



Справочное руководство по SCPI протоколу управления  
позиционеров серий PS.

Radioline

18 сентября 2020

## Содержание

<b>1 Общая информация.</b>	<b>3</b>
1.1 Протокол . . . . .	3
1.2 Используемые обозначения . . . . .	4
1.3 О единицах измерения расстояния. . . . .	4
1.4 О единицах измерения ускорения (времени разгона). . . . .	4
<b>2 Начало работы.</b>	<b>5</b>
<b>3 SCPI команды.</b>	<b>6</b>
3.1 Протокол и стек ошибок. . . . .	6
3.2 Прочие стандартные функции. . . . .	6
3.3 Информация об изделии. . . . .	7
3.4 Системные команды. . . . .	7
3.5 Состояние устройств. . . . .	7
3.6 Управление осями. Совместимость. . . . .	8
3.7 Управление осями. Информация и состояние. . . . .	8
3.8 Управления осями. Системные функции. . . . .	9
3.9 Управления осями. Программные пределы, установка нулей. . . . .	9
3.10 Управления осями. Движение. . . . .	10
3.11 Управление осями. Непрерывное сканирование . . . . .	11
<b>4 Система доставки уведомлений NCPI.</b>	<b>13</b>
4.1 Введение. . . . .	13
4.2 Пример . . . . .	14
4.3 Перечень команд NCPI . . . . .	14
<b>5 Примеры.</b>	<b>16</b>
5.1 Пример скрипта работы с изделием. (python3) . . . . .	16
5.2 Пример скрипта работы с изделием с использованием системы уведомлений. (python3) . . . . .	17
<b>6 Часто встречающиеся проблемы.</b>	<b>18</b>

## 1 Общая информация.

Настоящий документ описывает систему команд управления для реализации автоматического или ручного управления позиционером.

Основной функциональной единицей протокола является логическая ось изделия. Логическая ось образуется одним или более входящих в нее устройств, таких как сервоусилитель или модуль синхронизации. Команды делятся на команды осей, образующих их устройств и системные команды. Команды в руководстве отсортированы по функциональному назначению и даны с описанием аргументов и особенностей применения. В необходимых местах приводятся примеры использования.

В документе представлена общая справка по командам управления. Для конкретных конфигураций, типов оборудования, а также, в зависимости от функционального назначения логических осей изделия, часть команд может быть не реализована или игнорироваться. Особенности работы с конкретным типом изделия уточняйте у изготавителя.

### 1.1 Протокол.

Управление изделием осуществляется по протоколу SCPI std V1999.0 4.2.1, расширенному системой подписки на уведомления NCPI.

Протокол SCPI используется для выставления параметров, запросов состояния прибора, передачи команд. Функциональные подсистемы:

- **:SYSTem:** - общесистемные команды.
- **:AXIS#:** - функции логических осей.
- **:DEV#:** - функции аппаратных модулей (сервоусилителей и прочих приборов)

Расширение NCPI позволяет получать информацию об изменении состояния изделия в режиме уведомлений. Используется единственная функциональная подсистема:

- **:NOT:** - подписка на уведомления

Связь с изделием осуществляется по протоколу TCP/IP через разъём rj45 (ethernet) на корпусе контроллера. Система использует два tcp порта:

- Порт SCPI консоли 5025 (стандартный порт для SCPI-консолей).
- Порт NCPI консоли 5026.

При работе с прибором через библиотеку VISA поддерживается режим socket. Режим instrument не поддерживается.

## 1.2 Используемые обозначения

**COMMAND#** - символ решетки заменяет натуральное число (начиная от нуля).

**<argument>** - угловые скобки - аргументы команд.

**CMD:[SUBCMD]** - квадратные скобки - необязательная часть команды, необязательный аргумент.

В тексте используются кодовые вставки:

*IDN?	% Командный запрос
> RADIOLINE,XXX,XXX,XXX	% Ответ обозначается символом '>'.
	% - односторонний комментарий.

## 1.3 О единицах измерения расстояния.

Команды протокола работают с несколькими разными типами единиц измерения расстояния.

Оборот - оборот вала двигателя. Входной оборот системы вал-редуктор.

Импульс - дискрет обратной связи энкодера сервопривода. Существует постоянное соотношение количества импульсов к количеству оборотов.

Юнит(unit) - естественная единица, устанавливаемая исходя из требований системы (обычно, сантиметр для линейных осей и градус для поворотных).

