

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
5. Режимы работы устройства	14
5.1 Общее описание режимов	14
5.2 Режим драйвера	15
5.3 Автоматический режим	17
5.4 Режим контроллера	19
5.5 Программный режим	19
5.6 Подключение в режиме контроллера и программном режиме	20
5.7 Подключение порта RS-232.	22
5.8 Использование переходников USBtoCOM.	22
6. Система команд в режиме контроллера и программном режиме	23
7. Управление стандартными средствами ОС Windows®	29
8. Работа с OSM Programmer	32
9. Использование SDK под ОС Windows®.	36
10. Использование SDK под OCLinux.	37
11. Обновление встроенного программного обеспечения	38
12. Типичные неисправности и их устранение.	39
13. Гарантийные обязательства и техническая поддержка	41

1. Описание устройства

Устройство управления шаговыми двигателями OSM-17R/OSM-42R (далее – устройство, блок) является современным, надежным, доступным устройством, предназначенным для управления шаговыми двигателями небольшой мощности. Блок работает с 4-, 6-, 8-ми выводными шаговыми двигателями с током обмоток до 1.7А для версии 17R и 4А для версии 42R. 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 работать **автономно**, по записанной в него программе, управляться в **режиме драйвера** стандартными логическими сигналами «ШАГ», «НАПРАВЛЕНИЕ» и «РАЗРЕШЕНИЕ», работать в **автоматическом режиме**, вращая двигатель со скоростью, заданной потенциометром, а так же управляться в **режиме реального времени** по последовательному порту.

В качестве источника управляющих сигналов может служить персональный компьютер, внешний контроллер (например, контроллер серии OSK производства «Онитекс»), либо генератор импульсов (например, PG-03).

Внешний вид устройства показан на рисунке 1.1.

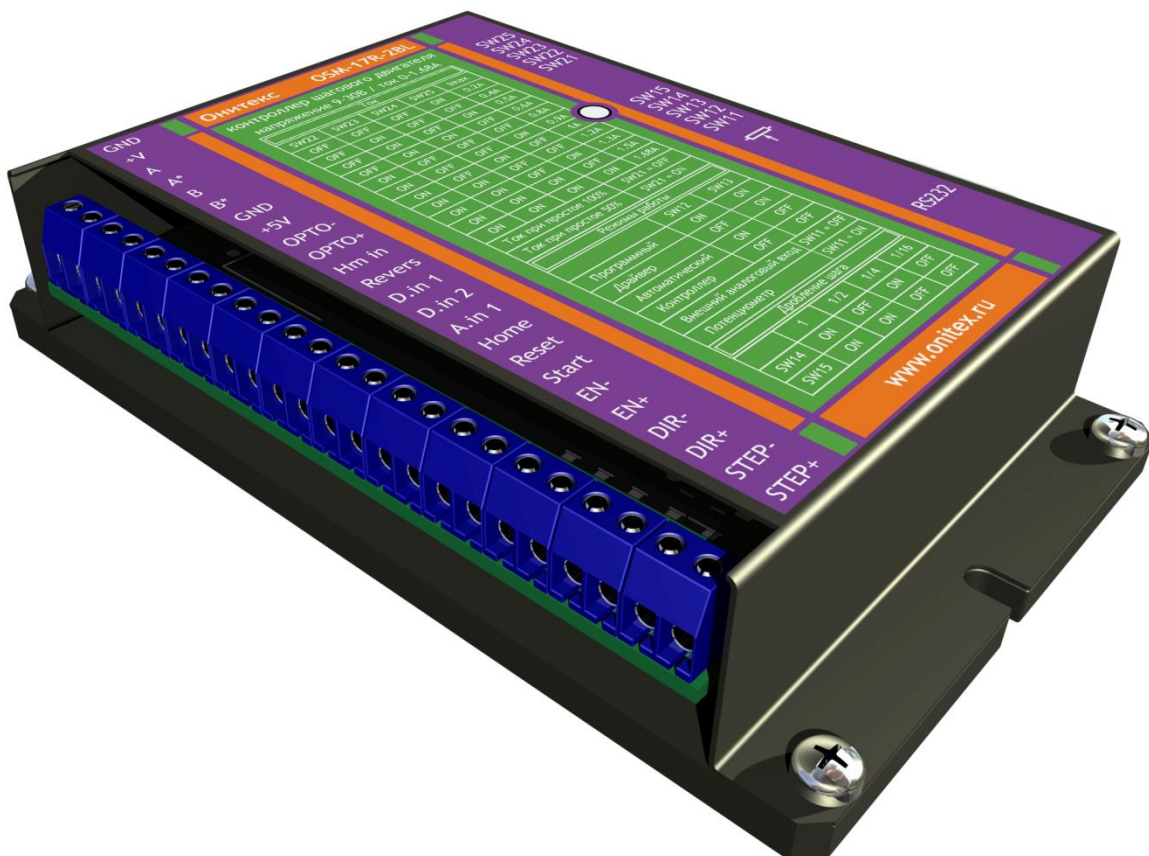


Рисунок 1.1 - Внешний вид контроллера OSM-17R

Габаритные и присоединительные размеры устройства указаны на рисунке 1.2

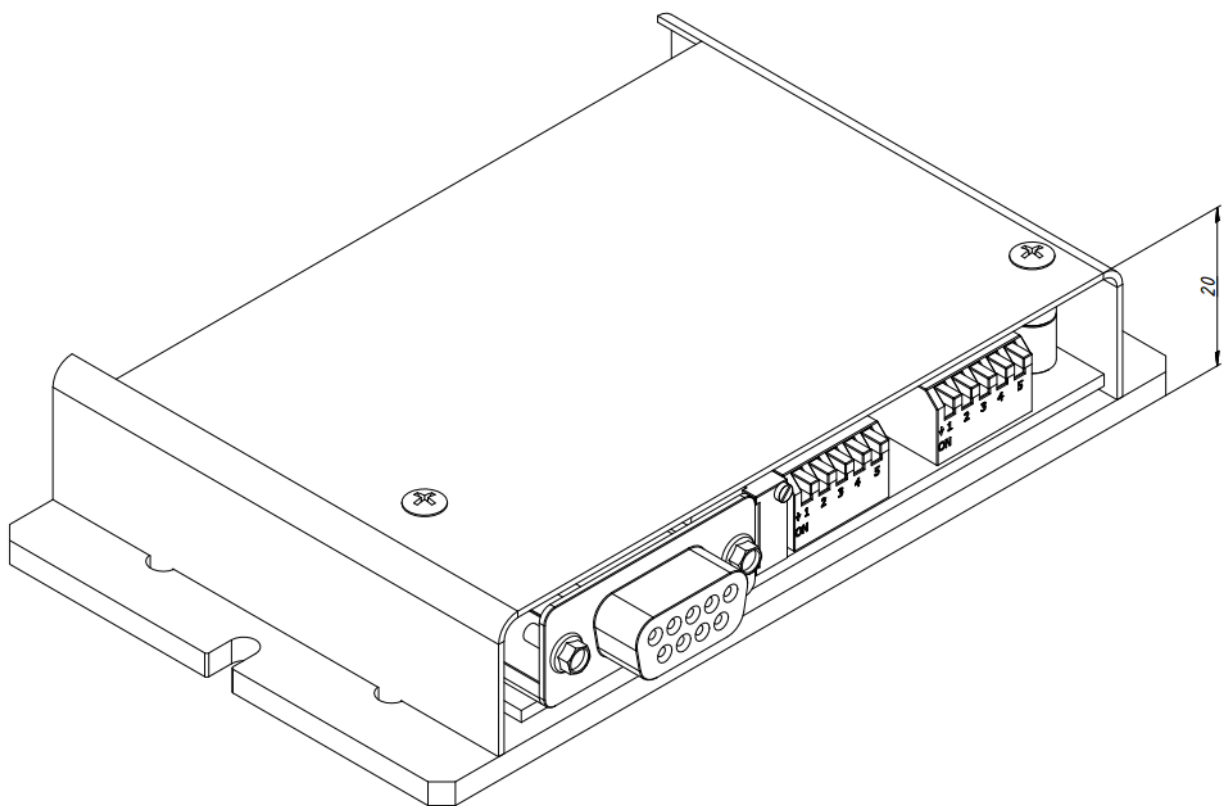
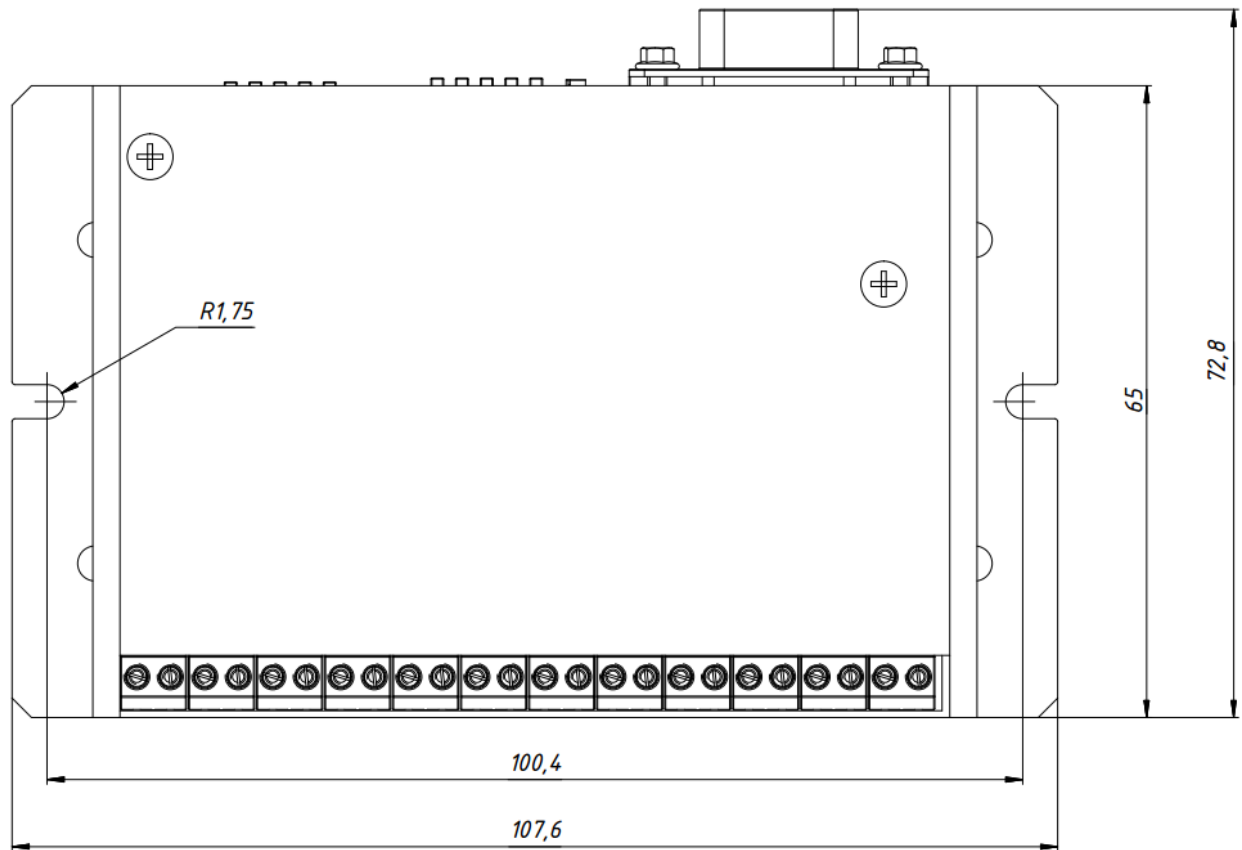


Рисунок 1.2 - Габаритные и присоединительные размеры устройства

Основные отличительные особенности устройства:

- установка скорости, ускорения, **режима дробления шага**, величины перемещения, направления движения шагового двигателя;
- дробление полного шага до 1/16;
- наличие энергонезависимой памяти для автономной работы контроллера;
- наличие потенциометра задания скорости, аналогового входа и разъема подключения внешнего потенциометра;
- функция поиска начального положения по отдельному датчику;
- автоматическое переключение направления вращения двигателя при поступлении сигнала от датчика реверса;
- три дополнительных **входа** для приема сигналов от внешних устройств (датчиков);
- один дополнительный **выход** для подачи сигналов внешним устройствам;

На рисунке 1.3 представлена структурная схема контроллера.

