

ОАО «Специальное Конструкторское Бюро Информационно-Измерительных Систем»  
Санкт-Петербург

# Универсальный протокол взаимодействия с ЛИР

---

**Описание протокола**

**ВЕРУ.406920.002ИС**

12.2018

**Оглавление**

|                                                                    |    |
|--------------------------------------------------------------------|----|
| Протокол .....                                                     | 4  |
| Описание .....                                                     | 4  |
| Стандартные команды: .....                                         | 8  |
| Системный модуль .....                                             | 12 |
| Описание .....                                                     | 12 |
| Расширенные команды .....                                          | 13 |
| Модуль датчика.....                                                | 17 |
| Описание .....                                                     | 17 |
| Расширенные команды .....                                          | 19 |
| Подмодули .....                                                    | 25 |
| Подмодуль инкрементный .....                                       | 25 |
| Подмодуль SSI .....                                                | 27 |
| Подмодуль BISS.....                                                | 29 |
| Подмодуль аналоговый .....                                         | 33 |
| Модуль RS485 .....                                                 | 34 |
| Описание .....                                                     | 34 |
| Расширенные команды .....                                          | 34 |
| Подмодули .....                                                    | 35 |
| Подмодуль Modbus RTU .....                                         | 35 |
| Подмодуль СППУ .....                                               | 38 |
| Подмодуль протокола ЛИР-915/6 .....                                | 41 |
| Модуль входов/выходов.....                                         | 46 |
| Описание .....                                                     | 46 |
| Расширенные команды .....                                          | 46 |
| Модуль виртуальных входов/выходов .....                            | 48 |
| Описание .....                                                     | 48 |
| Расширенные команды .....                                          | 53 |
| Модуль функциональных сигналов .....                               | 54 |
| Описание .....                                                     | 54 |
| Расширенные команды .....                                          | 56 |
| Модуль позиционирования.....                                       | 58 |
| Описание .....                                                     | 58 |
| Расширенные команды .....                                          | 60 |
| Подмодули .....                                                    | 63 |
| Подмодуль управления двигателем с помощью дискретных выходов ..... | 63 |
| Модуль обработки g кодов .....                                     | 64 |
| Описание .....                                                     | 64 |

|                                        |    |
|----------------------------------------|----|
| Расширенные команды .....              | 70 |
| Модуль зоны.....                       | 76 |
| Описание .....                         | 76 |
| Расширенные команды .....              | 77 |
| Модуль Ethernet .....                  | 78 |
| Описание .....                         | 78 |
| Расширенные команды .....              | 78 |
| Подмодули .....                        | 79 |
| Подмодуль Modbus RTU over TCP/IP ..... | 79 |
| Подмодуль Modbus TCP/IP .....          | 81 |
| Модуль графического интерфейса .....   | 83 |
| Описание .....                         | 83 |
| Расширенные команды .....              | 84 |
| Модуль математический .....            | 85 |
| Описание .....                         | 85 |
| Расширенные команды .....              | 87 |
| История изменений.....                 | 89 |

## Протокол

### Описание

Устройство поддерживает удаленное управление и конфигурацию с помощью унифицированного протокола, реализуемого поверх соединения через usb, RS-485 (Modbus RTU) или ethernet (Modbus RTU over TCP/IP или Modbus TCP/IP). При подключении через usb устройство определяется как hid. Обмен с устройством происходит с помощью hid report по 255 байт.

*Запрос:*

|                  |
|------------------|
| Пакет управления |
| hid report       |

*Ответ:*

|            |
|------------|
| Результат  |
| hid report |

При подключении через RS-485 пакет управления передается внутри функции, предназначенной для передачи данных в произвольных форматах протокола Modbus RTU.