*\* Количество импульсов в юните для оси может быть запрошено командой "AXIS#:SETTINGS:RATIO?".*

## 1.4 О единицах измерения ускорения (времени разгона).

Ускорения задаются как количество миллисекунд требуемых для разгона двигателя до номинального рабочего режима. Обратите внимание, что ускорение будет повышаться при снижении устанавливаемого времени разгона.

## 2 Начало работы.

SCPI протокол допускает работу в автоматическом и в ручном режимах в зависимости от источника команд управления. В автоматическом режиме источником команд является управляющее ПО. В ручном режиме команды вводятся пользователем через терминал ввода-вывода. Функционально эти режимы не отличаются и реализуются через установленное TCP соединение с SCPI портом (по умолчанию 5025) контроллера.

Ручной режим удобен для проверки соединения и отладки команд управления для последующего использования в автоматическом режиме, либо для выполнения простых команд при недоступности управляющего ПО.

Перед началом работы с протоколом необходимо проверить доступность контроллера в локальной сети и установить соединение с SCPI портом изделия. Для установки соединения в ручном режиме можно использовать утилиту *putty*(режим: *raw*) или аналоги.

Посылки SCPI протокола имеют представление ASCII строк, терминированных символом конца строки (LF: \n, dec:10, bin:0x0A).

Тестовый запрос идентификатора изделия:

```
*IDN?  
> RADIOLINE,XXX,XXX,XXX
```

Запрос статуса:

```
SYST:STAT? % Запрос статуса готовности изделия  
> 0 % Изделие готово к работе
```

Запрос информации от объекта оси.

```
SYST:AXESTOT? % Запрос количества осей.  
> 3 % В системе есть оси 0, 1, 2  
AXIS2:UPosition? % поскольку нумерация осей идёт от нуля  
> 0.378123
```

Установка параметров.

```
AXIS2:USPD 2 % Установка параметров и команды  
% не предполагают ответа
```

### 3 SCPI команды.

#### 3.1 Протокол и стек ошибок.

Запрос установочных данных о версии протокола и состоянии стека ошибок. Управление стеком ошибок.

**SYSTem:VERSion?**

Версия SCPI протокола.

**SYSTem:ERRor[:NEXT]?**

Возврат следующего сообщения об ошибке.

**SYSTem:ERRor:COUNt?**

Количество ошибок в стеке ошибок.

**\*CLS**

Очистка стека ошибок.

#### 3.2 Прочие стандартные функции.

**\*ESE**

Игнорируется.

**\*ESE?**

Игнорируется. (возврат единицы)

**\*ESR?**

Игнорируется. (возврат единицы)

**\*OPC**

Игнорируется.

**\*OPC?**

Игнорируется. (возврат единицы)

**\*RST**

Игнорируется.

**\*SRE**

Игнорируется.

**\*SRE?**

Игнорируется. (возврат единицы)

**\*STB?**

Игнорируется. (возврат единицы)

**\*WAI**

Игнорируется.

### 3.3 Информация об изделии.

Запрос неизменяемых параметров прибора и осей изделия.

**\*IDN?**

Запрос идентификатора изделия.

**SYSTem:DEVSTOTal?**

Количество зарегистрированных устройств.

**SYSTem:axes<sub>total</sub>?**

Количество зарегистрированных осей.

### 3.4 Системные команды.

Подмножество команд, влияющих на все оси и устройства в системе, состояние системы в целом.

**SYSTem:PRESet**

Сброс всех устройств в состояние по умолчанию, сброс всех настроек.

**SYSTem:IPADDR**

Установка IPv4 адреса изделия. В записи адреса вместо символа точки следует использовать символы запятых. Пример:

**SYST:IPADDR 192,168,1,42**

**SYSTem:STOP**

Выполнить остановку движения по всем осям.

**SYSTem:POWOFF**

Выполнить отключения питания исполнительных устройств на всех осях.

*\* Только для отдельных изделий. Для большинства изделий управление питанием не требуется.*

**SYSTem:STATus?**

Запрос статуса готовности системы

Система возвращает готовность если все входящие в её состав устройства и оси возвращают готовность.

Ответ:

0 – Система готова.

1 – Система не готова.

### 3.5 Состояние устройств.

Информация о состоянии аппаратных устройств.

**DEV#:IDN?**

Запрос идентификатора устройства.

**DEV#:PRESET**

Возврат параметров устройства к параметрам по умолчанию.

