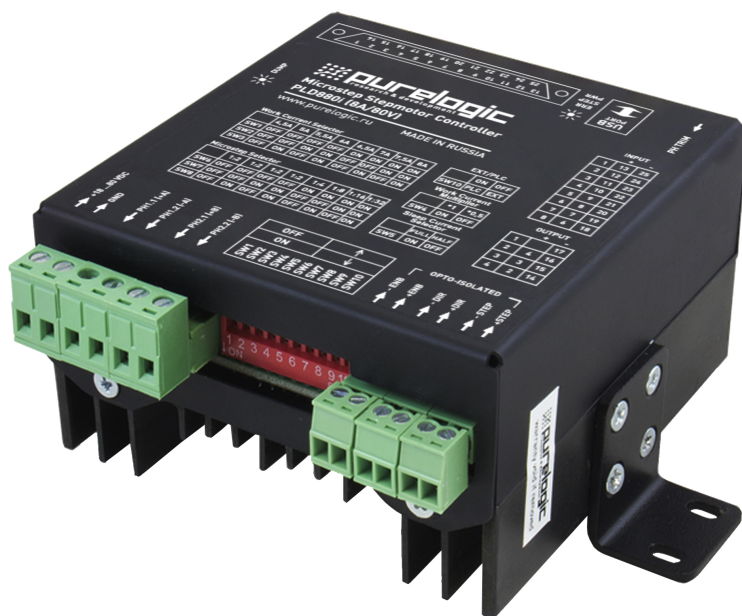


# ДРАЙВЕР ШАГОВОГО ДВИГАТЕЛЯ

# PLD8801



## СОДЕРЖАНИЕ

1. Введение .....	2
2. Характеристики и параметры продукции .....	3
3. Установка драйвера и вентиляция .....	6
4. Выбор режима работы модуля .....	7
5. Подключение устройства .....	7
6. Подключение ШД .....	9
7. Выбор токов и напряжений .....	11
8. Защитные функции .....	11
9. Выбор деления шага ШД .....	12
10. Описание языка программирования .....	12
11. Примеры программ для модуля .....	18
12. Маркировка, упаковка, хранение, транспортировка, утилизация .....	19
13. Гарантийные обязательства .....	21

## Используемые символы.



**Важная информация.**

Этот символ указывает на полезную дополнительную информацию.



**Внимание!**

Игнорирование таких предупреждений может привести к ошибкам или неправильному функционированию.

## Термины, аббревиатуры и сокращения.

В документе используются следующие термины, аббревиатуры и сокращения:

**БП** — блок питания.

**КЗ** — короткое замыкание.

**ПК** — персональный компьютер.

**ПО** — программное обеспечение.

**РЭ** — руководство по эксплуатации изделия.

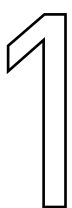
**СОЖ** — смазочно-охлаждающая жидкость.

**ЧПУ** — числовое программное управление.

**ШД** — шаговый двигатель.

**ШИМ** — широтно-импульсная модуляция.

**ЭДС** — электродвижущая сила.



## Введение.

- **Наименование товара:** драйвер шагового двигателя PLD880i.
- **Артикул:** PLD880i.
- **Комплект поставки:** драйвер шагового двигателя PLD880i — 1 шт.; регулировочная отвертка – 1 шт.; ответные части разъемов.

Разработано и произведено в России.



## Назначение документа.

Руководство по эксплуатации изделия (далее по тексту — РЭ) включает в себя общие сведения, предназначенные для ознакомления обслуживающего персонала с работой и правилами эксплуатации изделия «Драйвер шагового двигателя PLD880i» (далее по тексту — изделие или драйвер). Документ содержит технические характеристики, описание конструкции и принципа действия, а также сведения, необходимые для правильной эксплуатации изделия.

К работе с изделием допускаются лица, ознакомленные с настоящим руководством по эксплуатации. Изделие может обслуживать персонал, имеющий квалификационную группу по электробезопасности не ниже третьей.

В ходе эксплуатации изделия персоналу надлежит исполнять рекомендации, изложенные в отраслевой инструкции по защите от поражающего воздействия электрического тока.

Запрещается производить монтаж и демонтаж изделия при включенном электропитании изделия.

Предприятие-изготовитель оставляет за собой право производить непринципиальные изменения, не ухудшающие технические характеристики изделия. Данные изменения могут быть не отражены в тексте настоящего документа.

## 2 Характеристики и параметры продукции.

02

### Общие сведения.

PLD880i — современный и доступный микрошаговый драйвер со встроенным контроллером шагового двигателя (ШД).

Модуль состоит из двух объединенных на одной печатной плате устройств — драйвера ШД типа PLD86 и контроллера ШД типа PLC1Xi. Модуль может функционировать как обычный драйвер ШД, работающий по протоколу STEP/DIR/ENABLE и как контроллер ШД, работающий по программе автономно.

### Основные возможности драйвера PLD880i.

#### Часть контроллера:

Контроллер предназначен для управления ШД, согласно заданной программе и состоянию управляющих входов. В контроллере реализованы аппаратный разгон/торможение ШД, функции останова/запуска управляющей программы, ручное управление ШД (JOG), управление частотным преобразователем, отслеживание состояний 8 оптоходов, управление 4 оптовыми выходами.

Контроллер имеет собственный язык программирования, схожий с языком C (поддерживаются операторы циклов, условных/безусловных переходов, работа с переменными и пр.), что позволяет гибко построить управляющую программу и делает контроллер максимально функциональным.

Программирование, настройка и управление контроллером осуществляется через специальную программу разработки Purelogic R&D или в терминальном режиме через любую стандартную терминальную программу, в качестве интерфейса используется USB (виртуальный COM-порт).

Исполнительная программа хранится в энергонезависимой памяти контроллера. После программирования контроллера ПК не нужен, контроллер может работать автономно. Контроллер поддерживает удаленное обновление встроенного ПО для последующего расширения функционала.

#### Часть драйвера:

Драйвер построен с применением аналоговых и цифровых компонентов, что позволяет добиться максимальной функциональности. Драйвер имеет встроенные цепи защиты от КЗ обмоток ШД, от эффекта обратной ЭДС от ШД (дампер), автоматический компенсатор среднечастотного резонанса ШД, схему изменения формы фазовых токов с увеличением частоты (морфинг), защиту от переплюсовки напряжения питания (драйвер не включится) и схему плавного пуска ШД.

Устройство работает со стандартным протоколом управления STEP/DIR/ENABLE. Все управляющие входы драйвера оптоизолированы и совместимы с логическими уровнями 2.5 В, 3.3 В, 5 В. Также модуль снабжен режимом AUTO-SLEEP, который включается при отсутствии сигнала STEP.

Драйвер оптимально подходит для управления биполярными и униполярными шаговыми двигателями Purelogic R&D серий PL57 и PL86. Также возможна работа и с другими ШД.



**Внимание!**

Все подключения и изменения режимов работы устройства производить только при отключенном источнике питания.