*Запрос:*

|                      |      |   |                  |     |
|----------------------|------|---|------------------|-----|
| Адрес устройства     | 0x2B | 1 | Пакет управления | CRC |
| Сообщение Modbus RTU |      |   |                  |     |

*Ответ:*

|                      |      |   |           |     |
|----------------------|------|---|-----------|-----|
| Адрес устройства     | 0x2B | 1 | Результат | CRC |
| Сообщение Modbus RTU |      |   |           |     |

Передача пакета управления через Ethernet для протокола Modbus RTU over TCP/IP аналогична:

*Запрос:*

|                      |      |   |                  |     |
|----------------------|------|---|------------------|-----|
| Адрес устройства     | 0x2B | 1 | Пакет управления | CRC |
| Сообщение Modbus RTU |      |   |                  |     |

*Ответ:*

|                      |      |   |           |     |
|----------------------|------|---|-----------|-----|
| Адрес устройства     | 0x2B | 1 | Результат | CRC |
| Сообщение Modbus RTU |      |   |           |     |

Для протокола Modbus TCP/IP:

*Запрос:*

| Номер транзакции     | Идентификатор протокола | Длина поля данных | Адрес устройства | 0x2B | 1 | Пакет управления |
|----------------------|-------------------------|-------------------|------------------|------|---|------------------|
| Сообщение Modbus TCP |                         |                   |                  |      |   |                  |

*Ответ:*

| Номер транзакции     | Идентификатор протокола | Длина поля данных | Адрес устройства | 0x2B | 1 | Результат |
|----------------------|-------------------------|-------------------|------------------|------|---|-----------|
| Сообщение Modbus TCP |                         |                   |                  |      |   |           |

Структура пакета управления для всех типов соединения полностью идентична. Тип соединения определяет только максимальный допустимый размер пакета управления: для usb – 255 байта, для RS485/Ethernet – 251 байт.

Устройство разбито на отдельные модули, взаимодействие с которыми осуществляется с помощью пакета управления. Один пакет управления может содержать обращения к нескольким подмодулям или несколько команд к одному модулю. Ответный пакет управления содержит ответ на команды только из одного пакета управления запроса. Если ответы на управляющие команды не помещаются в пакет управления ответа, команды выполняются, но ответы теряются. Пользователь должен следить за посылаемым набором команд, исключая переполнение ответного пакета управления.

Пакет управления имеет следующую структуру:

| N <sub>p</sub>   | N1 | I1 | C1 | D1 | ... | Nn | In | Cn | Dn |
|------------------|----|----|----|----|-----|----|----|----|----|
| Пакет управления |    |    |    |    |     |    |    |    |    |

, где:

N<sub>p</sub> – количество команд в пакете,

N1..n – размер команды в байтах,

I1...n – номер модуля, к которому адресована команда,

C1...n – команда,

D1...n – область данных (0...n байт).

В ответ на команду к несуществующему модулю, устройство возвращает номер модуля + 0x80 и номер команды:

*Запрос:*

| N <sub>p</sub>   | ... | Nn | In | Cn | Dn | ... |
|------------------|-----|----|----|----|----|-----|
| Пакет управления |     |    |    |    |    |     |

*Ответ:*

| N <sub>p</sub> | ... | Nn | 0x80   In | Cn | ... |  |
|----------------|-----|----|-----------|----|-----|--|
| Результат      |     |    |           |    |     |  |

В ответ на несуществующую команду у адресуемого модуля, устройство возвращает номер модуля и команду + 0x80.

*Запрос:*

|                  |     |    |    |    |    |     |
|------------------|-----|----|----|----|----|-----|
| N <sub>p</sub>   | ... | Nn | In | Cn | Dn | ... |
| Пакет управления |     |    |    |    |    |     |

*Ответ:*

|                |     |    |    |           |     |  |
|----------------|-----|----|----|-----------|-----|--|
| N <sub>p</sub> | ... | Nn | In | 0x80   Cn | ... |  |
| Результат      |     |    |    |           |     |  |

Если устройство выполняет команду, оно отвечает подтверждением. Если команда не предусматривает возвращение каких-либо данных, то подтверждением является ack (0xF0).