**DEV#:STATus?**

Запрос статуса готовности устройства.

Ответ:

0 – Устройство готово.

1 – Устройство не готово.

**DEV#:ALM?**

Запрос alarm-кода устройства в случае его неисправности. \* *Формат ответа зависит от типа устройства.*

### 3.6 Управление осями. Совместимость.

В зависимости от типа сервопривода и функционального назначения осей, ось может поддерживать дополнительные функции.

**AXIS#:COMPat:SCAN?**

Запрос наличия на оси подсистемы синхронного сканирования.

**AXIS#:COMPat:REFSet?**

Запрос наличия функции установки текущего абсолютного положения (см. AXIS#:SETREF, AXIS#:SETUREF).

### 3.7 Управление осями. Информация и состояние.

**AXIS#:STATus:IDN?**

Запрос идентификатора оси.

**AXIS#:STATus:DEVS?**

Запрос кодов устройств, входящих в ось.

**AXIS#:STATus:POSIon?**

Запрос положения в импульсах энкодера.

**AXIS#:STATus:UPOSition?**

Запрос положения в юнитах.

**AXIS#:SETTINGS:RATIO?**

Количество импульсов энкодера в одном юните.

**AXIS#:SETTINGS:DEFSPeed?**

Скорость по умолчанию в оборотах в минуту.

**AXIS#:SETTINGS:DEFACCEl?**

Длительность разгона по умолчанию в миллисекундах.

**AXIS#:SETTINGS:MAXSPEed?**

Максимальная скорость в оборотах в минуту.

**AXIS#:SETTINGS:MINAccel?**

Минимальное времена разгона в миллисекундах.

**AXIS#:STATus[:STATus]?**

Запрос статуса готовности оси.

Ответ:

0 – Ось готова.

1 – Ось не готова.

**AXIS#:STATus:LSWItch?**

Запрос статуса концевых датчиков.

Ответ:

0 – нет срабатывания,

1 - левый,

2 - правый,

10 – выдается при срабатывании обоих концевиков (вероятна ошибка). \* *Только для изделий с концевыми датчиками. При отсутствии таковых всегда возвращает 0*

**AXIS#:STATus:OPcode?**

Запрос кода выполняемой операции.

Ответ:

0 - Нет операции,

1 - Выполняется перемещение,

-1 - Ось в состоянии инициализации

**3.8 Управления осями. Системные функции.****AXIS#:PRESET**

Возврат устройств оси к значениям по умолчанию.

**AXIS#:SON**

[Снять тормоз] и подать питание на двигатель.

\* *Только для отдельных изделий. Для большинства изделий управление питанием не требуется.*

**AXIS#:SOFF**

Снять питание с двигателя, [установить тормоз].

\* *Только для отдельных изделий. Для большинства изделий управление питанием не требуется.*

**3.9 Управления осями. Программные пределы, установка нулей.**

В отличии от других настроек, настройки нулей и пределов перемещения не сбрасываются при отключении питания.

**AXIS#:SETTINGS:UBACKLIMit <arg>****AXIS#:SETTINGS:UBACKLIMit?**

Задать/считать предел перемещения при движении в отрицательном направлении.

**AXIS#:SETTINGS:UFORWLIMit <arg>****AXIS#:SETTINGS:UFORWLIMit?**

Задать/считать предел перемещения при движении в положительном направлении.

**AXIS#:SETTINGS:ULIMITS <back>, <forw>**

Задать/считать пару пределов перемещения. back должен быть меньше forw.

**AXIS#:SETZERO**

Сброс абсолютного положения в ноль.

**AXIS#:SETREference <pos>**

Сброс абсолютного положения в заданное значение.

pos – новая текущая позиция в импульсах.

\* Поддерживается не для всех типов сервоусилителей. см. *AXIS#:COMPat:REFSet?***AXIS#:SETUREference <upos>**

Сброс абсолютного положения в заданное значение (в юнитах).

upos – новая текущая позиция в юнитах.

\* Поддерживается не для всех типов сервоусилителей. см. *AXIS#:COMPat:REFSet?*

## 3.10 Управления осями. Движение.

**AXIS#:SPEeed <rpm>****AXIS#:SPEeed?**

Установка и запрос скорости в об/мин.

**AXIS#:USPEeed <ups>****AXIS#:USPEeed?**

Установка и запрос скорости в unit/сек.