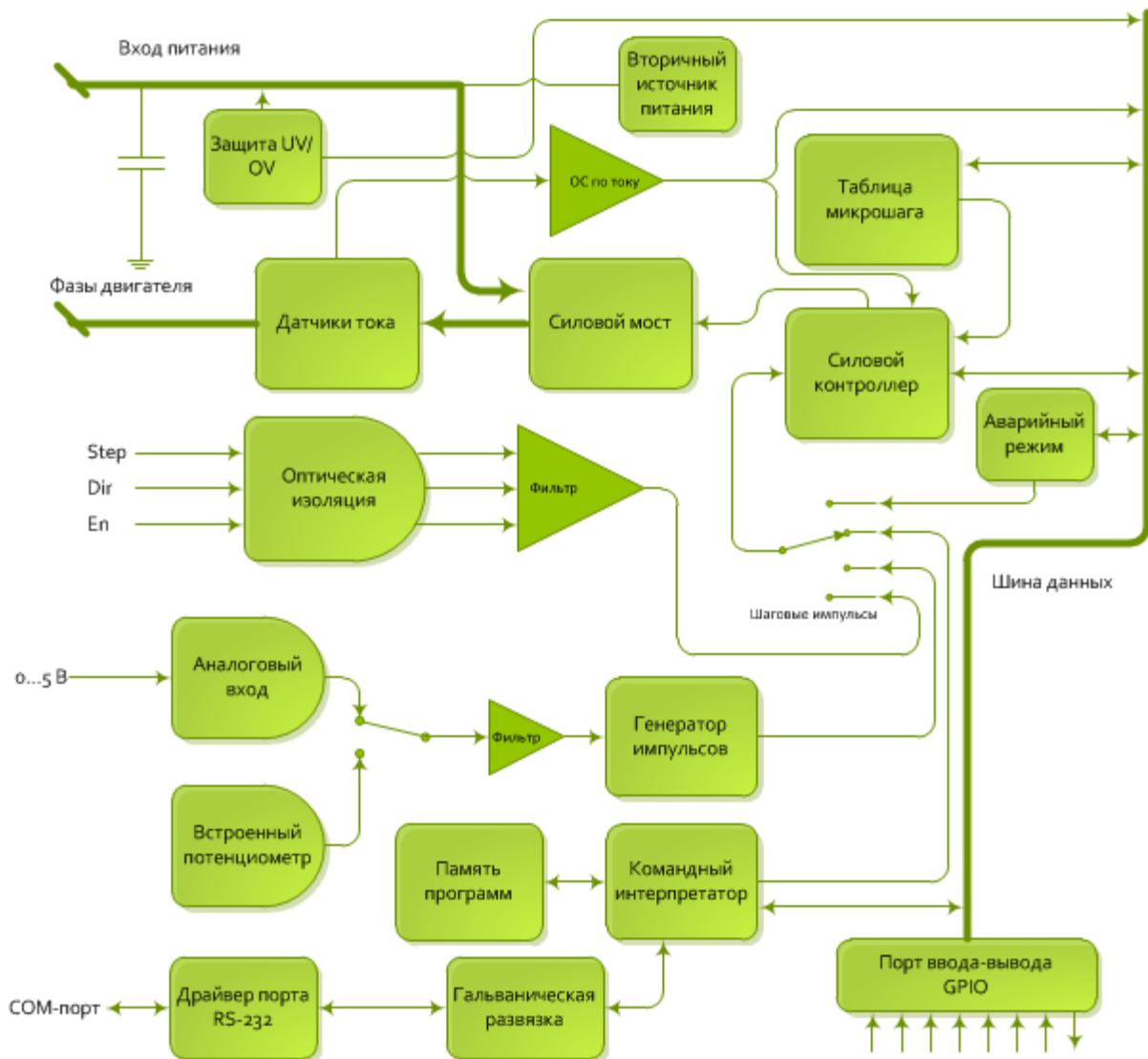


Рисунок 1.3 -Структурная схема контроллера

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

2. Технические характеристики

Технические характеристики устройства приведены в таблице 2.1.

Таблица 2.1- Технические характеристики устройства

Общие характеристики:		
Максимальный выходной ток, OSM-17R (OSM-42R)	1.7 (4.2)	А
Минимальный выходной ток, OSM-17R (OSM-42R)	0.1 (0.28)	А
Габаритные размеры, не более	108x73x20	мм
Массанетто, не более	0.2	кг
Коэффициенты дробления шага	1, 1/2, 1/4, 1/16	
Максимальная частота входа STEP, не менее	85 000	Гц
Диапазон частот встроенного генератора импульсов	1...10000	Гц
Дискретность установки частоты генератора	1	Гц
Напряжение питания, OSM-17R (OSM-42R)	8...30(10..45)	Вольт
Ошибка установки величины фазных токов, не более	+/-5	%
Оптически изолированные входы управления:		
Входное сопротивление	1.2	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:		
Напряжение гальванической изоляции, не менее	1000	Вольт
скорость обмена	9600...115200	бит/с
количество бит	8	
четность	чет	
стоповые биты	1	
управление потоком	аппаратное	
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	IP20	

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 розетки 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

Вниманию пользователей устройства OSM-17R.

Устройство предназначено для работы от стабилизированного или нестабилизированного источника питания с выходным напряжением 8-30 Вольт. Максимально допустимое напряжение питания устройства составляет 35 Вольт, включая флуктуации питающего напряжения и обратную ЭДС двигателя, таким образом, источник питания не должен выдавать больше 30 Вольт. Не допускается эксплуатация устройства при температурах, выходящих за пределы указанных в [таблице 2.1](#).

Вниманию пользователей устройства OSM-42R.

Устройство предназначено для работы от стабилизированного или нестабилизированного источника питания с выходным напряжением 10-44 Вольт. Максимально допустимое напряжение питания устройства составляет 48 Вольт, включая флуктуации питающего напряжения и обратную ЭДС двигателя, таким образом, источник питания не должен выдавать больше 44 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.

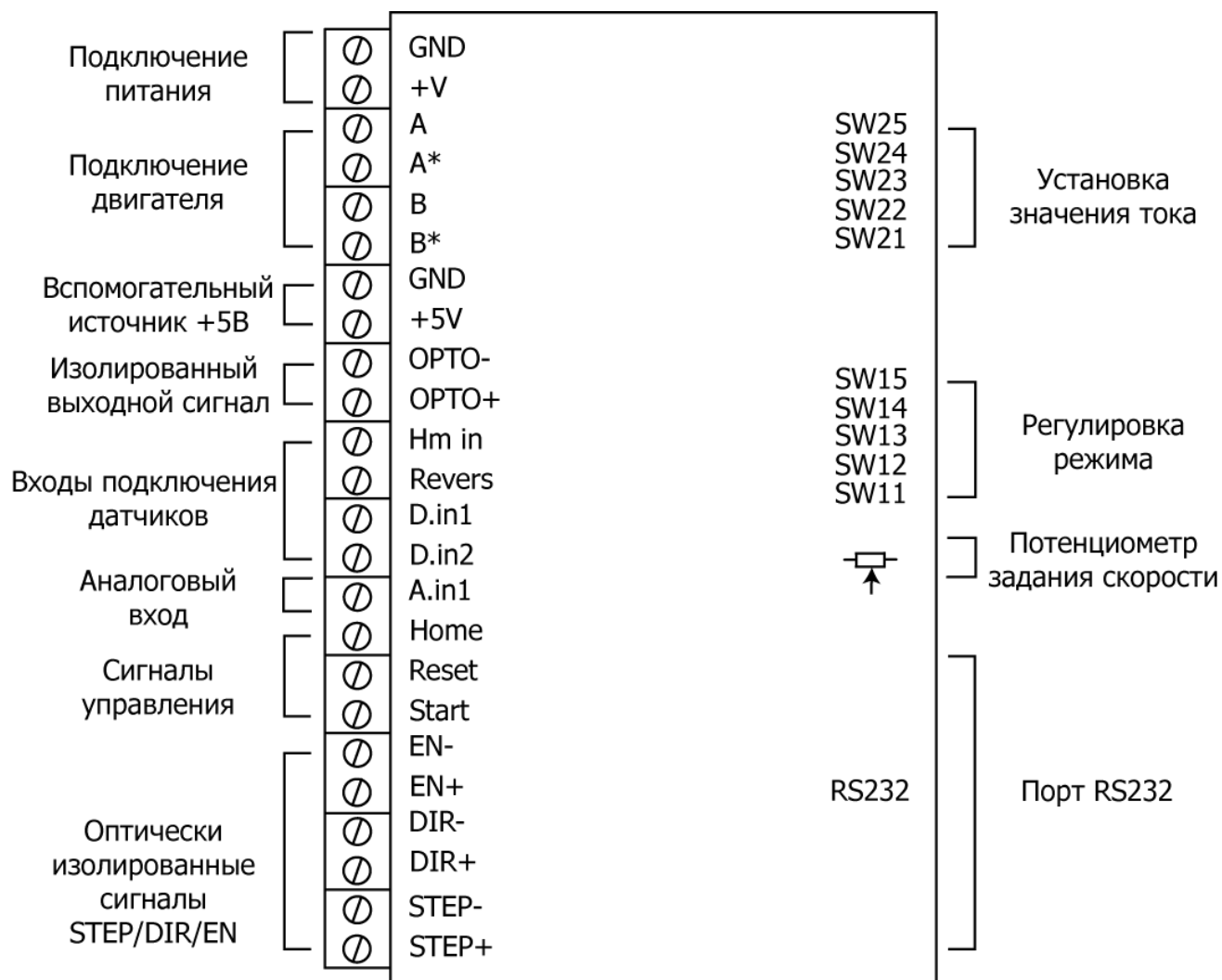


Рисунок 4.1.1 -Расположение управляющих входов и элементов управления

Функциональное назначение управляющих входов и выходов указано в таблице 4.1.1.