*Запрос:*

|                  |     |    |    |    |    |     |
|------------------|-----|----|----|----|----|-----|
| N <sub>p</sub>   | ... | Nn | In | Cn | Dn | ... |
| Пакет управления |     |    |    |    |    |     |

*Ответ:*

|                |     |    |    |    |      |  |
|----------------|-----|----|----|----|------|--|
| N <sub>p</sub> | ... | Nn | In | Cn | 0xF0 |  |
| Результат      |     |    |    |    |      |  |

, где:

0xF0 – ack.

Если предусматривает – возвращенные данные являются подтверждением.

*Запрос:*

|                  |     |    |    |    |    |     |
|------------------|-----|----|----|----|----|-----|
| N <sub>p</sub>   | ... | Nn | In | Cn | Dn | ... |
| Пакет управления |     |    |    |    |    |     |

*Ответ:*

|                |     |    |    |    |    |     |
|----------------|-----|----|----|----|----|-----|
| N <sub>p</sub> | ... | Nn | In | Cn | Dn | ... |
| Результат      |     |    |    |    |    |     |

Если устройство не может выполнить команду, оно отвечает отказом. Если команда не предусматривает возвращение каких-либо данных, то отказом является nack (0x0F).

*Запрос:*

|                  |     |                |    |    |    |     |
|------------------|-----|----------------|----|----|----|-----|
| N <sub>p</sub>   | ... | N <sub>n</sub> | In | Cn | Dn | ... |
| Пакет управления |     |                |    |    |    |     |

*Ответ:*

|                |     |                |    |    |      |     |
|----------------|-----|----------------|----|----|------|-----|
| N <sub>p</sub> | ... | N <sub>n</sub> | In | Cn | 0x0F | ... |
| Результат      |     |                |    |    |      |     |

, где:  
0x0F – nack.

Если предусматривает – поле данных отсутствует.

*Запрос:*

|                  |     |                |    |    |    |     |
|------------------|-----|----------------|----|----|----|-----|
| N <sub>p</sub>   | ... | N <sub>n</sub> | In | Cn | Dn | ... |
| Пакет управления |     |                |    |    |    |     |

*Ответ:*

|                |     |                |    |    |     |  |
|----------------|-----|----------------|----|----|-----|--|
| N <sub>p</sub> | ... | N <sub>n</sub> | In | Cn | ... |  |
| Результат      |     |                |    |    |     |  |

Диапазон значений и размер полей данных команд имеет ограничения согласно описанию соответствующего модуля и проверяется в момент выполнения команды, поэтому можно получить отказ выполнения на правильно сформированную команду, содержащую значение в поле данных, выходящее за диапазон допустимых.

Каждый модуль имеет набор стандартных команд, а также может иметь расширенные.

Системный модуль всегда имеет номер 0. С его помощью можно узнать количество модулей в системе, идентификатор устройства, его серийный номер, а также, аппаратную и программную версии с помощью соответствующих команд.

Каждый модуль может содержать несколько подмодулей. Количество подмодулей можно узнать у каждого модуля с помощью соответствующей команды. Также у каждого модуля и подмодуля можно узнать его тип и версию. Для каждого модуля настройки хранятся отдельно и с помощью соответствующей команды системного модуля можно узнать их состояние. Настройки подмодулей, принадлежащих одному модулю, хранятся в общей для них памяти и, соответственно, сохраняются для текущего выбранного подмодуля.

В некоторых устройствах несколько модулей могут использовать общую аппаратную часть. Такие модули называются конфликтующими. Соответственно, одновременно может быть включен только один из них. Список конфликтующих модулей доступен для каждого модуля с помощью соответствующей стандартной команды.