**AXIS#:ACCel <ms>****AXIS#:ACCel?**

Установка и запрос интервала времени разгона и торможения для операций типа MOV и JOG. Интервал времени выражен в миллисекундах. ms – интервал времени в миллисекундах.

**AXIS#:MOVe[:RELative] <distance>**

Относительное перемещение (в импульсах энкодера).

distance – количество импульсов. Знак определяет направление перемещения.

**AXIS#:MOVE:ABSolute <position>**

Абсолютное перемещение (в импульсах энкодера).

position – координата в импульсах.

**AXIS#:UMOVe[:RELative] <distance>**

Относительное перемещение (в unit).

distance – расстояние в юнитах. Знак определяет направление перемещения.

**AXIS#:UMOVe:ABSolute <position>**

Абсолютное перемещение (в unit).

position – координата в юнитах.

**AXIS#:JOG <direction>**

Движение с постоянной скоростью.

direction – 1 – вперёд, -1 – назад.

**AXIS#:STOP**

Остановить ось.

**AXIS#:UNSAFE:MOVE <distance>**

Относительное перемещение (в импульсах энкодера) с игнорированием пределов.

Данная команда может быть использована при наладке оборудования. Не рекомендуется для использования в штатном режиме.

**AXIS#:UNSAFE:UMOVe <distance>**

Относительное перемещение (в юнитах) с игнорированием пределов.

Данная команда может быть использована при наладке оборудования. Не рекомендуется для использования в штатном режиме.

### 3.11 Управление осями. Непрерывное сканирование

Функции системы непрерывного сканирования доступны для осей, на которых установлен модуль синхронизации. При активации режима непрерывного сканирования устройство синхронизации будет посыпать триггерные сигналы при проходе через установленные точки.

Также при проходе через установленные точки или после получения обратного триггера (см. NOTRIGMODE) генерируются уведомления (см. NOT:AXIS#:SCANPOINT). Для получения информации о проходе очередной точки на стороне клиента необходимо подключить соответствующее уведомление.

Шаг сканирования вычисляется по формуле:

$$\text{STEP} = \text{UMOVE} / (\text{POINTS} - 1)$$

**AXIS#:SCAN:MOVE <distance>****AXIS#:SCAN:MOVE?****AXIS#:SCAN:UMOVE <distance>****AXIS#:SCAN:UMOVE?**

Установка зоны сканирования.

distance – расстояние, обрабатываемое алгоритмом сканирования в импульсах/юнитах  
Функции с префиксом U здесь и далее - аналоги в юнитах.

**AXIS#:SCAN:FWRDzone <distance>**

**AXIS#:SCAN:FWRDzone?**

**AXIS#:SCAN:UFWRDzone <distance>**

**AXIS#:SCAN:UFWRDzone?**

Установка расстояния до первой точки.

distance – расстояние, пропускаемое алгоритмом перед началом сканирования в импульсах/юнитах.

**AXIS#:SCAN:BWRDzone <distance>**

**AXIS#:SCAN:BWRDzone?**

**AXIS#:SCAN:UBWRDzone <distance>**

**AXIS#:SCAN:UBWRDzone?**

Установка расстояния прохода после последней точки (только для режима SCAN:START).

distance – расстояние в импульсах/юнитах.

**AXIS#:SCAN:COMPSTART**

Активировать режим сканирования. Активации сканирования не приводит к началу движения. Данная команда включает режим выработки триггеров и передачи уведомлений о достижении точек.

**AXIS#:SCAN:POINTS <npnts>**

**AXIS#:SCAN:POINTS?**

Установка количества точек сканирования.

npnts – количество необходимых точек.

**AXIS#:SCAN:NOTRIGMODE <en>**

Включить/отключить ожидание обратного триггера перед выдачей уведомления о достижении очередной точки.

en:

0 - уведомление о достижении очередной точке будет послано после прихода обратного триггера (режим по умолчанию)

1 - ожидание обратного триггера отключено

**AXIS#:MANTRIGmode <en>**

Перевести ось в режим ручной генерации триггеров.

Настраивает синхронизатор в режим работы с командой AXIS#:TRIGGER.

en:

1 - включить ручной режим

0 - выключить ручной режим

**AXIS#:TRIGGER**

Сгенерировать триггер.

Перед использованием следует перевести ось в режим ручной генерации триггеров (см. AXIS#:MANTRIGmode).

**AXIS#:TRIGRETTIME?**

Запрос времени возврата обратного триггера в миллисекундах.