Запрещается установка размыкателя (выключателя) питания после источника питания (на линии питания драйвера). Устанавливать размыкатель допускается только до блока питания, со стороны ~220 В.

Запрещается последовательное подключение драйверов по питанию, допускается только соединение типа «звезда» (своя линия питания для каждого драйвера, подсоединяется к БП).

Запрещается соединение «-» источника питания с заземлением, массой, корпусом и т.д. Строго соблюдайте полярность подключения источника питания и управляющих сигналов.

Во время работы, драйвер и ШД могут нагреваться, в зависимости от установленного рабочего тока. Если температура ШД и радиатора контроллера превышает 60°C – необходимо обеспечить дополнительный отвод тепла при помощи вентилятора. Рабочий ток ШД указан на корпусе ШД, без необходимости не рекомендуется превышать это значение.

**Технические характеристики.**

Параметр	Значение
Напряжение питания	18...80 В (типичное значение 75 В)
Интерфейс управления	USB (виртуальный COM-порт, FT232RL) Сигналы STEP/DIR/ENABLE
Размер управляющей программы	2 кбайт, максимум
Макс. частота сигнала STEP контроллера	100 кГц
Частота сигнала STEP драйвера / фронт	макс. 270 кГц / передний
Макс. частота вращения вала ШД	42 об/сек или 2530 об/мин (PL86H113, 75 В, деление шага 1:32, 270 кГц)
Число входов	8, оптовоходы (светодиод + 1 кОм, оптопара)
Число выходов	4, оптовыходы (открытый коллектор, оптопара, 50 В/10 мА MAX)
Сопротивление изоляции	500 МОм
Рабочая температура	0...50°C
Вес модуля без упаковки	0.3 кг
Габаритные размеры (Ш x В x Г)	110 x 33 x 95 мм

02

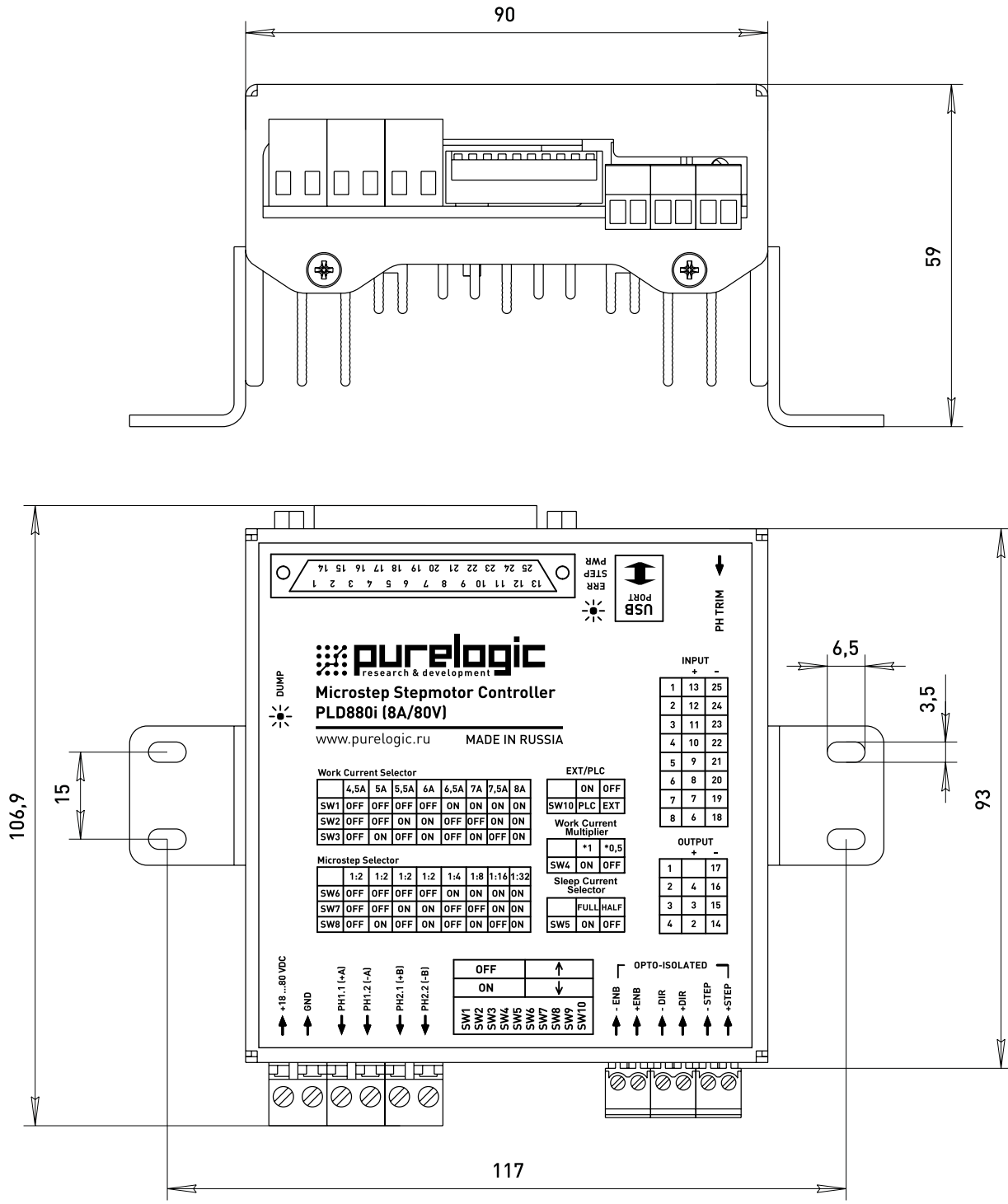


Рисунок 1 — Размеры драйвера ШД PLD880i.

# 3

## Установка драйвера и вентиляции.

С целью обеспечения оптимального теплового режима монтаж оборудования внутри стойки управления ЧПУ необходимо производить, придерживаясь схемы, приведенной ниже:

03

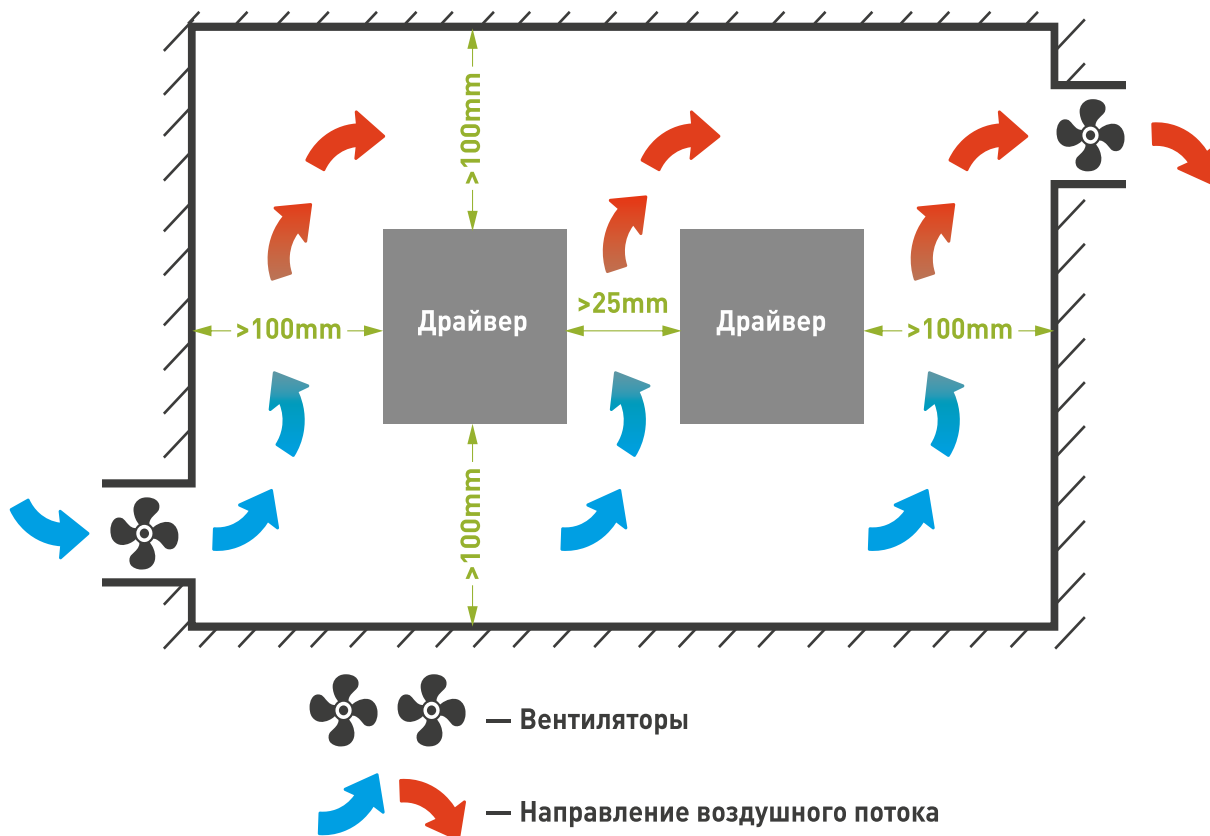


Рисунок 2 — Схема установки драйвера.

### Условия окружающей среды.

- 1) Рабочая температура: 0 ~ 45 °С.
- 2) Рабочая влажность окружающей среды: ниже 40% ~ 80% (без конденсации).
- 3) Температура хранения: -40 ~ 55 °С.
- 4) Влажность окружающей среды при хранении: ниже 80% (без конденсации).
- 5) Вибрация меньше 0.5G.
- 6) Предотвратить попадание влаги, прямых солнечных лучей, масляного тумана и солей, разъедающих сжиженных газов.

# 4 Выбор режима работы модуля.

Модуль состоит из двух объединенных на одной печатной плате устройств — драйвера ШД типа PLD880 и контроллера ШД типа PLC1Xi. Модуль может функционировать как обычный драйвер ШД, работающий по протоколу STEP/DIR/ENABLE (SW10=EXT) и как контроллер ШД, работающий по программе автономно (SW10=PLC) с программированием по USB.

05

# 5 Подключение устройства.

## Подключение входных сигналов.

Модуль поддерживает работу с 8 независимыми входными сигналами (IN1...IN8). Физически каждый вход представляет собой вход оптопары (светодиод + резистор 1 кОм). При подаче напряжения +5 В на оптовход происходит запрограммированное событие. Контакты 6, 7, 8, 9, 10, 11, 12, 13 - это «+» входа, контакты 18, 19, 20, 21, 22, 23, 24, 25 - это «-» входа. В качестве источника сигнала для оптовхода может служить любой концевой датчик, источник ТТЛ сигнала и пр. Зарезервированный вход IN9 соединен с сигналом аварии драйвера ШД и может использоваться для аварийной остановки программы. Полярность сигнала аварии изменяется переключателем SW9.

## Подключение внешних устройств.

Контроллер поддерживает управление 4 независимыми выходными сигналами (O1...O4). Физически каждый выход представляет собой выход оптопары (открытый коллектор).

Контакты 2, 3, 4, 5 - это «коллектор» выхода, контакты 14, 15, 16, 17 - это «эмиттер» выхода. Управление выходами происходит согласно управляющей программе.

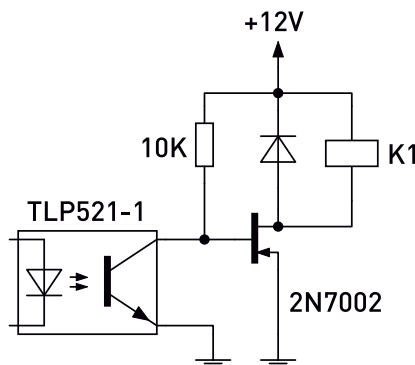


Рисунок 3 — Подключение реле к выходу управления.

## Подключение модуля к ПК.

Контроллер подключается к любому свободному порту USB ПК. После подключения в системе появляется виртуальный COM-порт (мост FT232RL). При первом подключении возможно понадобится установка драйверов.

На ПК должен быть установлен пакет Microsoft.NET framework версии не ниже 2.0. Он уже присутствует в ОС Windows XP SP3 и в Windows 7. Для управления модулем необходимо скачать программу с сайта Purelogic R&D.

## Подключение сигналов управления в режиме «драйвер».

Для управления модулем в режиме «драйвер» используются стандартные сигналы STEP/DIR/ENABLE. Сигналы подаются на дифференциальные оптоизолированные входы. DIP-переключатель SW10=EXT.

На рис. 4 схематично показано устройство дифференциальных входов управления и метод подключения к системе управления (контроллеру) с выходами типа «открытый коллектор».

Параметры сигнала STEP: рабочее напряжение 2.5 В, 3.3 В, 5 В (возможно, понадобится подключение дополнительного токоограничивающего резистора), ток потребления до 20 мА, минимальная длительность сигнала 2 мкс. Шаг ШД осуществляется по переднему фронту сигнала.

Параметры сигнала DIR: рабочее напряжение 2.5 В, 3.3 В, 5 В (возможно, понадобится подключение дополнительного токоограничивающего резистора), ток потребления до 20 мА, время срабатывания 200 нс до/после переднего фронта STEP.

Параметры сигнала ENABLE: рабочее напряжение 2.5 В, 3.3 В, 5 В (возможно, понадобится подключение дополнительного токоограничивающего резистора), ток потребления до 20 мА, время срабатывания 100 мкс. Логическая единица (подано напряжение на вход) — драйвер ШД выключен и обмотки ШД обесточены, ноль (ничего не подано или 0В на вход) — драйвер ШД включен и обмотки ШД запитаны.

