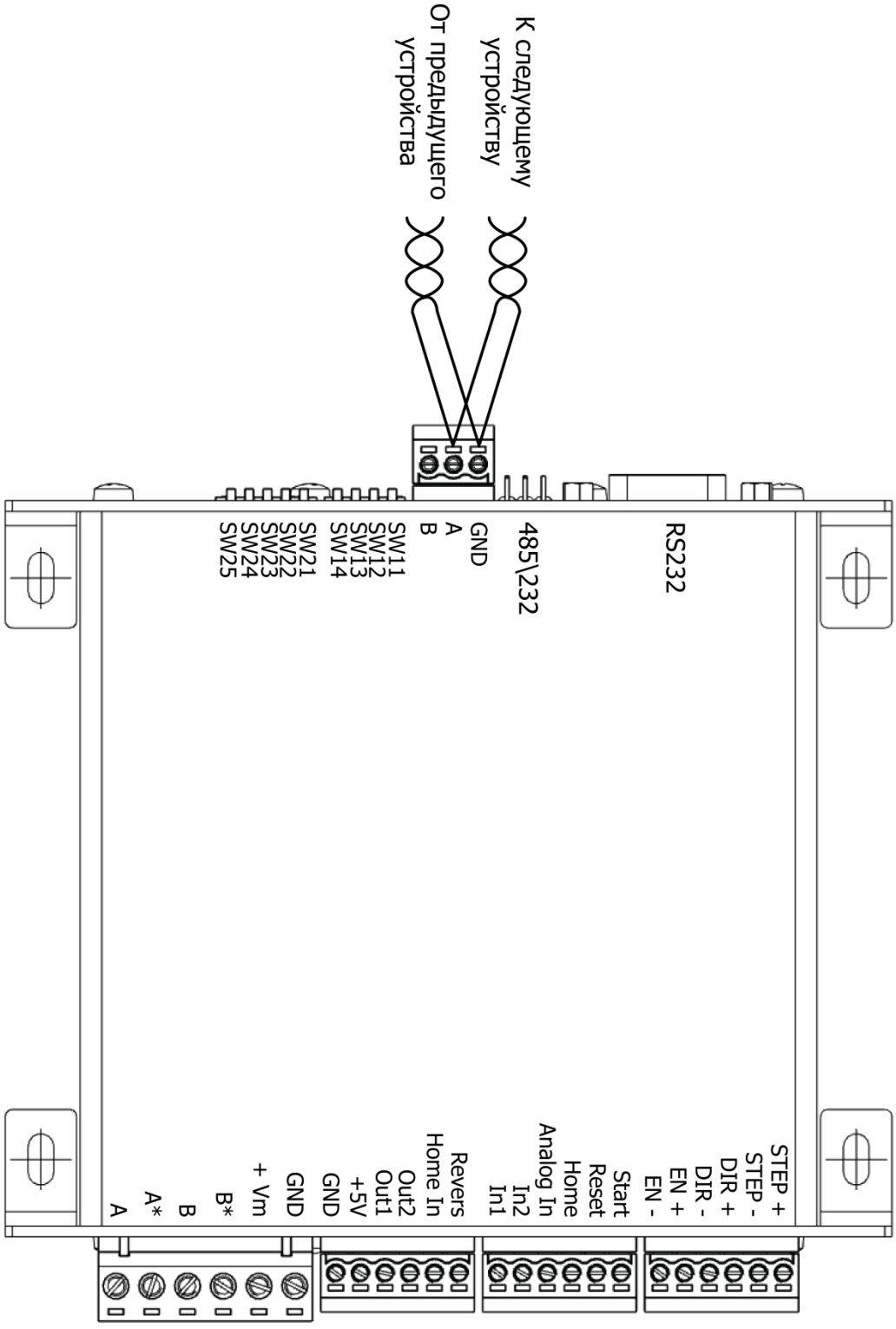



Рисунок 4.5.3 – Схема подключения нескольких контроллеров

Таким образом, монтаж сети RS-485 происходит методом «от точки к точке», с обязательной установкой терминального резистора в начале и конце линии.

# 4.6 Подключение интерфейса RS-232

Подключение к управляющему устройству или ПК при выбранном интерфейсе RS-232 осуществляется стандартным 9-контактным кабелем.

Распиновка кабеля для корректной работы устройства приведена в таблице 4.6.1:

| ПК | Контроллер OSM |
|---|---|
| 2 | 2 |
| 3 | 3 |
| 5 | 5 |

Другие контакты кабеля не используются.

# 4.7 Использование преобразователей интерфейсов

Устройство допускает соединение с управляющим ПК через переходники USB->RS-232 и USB->RS-485.

В случае, если управляющий ПК не оборудован портом RS-232, возможно подключение через переходник, позволяющий работать в режиме эмуляции COM-порта. Устройство спроектировано таким образом, что может работать с большинством переходников. Производителем гарантируется работа с переходниками на основе микросхемы CP2102, в других случаях работа через переходник возможна, но не гарантируется. В первую очередь это касается старых переходников, не совместимых с USB2.0.

В случае подключения через переходник USB->RS-485 доступны все возможности по управлению через порт RS-485, однако следует помнить, что в данном случае возможны задержки в передаче, и может потребоваться дополнительная настройка.

В общем случае для управления устройством от ПК через интерфейс RS-485 рекомендуется использовать переходник RS-232->RS-485, это позволит обеспечить максимальное качество передачи.

# 5. Режимы работы устройства

## 5.1 Общее описание режимов

Устройство с прошивкой OSM MB может работать в одном из трех режимов, устанавливаемых перемычками SW11-SW12 (таблица 5.1.1).

В режиме драйвера устройство управляется оптически изолированными сигналами «Шаг», «Направление» и «Разрешение». Перемещение на один шаг осуществляется по фронту сигнала на входе «Шаг».

В режиме контроллера устройство управляется с помощью команд по интерфейсу RS-232 или RS-485. Доступны команды задания скорости, ускорения, количества шагов, работы с датчиками. В автоматическом режиме осуществляется вращение двигателя со скоростью, устанавливаемой внешним подключенным потенциометром либо источником сигнала 0-5 Вольт. В программном режиме добавлена возможность вводить и сохранять исполнительную программу в памяти контроллера для независимой работы контроллера в автономном режиме. Программный режим доступен с прошивкой OSM ASCII.

Выбрать нужный режим работы устройства необходимо установкой переключателей SW12 и SW13 в соответствии с таблицей 5.1.

Таблица 5.1.1 – Режимы работы устройства

| Режимы работы | Драйвер | Автоматический | Контроллер |
|---|---|---|---|
| SW11 | OFF | **ON** | OFF |
| SW12 | **ON** | OFF | OFF |

## 5.2 Режим драйвера

Режим драйвера предназначен для управления двигателем с помощью сигналов «Шаг», «Направление» и «Разрешение». В данном режиме на устройство управления подаются шаговые импульсы, каждый из которых вызывает перемещение двигателя на один шаг. Этот режим позволяет использовать устройство в станке с ЧПУ или в паре с любым контроллером, выдающим шаговые импульсы. Использование дифференциальных оптически развязанных входов позволяет достичь высокой гибкости подключения для любых источников сигнала. На рисунке 5.2.1 показана структурная схема оптически изолированного высокочастотного входа STEP:

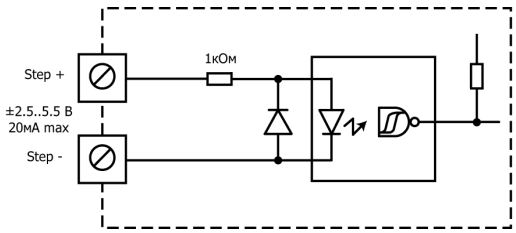
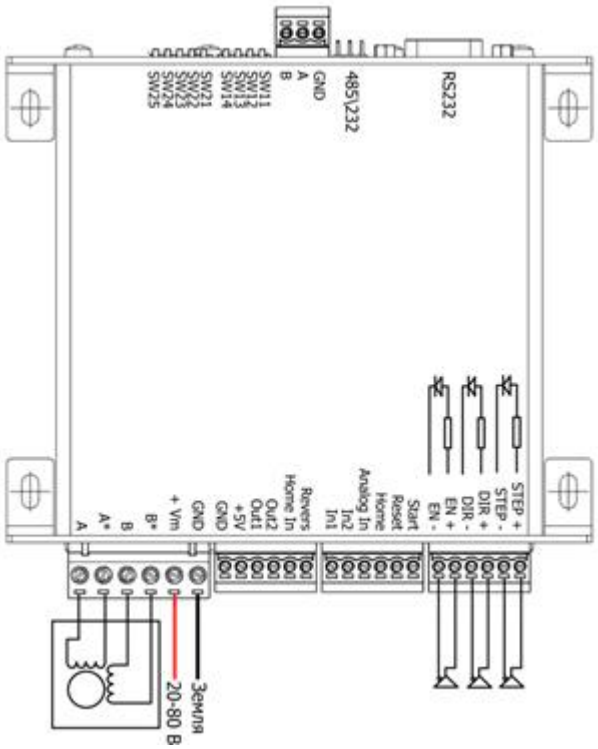
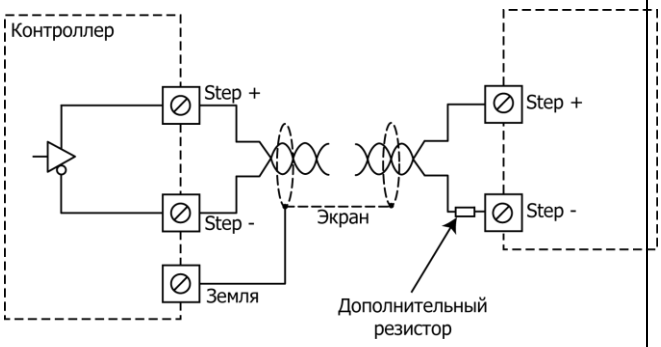
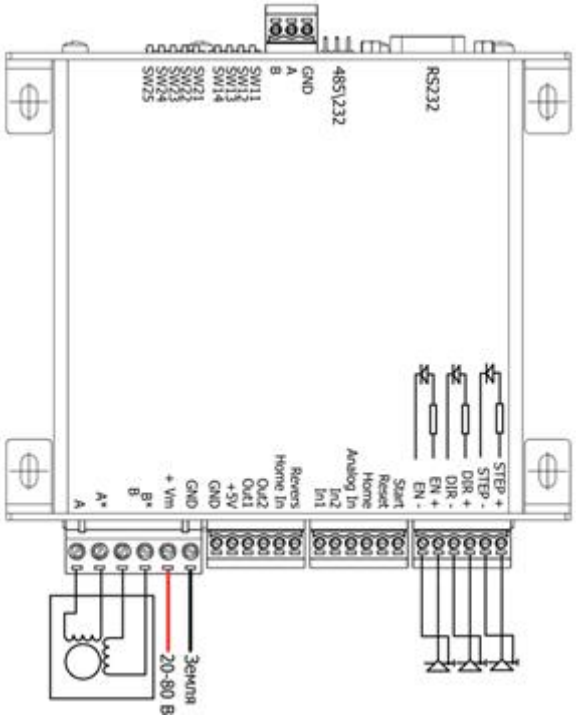
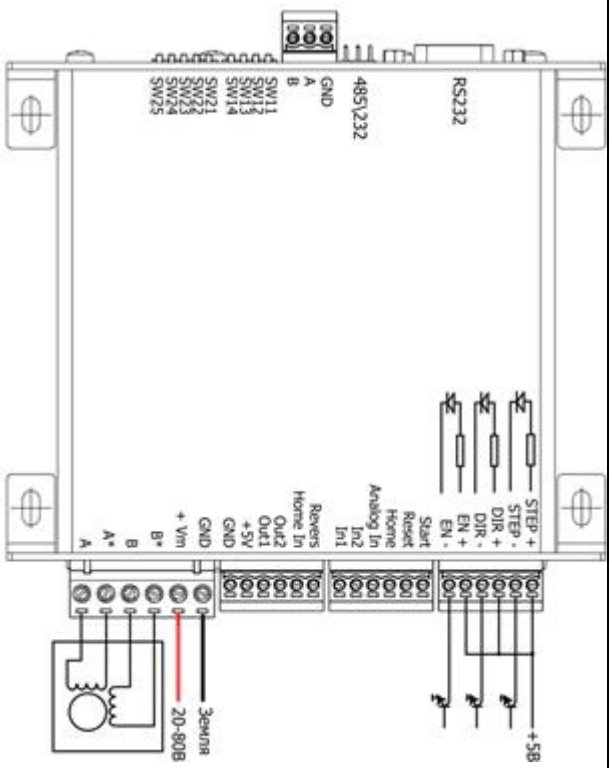
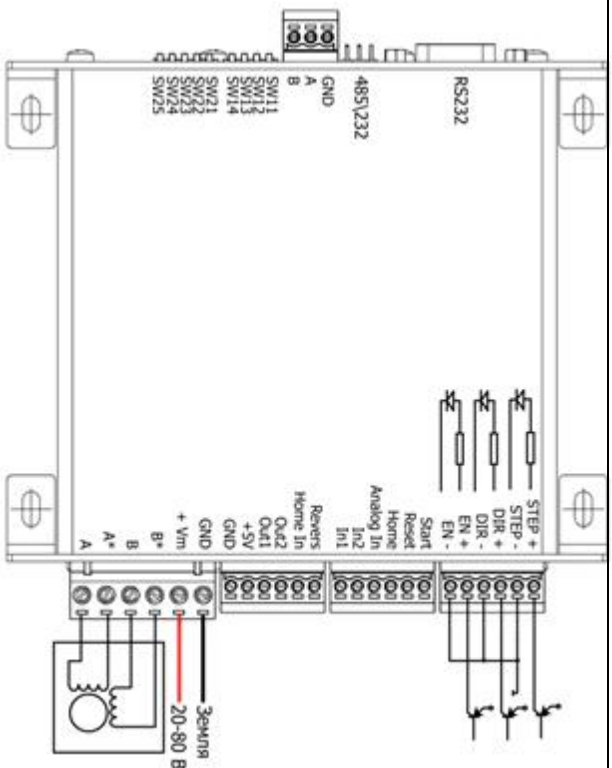



Рисунок 5.2.1 – Структурная схема оптически изолированного высокочастотного входа STEP

Таблица 5.2.1 - Возможные варианты подключения внешних управляющих контроллеров

| 1. Управление от контроллера с дифференциальным выходом. | |
|---|---|
|  | Подключение контроллера с дифференциальным выходом – наиболее помехозащищенное решение. Прямой и инверсный сигналы управляющего контроллера подключаются напрямую к положительным и отрицательным входам контроллера OSM.<br>  <br>Дополнительный резистор требуется, если напряжение сигнала превышает 9В, номинал резистора рассчитывается исходя из указанного тока через опторазвязку. Обратите внимание на необходимость применения витой пары и экранирование сигнального провода. |
| 2. Управление от контроллера TTL-выходом. | |
|  | При подключении управляющего контроллера с TTL-выходом необходимо объединить отрицательные входы контроллера OSM и соединить их с землей управляющего контроллера. Соединение с землей необходимо производить на стороне управляющего контроллера. |

| 3. Управление от контроллера с выходом типа «открытый коллектор» | |
|---|---|
|  | При подключении контроллера с выходом типа «открытый коллектор» необходимо объединить положительные входы контроллера OSM и соединить их с плюсом питания управляющего контроллера. Если используется питание выше +9В, необходимо использовать дополнительный токоограничивающий резистор на каждом входе. |
| 4. Управление от контроллера с выходом типа PNP | |
|  | При подключении контроллера с выходом типа PNP необходимо объединить отрицательные входы контроллера OSM и соединить их с плюсом питания управляющего контроллера. Если используется питание выше +9В, необходимо использовать дополнительный резистор на каждом входе. |