## 4 Система доставки уведомлений NCPI.

### 4.1 Введение.

Система NCPI позволяет получать информацию о событиях, без необходимости постоянного опроса состояния устройств в режиме уведомлений.

После подключения к серверу клиент должен активировать необходимые ему темы уведомлений с помощью команд описанных в настоящем разделе. Одновременно клиент может подписаться на любое количество тем, но не более одного раза на одну тему. По активации подписки, при возникновении события, сервер будет генерировать сообщения уведомлений в формате "Имя:Темы [Ответ]". Использование протокола TCP даёт гарантию доставки сообщений.

Система NCPI имеет три типа сообщений:

- Простое событие.
- Изменение состояния, событие с аргументом
- Уведомление о непрерывных параметрах.

#### Простое событие

Простое событие предполагает уведомление формата: SUBSYST:THEME и не несет никакой дополнительной информации, кроме факта возникновения события.

```
NOT:SUBSYST:THEME 1      % Активировать подписку  
NOT:SUBSYST:THEME 0      % Отменить подписку  
%  
> SUBSYST:THEME          % Формат уведомления
```

#### Уведомление об изменении состояния, событии с аргументом

Уведомление об изменении состояния возникает при изменении соответствующей переменной и передаёт связанную с ней строку, код или значение.

```
NOT:SUBSYST:THEME 1      % Активировать подписку  
NOT:SUBSYST:THEME 0      % Отменить подписку  
%  
> SUBSYST:THEME <argument> % Формат уведомления
```

#### Уведомления о изменении непрерывного параметра.

Данный тип предназначены для получения информации о непрерывных параметрах, изменяющихся с течением времени (например, текущее положение). Поддерживаются две стратегии получения уведомлений:

1. Уведомление с временным дискретом, выраженным в миллисекундах. Дискрет задаёт минимальное временной промежуток между уведомлениями. Система может высылать сообщения реже этого интервала. Кроме того существует минимальный интервал времени чаще которого сообщения не могут быть посланы (обычно этот интервал составляет несколько десятков миллисекунд).

```
NOT:SUBSYST:THEME TIMERED,1000 % Уведомления каждую секунду.  
NOT:SUBSYST:THEME 0          % Отменить подписку  
%  
SUBSYST:THEME <argument>    % Формат уведомления
```

2. Уведомление при изменении параметра на заданное значение. Как и в случае с уведомлениями по истечению интервалов времени и интервал может превышать установленное значение.

```
NOT:SUBSYST:THEME SMOOTH,5.34    % Уведомления с дискретом величин 5.34.
NOT:SUBSYST:THEME 0                % Отменить подписку
                                    %
SUBSYST:THEME <argument>        % Формат уведомления
```

## 4.2 Пример

```
NOT:AXIS2:OPSTAT 1    % Уведомления на изменение состояния оси 2
                      %
> AXIS2:OPSTAT 1    % Ось 2 начала движение
> AXIS2:OPSTAT 0    % Ось 2 завершила движение
                      %
NOT:AXIS2:OPSTAT 0    % Отменить подписку
```

## 4.3 Перечень команд NCPI

### **NOT:SYSTem:STATus**

Уведомление о готовности системы.

Тип: изменение состояния.

Система считается готовой, если ни одно из устройств не сигнализирует об ошибке.

Параметр:

0 – Готов

1 – Не готов

### **NOT:AXIS#:STATus**

Уведомление о готовности оси.

Тип: изменение состояния.

Ось считается готовой, если ни одно из устройств оси не сигнализирует об ошибке.

Параметр:

0 – Готов

1 – Не готов

### **NOT:DEV#:STATus**

Уведомление о готовности устройства.

Тип: изменение состояния

Параметр:

0 – Готов

1 – Не готов

### **NOT:AXIS#:POsition**

Уведомление о текущем положении по оси.

Тип: непрерывный параметр

**NOT:AXIS#:UPOSition**

Уведомление о положении по оси в единицах unit.

Тип: непрерывный параметр

**NOT:AXIS#:OPSTATus**

Уведомление о статусе операции.

Тип: изменение состояния

Параметр:

- 0 – Нет операции
- 1 – Задание на движение
- 2 – Режим сканирования

**NOT:AXIS#:OPSTOPtype**

Уведомление о типе прерывания последней операции.

Тип: изменение состояния

Параметр:

- 0 – операция начата
- 1 – нормальное завершение операции
- 2 – операция завершена командой stop
- 3 – аварийное завершение операции

**NOT:AXIS#:SCAN:POINT**

Уведомление о достижении новой точки в режиме сканирования.