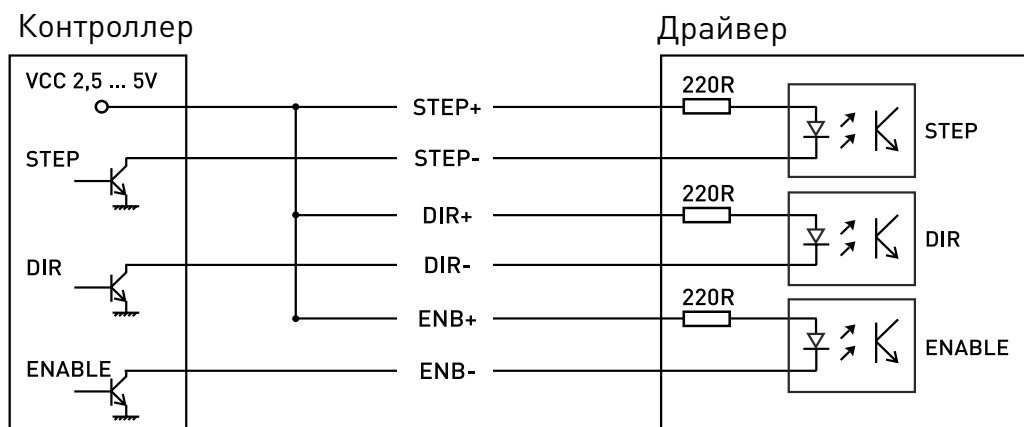


Рисунок 4 — Устройство дифференциальных входов.



## 6

## Подключение ШД.

Модуль оптимально подходит для управления биполярными и униполярными шаговыми двигателями Purelogic R&D серий PL57, PL86.

Подключение ШД к драйверу осуществляется согласно рис. 5 (клеммы PH1.1[+A], PH1.2[-A] и PH2.1[+B], PH2.2[-B]). Драйвер имеет защиту от неправильного подключения обмоток ШД и от КЗ обмоток ШД между собой / на «+» питания.

Подключение ШД производства Purelogic R&D к драйверу осуществляется согласно рис. 4. Обратите внимание, если поменять местами пары (фазы) ШД PH1.x<>PH2.x, то двигатель начнет вращаться в противоположную сторону (аналог инверсии сигнала DIR).

Длина проводов, идущих к ШД от драйвера, не должна превышать 10 метров. Более длинные провода могут привести к сбоям в работе драйвера. Настоятельно рекомендуется пофазно переплести между собой провода ШД, полученные жгуты уложить в экранирующие металлические оплетки. Оплетки и корпус ШД должны быть заземлены.


PH1.1 [+A]		Красный (RED)
PH1.2 [-A]		Оранжевый (ORG)
PH2.1 [+B]		Голубой (BLU)
PH2.2 [-B]		Зелёный (GRN)

Рисунок 5 — Подключение ШД производства Purelogic R&D.

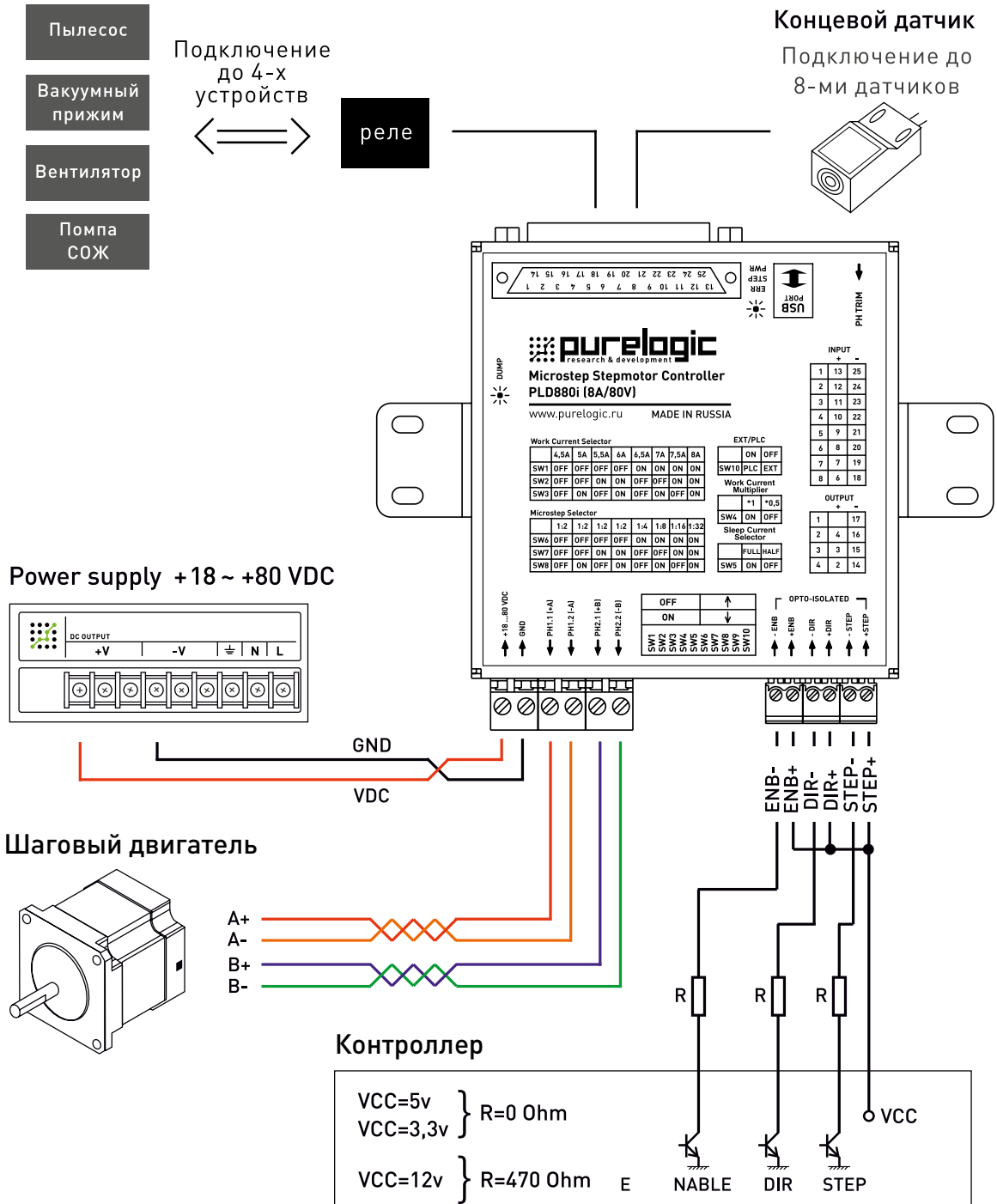


Рисунок 6 — Подключение драйвера-контроллера PLD880i.

## 7

## Выбор токов и напряжений.

Выбор максимального напряжения питания драйвера зависит от применяемого ШД и желаемой максимальной скорости его вращения. Расчет оптимального напряжения питания для данного ШД производится по формуле  $U=32 \cdot V$  (индуктивность фазы ШД в мГн), но не более 80 В.

Ток источника питания нужно выбирать с расчетом 50...70% от заявленного тока обмотки ШД. Установку рабочего тока необходимо осуществлять DIP-переключателями SW1, SW2, SW3.

Потенциометром PHASE TRIM осуществляется подстройка фазных токов (плавности хода ШД) при обнаружении вибраций и неравномерности шага на низких оборотах вращения.