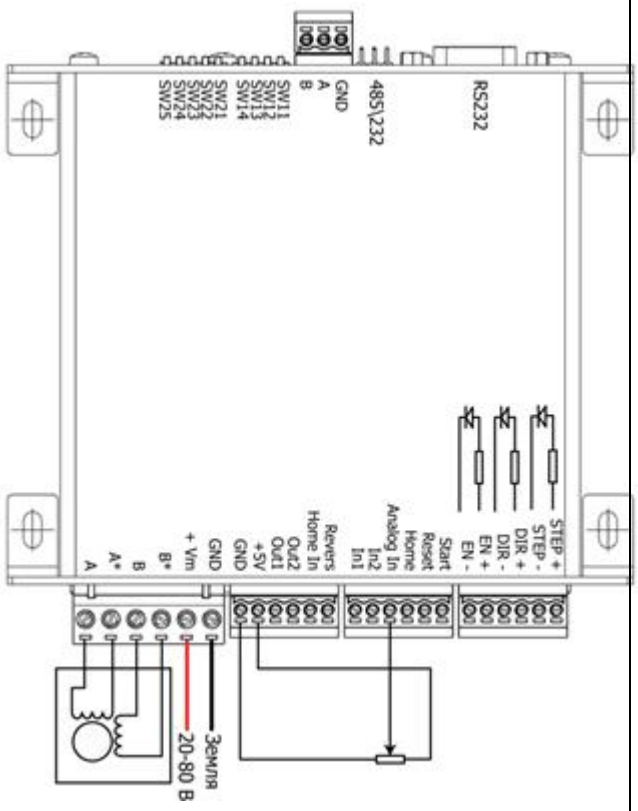
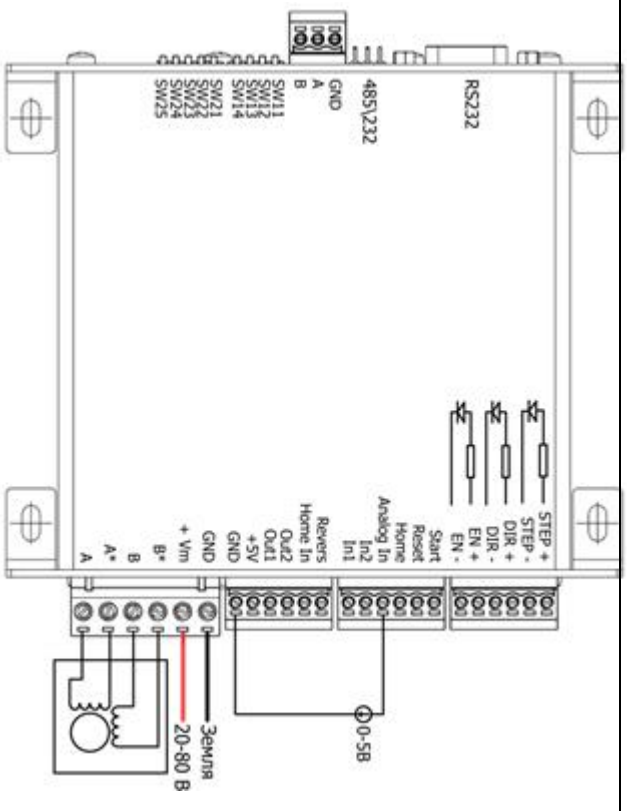
# 5.3 Автоматический режим

В автоматическом режиме происходит вращение двигателя со скоростью, задаваемой внешним аналоговым сигналом. Разрядность АЦП контроллера составляет 8 бит.

В контроллере имеется две возможности работы в автоматическом режиме: работа от внешнего аналогового сигнала 0..5В либо от внешнего потенциометра.

Схемы подключения устройств показаны в таблице 5.3.1.

Таблица 5.3.1 - Схемы подключения устройства в автоматическом режиме

| | |
|---|---|
|  1. |  2. |
| Варианты подключения в автоматическом режиме:<br>1. Подключение внешнего потенциометра.<br>2. Подключение источника напряжения. | |

Зависимость частоты от входного напряжения вычисляется по формуле:

$$f(v) = \begin{cases} 1 + v, & v < 100 \\ 1 + v + \frac{2}{5}(v - 100)^2, & v \geq 100, \end{cases}$$

где v – оцифрованное значение напряжения на аналоговом входе (от 0 до 255).

# 5.4 Режим контроллера

Режим контроллера предназначен для управления двигателем в режиме реального времени по оптоизолированному интерфейсу RS-232 или RS-485. Управление происходит с помощью набора несложных команд по протоколу Modbus. В данном режиме можно:

- Задать бесконечное движение с определенной скоростью.
- Задать поворот на определенное количество шагов с заданной скоростью.
- Установить значение ускорения при движении.
- Задать движение с необходимой скоростью и ускорением до момента срабатывания датчиков.
- Установить сигнал для внешних устройств на выходе OPTO.

При управлении по протоколу Modbus возможна адресация 32-х устройств на линии. Адрес устройства в режиме контроллера выставляется переключателем SW2. Код адреса состоит из пятибитного двоичного кода, выставленного переключателем, плюс единица. Поднятый переключатель соответствует нулю, опущенный – единице. Таким образом, все поднятые переключатели соответствуют двоичному коду числа 0 (`00000` в бинарном виде), и адресу 1 (0 + 1), все опущенные переключатели соответствуют двоичному коду числа 31 (`11111` в бинарном виде), и адресу 32 (31 + 1).

Таблица 5.4.1 – Установка адреса прибора.

| Адрес | SW2.1 | SW2.2 | SW2.3 | SW2.4 | SW2.5 | Адрес | SW2.1 | SW2.2 | SW2.3 | SW2.4 | SW2.5 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | OFF | OFF | OFF | OFF | OFF | 17 | **ON** | OFF | OFF | OFF | OFF |
| 2 | OFF | OFF | OFF | OFF | **ON** | 18 | **ON** | OFF | OFF | OFF | **ON** |
| 3 | OFF | OFF | OFF | **ON** | OFF | 19 | **ON** | OFF | OFF | **ON** | OFF |
| 4 | OFF | OFF | OFF | **ON** | **ON** | 20 | **ON** | OFF | OFF | **ON** | **ON** |
| 5 | OFF | OFF | **ON** | OFF | OFF | 21 | **ON** | OFF | **ON** | OFF | OFF |
| 6 | OFF | OFF | **ON** | OFF | **ON** | 22 | **ON** | OFF | **ON** | OFF | **ON** |
| 7 | OFF | OFF | **ON** | **ON** | OFF | 23 | **ON** | OFF | **ON** | **ON** | OFF |
| 8 | OFF | OFF | **ON** | **ON** | **ON** | 24 | **ON** | OFF | **ON** | **ON** | **ON** |
| 9 | OFF | **ON** | OFF | OFF | OFF | 25 | **ON** | **ON** | OFF | OFF | OFF |
| 10 | OFF | **ON** | OFF | OFF | **ON** | 26 | **ON** | **ON** | OFF | OFF | **ON** |
| 11 | OFF | **ON** | OFF | **ON** | OFF | 27 | **ON** | **ON** | OFF | **ON** | OFF |
| 12 | OFF | **ON** | OFF | **ON** | **ON** | 28 | **ON** | **ON** | OFF | **ON** | **ON** |
| 13 | OFF | **ON** | **ON** | OFF | OFF | 29 | **ON** | **ON** | **ON** | OFF | OFF |
| 14 | OFF | **ON** | **ON** | OFF | **ON** | 30 | **ON** | **ON** | **ON** | OFF | **ON** |
| 15 | OFF | **ON** | **ON** | **ON** | OFF | 31 | **ON** | **ON** | **ON** | **ON** | OFF |
| 16 | OFF | **ON** | **ON** | **ON** | **ON** | 32 | **ON** | **ON** | **ON** | **ON** | **ON** |

## 5.5 Программный режим

Программный режим работы предоставляет возможность автономной работы устройства по заранее записанной в него программе. Последовательность команд составляется на ПК, записывается в контроллер через интерфейс RS-232 и сохраняется в энергонезависимой памяти контроллера. Это позволяет реализовать на базе контроллера полностью автономное устройство, реализующее требуемый пользователю технологический процесс.

Программный режим работы доступен при использовании прошивки OSM ASCII. Документация на использование контроллера в программном режиме распространяется отдельно и полностью соответствует документации на контроллеры типа OSM-17R и OSM-42R. При помощи обновления программного обеспечения можно менять прошивки устройства с версии OSM MB, описанной в настоящем руководстве, на версию OSM ASCII, и наоборот.

## 5.6 Подключение в режиме контроллера и программном режиме

В режиме контроллера и программном режиме к устройству можно подключать два датчика на цифровые входы, аварийный датчик, датчик реверса.

Общие схемы подключения блока в режиме контроллера показаны на рисунках 5.6.1 и 5.6.2.