Таблица 4.1.1 - Функциональное назначение управляющих входов и выходов

Вход/Выход	Программный/ контроллера	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		
ОРТО+/ОРТО-	Выход типа «общий коллектор» для управления внешними устройствами	<i>Не используется</i>	
Hm in	Вход датчика начального положения	<i>Не используется</i>	
Revers	Смена направления вращения	Смена направления вращения	<i>Не используется</i>
D.in1	Дискретный вход 1	<i>Не используется</i>	
D.in2	Дискретный вход 1	<i>Не используется</i>	
A.in1	<i>Не используется</i>	Аналоговый вход	<i>Не используется</i>
Home	Старт поиска начального положения	<i>Не используется</i>	
Reset	Сброс контроллера	<i>Не используется</i>	
Start	Начало выполнения последовательности команд	<i>Не используется</i>	
EN+/EN-	Включение напряжения на обмотках		
DIR+/DIR-	Дополнительный оптоизолированный вход D	Направление вращения	
STEP+/STEP-	Дополнительный оптоизолированный вход S	<i>Не используется</i>	Вход шаговых импульсов
RS232	Порт RS-232	<i>Не используется</i>	

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

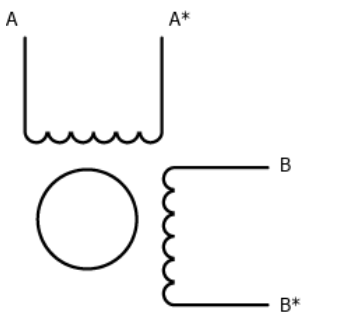
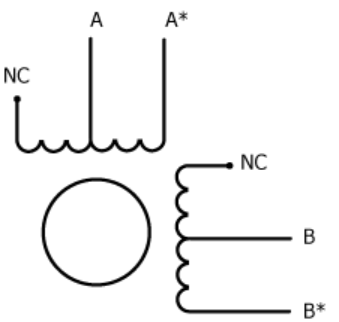
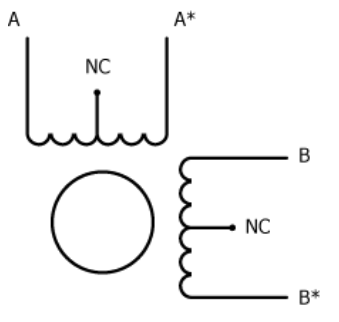
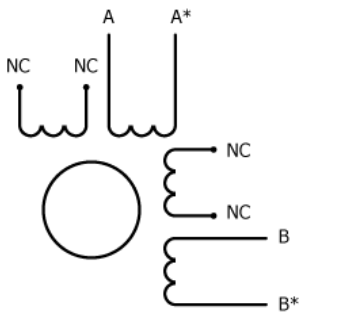
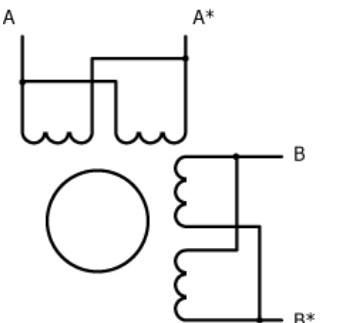
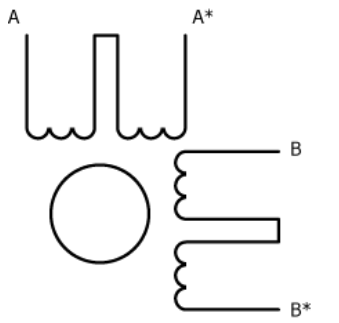
Таблица 4.1.2 – Структурные схемы управляющих входов

Входы	Назначение	Структурная схема
STEP DIR EN	Входы дифференциальные оптически-изолированные	
A.in	Вход аналоговый неизолированный	
Hm in Revers D.in1 D.in2 A.in1 Home Reset Start	Вход сигнальный с подтяжкой неизолированный	
ОРТО	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 А, А*, В, В* в соответствии с таблицей 4.2.1.

Таблица 4.2.1 – Способы включения фаз двигателя

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

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. Ток двигателя выставляется с помощью переключателей SW22 – SW25 (таблица 4.3.1). Ток 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

OSM-17R

Макс ток, А	0,10	0,22	0,34	0,45	0,56	0,67	0,80	0,90	1,00	1,10	1,20	1,30	1,40	1,50	1,68
SW22	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON	ON	ON	ON	ON	ON	ON
SW23	OFF	OFF	OFF	ON	ON	ON	ON	OFF	OFF	OFF	OFF	ON	ON	ON	ON
SW24	OFF	ON	ON	OFF	OFF	ON	ON	OFF	OFF	ON	ON	OFF	OFF	ON	ON
SW25	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON

OSM-42R

Макс ток, А	0,28	0,56	0,86	1,20	1,40	1,70	2,00	2,30	2,60	2,80	3,20	3,50	3,80	4,00	4,20
SW22	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON	ON	ON	ON	ON	ON	ON
SW23	OFF	OFF	OFF	ON	ON	ON	ON	OFF	OFF	OFF	OFF	ON	ON	ON	ON
SW24	OFF	ON	ON	OFF	OFF	ON	ON	OFF	OFF	ON	ON	OFF	OFF	ON	ON
SW25	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	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).

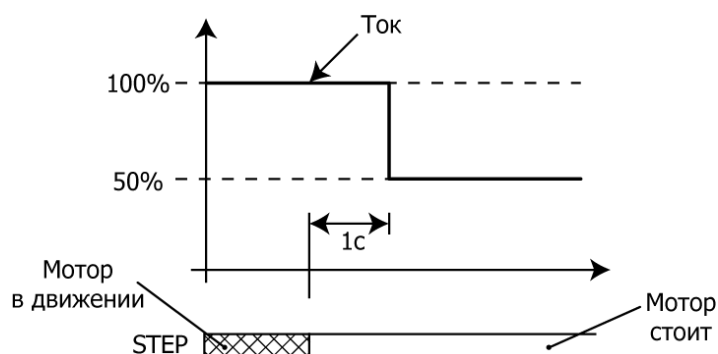


Рисунок 4.3.1 – Режим SLEEP

Включение режима SLEEP происходит переключателем SW1.

Ток при простое 100% SW21 = OFF

Ток при простое 50% SW21 = ON

Выбор режима дробления шага осуществляется джамперами. Соответствия между режимом и состоянием DIP-переключателя описано в таблице 4.3.2. Состояние «ON» соответствует опущенному переключателю.

Таблица 4.3.2 – Выбор режима дробления шага

Дробление шага	1	1/2	1/4	1/16
SW14	ON	OFF	ON	OFF
SW15	ON	ON	OFF	OFF

5. Режимы работы устройства

5.1. Общие описание режимов

Устройство может работать в одном из четырех режимов (рисунок 5.1.1), устанавливаемых переключками SW12-SW13 (таблица 5.1.1).



Рисунок 5.1.1 – Режимы работы устройства

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. Доступны команды задания скорости, ускорения, количества шагов, работы с датчиками.

В **программном** режиме доступны те же команды, но добавлена возможность сохранять исполнительную программу в памяти контроллера для независимой работы контроллера в автономном режиме.

В **автоматическом** режиме осуществляется вращение двигателя со скоростью, устанавливаемой встроенным потенциометром, внешним подключенным потенциометром либо источником сигнала 0-5Вольт.

Выбрать нужный режим работы устройства необходимо установкой переключателей SW12 и SW13 в соответствии с таблицей 5.1.1.

Таблица 5.1.1 – Режимы работы устройства

Режимы работы	Программный	Драйвер	Автоматический	Контроллер
SW12	ON	OFF	ON	OFF
SW13	ON	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 или микрошаг (в зависимости от положения переключателей микрошага). 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:

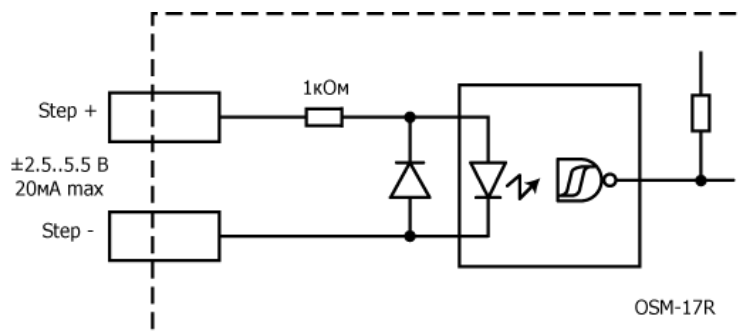
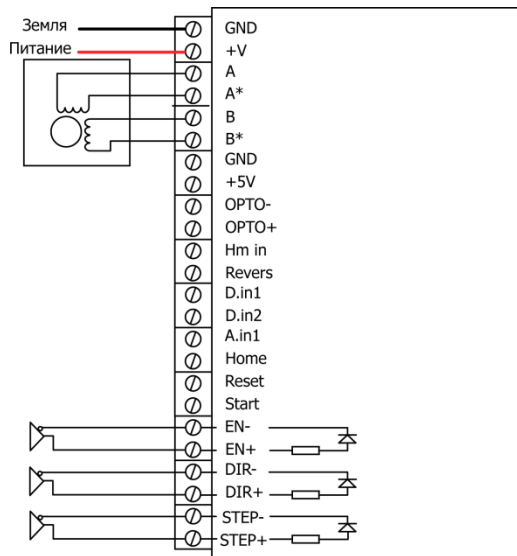


Рисунок 5.2.1 – Структурная схема оптически изолированного высокочастотного входа STEP

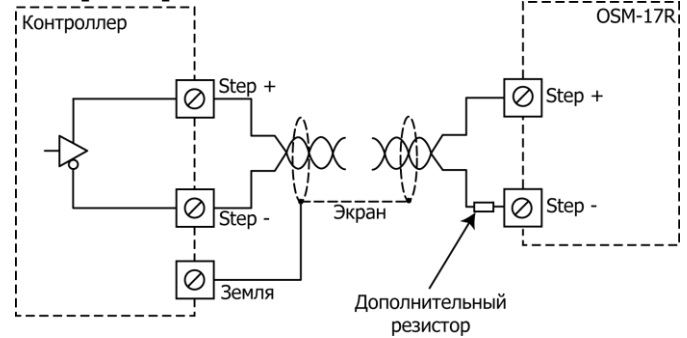
В таблице 5.2.1 показаны возможные варианты подключения внешних управляющих контроллеров.

Таблица 5.2.1 - Возможные варианты подключения внешних управляющих контроллеров

1. Управление от контроллера с дифференциальным выходом.

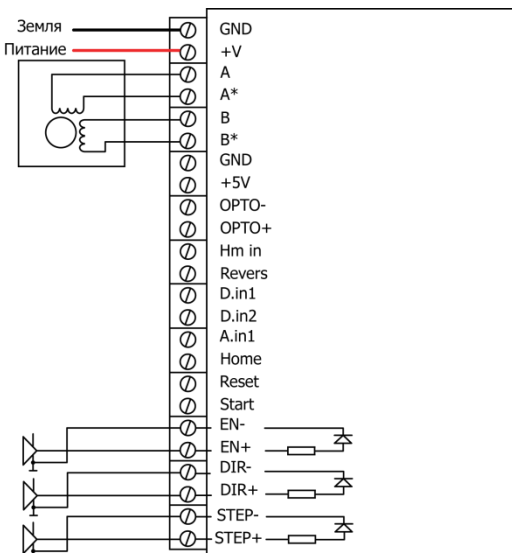


Подключение контроллера с дифференциальным выходом – наиболее помехозащищенное решение. Прямой и инверсный сигналы управляющего контроллера подключаются напрямую к положительным и отрицательным входам контроллера OSM.



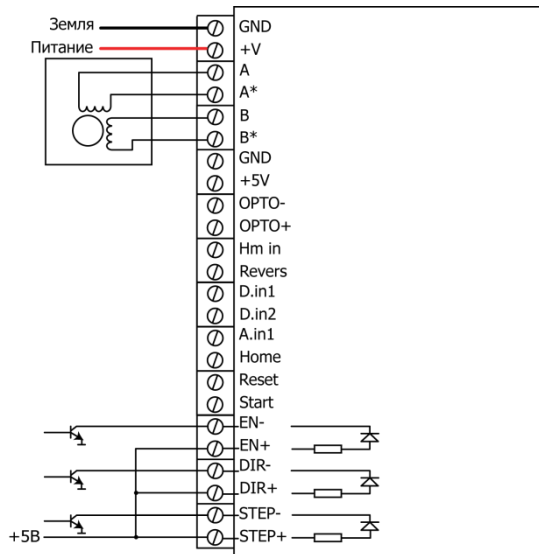
Дополнительный резистор требуется, если напряжение сигнала превышает 9В, номинал резистора рассчитывается исходя из указанного тока через опторазвязку. Обратите внимание на необходимость применения витой пары и экранирование сигнального провода.