Алгоритм подстройки следующий: 1) Потенциометр устанавливается в крайнее левое положение (минимальное значение). 2) Подключается ШД, подается напряжение питания, генератором частоты STEP устанавливается частота, на которой заметны вибрации. 3) Плавно вращая потенциометр PHASE TRIM по часовой стрелке, устраняем вибрацию. Возможно, что полностью устранить вибрации не получится, из-за особенностей конкретного ШД. Возможно, убрав вибрации на одной частоте, они появятся на другой. Поэтому необходимо выбрать наиболее приемлемый вариант настройки для конкретного применения.

При отсутствии сигнала STEP больше 1 секунды, драйвер переходит в спящий режим (режим AUTO-SLEEP) и снижает ток обмотки на значение, установленное DIP-переключателем SW6. Это позволяет уменьшить нагрев ШД при простое и снизить потребление тока.

Все переключения осуществлять при выключенном питании драйвера.

## 8

## Защитные функции.

Защита от переплюсовки напряжения питания — драйвер не включится.

Защита от КЗ обмоток ШД — от неправильного подключения обмоток ШД и от КЗ обмоток ШД между собой / на «+» питания. В некоторых случаях при межвитковом КЗ обмотки ШД, защита от КЗ не срабатывает, поскольку не происходит превышение допустимого аварийного тока. Это происходит потому, что при таком КЗ сопротивление обмотки не становится равным 0 и драйвер продолжает поддерживать в ней заданный ток.

Защита от КЗ на плате драйвера — драйвер выключается, если потребляемый ток превышает максимальное значение.

Защита от скачков питающего напряжения — защита включается при подаче напряжения питания больше 90 В, для защиты драйвера от эффекта «обратной ЭДС» ШД.

Встроенный демпфер — устройство компенсации обратной ЭДС от ШД.

Плавный пуск ШД — после включения напряжения питания или подачи сигнала ENABLE, ток в обмотках ШД нарастает постепенно. Это позволяет исключить характерный «удар» при включении ШД.

Подстройка фазных токов — исключение вибраций на низких частотах.

Автоматический компенсатор среднечастотного резонанса ШД — электронная схема, позволяющая устранить эффект среднечастотного резонанса. Включается/отключается автоматически при обнаружении резонанса. Резонанс обычно проявляется в диапазоне 6-12 об/сек.

Защита от перегрева — драйвер имеет встроенный датчик температуры, который измеряет температуру радиатора. При достижении температуры радиатора ~60°C, драйвер отключается. Драйвер включится после того, как температура упадет до ~50°C.

После включения питания загораются зеленый светодиод PWR (индикация поданного напряжения питания), красный светодиод ERR. При отсутствии аварии, через 1 сек. красный светодиод ERR гаснет. При наличии частоты STEP, меняется интенсивность свечения светодиода STEP.

При срабатывании защиты, драйвер отключает обмотки ШД, загорается красный светодиод ERR ошибки, желтый светодиод STEP гаснет и выдается сигнал ошибки на IN9 контроллера.

Переключатель SW9 инвертирует сигнал ошибки IN9. Драйвер восстанавливает работоспособность после устранения причины аварии и выключения/включения напряжения питания.

## 9 Выбор деления шага ШД.

Драйвер позволяет электронным способом делить физический шаг ШД на целое значение – 2, 4, 8, 16 и 32. Таким образом повышается точность позиционирования. Режим с делением шага называется режимом микрошага. Режим микрошага позволяет уменьшить резонансы ШД на низких оборотах.

Рекомендуется использовать драйвер в режиме деления шага 1:8, 1:16 или 1:32. В этих режимах ШД вращается максимально плавно, низкочастотные вибрации минимальны и оптимально работает автоматический компенсатор среднечастотного резонанса ШД.

Переключение режима деления шага осуществляется DIP-переключателями SW6, SW7, SW8. Выбор деления шага необходимо осуществлять только при выключенном питании драйвера.

## 10 Описание пунктов меню программы управления.

Для управления контроллером разработан специальный язык программирования. На этом языке пользователем может быть написана программа, выполняющая желаемую последовательность действий. Интерпретатор поддерживает работу с внутренними переменными, аппаратными входами и выходами, а также условные операторы и переходы по строкам программы.

Интерпретатор языка не чувствителен к регистру написания команд. Команда должна быть записана в одну строку, без переноса на другую строку.

## Арифметические и логические операции.

Язык программирования не поддерживает различные типы данных. Все переменные, значения входов/выходов и пр. имеют единый тип данных – целое число в диапазоне от -2147483648 до 2147483647. При этом, если значение должно быть интерпретировано как логическое, то число 0 принимается за «ложь», а любое ненулевое число – «истина».

Язык поддерживает следующие математические и логические операторы, перечисленные ниже (в порядке убывания приоритетов вычисления):

«-»	Унарный минус. Изменения знака операнда
«!»	Логическое отрицание
«*», «/»	Арифметическое умножение и деление
«+», «-»	Арифметическое сложение и вычитание
«<», «<=», «>», «>=»	Логические операции сравнения
«=», «!=»	Проверка на равенство/неравенство
«&»	Побитовое «И»
« »	Побитовое «ИЛИ»
«&&»	Логическое «И»
«  »	Логическое «ИЛИ»

Описанные выше операции являются подмножеством операций языка программирования C и переменные имеют тип «int».

Результаты вычисления выражений также полностью совпадают с аналогичными выражениями языка C.

## Предопределенные переменные.

Язык программирования имеет возможность работать с переменными. Переменная – это некоторая область памяти, которая может хранить одно значение (число). Каждая переменная имеет свое имя.

Контроллер оснащен несколькими аппаратными (дискретными) входами и выходами, управлять которыми можно из пользовательской программы. С точки зрения языка программирования аппаратные входы/выходы являются обычными переменными с одним ограничением: если в переменную, связанную с конкретным аппаратным входом, произвести запись некоторого значения, то ничего не произойдет.

При последующем чтении данных из этой переменной программа получит реальное значение входа (0 или 1), а не записанные туда данные. Интерпретатор не позволяет использовать произвольные имена для переменных.

Контроллер имеет следующие предопределенные переменные:

v1..v9 – переменные общего назначения. При старте пользовательской программы они имеют неопределенные (случайные) значения. Их можно использовать для любых целей.

inp1..inp8 – эти 8 переменных «связаны» с аппаратными входами. Если на соответствующем входе установлен логический «0», переменная имеет нулевое значение. В противном случае значение переменной будет неизвестным, но отличным от нуля числом. Запись в эту переменную игнорируется.

out1..out5 – эти 5 переменных «связаны» с аппаратными выходами. При записи ненулевого значения в переменную, на соответствующем выходе появится логическая единица. При записи нуля выход также становится нулевым. Чтение из этой переменной возвращает текущее состояние выхода.

rel1..rel3 – эти 3 переменных «связаны» с аппаратными релейными выходами. При записи ненулевого значения в переменную, на соответствующем выходе появится логическая единица. При записи нуля выход также становится нулевым. Чтение из этой переменной возвращает текущее состояние выхода.