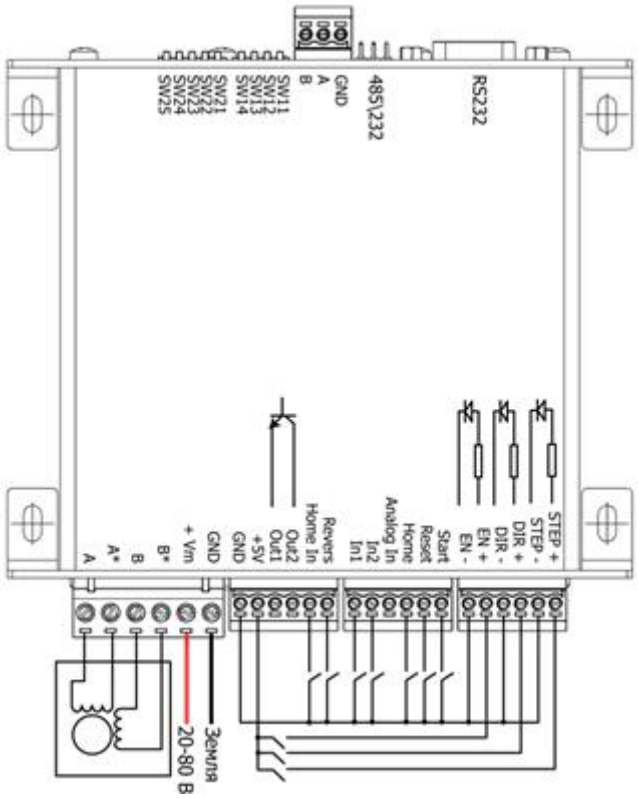



Рисунок 5.6.1 – Схема подключения датчиков типа «сухой контакт» в режиме контроллера

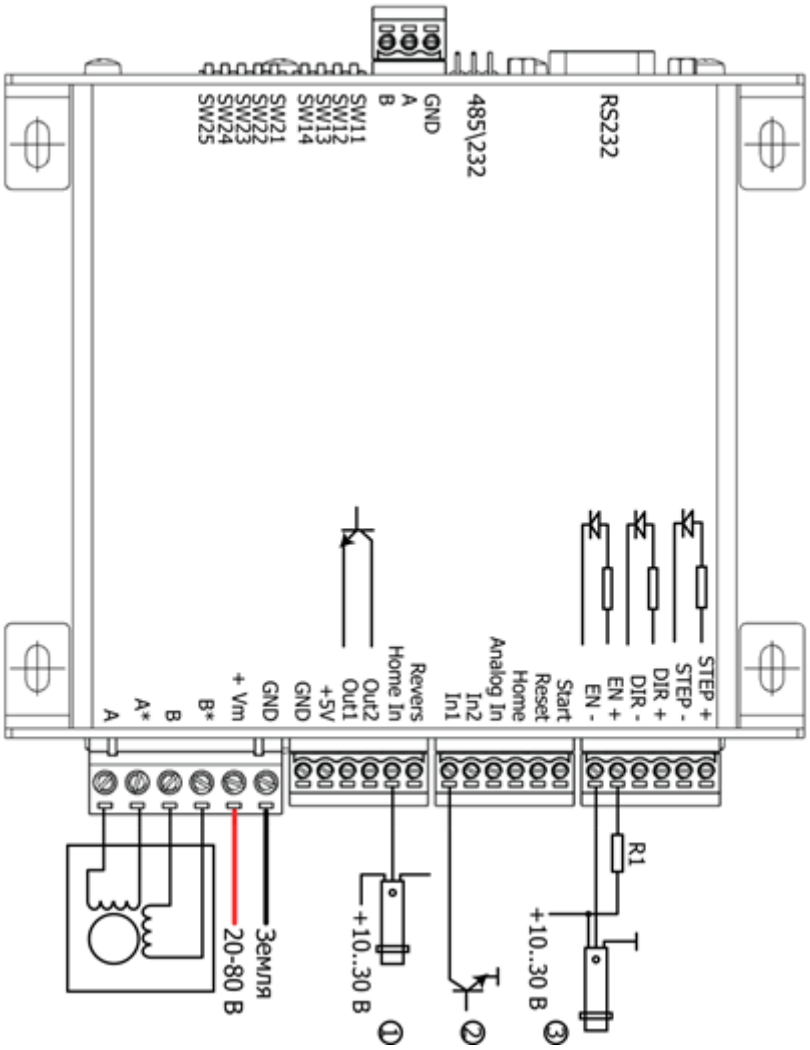
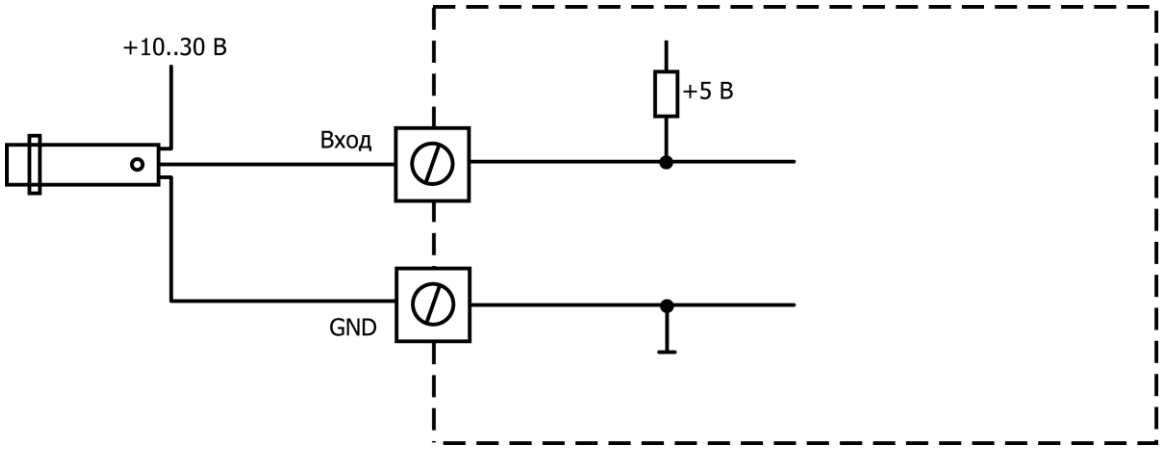
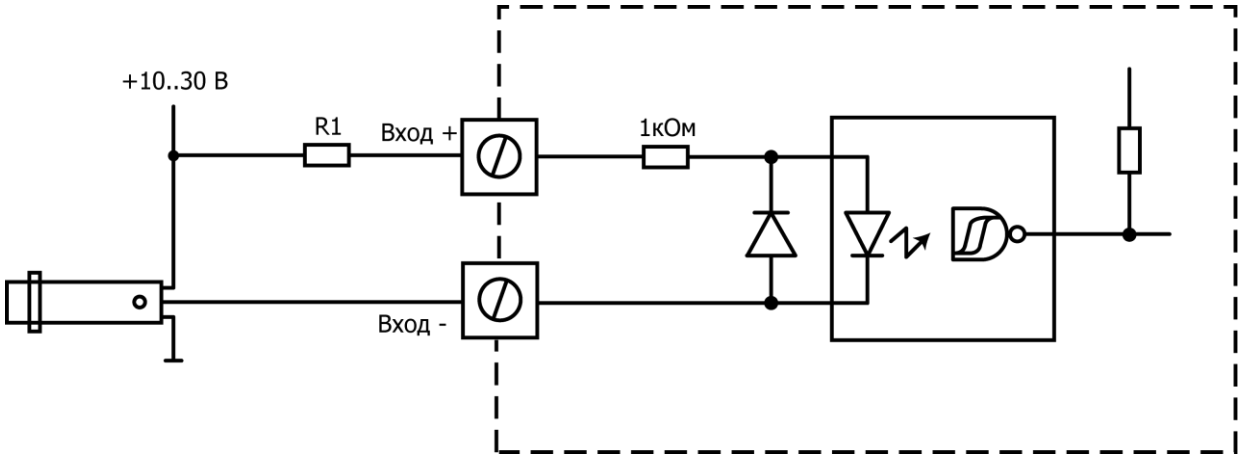



Рисунок 5.6.2 – Примеры подключения PNP, NPN и OK датчиков в режиме контроллера

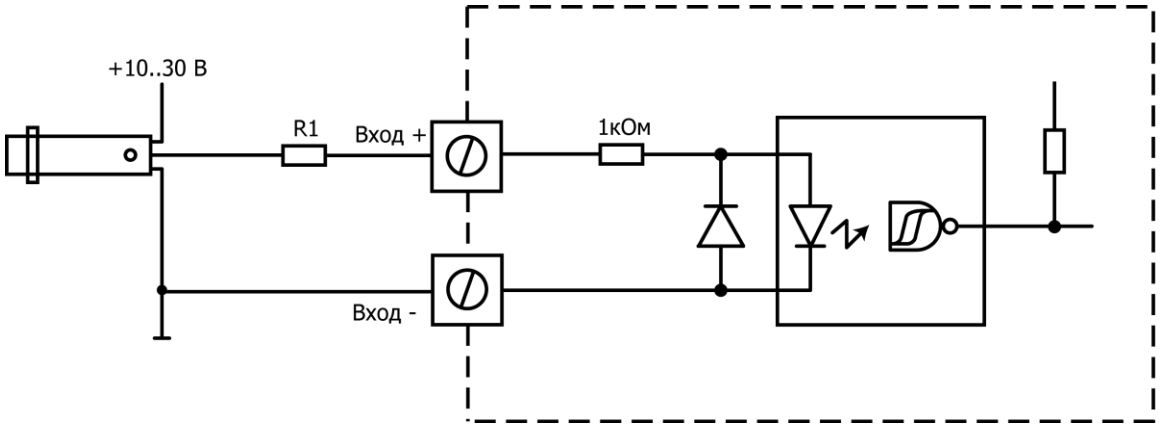
В режиме контроллера и программном режиме возможно подключение различных датчиков. Схемы подключения датчиков к входам контроллера приведены на рисунке 5.6.3.