2. Управление от контроллера с TTL-выходом.



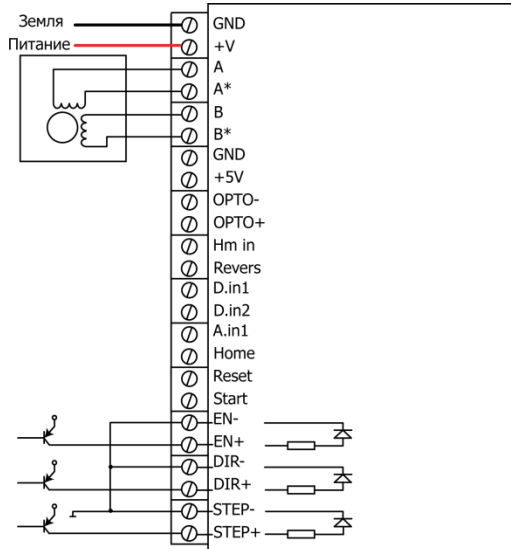
При подключении управляющего контроллера с TTL-выходом необходимо объединить отрицательные входы контроллера OSM и соединить их с землей управляющего контроллера. Соединение с землей необходимо производить на стороне управляющего контроллера.

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.

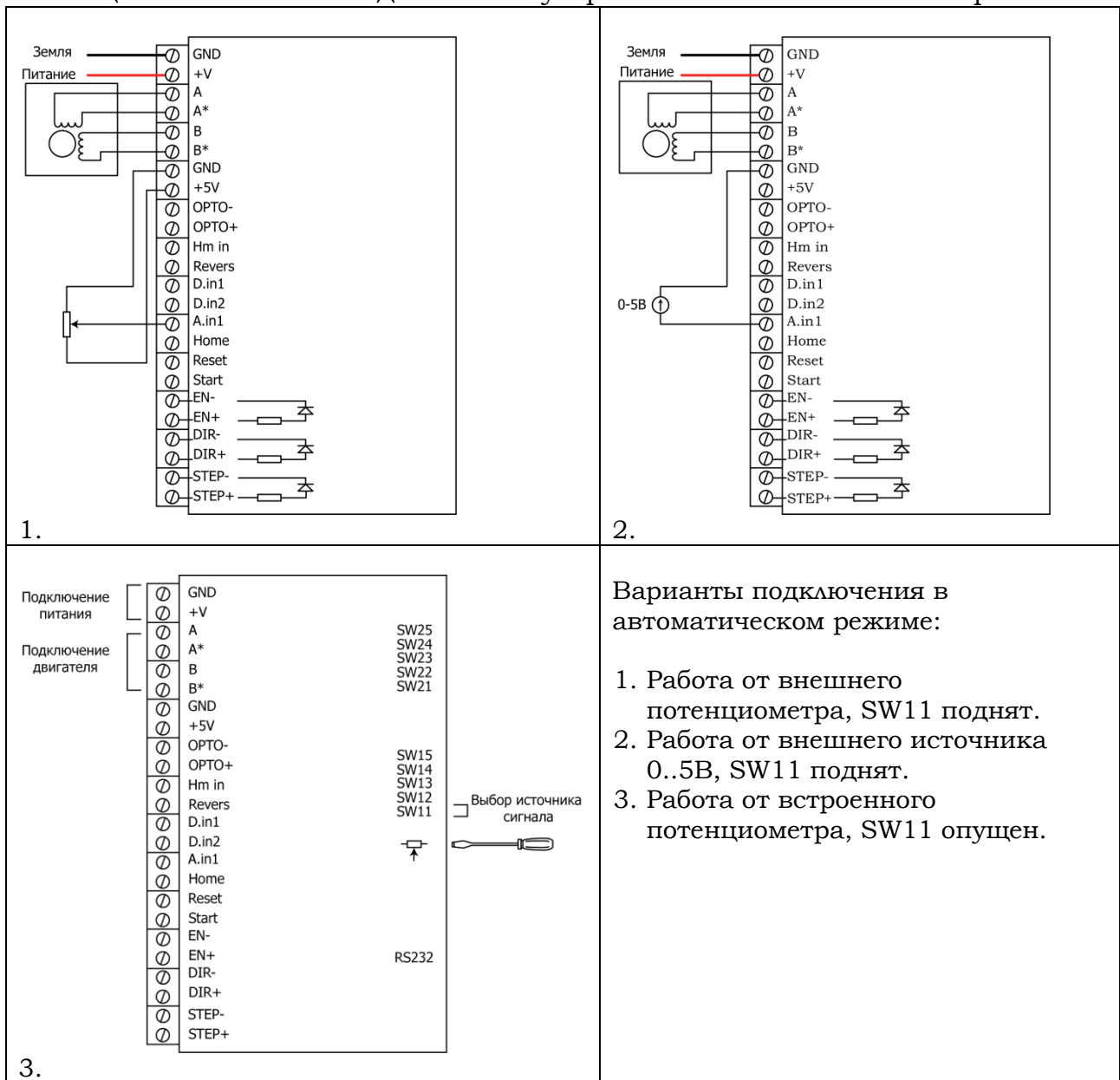
5.3 Автоматический режим

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

В контроллере имеется три возможности работы в автоматическом режиме: работа от встроенного многооборотного потенциометра, от внешнего аналогового сигнала 0..5В либо от внешнего потенциометра. Выбор между встроенным потенциометром и внешним аналоговым сигналом осуществляется с помощью переключателя SW11. Включение переключателя подключает встроенный потенциометр к аналоговому входу.

Схемы подключения устройств показаны в таблице 5.3.1.

Таблица 5.3.1 - Схемы подключения устройства в автоматическом режиме



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.

Управление происходит с помощью набора несложных **команд**. 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.

В режиме контроллера команды записываются в оперативную память устройства. При подаче повторной команды «Старт» повторяется последняя переданная в устройство последовательность команд. Подробнее система команд будет описана ниже в настоящем руководстве.

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.

В программном режиме устройство может использовать показания датчиков для своей работы, в частности, имеются команды движения до датчиков или ожидание сигнала до датчика. В качестве датчиков служат входы контроллера, в том числе два оптически изолированных входа STEP и DIR.

Так же предусмотрена возможность использования циклов и команд безусловного перехода к метке.

Работа по записанной в память программе начинается при замыкании кнопки входа Старт либо после получения команды ST. Так же существует возможность автоматического старта выполнения программы после подачи питания на контроллер. Замыкание контакта «Reset» вызывает остановку программы и возвращение указателя в нулевой адрес.

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 рисунке 5.6.1.

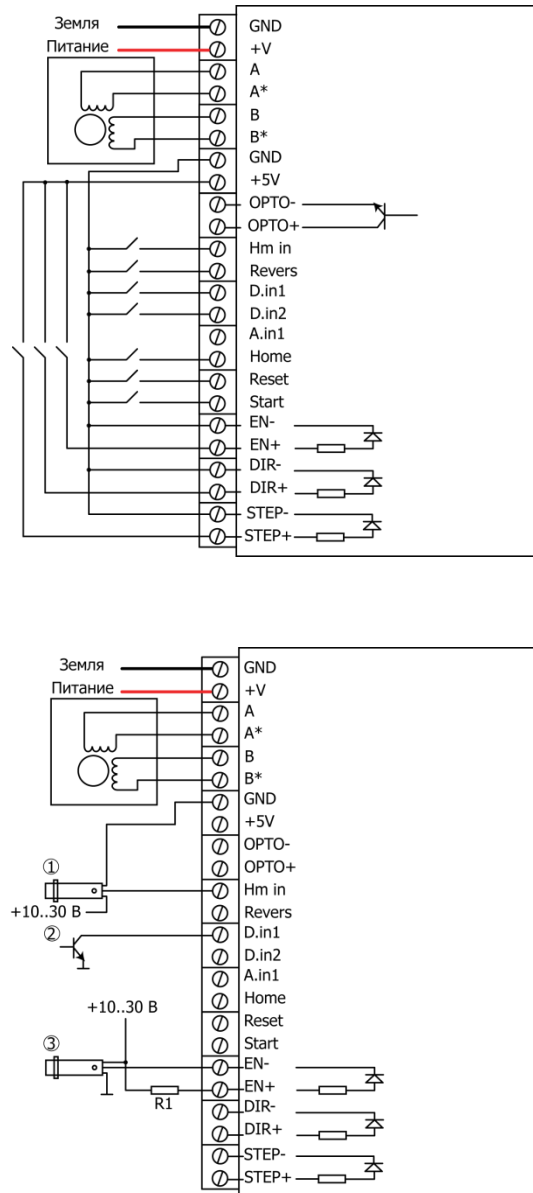


Рисунок 5.6.1 - Схемы подключения блока в режиме контроллера

В режиме контроллера и программном режиме возможно подключение различных датчиков. Схемы подключения датчиков к входам контроллера приведены на рисунке 5.6.2.

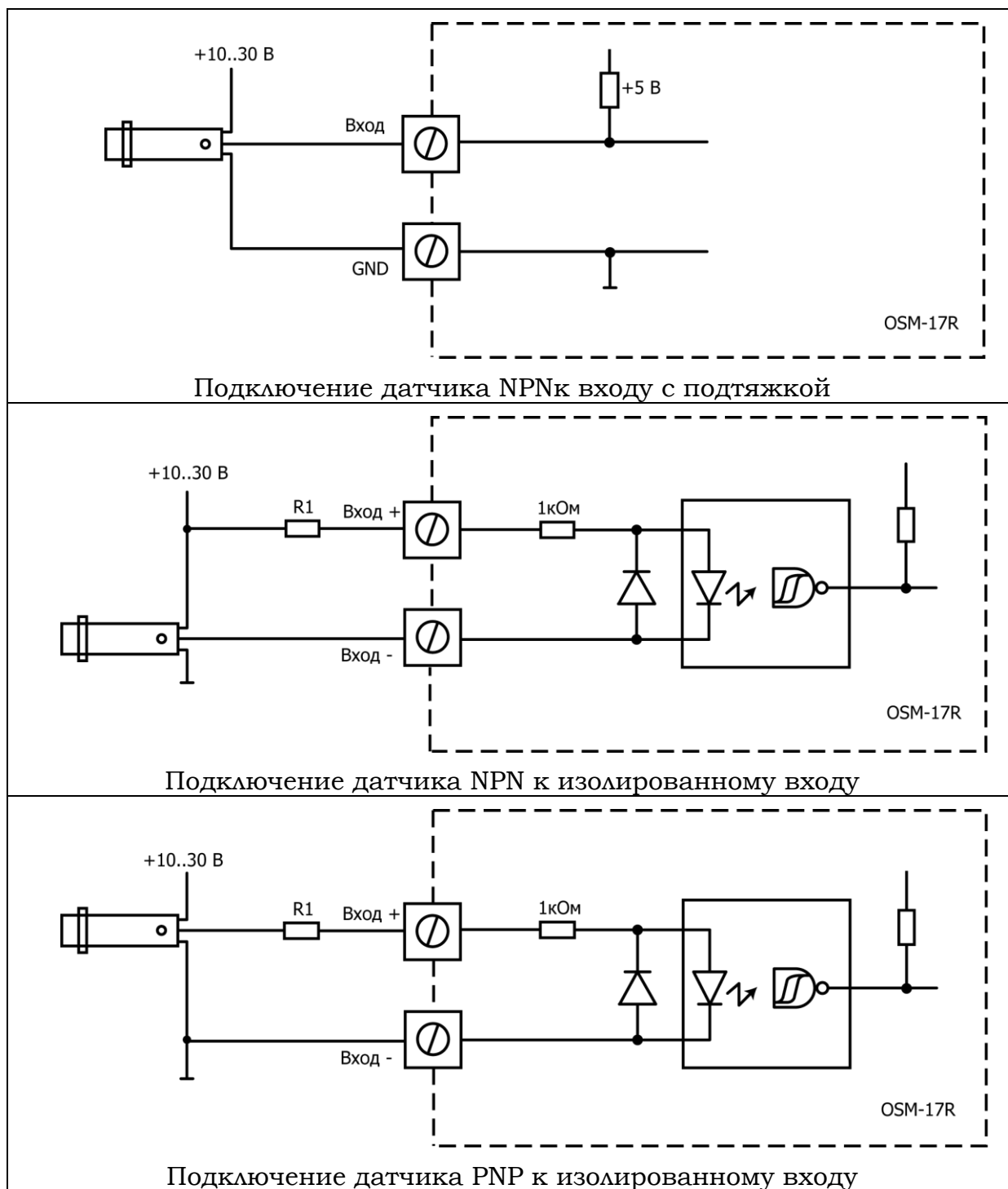


Рисунок 5.6.2 - Схемы подключения датчиков к входам контроллера

5.7 Подключение порта 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 5.7.1.