PWM1 – переменная, определяет скважность управляющего ШИМ сигнала. Может принимать значения в диапазоне от 0 до 100 (попытка записи в переменную числа вне диапазона приведет к установке ближайшего значения из разрешенного диапазона). Изменяя значение переменной от 0...100, можно изменять напряжение на клемме OUT в диапазоне 0...10 В.

Emerg – переменная экстренной остановки программы, если одной кнопки E-STOP недостаточно. Переменной можно присвоить любое значение (выражение), когда значение переменной станет ненулевым (т.е. истинным) программа остановится.

Position – текущее положение, выраженное в шагах привода. Этой переменной может быть присвоено любое значение и во время перемещений значение будет автоматически корректироваться. Это позволяет, например, определить длину предыдущего перемещения, если оно было прервано сработавшим датчиком.

Помимо перечисленных переменных, в контроллере существуют также встроенные служебные переменные, которые отвечают за настройку параметров контроллера. Изменять их можно также, как и любые другие переменные. Их имена начинаются с символов «P\_». Значения этих переменных автоматически сохраняются в ПЗУ одновременно с сохранением программы. В данной версии контроллера определены следующие параметры:

P\_STEPS\_MM – количество импульсов «STEP» на каждый миллиметр перемещения. Например, если двигатель имеет 200 шагов на оборот (стандартный ШД с угловым шагом 1.8°, например, PL57H76), шаг винта 5 мм и драйвер ШД настроен на деление шага на 8, то количество импульсов должно быть установлено в  $(200*8)/5 = 320$ .

P\_FEED\_MAX – максимально разрешенная для данной системы подача в мм/мин. Это значение будет использовано при перемещении, если в команде явно не указана другая скорость подачи. Кроме того, если в команде перемещения будет задана скорость подачи выше указанной в данном параметре, то она автоматически будет ограничена до P\_FEED\_MAX.

P\_FEED\_JOG – максимально разрешенная для данной системы подача в мм/мин в режиме ручного управления JOG.

P\_ACC\_TIME – время в миллисекундах, за которое двигатель должен разогнаться до скорости P\_FEED\_MAX. Задаёт профиль ускорения двигателя.

P\_BUT\_START – задает номер входа, на котором установлена кнопка «СТАРТ». 0 – кнопки «СТАРТ» нет.

P\_BUT\_STOP – задает номер входа, на котором установлена кнопка «СТОП». 0 – кнопки «СТОП» нет.

P\_BUT\_PLUS – задает номер входа, на котором установлена кнопка «ВПЕРЕД» в режиме ручного управления JOG. 0 – кнопки «ВПЕРЕД» нет.

P\_BUT\_MINUS – задает номер входа, на котором установлена кнопка «НАЗАД» в режиме ручного управления JOG. 0 – кнопки «НАЗАД» нет.

### Математические выражения.

Выражением называется некоторая последовательность чисел, переменных и математических операций, которую можно вычислить. Для изменения порядка вычисления в выражении могут использоваться традиционные круглые скобки.

Примеры выражений:

$V1 + (5 * V3) - 10 <= 15$  или  $!inp1 \ \&\& \ inp5$

Выражения делятся на:

Однократно вычисляемые – выражение заключено в круглые скобки (), или скобки вообще не используются.

Динамически отслеживаемые – выражение заключено в квадратные скобки [].

Пример однократно вычисляемого выражения:

SET v2, 2

SET v3, 3

SET v1, [v2 + v3]

В данном примере в переменную v1 будет помещен результат сложения v2 и v3 и теперь значение переменной v1 станет равным 5.

Пример динамически отслеживаемого выражения:

SET v2, 2

SET v3, 3

SET v1, [v2 + v3]

SET v4, v1

SET v2, 3

SET v5, v1

В данном примере в переменную v1 будет помещен не результат сложения v2 и v3, а формула «v2+v3». В четвертой строке примера происходит присваивание переменной v4 значения, хранящегося в v1. И именно в данный момент будет вычислено значение переменной v1 по формуле v2+v3 и результат (5) будет помещен в v4. В дальнейшем мы можем изменить значение переменной v2 (5-я строка) и теперь значение переменной v1 станет равным 6.

### Метки.

Перед любой командой может быть помещена метка, или, говоря другими словами, «название» данной строки программы. Метка может быть использована для произвольного перехода между строками программы. Метка должна состоять из латинских букв и/или цифр, причем первым символом обязательно должна быть буква или символ «\_», а в конце метки должно стоять двоеточие.

Примеры правильной метки: «M:», «\_Label1:».

### Комментарии.

Символ «#» в любом месте строки означает начало комментария. Любые символы после «#» не воспринимаются как команда.

Пример комментирования:

SET v2, 2 # пусть v2=2;

### Команды языка.

**Команда – MOVE C(<выражение>) L(<выражение >) F(<выражение>) D(<выражение >).**

Данная команда выполняет перемещение. Команда имеет 4 аргумента:

C(<выражение>) – выполняется перемещение, пока указанное выражение ложно. Выражение всегда интерпретируется как динамическое, т.е. независимо от того, задана ли команда в виде MOVE C([inp2]) или MOVE C(inp2), выражение в скобках будет перепроверяться постоянно в течение выполнения команды и в момент, когда оно станет истинно (говоря другими словами – станет ненулевым, пусть «1»), перемещение будет остановлено.

D(<выражение>) – задает направление перемещения. Если выражение истинно, то выход «DIR» устанавливается в «1», если ложно, то «DIR» имеет низкий уровень. Это выражение может быть задано динамически, т.е. если в процессе движения оно изменит свое значение, то произойдет изменение направления вращения двигателя.

L{<выражение>} – выполняется перемещение на расстояние, заданное в сотых долях миллиметра, т.е. 100 единиц это 1 мм. При задании длины перемещения признак динамического выражения игнорируется. Это означает, что даже если длина задана в виде L{[v1\* 100 + v4]}, указанное выражение будет вычислено только один раз перед началом движения. Выражение может быть и отрицательным. В этом случае направление перемещения будет изменено на обратное.

F{<выражение>} – задает желаемую скорость линейного перемещения в мм/мин. Так же, как и в случае с длиной перемещения, признак динамического выражения игнорируется, т.е. F{v1} и F{[v1]} полностью идентичны. Скорость подачи будет вычислена один раз перед началом движения, и во время перемещения будет оставаться неизменной. Если задана скорость подачи, большая, чем параметр P\_FEED\_MAX, то будет установлена максимальная разрешенная подача, равная P\_FEED\_MAX.

Любой из параметров команды MOVE может быть опущен, тогда вместо него будет подставлено значение по умолчанию.

Обязательными являются хотя бы один из параметров C или L. Такое построение команды дает возможность реализовывать, например, бесконечные перемещения, или бесконечные перемещения, пока не произойдет событие и пр.