Подключение датчика NPN к входу с подтяжкой




Подключение датчика NPN к изолированному входу




Подключение датчика PNP к изолированному входу

Рисунок 5.6.3 - Схемы подключения датчиков к входам контроллера

# 5.7 Использование счетчиков и энкодера

В режиме контроллера устройство имеет возможность работы со счетчиками срабатывания внешних датчиков. Иными словами, можно получить число срабатываний датчиков, подключенных к входам EN и Revers. Для этого необходимо включить соответствующее прерывание записью числа 1 в нужный бит регистра INT_EN.

Вход EN имеет только один режим счетчика, который включается записью значения 1 в бит 0 регистра INT_EN. После включения данного прерывания после каждого срабатывания входа EN будет увеличиваться значение регистра EN_counter. При необходимости этот регистр можно обнулить. Следует иметь в виду, что счетчик срабатываний по входу EN имеет размер 16 бит, в отличие от счетчика входа Revers, который имеет размерность 32 бита. Кроме того, частота срабатываний счетчика EN меньше, т.к. она ограничена скоростью срабатывания оптопары гальванической развязки. Максимальная частота по входу EN не должна превышать 500 Гц.

Вход Revers имеет четыре режима работы, конфигурируемых регистром INT_mode.

1. Режим реверсирования. В данном режиме прерывание по входу Revers приводит к смене направления вращения двигателя.
2. Режим счетчика. Прерывание в данном режиме приводит к увеличению счетчика INT_counter.
3. Режим счетчика с защитой от дребезга контактов. Логика работы в данном режиме соответствует режиму 2, но присутствует защита от дребезга. Данный режим применяется при использовании счетчика с механическими контактами.
4. Режим энкодера. Данный режим используется для подключения энкодера. Канал А подключается к входу Revers, канал В – к входу STEP. В зависимости от направления вращения, прерывание по сигналу энкодера приводит к увеличению или уменьшению регистра Encoder_position.

Вход Revers является более высокоскоростным, но не имеет гальванической развязки.

# 6. Управление устройством по протоколу Modbus

## 6.1 Описание протокола

Modbus - коммуникационный протокол, основанный на клиент-серверной архитектуре. Широко используется для передачи данных последовательных линиях связи, основанных на интерфейсах RS-485, RS-422, RS-232. В начале развития применялся интерфейс RS-232, как один из наиболее простых промышленных интерфейсов для последовательной передачи данных. В настоящее время протокол часто используется поверх интерфейса RS-485. Широкая распространенность протокола Modbus, обусловленная его простотой и надежностью, позволяет легко интегрировать устройства, поддерживающие Modbus, в единую сеть.

Основной особенностью протокола является наличие в сети одного ведущего устройства - Master. Только ведущее устройство может опрашивать остальные устройства сети, которые являются ведомыми (Slave). Подчиненное устройство не может самостоятельно инициировать передачу данных или запрашивать какие-либо данные у других устройств, работа сети строится только по принципу «запрос-ответ». Мастер может так же выдать широковещательный запрос, адресованный всем устройствам в сети, в таком случае ответное сообщение не посылается.

Существует три типа протокола Modbus: Modbus ASCII, Modbus RTU и Modbus TCP. Устройства Onitex поддерживают протокол Modbus RTU.

Пакет данных в Modbus выглядит следующим образом:

**Адрес | Код функции | Данные | Контрольная сумма.**

**Адрес** - это поле, содержащее номер устройства. Каждое устройство в сети должно иметь уникальный адрес. Устройство отвечает только на те запросы, которые поступают по его адресу, что позволяет избежать конфликтов. При этом ведомое устройство в своем ответе также посылает поле Адрес, кроме широковещательного запроса.

**Код функции** содержит номер функции Modbus (о функциях будет сказано ниже). Функция может запрашивать данные или давать команду на определенные действия. Коды функций являются числами в диапазоне от 1 до 127. Коды от 128 до 255 зарезервированы для пересылки в ответном сообщении информации об ошибках.

В поле **Данные** содержится информация, которую передает мастер ведомому, либо наоборот, в случае ответного сообщения. Длина этого поля зависит от типа передаваемых данных.

Поле **Контрольная сумма** является важным элементом протокола: в нем содержится информация, необходимая для проверки целостности сообщения и отсутствия ошибок передачи. В контроллере используется контрольная сумма CRC16.

## 6.2 Типы данных Modbus

Протоколом Modbus определено несколько типов данных.

| | Тип данных | Тип доступа |
|---|---|---|
| Дискретные входы (Discrete Inputs) | один бит | только чтение |
| Регистры флагов (Coils) | один бит | чтение и запись |
| Регистры ввода (Input Registers) | 16-битное слово | только чтение |
| Регистры хранения (Holding Registers) | 16-битное слово | чтение и запись |

В устройствах OSM используется тип Holding Registers.

## 6.3 Функции Modbus

Существует три типа функций:

- Стандартные. Описание этих функций опубликовано и утверждено Modbus-IDA. Эта категория включает в себя как опубликованные, так и свободные в настоящее время коды.
- Пользовательские. Два диапазона кодов (от 65 до 72 и от 100 до 110), для которых пользователь может создать произвольную функцию.
- Зарезервированные. В эту категорию входят коды функций, не являющиеся стандартными, но уже используемые в устройствах, производимых различными компаниями. К этим кодам относятся 9, 10, 13, 14, 41, 42, 90, 91, 125, 126 и 127.

При использовании режима Modbus RTU сообщение начинается с интервала тишины, равного времени передачи 3.5 символов, при заданной скорости обмена. Первым полем передается адрес устройства. Вслед за последним передаваемым символом также следует интервал тишины продолжительностью не менее 3.5 символов. Новое сообщение может начинаться после этого интервала. Фрейм сообщения передаётся непрерывно. Если интервал тишины продолжительностью 1.5 символа возник во время передачи фрейма, принимающее устройство должно игнорировать этот фрейм как неполный. Если новое сообщение начнется раньше интервала 3.5 символа, принимающее устройство воспримет его как продолжение предыдущего сообщения. В этом случае устанавливается ошибка CRC (несовпадение контрольной суммы).

Для чтения значений используются функции с кодами 1—4 (0x01–0x04):

- 1 (0x01) — чтение значений из нескольких регистров флагов (Read Coil Status)
- 2 (0x02) — чтение значений из нескольких дискретных входов (Read Discrete Inputs)
- 3 (0x03) — чтение значений из нескольких регистров хранения (Read Holding Registers)
- 4 (0x04) — чтение значений из нескольких регистров ввода (Read Input Registers)

В контроллерах OSM используются функции `Read Holding Registers`.

Запрос состоит из адреса первого элемента таблицы, значение которого требуется прочитать, и количества считываемых элементов. Адрес и количество данных задаются 16-битными числами, старший байт каждого из них передается первым.

В ответе передаются запрошенные данные. Количество байт данных зависит от количества запрошенных элементов. Перед данными передается один байт, значение которого равно количеству байт данных.

Запись одного значения происходит при помощи следующих функций:

- 5 (`0x05`) — запись значения одного флага (`Force Single Coil`)
- 6 (`0x06`) — запись значения в один регистр хранения (`Preset Single Register`)

Команда состоит из адреса элемента (2 байта) и устанавливаемого значения (2 байта). Если команда выполнена успешно, ведомое устройство возвращает копию запроса.

Запись нескольких значений задается функциями:

- 15 (`0x0F`) — запись значений в несколько регистров флагов (`Force Multiple Coils`)
- 16 (`0x10`) — запись значений в несколько регистров хранения (`Preset Multiple Registers`)

В устройствах OSM используются функции `Preset Single Register` и `Preset Multiple Registers`.

Команда состоит из адреса элемента, количества изменяемых элементов, количества передаваемых байт устанавливаемых значений и самих устанавливаемых значений. В ответе ведомый передает начальный адрес и количество изменённых элементов.

В случае ошибки ведомое устройство возвращает ее код.

- 01 – неверно указана функция.
- 02 – ошибка адреса данных.
- 03 – ошибка значения: величина, содержащаяся в поле данных, недоступна устройству.
- 04 – ошибка обращения.
- 05 – обработка запроса требует много времени.
- 06 – устройство занято и не может выполнить программу.
- 07 – устройство не может выполнить программную функцию.

# 6.4 Управление контроллерами OSM

Управление контроллером осуществляется путем записи значений в регистры Modbus. В контроллере предусмотрены три типа регистров: 8 бит, 16 бит и 32 бита. Все регистры, кроме счетчика текущей позиции, являются беззнаковыми. Регистры размером 32 бита образованы последовательной записью в два регистра 16 бит при помощи функции №16.