Таблица 5.7.1 – Распиновка кабеля

ПК	Контроллер OSM
2	2
3	3
5	5

Другие контакты кабеля не используются.

5.8 Использование переходников USBtoCOM.

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

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.

6. Система команд в режиме контроллера и программном режиме

Обмен ПК с устройством осуществляется по принципу «Команда» - «Ответ». Ответ содержит информацию о выполнении или невыполнении команды (ошибке) с указанием номера ошибки. Команды обмена по последовательному порту, представленные в данном разделе, являются командами низкого уровня. Данные команды предназначены для программирования устройства из терминала либо программы, написанной пользователем, или для управления в реальном времени. Для удобного программирования устройства в программном режиме можно воспользоваться удобной надстройкой высокого уровня, входящей в состав OSMProgrammer, описанной в соответствующем разделе. Данная программа транслирует высокоуровневые команды в исполнительный код для контроллера.

Команда, посылаемая в устройство, состоит из двух байт мнемоники команды (буквенных символов) и, если необходимо, от 1 до 10 байт данных (символов цифр). Полный список исполнительных команд сведен в таблице 6.1.

Таблица 6.1 – Синтаксис команд

	Команда	Параметр	Синтаксис	Описание команды
1	LOAD		LM	Указатель начала последовательности команд
2	START		ST	Запуск выполнения
3	START_POWER_ON			Автоматический старт при включении контроллера
4	FINAL		FN	Окончание программы без старта выполнения
5	FSTART		FS	Окончание программы и старт выполнения
6	DIR_L		DL	Установка вращения влево
7	DIR_R		DR	Установка вращения вправо
7	REVERS		RS	Реверс
8	ACCEL	-3000...3000	AL (-) x	Ускорение от -3000 до 3000
9	SPEED	0...20 000	SDx	Скорость
10	START_SPEED	0...3 000	SSx	Начальная скорость
12	MOVE	0...4294967295	MV (x)	Выполнить x шагов (без параметра –бесконечно)
13	STOP		SP	Остановка двигателя
14	STOP_N	0...100 000	SPx	Пауза длительностью x миллисекунд
15	LABLE		LL	Метка
16	JUMP_N	1...65535	JPx	Цикл от метки x раз
17	NUM_LABEL	1...65535	NL	Нумерованная метка
21	ENABLE		EN	Включить питание обмоток
22	DISABLE		DS	Выключить питание обмоток
23	SETOUT		SF	Установить выходной сигнал
24	CLEAROUT		CF	Снять выходной сигнал
25	WAITGPIN1		WL	Дождаться сигнала от входа 1 (Int1)
26	WAITGPIN2		WH	Дождаться сигнала от входа 2 (Int2)
27	MOVE_GPIN1	0...4294967295	ML (x)	Движение до сигнала от дискретного входа 1 (Int1),

	Команда	Параметр	Синтаксис	Описание команды
				но не более x шагов (без параметра – бесконечно)
28	MOVE_GPIN2	0...4294967295	MH (x)	Движение до сигнала от дискретного входа 2 (Int2), но не более x шагов (без параметра – бесконечно)
29	MOVE_HOME	0...4294967295	HM (x)	Движение до сигнала от входа поиска начального положения (Hm.In), но не более x шагов (без параметра – бесконечно)
30	MOVE_STEP	0...4294967295	MS (x)	Движение до сигнала от входа STEP, но не более x шагов (без параметра – бесконечно)
31	MOVE_DIR	0...4294967295	MD (x)	Движение до сигнала от входа DIR, но не более x шагов (без параметра – бесконечно)
32	REGREAD	0...11	RR	Чтение состояния контроллера 0 – регистр порта ввода (весь порт, 5 бит) 1 – регистр счетчика перемещения (оставшееся число шагов до завершения команды MOVE_N) 2 – регистр абсолютного числа шагов 3 - вход датчика НП (отдельно) 4 - вход DIR 5 - вход EN 6 - вход In1 7 - вход In2 8 - вход START 9 - вход STEP 10 - текущая скорость 11 – значение регистра XR
33	VERSION		VR	Чтение кода модификации контроллера
34	ADCSPDSET		AD	Установка скорости согласно величине напряжения на аналоговом входе
35	CLI		CI	Запрет прерывания по сигналу реверса
36	SEI		SI	Разрешение прерывания по сигналу реверса
37	SET_DIR_EXT		DE	Установить направление согласно внешнему сигналу на входе DIR
38	CURRENTSET	0...255	CSx	Установка значения тока двигателя
39	DELAYMS	0...65 500	DMx	Пауза в выполнении команд в xмс
40	SENDCOMMAND	0...255	SCx	Посылка команды x через порт RS232 в формате "\$xxx".

	Команда	Параметр	Синтаксис	Описание команды
41	RESET		RX	Аппаратный сброс контроллера
42			XR	Переключение входа Revers в режим счетчика
43	BAUDRATE	207, 103, 51, 34, 16	BR	Скорость обмена по RS232: 9600, 19200, 38400, 57600, 115200.

При получении символа «>» контроллер анализирует полученную команду и отвечает ПК о результате выполнения команды. Символ «<» до передачи символа «>» аннулирует ранее выданную строку.

Варианты ответа приведены в таблице 6.2.

Таблица 6.2 – Варианты ответа контроллера

Ответ	Содержание сообщения
e3	Ошибка кода исполнительной программы
end	Завершение исполнительной программы
e5	Ошибка приема по COM-порту
e6	Ошибка команды
e9	Ошибка данных команды

Описание команд приведено в таблице 6.3.

Таблица 6.3 – Описание команд

1	ENTER LOAD MODE	Данная команда является указателем начала загрузки программы в память контроллера. После получения данной команды контроллер входит в режим загрузки и сохраняет введенные далее команды в буфер. После получения команды FINAL или FINAL&START (FN или FS) команды будут перезаписаны в энергонезависимую память контроллера. В режиме контроллера записанная последовательность остается в оперативной памяти.
2	FINAL	Команда является указателем окончания последовательности команд, введенных пользователем. В программируемом режиме после получения данной команды записанная последовательность переписывается в энергонезависимую память.
3	START	Команда служит для старта выполнения записанной последовательности команд. В режиме контроллера команда ST повторяет ранее введенную последовательность команд. В программируемом режиме начинает выполняться последовательность, записанная в энергонезависимой памяти. Команда обычно используется в конце последовательности команд перед командой FINAL(FN). В большинстве случаев можно заменить команды START и FINAL одной командой FS.
4	FINAL&START	Обозначает окончание последовательности команд и одновременный старт выполнения.
5	DIR L DIR R	Устанавливает направление вращения двигателя.
6	REVERS	Изменяет текущее направление вращения на противоположное. Дальнейшее применение команд задания направления устанавливает направление вращения согласно команде.

7	ACCEL	От -3000 до 3000 шаг/с
	Устанавливает значения ускорения. Минус означает замедление движения, плюс – ускорение. Начальное значение задается командой START SPEED, конечное значение – командой SPEED. Двигатель стартует со скоростью STARTSPEED, разгоняется с ускорением ACCEL до значения SPEED и продолжает работу с данной скоростью. Команда используется, в том числе, при необходимости разогнать двигатель до больших оборотов, так как шаговый двигатель не может стартовать сразу с больших скоростей.	
8	SPEED	От 1 до 20 000 шаг/с
	Устанавливает значение скорости в шаг/с. При использовании команды ACCEL задает конечную скорость.	
9	START SPEED	
	Начальная скорость при разгоне с ускорением.	
	MOVE	От 1 до 4294967295шагов
	Движение двигателя. Без параметра команда вызывает бесконечное движение двигателя, с параметром – перемещение на заданное число шагов (микрошагов).	
10	STOP	
	Остановка двигателя.	
11	STOP N	
	Остановка двигателя на x миллисекунд.	
12	LABEL	
	Установка нумерованной метки.	
13	JUMP	
	Безусловный переход к метке нумерованной метке LABEL. Переход осуществляется x раз (параметр x от 1 до 255). Позволяет организовать цикл. Команда без параметра – бесконечный цикл.	
18	ENABLE	
	Включает напряжение питания на двигателе. Ротор становится в режим удержания. По умолчанию напряжение на обмотках включено, для его снятия необходимо использовать команду DISABLE.	
19	DISABLE	
	Снимает питание с обмоток двигателя. Ротор выходит из режима удержания.	
20	SETGROUT	
	Замыкает сигнальный выход OPTO на землю (устанавливает сигнал).	
21	CLEARGROUT	
	Снимает сигнал OPTO.	
22	WAITGPIN1	
	Пауза в выполнении программы в ожидание сигнала на входе D.In1.	
23	WAITGPIN2	
	Пауза в выполнении программы в ожидание сигнала на входе D.In2.	
24	MOVE GPIN1	
	Движение до срабатывания датчика In1. Без параметра движение происходит до срабатывания датчика без ограничения по времени. При необходимости можно задать максимальное число шагов, которые может проделать двигатель. В таком случае движение прекращается по наиболее раннему событию.	
25	MOVE GPIN2	
	То же, что GPIN1, но для входа D.In2.	
26	MOVE HOME INPUT	
	То же, что GPIN1, но для входа Home.	
27	MOVE ISO STEPINPUT	
	То же, что GPIN1, но для входа STEP.	