### Примеры использования команды MOVE:

MOVE L(1000) – переместиться «вперед» на 10 мм с максимальной подачей;

MOVE L(-100) F(10) – переместиться «назад» на 1 мм с подачей 10 мм/мин;

MOVE L(-1000) D(0) – переместиться «вперед» на 10 мм, максимальная подача;

MOVE L(500) D([inp1]) – переместиться на 5 мм, максимальная подача, с динамическим отслеживанием направления по входу inp1;

MOVE C(inp1) D(0) – перемещаться «назад», пока на первом входе не появится «1»;

MOVE C(!inp1) D(1) F(10) L(100) – подача 10 мм/мин, движемся «вперед», пока на первом входе держится высокий уровень, но не более 1 мм.

MOVE C(inp1) D(inp2) – перемещаться в направлении, заданном входом IN2, пока на входе IN1 держится низкий уровень. При изменении значения второго входа во время движения, направление перемещения также изменяется.

### Команда – WAIT C{<выражение>} T{<выражение >}.

Данная команда выполняет ожидание. Команда может иметь 1 или 2 аргумента:

C{<выражение>} – выполняется ожидание, пока указанное выражение ложно. Выражение всегда интерпретируется, как динамическое, т.е. независимо от того, задана ли команда в виде WAIT C([inp2]) или WAIT C(inp2), выражение в скобках будет перепроверяться постоянно в течение выполнения команды и в момент, когда оно станет истинно (говоря другими словами – станет ненулевым), ожидание будет остановлено.

T{<выражение>} – задает желаемое время ожидания в миллисекундах.

Любой из параметров команды WAIT может быть опущен.

### Примеры использования команды WAIT:

WAIT T(100) – ждать 0.1 секунды;

WAIT C(inp1) T(3000) – ожидать 3 секунды, либо пока на входе IN1 не появится логическая единица.

### Команда – SET <переменная>,<выражение>.

Данная команда присваивает переменной заданное выражением значение. Если выражение задано в квадратных скобках, происходит «связывание» переменной с указанной в выражении формулой и при каждом обращении к переменной, ее значение будет пересчитываться.



**Примеры использования команды SET:**

SET v1, 5 – в переменную v1 помещаем число 5;  
 SET v2, v1 + 1 – в переменную v2 помещается число, равное значению V1+1;  
 SET v3, [v1 + 1] – в переменную v3 помещается формула «v1+1».

**Команда – ALIAS <переменная>, <новое имя>.**

Данная команда присваивает переменной еще одно альтернативное имя. Оно не сохраняется в энергонезависимой памяти и действительно только в течение одного сеанса работы. Может быть использовано внутри пользовательской программы для упрощения понимания программы.

**Примеры использования команды ALIAS:**

```
ALIAS v1, counter
#пусть переменная v1 называется counter;
LOOP: SET counter, counter + 1
#увеличим счетчик counter на 1;
MOVE L(10)
#двигаемся на 0.1 мм;
IF (counter < 10) JMP LOOP:
#переход на метку LOOP: пока counter<10;
```

**Команда – IF (<выражение>) <команда>.**

Данная команда проверяет условие, заданное выражением в скобках, и, если оно истинно, т.е. не равно нулю, выполняется <команда>. В качестве команды может быть задана любая команда языка.

**Примеры использования команды IF:**

IF (v1 = 10 && (v2 + v3) < 0) JMP Label1: – если переменная v1 равна 10 и при этом v1 + v3 меньше нуля, перейти на строку с меткой «Label1:»;  
 IF (inp1) IF (10 + v1 > 100) MOVE L(v1 \* 100) C[(!inp1)] – если на входе IN1 высокий уровень и если (10+v1)>100, то перемещаться на расстояние не более V1 мм, пока на входе IN1 не появится «0».

**Команда – MEM.**

Сохраняет текущие значения аппаратных выходов во внутренней памяти. Эти значения будут автоматически восстановлены из памяти при завершении работы программы независимо от причины остановки (кнопка E-STOP или нормальное завершение программы).

**Пример использования команды MEM:**

Необходимо, чтобы после окончания программы выходы №1 и №2 были нулевыми, а №3 и №4 – активными.

В начале программы напишем:

```
set out1, 0
set out2, 0
set out3, 1
set out4, 1
mem
```

После выполнения программы или ее остановки (нормальное завершение программы или кнопка E-STOP), выходы вернуться к состоянию, в котором находились на момент выполнения команды MEM (в этом примере – out1 и out2 станут нулевыми, а out3 и out4 – активными).

### Команда – JMP <метка>.

Переход на указанную метку в программе. Строка с такой меткой должна существовать в программе, иначе выполнение программы будет прервано при попытке выполнения перехода.

### Примеры использования команды JMP:

JMP L1: – переход на метку L1:

### Команда – ON.

Включить драйвер ШД, т. е. установить выход Enable в 1.

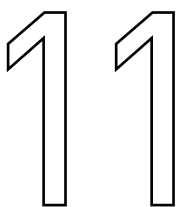
### Команда – OFF.

Выключить драйвер ШД, т. е. установить выход Enable в 0.

### Команда – SHOW.

Выводит значение указанной переменной из памяти. Если имя переменной не задано — выводит текущую программу из памяти.

SHOW inp1 – выводит значение переменной inp1.



## Примеры программ для модуля.

### Программа поиска «Нуля».

На входе IN1 установлен концевой выключатель, с активным низким сигналом.

```
MOVE D(0) C([inp1])
```

#Назад с макс. подачей до концевика, проскакиваем его

```
MOVE D(1) C([inp1]) F(10)
```

#Медленно возвращаемся, пока не «сойдем» с датчика.

### Пример 1.

#Назначим переменным новые имена

```
ALIAS V1, OLD_FEED
```

```
ALIAS V2, COUNTER
```

#Сохраним текущее значение подачи в обычную переменную

```
SET OLD_FEED, P_FEED_MAX
```

#теперь можем смело изменить подачу для данной программы

```
SET P_FEED_MAX, 100
```

#Выполним 10 перемещений вперед и назад

```
SET COUNTER, 10
```

```
ON
```

```
LOOP:
```

```
MOVE L(100) F(P_FEED_MAX / 10)
```

```
MOVE L(-100)
```

```
SET COUNTER, COUNTER - 1
```

```
IF (COUNTER > 0) JMP LOOP:
```

```
OFF
```

#Восстановим старое значение подачи

```
SET P_FEED_MAX, OLD_FEED
```

### Пример 2.

```
ALIAS v1,x
SET x,10
lab1: MOVE L(100*x) F(x)
SET x,x-1
IF (x>=0) JMP lab1:
MOVE L(10)
lab2: MOVE L(10) D(0)
```

### Пример 3.

```
alias v1, x
set x, 5
lab1: MOVE L(100*x) F(x) D(x& 1)
SET x,x-1
IF (x>0) JMP lab1:
```

### Пример 4.

```
alias v1,x
alias v2,y
set x,5
SET y,[x * 100]
lab1: MOVE L(y) F(x * 10) D(!{x& 1})
SET x,{x-1}
IF (x>0) JMP lab1:
```

### Пример 5.

```
SET Emerg, [inp3 && (inp2 ||inp4)]
```

Аварийное завершение программы произойдет, если станет активным вход №3 одновременно с любым из входов №2 или №4.