Все регистры доступны как для чтения, так и для записи, однако запись в регистр `Speed_Current` не разрешена, так как это может нарушить режимы ускорения.

Управление двигателем заключается в записи параметров движения и команды. При инициализации контроллера необходимо записать в него текущие параметры настройки, такие, как ток простоя, коэффициент дробления и другие, если требуются значения, отличные от установленных по умолчанию.

Далее необходимо записать параметры движения: скорость, и, если требуется, ускорение. Если ускорение не требуется, то в регистр ускорения надо записать значение «`0`». Кроме того, если контроллер должен совершить определенное число шагов, необходимо записать это значение в регистр `Steps_Number`.

После этого можно подать команду движения, записав ее номер в регистр `Command` . Команды движения приведены в отдельной таблице. Существуют четыре основных типа команд движения.

1. Бесконечное движение. Двигатель движется до подачи команды остановки.
2. Движение на заданное число шагов. Двигатель отрабатывает заданное число шагов и ждет следующей команды.
3. Движение до срабатывания датчика без ограничения по времени. Двигатель движется до тех пор, пока не сработает датчик, определенный в команде (см. список команд).
4. Движение до датчика с ограничением числа шагов (двигатель остановится в одном из двух случаев: срабатывание датчика или отработка заданного числа шагов).

В случае если движение должно быть ограничено определенным числом шагов, это значение записывается в регистр `Steps_Number`.

Таблица 6.4.1 – Регистры контроллера

| Регистры хранения (Holding) 8 бит бит без знака (unsigned int08) | | | |
|---|---|---|---|
| Адрес | Название | Описание | Значения |
| 0 | Adress | Адрес устройства | Адрес устройства 8 бит |
| 1 | Baud_Rate_Index | Скорость обмена | 0 – 1200 кбит/с<br>1 – 2400кбит/с<br>2 – 4800 кбит/с<br>3 – 9600 кбит/с<br>4 – 19200 кбит/с<br>5 – 38400 кбит/с<br>6 – 57600 кбит/с<br>7 – 115200 кбит/с<br>По умолчанию 57600 кбит/с |
| 2 | RTS_Delay | Задержка RTS | Параметр настройки задержек переключателя направления передачи |
| 3 | Enable | Включение тока в обмотках | 0 – двигатель обесточен<br>1 – двигатель включен |
| 4 | Direction | Направление | 0 – CW<br>1 – CCW<br>По умолчанию – 0. |
| 5 | Command | Номер команды | См. список команд |
| 6 | Output | Состояние выхода | 0 – выход выключен<br>1 – выход включен |
| 7 | Microstep | Микрошаг | Коэффициент дробления шага:<br>1 – полный шаг<br>2 – 1/2 шага<br>4 - 1/4 шага<br>16 – 1/16 шага<br>По умолчанию – 1. |
| 8 | Inputs | Состояние входов | 5 бит состояния входов (см. табл. 6.4.3) |
| 9 | Sleep_Current | Ток в режиме простоя | Ток в простое в процентах от полного тока двигателя, от 0 до 100% с шагом 1%:<br>0 – 0% (двигатель в простое обесточен)<br>1 – 1%<br>2 – 2%<br>…<br>50 – 50%<br>…<br>98 – 98%<br>99 – 99%<br>100 – 100% (ток в простое не снижается)<br>По умолчанию – 50% |

| | | | |
|---|---|---|---|
| 10 | INT_EN | Разрешение прерываний по входам | Прерывания по входам разрешаются установкой значения 1 в соответствующий бит прерывания.<br>Бит 0 – прерывание по входу EN<br>Бит 1 – прерывание по входу Revers<br>Запись значения 0 – все прерывания выключены, запись значения 3 – оба прерывания включены.<br>По умолчанию значения битов прерывания 0, прерывания отключены. |
| 11 | INT_MODE | Режим работы прерывания по входу Revers. | Устанавливает режим работы прерывания:<br>0 – реверсирование двигателя по изменению сигнала входа Revers<br>1 – режим счетчика. По прерыванию увеличивается значение регистра INT1_Counter.<br>2 – режим счетчика с защитой от дребезга контактов. Используется в случае механического контакта на входе Revers.<br>3 – режим энкодера (второй канал подключается на вход STEP). |
| 12 | SYSTEM_ID | Текущая версия прошивки | Параметр только для чтения. Содержит текущую версию прошивки. Доступно, начиная с версии 10 |
| Регистры хранения (Holding) 16 бит без знака (unsigned int16) | | | |
| Адрес | Название | Описание | Значения |
| 16384 | UART_Delay | Задержка приемопередатчика | Служебный параметр |
| 16385 | Speed | Скорость двигателя в Гц | Скорость двигателя от 1 до 20000 Гц |
| 16386 | Start_Speed | Стартовая скорость | Стартовая скорость двигателя |
| 16387 | Accel | Ускорение | Величина ускорения от 0 до 3000 |
| 16388 | End_Speed | Конечная скорость | Конечная скорость при замедлении, должна быть меньше рабочей скорости |
| 16389 | Current | Ток двигателя | Ток двигателя в миллиамперах. Значения от 0 до 1700 для модели OSM-17 и от 0 до 4000 для модели OSM-42<br>По умолчанию – 0 мА. |
| 16390 | Speed_Current | Текущая скорость | Текущая скорость двигателя в каждый момент времени, параметр только для |

| | | | |
|---|---|---|---|
| | | | чтения |
| 16391 | Steps_Bef_Decel | Число шагов при замедлении | Число шагов, определяющее длительность замедления |
| 16392 | Sleep_Time | Время до перехода в спящий режим | Время до перехода в спящий режим в миллисекундах, от 0 до 2000мс. Если после последнего шага прошло более Sleep_Time мс, ток двигателя уменьшается до величины, определенной в параметре Sleep_Current.<br>По умолчанию – 2000мс |
| 16393 | EN_counter | Счетчик прерываний по входу EN | Значение регистра увеличивается при срабатывании прерывания по входу EN. Прерывание по входу EN должно быть включено в регистре INT_EN. |
| Регистры хранения (Holding) 32 бит без знака (unsigned int32) | | | |
| Адрес | Название | Описание | Значения |
| 32768 | Steps_Number | Число шагов | Число шагов, используемое в командах движения |
| 32772 | Steps_Counter | Число шагов до завершения команды | Счетчик, показывающий оставшееся число шагов до завершения вращения при использовании команд с заданным числом шагов |
| 32776 | INT1_counter | Счетчик прерываний по входу Revers | Значение регистра увеличивается при срабатывании прерывания по входу Revers. Прерывание по входу Revers должно быть включено в регистре INT_EN. |
| Регистры хранения (Holding) 32 бит со знаком (signed int32) | | | |
| Адрес | Название | Описание | Значения |
| 32770 | Position | Позиция двигателя | Счетчик текущей позиции, при повороте на 1 шаг увеличивается или уменьшается на единицу в зависимости от направления вращения. |
| 32778 | Encoder_position | Позиция, считанная с энкодера | Считывает текущую позицию с инкрементального энкодера, канал А подключается к входу Revers, канал В – к входу STEP. |

Таблица 6.4.2 – Команды движения

| Номер | Команда | Датчик остановки | Число шагов | Описание команды |
|---|---|---|---|---|
| 0x00 | STOP | – | - | Остановка двигателя |
| 0x01 | MOVE | – | - | Бесконечное движение. |
| 0x02 | MOVE N | – | Steps_Number | Движение Steps_Number шагов |
| 0x03 | MOVE_STEP | STEP | - | Движение до срабатывания датчика, подключенного ко входу STEP. |
| 0x04 | MOVE_DIR | DIR | - | Движение до срабатывания датчика, подключенного ко входу DIR. |
| 0x05 | ADC_SPEED | – | - | Команда устанавливает текущее значение скорости в соответствии со значением напряжения на аналоговом входе 0..5В от 1Гц до 10кГц. |
| 0x06 | WL | – | - | Системная команда |
| 0x07 | WH | – | - | Системная команда |
| 0x08 | REVERS | – | - | Реверс |
| 0x09 | MOVE_IN1 | IN1 | - | Движение до срабатывания датчика, подключенного ко входу In1. |
| 0xA | MOVE_IN2 | IN2 | - | Движение до срабатывания датчика, подключенного ко входу In2. |
| 0xB | FIND_HOME | HomeIn | - | Поиск начального положения (движение до сигнала датчика HomeIn) |
| 0xC | RESET | – | - | Перезагрузка контроллера. Все параметры сбрасываются, движение прекращается. |
| 0xD | MOVE_IN1_N | IN1 | Steps_Number | Движение до срабатывания датчика, подключенного ко входу In1, но не более Steps_Number шагов. |
| 0xE | MOVE_IN2_N | IN2 | Steps_Number | Движение до срабатывания датчика, подключенного ко входу In1, но не более Steps_Number шагов. |
| 0xF | FIND_HOME_N | HomeIn | Steps_Number | Поиск начального положения (движение до сигнала датчика HomeIn, но не более Steps_Number шагов). |
| 0x10 | MOVE_STEP_N | STEP | Steps_Number | Движение до срабатывания датчика, подключенного ко входу STEP, но не более Steps_Number шагов. |
| 0x11 | MOVE_DIR_N | DIR | Steps_Number | Движение до срабатывания датчика, подключенного ко входу DIR, но не более Steps_Number шагов. |
| 0x12 | MAKE_STEP | – | - | Сделать один шаг. |
| 0x13 | SAVE_PARAMETERS | – | - | Сохранить параметры движения, микрошага и тока в энергонезависимую память |