28	MOVE ISO DIRINPUT
	То же, что GPIN1 , но для входа DIR.
29	REGISTERREAD
	Чтение состояния контроллера: 0 – регистр порта ввода (весь порт, 7 бит); 1 – регистр счетчика перемещения (оставшееся число шагов до завершения команды MOVE_N); 2 – регистр абсолютного числа шагов; 3 - вход датчика НП; 4 - вход DIR; 5 - вход EN; 6 - вход In1; 7 - вход In2; 8 - вход START; 9 - вход STEP; 10 - текущая скорость; 11 – счетчик RX. С помощью данной команды можно узнать текущее состояние контроллера: уровни на его входах-выходах, скорость и т.п. Это позволяет организовать обратную связь с устройством.
30	VERSION
	Считывает номер версии встроенного ПО. Версии нумеруются целым числом. Если номер прошивки в контроллере ниже, чем номер релиза, выпущенного производителем, можно обновить прошивку при помощи утилиты OSM Update.
31	ADCSPDSET
	Установка скорости согласно величине напряжения на аналоговом входе. Используется, если по логике выполнения программы скорость должна быть считана с внешнего аналогового сигнала или потенциометра. Зависимость частоты от входного напряжения указана в разделе «Автоматический режим».
32	CLI
	Запрет прерывания по сигналу реверса. По умолчанию в контроллере включен реверс от датчика, подключенного к соответствующему входу. При срабатывании этого датчика происходит смена направления на противоположное. Если требуется выключить эту возможность, необходимо применить данную команду. Обратное включение прерывания происходит командой SEI.
33	SEI
	Разрешение прерывания по сигналу реверса (см. CLI).
34	SETDIREXT
	Установить направление согласно внешнему сигналу на входе DIR.
35	CURRENTSET
	Установка значения тока двигателя.
36	DELAYMS
	Пауза в выполнении команд в x мс.
37	SENDCOMMAND
	Посылка команды xxx через порт RS232 в формате "\$xxx;". Команда дает возможность использовать контроллер в режиме ведущего устройства, соединяя его с другими устройствами через интерфейс RS-232. Пользователь имеет возможность разработать систему команд, используя числовые коды от 1 до 255, и настроить второе устройство на прием и выполнение данных команд.
38	RESET
	Аппаратный сброс контроллера.

39	XR Переключение входа Revers в режим счетчика. Данная команда позволяет переназначить вход реверс для использования его в качестве счетчика внешних событий. Каждое срабатывание датчика вызывает увеличение значения в регистре XR на единицу.
40	XO Обнуление значения регистра XR.
41	BAUDRATE Команда устанавливает скорость обмена по RS232из следующего ряда скоростей: 9600, 19200, 38400, 57600, 115200 кбит/с. Остальные настройки порта остаются теми же. После получения и обработки команды новое значение скорости устанавливается в качестве текущей и записывается в энергонезависимую память. При следующем включении устройства скорость устанавливается согласно значению, записанному в память.

Возможно два варианта программирования и управления блоком. В первом варианте производится работа с низкоуровневыми командами (двухсимвольными мнемониками). Простейшим примером может служить работа через [HyperTerminal](#). Используя описание протокола обмена и перечень мнемоник, можно написать собственное программное обеспечение, самостоятельно или с применением [SDK Onitex](#).

Второй, наиболее удобный вариант программирования, предусматривает использование редактора [OSM Programmer](#). В данном случае пользователю предоставляется возможность максимально удобной работы с блоками программируемыми контроллерами OSM.

7. Управление стандартными средствами ОС Windows®

Данное устройство имеет возможность управления стандартными средствами Windows (с помощью программы HyperTerminal). Это позволяет произвести пусконаладочные работы, не имея под рукой специального программного обеспечения, пользуясь только средствами операционной системы, что повышает гибкость и мобильность применения контроллера.

7.1. Запустить программу HyperTerminal.

Программа запускается из меню Windows «Пуск/Программы/Стандартные/Связь/HyperTerminal». После старта появится окно с предложением ввести название нового подключения (рисунок 7.1).

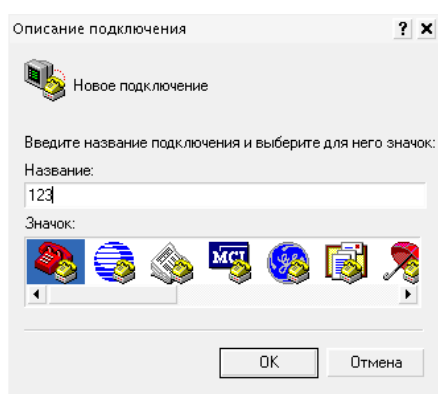


Рисунок 7.1 – Окно описания подключения

Необходимо ввести любое название подключения и нажмите ОК.

7.2. Указать используемый для подключения последовательный порт.

После ввода описания подключения появляется окно параметров подключения. Предположим, связи с устройством используется порт COM1 (рисунок 7.2).

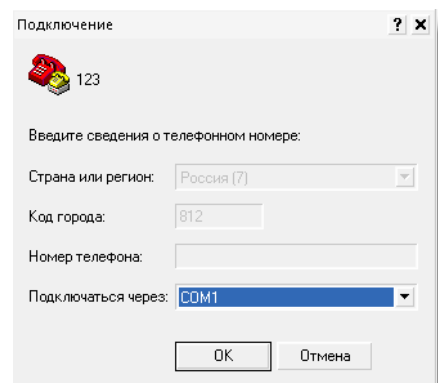


Рисунок 7.2 – Окно подключения

Необходимо выбрать номер COM-порта в выпадающем списке «Подключаться через: » и нажать ОК.

7.3. Настроить параметры используемого последовательного порта. После выбора последовательного порта, например, COM1, программа предлагает установить его параметры (рисунок 7.3).

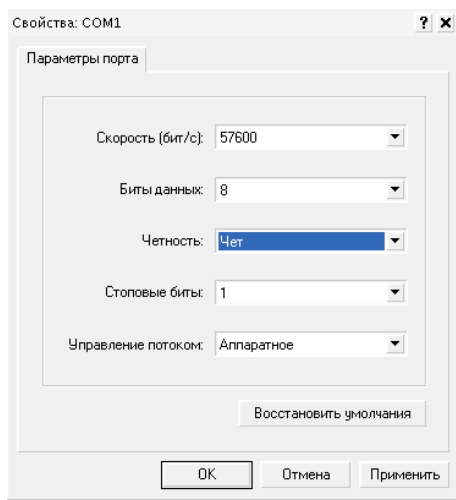


Рисунок 7.3 – Параметры порта

Для работы с устройствами OSM задайте скорость 57600 бит/с, 8 бит данных, четность «чет», 1 стоповой бит, в выпадающем списке «управление потоком» выберете «аппаратное». Нажать ОК.

7.4. Для перепрограммирования контроллеров, ввести систему команд в поле терминала. Вводимый текст и сообщения должны отображаться в окне терминала.

При закрытии программы HyperTerminal после первого запуска появляется окно с предложением сохранить созданное соединение (123). Можно сохранить его и использовать для старта терминала. После сохранения в папке «Пуск/Программы/Стандартные/Связь/HyperTerminal» появится значок соединения с названием 123.ht. Щелчком по значку открывается окно терминала с установленными ранее параметрами.

Ниже приведены примеры низкоуровневых управляющих последовательностей в ASCII мнемониках. Примеры использования команд высокого уровня приведены в разделе «Работа с OSM Programmer»

Установка скорости 100Гц и количество шагов 200:

```
LM>
SD100>
MV200>
FS>
```

Установка скорости 113 Гц, направление вращения «влево», 250 шагов, реверс, и 56 шагов вращение со скоростью 10 Гц:

```
LM>
DL>
SD113>
MV250>
RS>
SD10>
MV56>
FS>
```

Для простого бесконечного движения с заданной скоростью необходимы следующие команды:

```
LM>
DL>
SD100>
MV>
FS>
```

Добавление команд:

1. Ожидание включения датчика.
2. Реверс.
3. Цикл.

где `xxxx` и `zzzzz` - число шагов, которые надо сделать в том и другом направлении.

```
LM>
EN>
LL>
WL>
DL>
SD100>
MVxxxx>
RS>
MVzzzzz>
JP255>
FS>
```

Для более удобного редактирования программ рекомендуется использовать среду OSM Programmer, работа с которой описана в [главе 8](#). Работа в этой среде происходит на уровне команд, а не на уровне протокола, кроме того, в программе присутствует большое число дополнительных функций.

8. Работа с OSM Programmer

Наиболее удобной средой для программирования контроллеров OSM является OSM Programmer. Данная программа позволяет редактировать исходный код программы при помощи собственного редактора, выбирать команды из дерева команд, проводить проверку кода на наличие ошибок и загружать сформированную последовательность в контроллер через последовательный интерфейс.

Необходимо помнить, что в данном руководстве приводится описание общих принципов работы программы. Программа постоянно совершенствуется, поэтому для получения наиболее свежей информации необходимо обратиться к документации, поставляемой с каждой версией программы, и встроенной справочной системе.

Для установки программы необходимо запустить установочный файл и следовать инструкции установщика. Возможно, при установке потребуется обновить .NET Framework.

Окно программы состоит из дерева команд, панели настроек, окна проекта, окна терминала и окна вывода. Кроме того, в программе предусмотрено дополнительное окно подсказок под деревом команд.

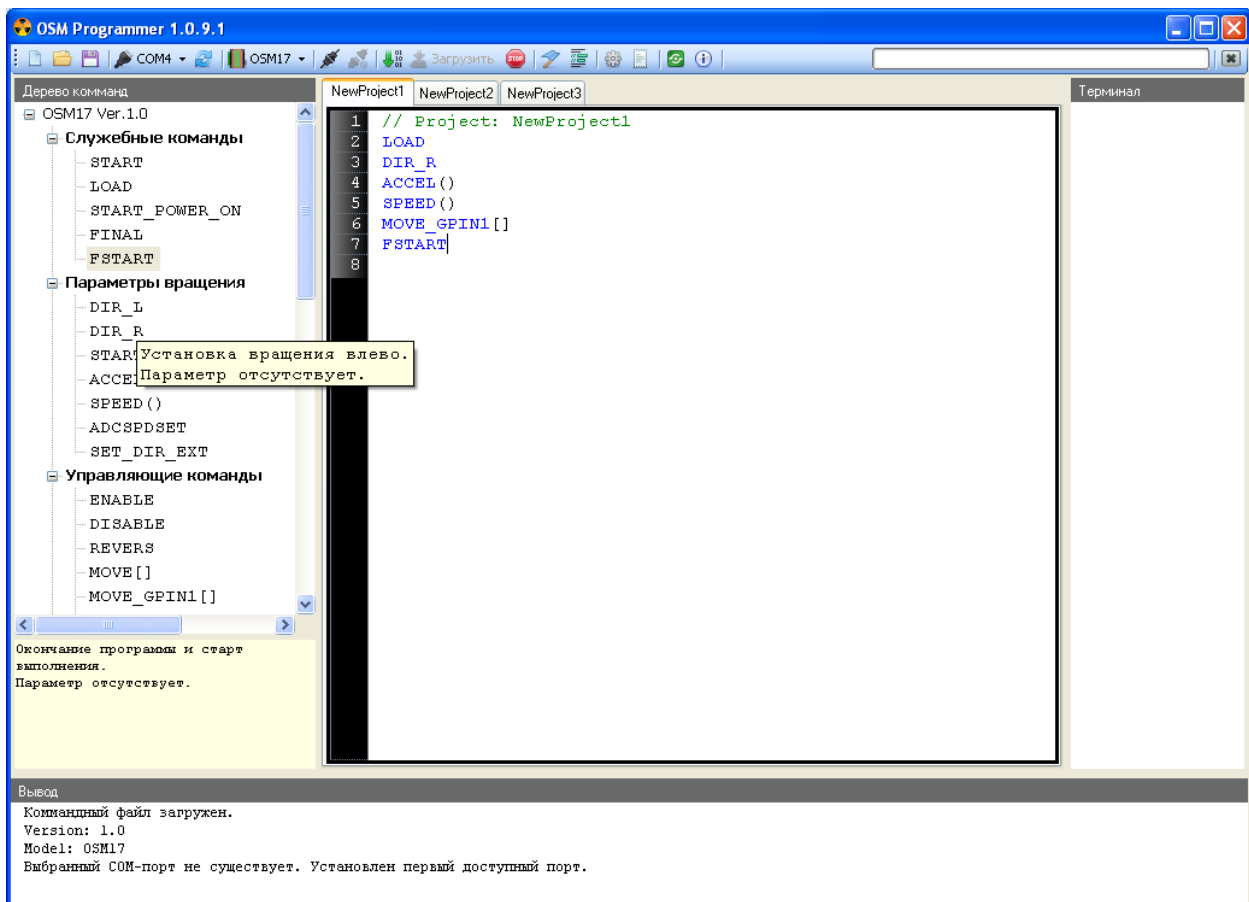


Рисунок 8.1 – Окно OSM Programmer

Редактор позволяет работать с несколькими проектами, используя вкладки окна проектов. В заголовке каждой вкладки отображается название текущего проекта.