Тип: событие с аргументом

Возвращает номер точки в теле уведомления.

**NOT:AXIS#:SCAN:TRIGERROR**

Уведомление об ошибке в системе синхронизации при ожидании обратного триггера.

Момент генерации следующего триггера настал ранее, чем вернулся обратный.

Тип: событие

**NOT:AXIS#:SCAN:LSEWItch**

Уведомление об изменении статуса концевых датчиков.

Тип: изменение состояния

Параметр:

- 0 – нет срабатывания
- 1 - левый
- 2 - правый
- 10 – выдается при срабатывании обоих концевиков (вероятна ошибка)

## 5 Примеры.

### 5.1 Пример скрипта работы с изделием. (python3)

```
#!/usr/bin/env python3

import sys
import time
import socket
import threading

HOST = '192.168.1.224' # ip address
SCPI_PORT = 5025
NCPI_PORT = 5026

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
n = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, SCPI_PORT))
n.connect((HOST, NCPI_PORT))

# request identifier
s.send("AXISO:STAT:IDN?\r\n".encode("utf-8"))
answer = s.recv(1024).decode("utf-8")
print("axis0_idn_is_{}".format(answer))

# set speed : 1 unit per second
s.send("AXISO:USPE=1\r\n".encode("utf-8"))

# set acceleration : 2 seconds for acceleration to nominal speed
s.send("AXISO:ACCEL=2000\r\n".encode("utf-8"))

# absolute move to coordinate 4u
print("move_to_4u")
s.send("AXISO:UMOV:ABS=4\r\n".encode("utf-8"))

time.sleep(5)

# absolute move to coordinate 0u
print("move_to_0u")
s.send("AXISO:UMOV:ABS=0\r\n".encode("utf-8"))
```

## 5.2 Пример скрипта работы с изделием с использованием системы уведомлений. (python3)

```
#!/usr/bin/env python3

import sys
import time
import socket
import threading

HOST = '192.168.1.224' # ip address
SCPI_PORT = 5025
NCPI_PORT = 5026

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
n = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, SCPI_PORT))
n.connect((HOST, NCPI_PORT))

s.send("AXISO:STAT:IDN?\r\n".encode("utf-8"))
answer = s.recv(1024).decode("utf-8")
print("axis0 idn is {}".format(answer))

print("enable notification")
n.send("NOT:AXISO:UPOS:SMOOTH,0.1\r\n".encode("utf-8"))
n.send("NOT:AXISO:OPSTAT:1\r\n".encode("utf-8"))

cancel = False
def foo():
    n.settimeout(0.5)
    while cancel is False:
        try:
            message = n.recv(1024).decode("utf-8")
        except:
            pass
        sys.stdout.write("notification: {}".format(message))

thr = threading.Thread(target=foo)
thr.start()

print("move to 4u")
s.send("AXISO:UMOV:ABS:4\r\n".encode("utf-8"))

time.sleep(5)

print("move to 0u")
s.send("AXISO:UMOV:ABS:0\r\n".encode("utf-8"))

time.sleep(5)
cancel = True
```

## **6 Часто встречающиеся проблемы.**

### **Соединение не установлено**

Убедитесь в доступности блока управления в локальной сети. Проверьте аппаратное подключение. Произведите ping изделия. Убедитесь, что файрвол на вашей машине не блокирует соединение. При необходимости перезапустите блок управления.

### **Изделие не реагирует на команды управления.**

Проверьте формат и текст команды. Убедитесь, что команда терминирована символом конца строки LF.

### **Изделие отвечает на команды, но не двигается.**

Проверьте стек ошибок протокола SCPI. Проверьте статус системы и осей с помощью команд:

```
SYST:STAT?  
AXIS\#:STAT?
```

При возникновении ошибок свяжитесь с изготавителем.

### **Ось движется в одну и ту же сторону вне зависимости от заданного направления движения.**

Проверьте работу UNSAFE подмножества команд движения. Убедитесь, что пределы оси выставлены верно и текущее положение не сбито. При необходимости переопределите ноль/корректируйте пределы.

### **В режиме непрерывного сканирования не приходит уведомлений о проходимых точках.**

Убедитесь, что ось поддерживает режим непрерывного сканирования. Проверьте правильность монтажа триггерных цепей. Убедитесь в корректности настроек режима сканирования.