Таблица 6.4.3 – Регистр входных сигналов

| Бит | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Значение | Step | En | Dir | In2 | In1 | HomeIn |

## 6.5 Примеры использования протокола.

Приведем пример использования протокола Modbus для управления двигателем с током обмоток 1.7А при помощи контроллера OSM-88RA. Предположим, двигатель должен будет найти начальное положение механизма, дойдя до датчика, подключенного к входу `HomeIn`, однако по условиям эксплуатации механизма, если число шагов превысит 15000, двигатель должен так же остановиться. Движение должно происходить с дроблением 1/16, с полным током, на скорости 4000 Гц. После движения двигатель должен быть обесточен.

Предположим, контроллер только что включен и параметры не загружены. Для реализации данной операции необходимо:

1. Задать значение тока двигателя 1.7А записью значения `1700` в регистр `Current` при помощи функции записи одного регистра.
2. Задать значение дробления шага записью числа `16` в регистр `Microstep`.
3. Параметры тока покоя 50% и времени перехода в режим покоя 2000мс нас устраивают, поэтому их можно оставить без изменений.
4. Задать скорость двигателя записью значения `4000` в регистр `Speed`.
5. Задать значение максимального числа шагов записью значения `15000` в регистр `Steps_Number`. Данный регистр является 32-битным, поэтому запись необходимо производить функцией записи в несколько регистров (№16).
6. Задать направление движения записью `1` в регистр `Direction`.
7. Подать команду движения до сигнала `HOME` (номер команды `0xF, FIND_HOME_N`) записью ее кода в регистр `Command`. Двигатель начнет движение.
8. Во время движения можно контролировать число шагов, оставшихся до остановки двигателя при остановке по достижению заданного числа шагов, в регистре `Steps_Counter`. Если же двигатель остановится по датчику, можно проверить содержимое регистра `Inputs` – бит сработавшего датчика будет равен 0.
9. После остановки двигатель должен быть обесточен. Это можно сделать записью числа `0` в регистр `Enable`. Можно сделать это и автоматически, записав `0` в регистр `Sleep_Current`, в таком случае контроллер сам снимет напряжение с обмоток по прошествии `Sleep_time` секунд с момента последнего шага. Однако при этом следует помнить, что в дальнейшем может потребоваться изменение этого параметра, так что в некоторых случаях проще записать 0 в регистр `Enable`.

В результате данных команд двигатель нашел исходное положение. Можно продолжить работу, к примеру, дав двигателю команду вращения в противоположном направлении с ускорением 1000 шаг/с на 567812 шага с остановкой в случае срабатывания датчика In1. Поскольку параметры контроллера уже загружены, можно ограничиться изменением параметров вращения (если, конечно, другие параметры не должны быть изменены).

Для реализации данного алгоритма необходимо:

1. Изменить направление вращения записью `0` в регистр `Direction`.
2. Если обмотки были обесточены, включить их записью числа 1 в регистр `Enable`.
3. Опционально можно обнулить счетчик позиции записью 0 в регистр `Position`.
4. Записать значение нового числа шагов `567812` в регистр `Steps_Number`.
5. Записать значение стартовой скорости `200` в регистр `Start_Speed`.
6. Записать значение максимальной скорости `10000` в регистр `Speed`.
7. Задать ускорение `1000` в регистре `Accel`.
8. Дать команду `MOVE_IN1_N` записью ее кода в регистр `Command`.

Теперь двигатель сделал 567812 шагов, либо остановился раньше по сигналу с датчика. Общее число шагов можно узнать в регистре `Position`. Предположим, двигатель должен снова сделать 567812 шагов в случае, если датчик не сработал, а в случае, если датчик сработал, вернуться в начальное положение. Для реализации данного алгоритма необходимо:

1. Считать значение входов из регистра `Inputs` либо число пройденных двигателем шагов из регистра `Position`.
2. Если из данных п.1 можно сделать вывод, что двигатель остановился по датчику, изменить направление записью `0` в регистр `Direction` и дать команду `FIND_HOME` записью `0xB` в регистр `Command`.
3. Если же двигатель сделал полностью `567812` шагов, можно просто подать команду `MOVE_IN1_N`, так как все остальные параметры записаны ранее и не изменялись.

Таким образом, протокол управления OSM MB является гибким для решения большого количества задач при управлении с центрального контроллера.

# 6.6 Использование программы Modbus Terminal

Управлять устройством возможно с помощью программы Onitex Modbus Terminal. Данная программа позволяет быстро освоить основные принципы управления устройствами OSM MB по протоколу Modbus RTU, проверить корректную работу устройства и быстрее написать собственное программное обеспечение, используя SDK.

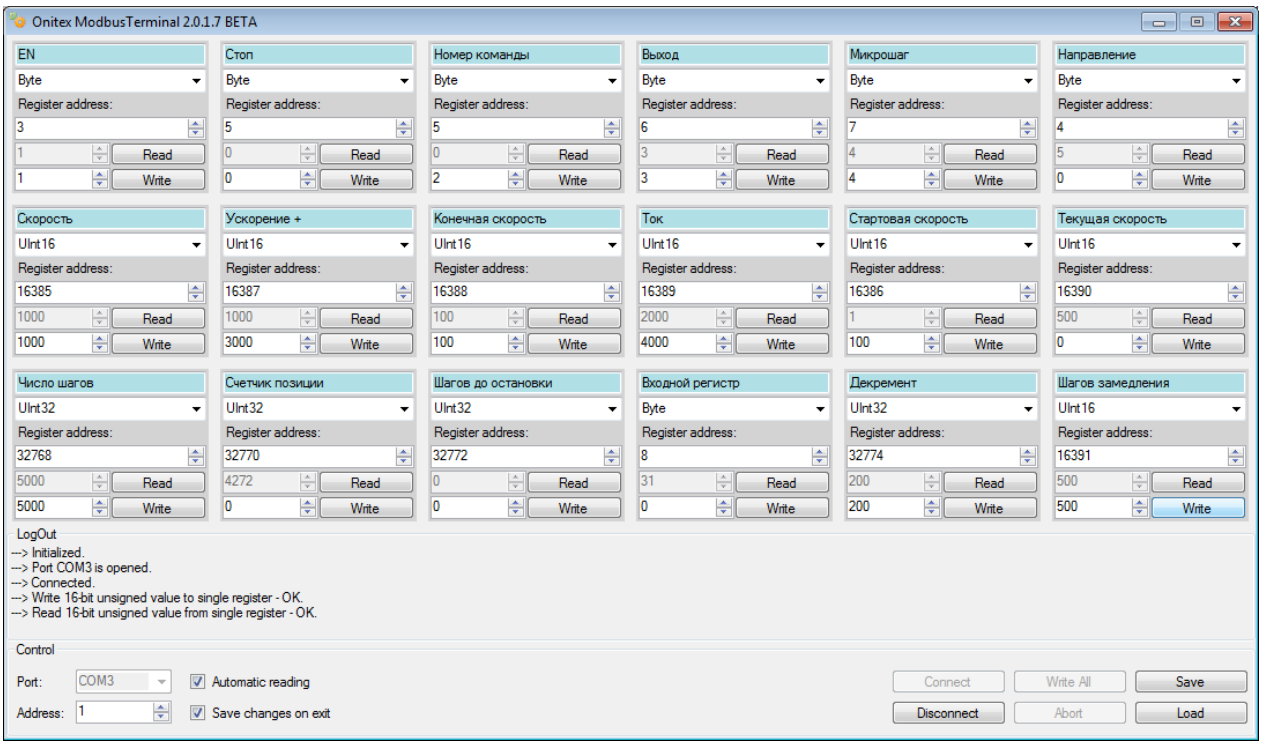



Рисунок 6.6.1 – Окно программы.