Для начала работы с проектом необходимо выбрать используемый контроллер, так как набор команд может незначительно отличаться в различных версиях. Текст программы набирается в окне проекта. Для удобства все команды перечислены в дереве проектов в левой части окна. При наведении кнопки мыши на команду высвечивается краткое описание команды. Команда добавляется в проект по двойному щелчку. Существует четыре типа команд:

- Команды без параметра. В данных командах отсутствует числовой параметр, примером может служить команда задания направления вращения `DIR_R`.
- Команды с необязательным параметром. Параметр в таких командах указывается в квадратных скобках. Если параметр не нужен, скобки необходимо оставить пустыми. Примером команды с необязательным параметром является команда `MOVE[]`. Без параметра эта команда выполняет бесконечное движение, с параметром – движение на определенное число шагов.
- Команда с обязательным параметром. В случае обязательного наличия параметра у команды он указывается в круглых скобках. Примером может служить команда `SPEED()`, задающая скорость движения.
- Команда с несколькими параметрами. Параметры указываются в круглых скобках через запятую. Примером является команда `ACCEL(100,120,1000)`. Первый параметр задает начальную скорость, второй параметр – конечную, третий – ускорение.

После написания текста программы можно выполнить проверку кнопкой «Построить программу без загрузки». В случае наличия ошибок в тексте программы в окне вывода появляется отчет, содержащий перечень ошибок с указанием номера строки, содержащей ошибку, и ее описание. Строка с ошибкой подсвечивается красным цветом. Необходимо помнить, что программа не выводит ошибки, связанные с логикой исполнения команд. Наиболее частые ошибки связаны с неверным указанием параметров: отсутствием обязательного параметра или выхода значения за допустимые пределы.

Полный список команд, которые поддерживает контроллер, можно найти в [разделе 6](#). Там же приводятся описания каждой команды. В отличие от программирования на уровне протокола обмена, которое описано в разделе 7, программирование через OSM Programmer отличается тем, что вместо мнемоник команд используется более высокоуровневые команды, что позволяет значительно упростить написание алгоритма.

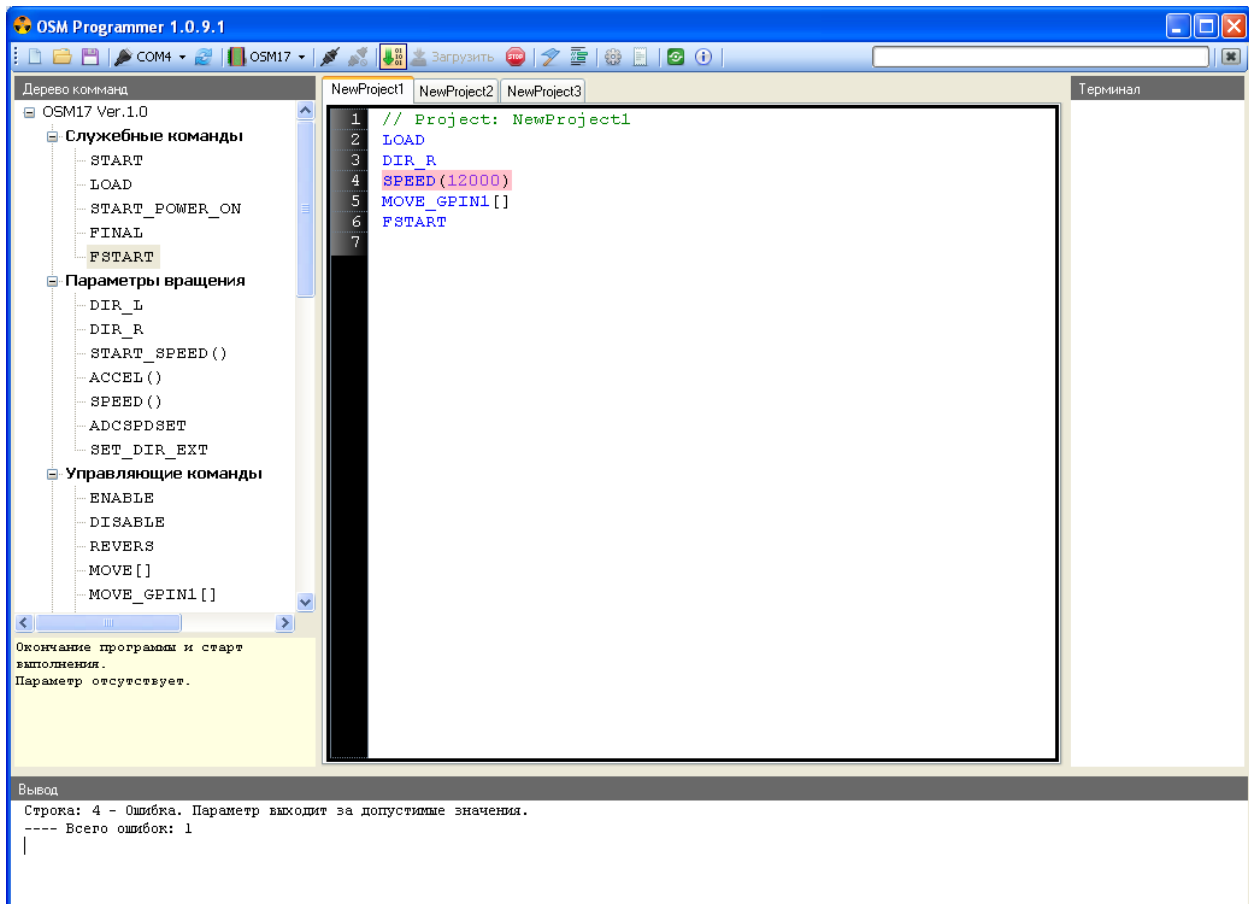


Рисунок 8.2 – Выделение ошибки

Если последовательность команд не содержит ошибок, в окне вывода появляется отчет об успешном выполнении построения программы с указанием размера программы и доступной памяти контроллера.

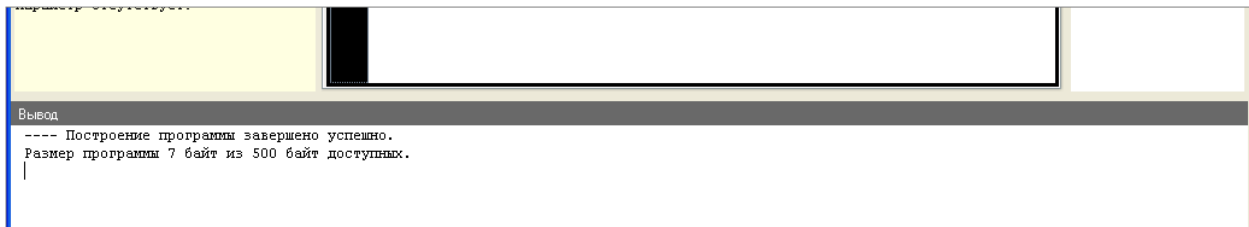


Рисунок 8.3 – Отчет построения программы

После успешного построения программы можно приступить к загрузке программы в контроллер. Необходимо подключиться к COM-порту, через который осуществляется обмен с контроллером, и нажать кнопку «Загрузить». Выбор порта осуществляется нажатием соответствующей кнопки в панели настроек. Программа предлагает к выбору только доступные порты. В случае, если требуемый порт отсутствует в списке, следует проверить его использование другими программами в текущий момент. Кроме того, в случае использования переходника USB->COM (виртуальный COM-порт) отсутствие нужного порта может означать проблему с драйверами или преобразователем. Необходимо убедиться в том, что виртуальный COM-порт правильно отображается в диспетчере устройств Windows. При необходимости можно обновить список доступных портов кнопкой «Обновить список портов».

В случае, если контроллер подключен и готов к приему команд, программа будет загружена в память, в противном случае будет выдано сообщение об ошибке. Необходимо помнить, что программа предназначена для работы в программируемом режиме. Если устройство находится в режиме контроллера, последовательность команд будет записана в оперативную память, а не в энергонезависимую, кроме того, в режиме контроллера не все команды являются применимыми.

Обмен контроллера и программы можно наблюдать в терминале. Так же в терминал выводятся сообщения контроллера об ошибках. Подробнее об обмене с контроллером через терминал можно прочитать в соответствующем разделе.

Для облегчения работы с программой предусмотрена возможность комментирования кода. Комментарий можно вставить после символа `//текст комментария`. Все, что находится справа от символа комментария, будет проигнорировано обработчиком.

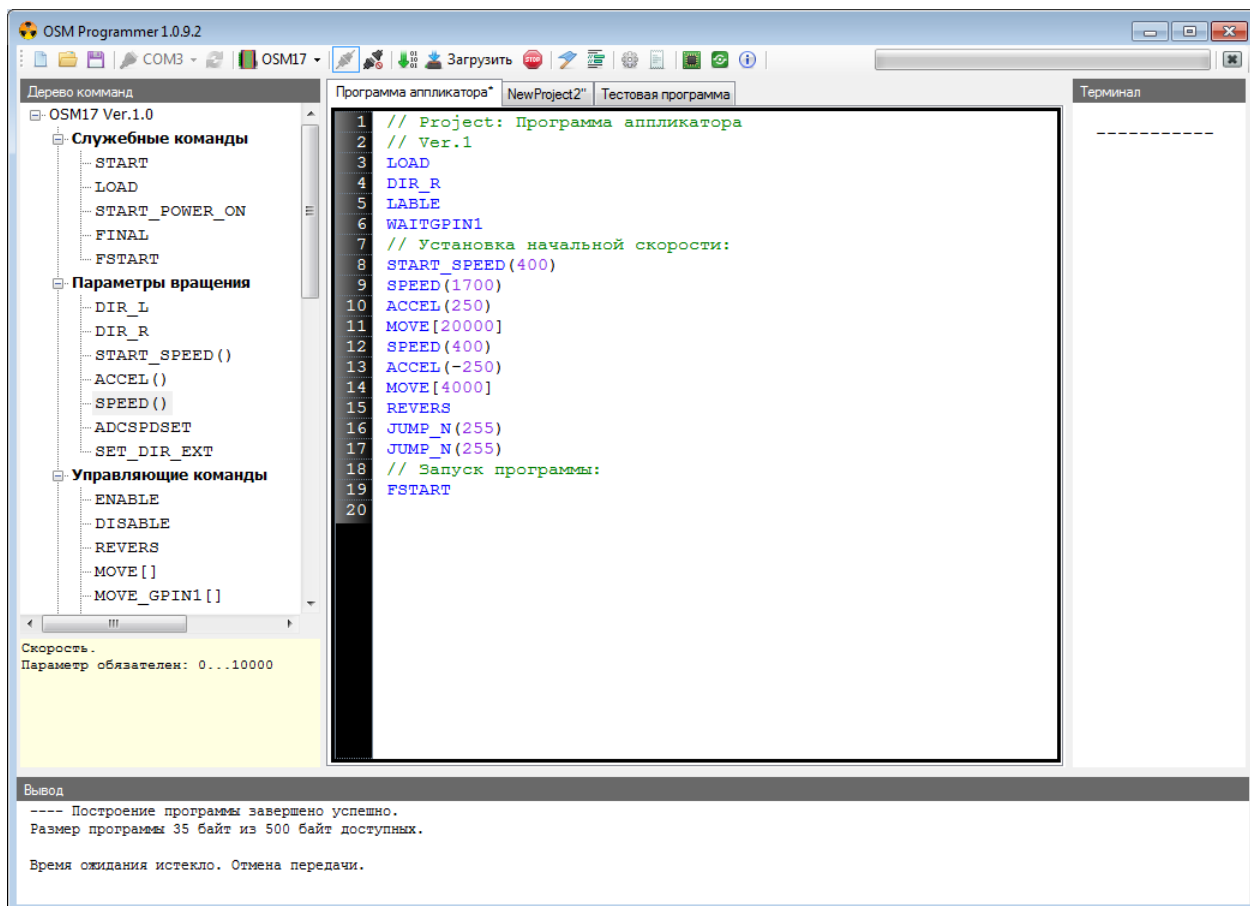


Рисунок 8.4 – Использование комментариев

Символ комментария можно так же использовать для временного исключения команды из кода программы при отладке.