# 12 Маркировка, упаковка, хранение, транспортировка, утилизация.

## 12.1. Маркировка изделия.

Маркировка изделия содержит:

- товарный знак;
- наименование или условное обозначение (модель) изделия.

Маркировка потребительской тары изделия содержит:

- товарный знак предприятия-изготовителя;
- условное обозначение и серийный номер;
- год и месяц упаковывания.

## 12.2. Упаковка.

К заказчику изделие доставляется в собранном виде. Оборудование упаковано в картонный короб. Все разгрузочные и погрузочные перемещения вести с особым вниманием и осторожностью, обеспечивающими защиту от механических повреждений.

При хранении упакованного оборудования необходимо соблюдать следующие условия:

- не хранить под открытым небом;
- хранить в сухом и незапыленном месте;
- не подвергать воздействию агрессивных сред и прямых солнечных лучей;
- оберегать от механических вибраций и тряски;
- хранить при температуре от  $-20^{\circ}\text{C}$  до  $+40^{\circ}\text{C}$ , при влажности не более 60%.

## 12.3. Условия хранения изделия.

Изделие без упаковки должно храниться в условиях по ГОСТ 15150-69, группа 1Л (Отапливаемые и вентилируемые помещения с кондиционированием воздуха) при температуре от  $-20^{\circ}\text{C}$  до  $+65^{\circ}\text{C}$  и относительной влажности воздуха не более 90% (при  $+20^{\circ}\text{C}$ ).

Помещение должно быть сухим, не содержать конденсата и пыли. Запыленность помещения должна быть в пределах санитарной нормы. В воздухе помещения для хранения изделия не должно присутствовать агрессивных примесей (паров кислот, щелочей). Требования по хранению относятся к складским помещениям поставщика и потребителя.

При длительном хранении изделие должно находиться в упакованном виде и содержаться в отапливаемых хранилищах при температуре окружающего воздуха от  $+10^{\circ}\text{C}$  до  $+25^{\circ}\text{C}$  и относительной влажности воздуха не более 60% (при  $+20^{\circ}\text{C}$ ).

При постановке изделия на длительное хранение его необходимо упаковать в упаковочную тару предприятия-поставщика.

Ограничения и специальные процедуры при снятии изделия с хранения не предусмотрены. При снятии с хранения изделие следует извлечь из упаковки.

## 12.4. Условия транспортирования.

Допускается транспортирование изделия в транспортной таре всеми видами транспорта (в том числе в отапливаемых герметизированных отсеках самолетов) без ограничения расстояний. При перевозке в железнодорожных вагонах вид отправки — мелкий малотоннажный. При транспортировании изделия должна быть предусмотрена защита от попадания пыли и атмосферных осадков.

### Климатические условия транспортирования

Влияющая величина	Значение
Диапазон температур	от $-40^{\circ}\text{C}$ до $+60^{\circ}\text{C}$
Относительная влажность, не более	90% при $+35^{\circ}\text{C}$
Атмосферное давление	от 70 до 106.7 кПа (537-800 мм рт. ст.)

## 12.5. Подготовка к транспортированию.

Изделие должно быть закреплено для обеспечения устойчивого положения, исключения взаимного смещения и ударов. При проведении погрузочно-разгрузочных работ и транспортировании должны строго выполняться требования манипуляционных знаков, нанесенных на транспортной таре.

## 12.6. Утилизация.

Изделие не содержит в своем составе опасных или ядовитых веществ, способных нанести вред здоровью человека или окружающей среде, и не представляет опасности для жизни, здоровья людей и окружающей среды по окончании срока службы.

В этой связи утилизация изделия может производиться по правилам утилизации общепромышленных отходов. Утилизация осуществляется отдельно по группам материалов: пластмассовым элементам, металлическим частям и крепежным деталям.

Содержание драгоценных металлов в компонентах изделия (электронных платах, разъемах и т.п.) крайне мало, поэтому их вторичную переработку производить нецелесообразно.

# 13 Гарантийные обязательства.

Гарантийный срок службы составляет 12 месяцев со дня приобретения. Гарантия сохраняется только при соблюдении условий эксплуатации и регламентного обслуживания.

### 1. Общие положения

1.1. Продавец не предоставляет гарантии на совместимость приобретаемого товара и товара, имеющегося у Покупателя, либо приобретенного им у третьих лиц.

1.2. Характеристики изделия и комплектация могут изменяться производителем без предварительного уведомления в связи с постоянным техническим совершенствованием продукции.

### 2. Условия принятия товара на гарантийное обслуживание

2.1. Товар принимается на гарантийное обслуживание в той же комплектности, в которой он был приобретен.

### 3. Порядок осуществления гарантийного обслуживания

3.1. Гарантийное обслуживание осуществляется путем тестирования (проверки) заявленной неисправности товара.

3.2. При подтверждении неисправности проводится гарантийный ремонт.

4. Гарантия не распространяется на стекло, электролампы, стартеры и расходные материалы, а также на:

4.1. Товар с повреждениями, вызванными ненадлежащими условиями транспортировки и хранения, неправильным подключением, эксплуатацией в штатном режиме либо в условиях, не предусмотренных производителем (в т.ч. при температуре и влажности за пределами рекомендованного диапазона), имеющий повреждения вследствие действия сторонних обстоятельств (скачков напряжения электропитания, стихийных бедствий и т.д.), а также имеющий механические и тепловые повреждения.

4.2. Товар со следами воздействия и (или) попадания внутрь посторонних предметов, веществ (в том числе пыли), жидкостей, насекомых, а также имеющий посторонние надписи.

4.3. Товар со следами несанкционированного вмешательства и (или) ремонта (следы вскрытия, кустарная пайка, следы замены элементов и т.п.).

4.4. Товар, имеющий средства самодиагностики, свидетельствующие о ненадлежащих условиях эксплуатации.

4.5. Технически сложный Товар, в отношении которого монтажно-сборочные и пуско-наладочные работы были выполнены не специалистами Продавца или рекомендованными им организациями, за исключением случаев, прямо предусмотренных документацией на товар.

4.6. Товар, эксплуатация которого осуществлялась в условиях, когда электропитание не соответствовало требованиям производителя, а также при отсутствии устройств электрозащиты сети и оборудования.

4.7. Товар, который был перепродан первоначальным покупателем третьим лицам.

4.8. Товар, получивший дефекты, возникшие в результате использования некачественных или выработавших свой ресурс запасных частей, расходных материалов, принадлежностей, а также в случае использования не рекомендованных изготовителем запасных частей, расходных материалов, принадлежностей.

## КОНТАКТЫ

---

8(800) 555-63-74 бесплатные звонки по РФ

+7 (495) 505-63-74 – Москва

+7 (473) 204-51-56 – Воронеж

394033, Россия, г. Воронеж, Ленинский пр-т, 160 офис 149

Пн-Чт: 8:00-17:00

Пт: 8:00-16:00