Программа представляет собой карту регистров, каждому из которых можно задать адрес, тип значения и название. В каждом регистре имеется возможность чтения и записи значения. В окне «LogOut» можно наблюдать вывод лога по результатам каждого действия, в т. ч. и возникшие ошибки.

Для начала работы с программой необходимо установить адрес порта ПК и адрес целевого устройства, и нажать кнопку «Connect». После этого можно производить чтение и запись в требуемые регистры. Устройство спроектировано таким образом, что запись неправильных значений не приводит к возникновению неисправности, что упрощает отладку и изучение устройства. Тем не менее, необходимо внимательно следить, чтобы ток, заданный посредством записи значения в регистр `16389` (`Current`) не превышал ток двигателя, иначе двигатель может быть испорчен.

Стандартная последовательность записи параметров в регистры указана в примере в главе 6.5. Во время движения двигателя можно считывать любые параметры, такие, как число шагов до остановки, счетчик шагов, а так же изменять команды и параметры движения. При необходимости можно сохранить названия и адреса используемых регистров кнопкой «Save».

Программа написана с использованием OsmModbusDriver_SDK и может служить примером использования SDK.

# 7. Использование SDK

Для быстрого добавления возможности управления устройством с помощью собственного программного обеспечения для данного устройства предусмотрен комплект разработчика OsmModbusDriver_SDK.

Комплект разработчика предназначен для быстрого добавления функций управления устройством в проект программного обеспечения разрабатываемого под платформу .NET Framework 3.5. Таким образом, поддерживаются такие среды разработки программного обеспечения как Microsoft Visual Studio 2008, Microsoft Visual Studio 2010, SharpDevelop, MonoDevelop, а также Borland Developer Studio.

В состав OsmModbusDriver_SDK входит:

- библиотека `OsmModbusDriver.dll` под платформу .NET Framework 3.5;
- полная документация в формате Microsoft Compiled HTML Help;
- пример использования библиотеки в виде готового проекта с задокументированным исходным текстом на языке C# в формате Visual Studio 2008 (совместим также с Visual Studio 2010 и SharpDevelop).

Библиотека `OsmModbusDriver.dll` использует возможности свободной библиотеки `NModbus.dll` для поддержки протокола, но, помимо стандартных операций синхронного чтения/записи регистров, предоставляет более удобный для управления данным устройством API и позволяет легко задействовать в программном проекте следующий набор функциональных возможностей:

- Поддержка стандартной реализации протокола Modbus RTU Serial Master, используемой в устройствах OSM MB.
- Функции для чтения/записи регистров с обязательной явной типизацией данных для минимизации возможности появления ошибок.
- Автоматическое преобразование различных типов целочисленных данных (`byte`, `unsigned int16`, `unsigned int32`, `signed int32`) для корректной записи в 16 битные регистры устройства (аналогично реализована функция чтения).
- Автоматическое обеспечение многопоточности на уровне библиотеки:
    - синхронизация с основным потоком программы обеспечивается со стороны библиотеки;
    - операции работы с портом, операции чтения из регистром и операции записи в регистры выполняются в отдельных потоках, что приводит к существенному повышению производительности даже на однопроцессорных системах;
    - реализованный контроль над потоками чтения/записи позволяет отловить момент некорректной обработки операций чтения/записи со стороны устройства (например, в случае отключения питания) и произвести отмену операции без задержек и зависаний в работе программы;

    - возможность выставления времени таймаута от отправки запроса до ответа контроллера с выходом из операции по таймауту.
- Автоматическое определение доступных для использования портов в системе.
- Улучшенная работа с серийным портом.
- Использование кодов ошибок принятых в стандарте Modbus RTU.
- Возможность ведения лога, с выводом сообщений на форму и/или в файл.

Полную документацию на SDK Onitex, а так же библиотеки и программное обеспечение, можно скачать с сайта производителя.

# 8. Обновление встроенного программного обеспечения

Контроллеры OSM имеет функцию удаленного обновления встроенного ПО, что позволяет пользователю скачивать с сайта производителя свежую версию программного обеспечения с дополнительной функциональностью, а также удаленно исправлять возможные ошибки в работе. Кроме того, существует возможность расширения функциональности устройства по договоренности с производителем. Обновление встроенного ПО происходит с помощью утилиты OSM Update.

Программа OSM Update версии Portable не требует установки. Для установки стандартной программы OSM Update необходимо запустить установщик Setup. После запуска программы необходимо выбрать порт, установить скорость 57600 кбит/с, а также выбрать скачанный файл прошивки контроллера. Контроллер должен быть подключен к ПК.

Внимание! Обновление встроенного ПО возможно только через порт RS-232. Перед обновлением необходимо перевести контроллер в режим работы через RS-232, обновление через RS-485 невозможно.

Для входа в режим обновления программного обеспечения контроллер необходимо установить в автоматический режим и замкнуть на землю вход `FindHome`. После подачи питания контроллер будет готов к загрузке. Загрузка происходит после нажатия кнопки «Update». После загрузки обновленного ПО необходимо отсоединить вход `FindHome` и перезагрузить контроллер включением-выключением питания или замыканием контакта `Reset`. После этого контроллер будет готов к работе.

С помощью данной программы можно записать в контроллер прошивку OSM ASCII, что позволит использовать контроллер в качестве универсального контроллера со встроенным ПЛК. При необходимости управления через Modbus можно в любой момент прошить контроллер прошивкой OSM MB.

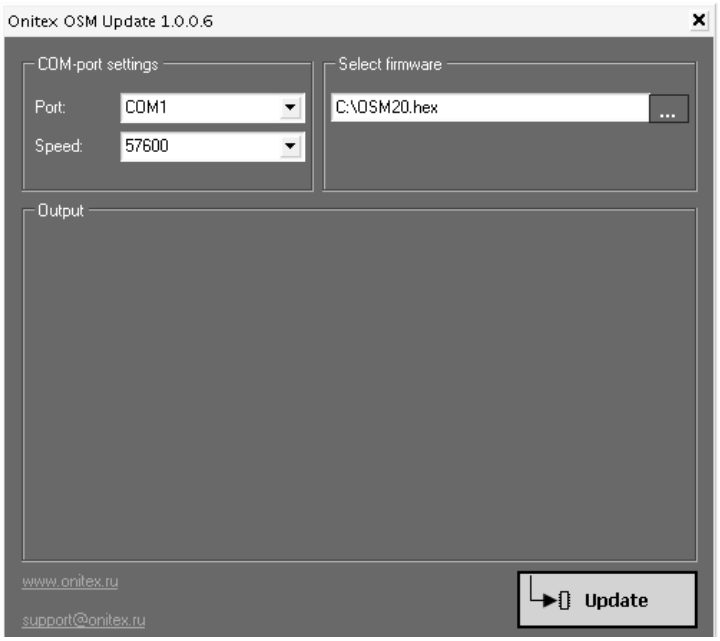



Рисунок 8.1 – Окно программы OSM Update

Внимание! Выключение питания во время процесса загрузки обновлений может повлечь за собой сбой программного обеспечения и неработоспособность контроллера. Выключение питания в процессе обновления и вызванная этим неисправность не является гарантийным случаем. Необходимо следить за правильным выбором файла обновления, так как прошивка от другого устройства может привести к нестабильной работе контроллера или вызвать неработоспособность изделия.

# 9. Гарантийные обязательства и техническая поддержка

Фирма-производитель гарантирует качество устройства и соответствие его характеристик заявленным, и дает гарантию пять лет на данное устройство, при условии соблюдения правил эксплуатации, изложенных в настоящем руководстве.

Гарантия не распространяется в следующих случаях:

- Попытки самостоятельного ремонта или модификации устройства.
- Наличие механических повреждений.
- Повреждения в результате действия непреодолимой силы.
- Нарушение правил и условий эксплуатации.
- Повреждения, вызванные некорректной работой внешних устройств (например, выходом из строя источника питания).

Производитель не берет на себя ответственность за повреждения стороннего оборудования, вызванные корректной или некорректной работой данного устройства.

Задать вопросы, высказать пожелания, а так же получить технические консультации, вы можете по адресу электронной почты support@onitex.ru, или по телефону (812) 928-65-54

Отметки отдела выходного контроля предприятия-изготовителя.

Штамп ОТК Инженер ОТК_____

Дата проверки _____

Серийный номер изделия_____

Настоящим подтверждается, что изделие прошло выходной контроль предприятия и признано годным к эксплуатации.

Контакты производителя:
ООО «Онитекс», г. Санкт-Петербург.
(812) 928-65-54
support@onitex.ru технический отдел
http://onitex.ru последние обновления