Устройства OSM спроектированы таким образом, что в случае отсутствия программы OSM Programmer, можно работать с устройством через стандартные средства ОС Windows или любую подходящую терминальную программу с возможностью обмена ASCII сообщениями.

9. Использование SDK под ОС Windows®.

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

Комплект разработчика предназначен для быстрого добавления функций управления устройством в проект программного обеспечения разрабатываемого под платформу .NETFramework 3.5. Таким образом, поддерживаются такие среды разработки программного обеспечения как MicrosoftVisualStudio 2008, MicrosoftVisualStudio 2010, SharpDevelop, MonoDevelop, а также BorlandDeveloperStudio.

В состав OsmControl_SDK входит:

- библиотека OsmControl.dll под платформу .NET Framework 3.5;
- полная документация в формате MicrosoftCompiledHTMLHelp;
- пример использования библиотеки в виде готового проекта с задокументированным исходным текстом на языке C# в формате VisualStudio 2008 (совместим также с VisualStudio 2010 и SharpDevelop).

Библиотека OsmControl.dll предоставляет удобный для управления устройствами OSM API и позволяет легко задействовать в программном проекте следующий набор функциональных возможностей:

- Поддержка стандартной реализации протокола OSM.
- Функции для чтения/записи с обязательной явной типизацией данных для минимизации возможности появления ошибок.
- Автоматическое обеспечение многопоточности на уровне библиотеки:
 - синхронизация с основным потоком программы обеспечивается со стороны библиотеки;
 - операции работы с портом, операции чтения в регистры 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 и выходом из операции по таймауту.
- Автоматическое определение доступных для использования портов в системе.
- Улучшенная работа с последовательный портом.
- Использование кодов ошибок OSM.
- 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.

10. Использование SDK под OCLinux.

Для упрощения работы с последовательным портом можно использовать кроссплатформенную библиотеку `OsmControl.dll`, работающей в среде ОС Windows 7, Windows XP, Linux, MacOS. Библиотека предоставляет в распоряжение пользователя простой функционал для осуществления взаимодействия с последовательным COM портом, в том числе открытие/закрытие порта, передача и приём данных. Данная библиотека поддерживает возможность выбора номера порта, передачу данных в виде строки и посимвольное считывание данных из порта. В таблице 10.1 представлены основные методы, реализованные в данной библиотеке.

Методы	Описание
<code>openPort(int):int</code>	Посылает команду на открытие порта. Параметр - номер желаемого порта. Возвращаемые значения: 1 - успешное открытие порта, -1 - порт не был открыт.
<code>Close()</code>	Посылает команду на закрытие (открытого) порта.
<code>readData(char*,int):int</code>	Посылает команду на чтение данных из порта. Параметры: <code>char*</code> - указатель на первый элемент считанного элемента, <code>Int</code> - количество байт для считывания. Возвращает число прочитанных байтов. В случае, если не удалось считать данные или количество прочитанных байтов равно 0, возвращает -1. В случае, если обнаружена команда конца приёма записи данных (<code>end</code>) посылает 1. В случае, если порт возвратил код ошибки (<code>e5</code> , <code>e6</code>), возвращает -2.
<code>writeToPort(char*):int</code>	Посылает в порт список команд, представленных в виде массива символов. Параметр - указатель на первый элемент массива. Возвращаемые значения: 1 - данные были успешно записаны в порт, -1 - данные не были записаны в порт, -2 - при записи командах порт вернул код ошибки (<code>e5</code> , <code>e6</code>).

Более детальное описание методов, а также пример по их использованию можно найти в документации, поставляемой с каждой версией библиотеки.

11. Обновление встроенного программного обеспечения

Контроллер OSM-17R-2BL/OSM-42R-2BL имеет функцию удаленного обновления встроенного ПО, что позволяет пользователю скачивать с сайта производителя свежую версию программного обеспечения с дополнительной функциональностью, а так же 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 (рисунок 11.1).

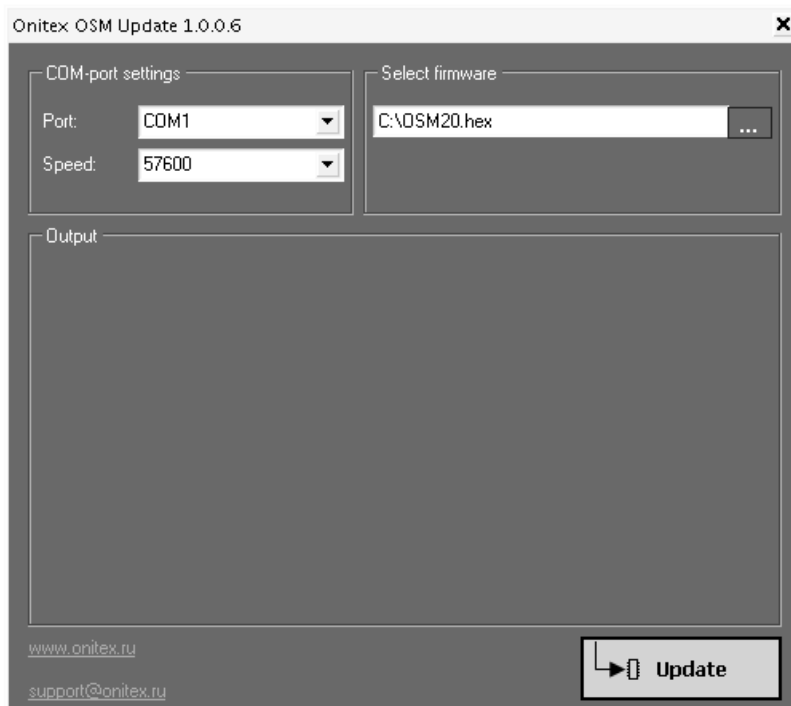


Рисунок 11.1 – Внешний вид программы OSM Update

Программа OSM Update не требует установки. После запуска необходимо выбрать порт, установить скорость 57600 кбит/с, а также выбрать скачанный файл прошивки контроллера. Контроллер должен быть подключен к ПК. Для входа в режим обновления программного обеспечения контроллер необходимо установить в автоматический режим и замкнуть на землю вход Home. После подачи питания контроллер будет готов к загрузке. Загрузка происходит после нажатия кнопки «Update».

После загрузки обновленного ПО необходимо отсоединить вход Home и перезагрузить контроллер включением-выключением питания или замыканием контакта Reset. После этого контроллер будет готов к работе.

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.

12. Типичные неисправности и их устранение.

В данном разделе приведен перечень наиболее распространенных неисправностей контроллера и способов их устранения. В случае обнаружения проблемы, не описанной в настоящем руководстве, необходимо обратиться к поставщику за получением консультации. Не допускается производить самостоятельный ремонт оборудования.

	Неисправность	Возможная причина	Устранение
1	После подачи питания контроллер не реагирует на команды (не выдает эхо), не запускает поиск начального положения, не работает в автоматическом режиме	Неправильная полярность питания. Отсутствие питания. Питание ниже или выше необходимого.	Проверить источник питания. Проверить соединительные кабели. Обратиться в техподдержку.
2	Контроллер ищет начальное положение, работает в автоматическом режиме, но не реагирует на сигналы по СОМ порту	Неверная распиновка кабеля. Выбран не тот СОМ-порт. Неверные настройки СОМ-порта. Повреждение кабеля.	Проверить распиновку кабеля. Проверить правильность выбора порта и настройки. Проверить целостность кабеля. В случае использования переходника попробовать другой переходник, проверить правильность установки драйверов. Проверить, не изменялась ли настройка порта при помощи команд. Для этого проверить работу на всех скоростях, поддерживаемых устройством.
3	Контроллер выдает непонятные символы по СОМ-порту	Неправильная настройка порта.	Проверить скорость и четность в настройках порта.
4	Команды принимаются, но двигатель не вращается. В автоматическом режиме вращения так же нет	Неправильное подключение двигателя. Слишком маленький ток.	Проверить подключение. Возможно, перепутаны фазы. Проверить соединение концов фаз при последовательном способе включения восьмифазного двигателя.
5	Команды принимаются, но двигатель не вращается. В автоматическом режиме вращение происходит, поиск начального положения работает и т.п.	Ошибка в алгоритме.	Проверить алгоритм, заданный в последовательности команд. Возможно, в алгоритме снимается напряжение с обмоток.

6	Двигатель очень сильно греется	Слишком большой ток.	Температура шагового двигателя может достигать 60-70 градусов. Если температура выше, необходимо проверить ток. Если ток выставлен правильно, но двигатель греется выше 100 градусов, проверить двигатель на сопротивление обмоток (возможно замыкание) и обратиться в техподдержку.
7	Момент ниже ожидаемого	Неверно выставлен ток Слишком низкое напряжение питания Неверный расчет момента	Проверить значение тока, уменьшение тока снижает момент двигателя. Момент двигателя падает с ростом оборотов. Для уменьшения падения момента на высоких оборотах можно увеличить питание (в пределах рабочего). Проверить расчеты требуемого момента двигателя с учетом того, что момент двигателя значительно падает на высоких оборотах. Сверить расчеты с графиком зависимости момента от частоты, приведенных исполнителем. В случае уверенности в правильности всех расчетов обратиться в техподдержку.
8	Двигатель не разгоняется до нужных скоростей	См. п. 7 – падение момента с ростом оборотов Двигатель конструктивно не способен работать на высоких оборотах. Слишком быстрое ускорение.	См. п. 7. Проверить характеристики двигателя и его график зависимости момента от частоты. Если график недоступен, обратить внимание на индуктивность и сопротивление обмоток. Высокое значение индуктивности и сопротивления свидетельствует о значительном падении момента с ростом частоты и невозможности работы двигателя на высоких частотах. Использовать более плавное ускорение
9	В режиме драйвера двигатель не вращается при подаче сигнала Step	Слишком низкая длительность импульса сигнала. Подан сигнал на вход EN	Увеличить скважность сигнала или ширину импульса. Снять сигнал со входа EN.