Так же каждый модуль может иметь входные и выходные функциональные сигналы. Функциональные сигналы позволяют привязывать внутренние функции модулей к физическим входам/выходам устройства. Описание сигналов для каждого модуля находится в его документации. Список всех доступных сигналов устройства, а так же их настройку производят с помощью модуля функциональных сигналов.

**Стандартные команды:**

В выключенном состоянии каждый модуль отвечает на команды только до “запрос текущего режима модуля”.

**Запрос информации о модуле (0x00):**

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x00 |
|---|---|------|

*Ответ:*

|   |   |      |    |     |
|---|---|------|----|-----|
| 5 | I | 0x00 | id | ver |
|---|---|------|----|-----|

, где:

id – идентификатор типа модуля,

ver – версия интерфейса.

Ниже перечислено соответствие идентификаторов типам модулей:

0 – системный модуль,

1 – модуль датчика,

2 – модуль порта RS485,

3 – модуль входов/выходов,

4 – модуль виртуальных входов/выходов,

5 – модуль функциональных сигналов,

6 – модуль позиционирования,

7 – модуль обработки g кодов,

8 – модуль телеметрии,

9 – модуль зоны,

10 – модуль Ethernet,

11 – модуль графического интерфейса,

12 – модуль математический.

Версия модуля передается умноженной на 10.

**Запрос количества подмодулей (0x01)**

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x01 |
|---|---|------|

*Ответ:*

|   |   |      |   |
|---|---|------|---|
| 4 | I | 0x01 | N |
|---|---|------|---|

, где:

N – количество подмодулей.



**Запрос информации об подмодуле (0x02)***Запрос:*

|   |   |      |   |
|---|---|------|---|
| 4 | I | 0x02 | N |
|---|---|------|---|

, где:

N – номер подмодуля.

*Ответ:*

|   |   |      |    |     |
|---|---|------|----|-----|
| 5 | I | 0x02 | id | ver |
|---|---|------|----|-----|

, где:

id – идентификатор типа подмодуля,

ver – версия подмодуля.

Для подмодулей идентификаторы определяются в модуле, к которому они относятся.

**Запрос текущего состояния модуля (0x03)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x03 |
|---|---|------|

*Ответ:*

|   |   |      |       |
|---|---|------|-------|
| 4 | I | 0x03 | state |
|---|---|------|-------|

, где:

state – текущее состояние модуля:

0 – включен,

1 – выключен.

**Установить состояние модуля (0x04)***Запрос:*

|   |   |      |       |
|---|---|------|-------|
| 4 | I | 0x04 | state |
|---|---|------|-------|

, где:

state – состояние модуля (см. описание интерфейса).

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x04 | ack/nack |
|---|---|------|----------|

**Запрос текущего режима модуля (0x05)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x05 |
|---|---|------|

*Ответ:*

|   |   |      |      |
|---|---|------|------|
| 4 | I | 0x05 | mode |
|---|---|------|------|

, где:

mode – текущее режим работы модуля (см. описание интерфейса).

**Установить режим работы модуля (0x06)***Запрос:*

|   |   |      |      |
|---|---|------|------|
| 4 | I | 0x06 | mode |
|---|---|------|------|

, где:

mode – режим работы модуля (см. описание модуля).

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x06 | ack/nack |
|---|---|------|----------|

**Запрос текущих настроек модуля (0x07)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x07 |
|---|---|------|

*Ответ:*

|       |   |      |                     |     |                     |
|-------|---|------|---------------------|-----|---------------------|
| 3 + n | I | 0x07 | config <sub>1</sub> | ... | config <sub>n</sub> |
|-------|---|------|---------------------|-----|---------------------|

, где:

config – текущие настройки модуля (см. описание модуля),

n – размер настроек в байтах.

**Установить настройки модуля (0x08)***Запрос:*

|       |   |      |                     |     |                     |
|-------|---|------|---------------------|-----|---------------------|
| 3 + n | I | 0x08 | config <sub>1</sub> | ... | config <sub>n</sub> |
|-------|---|------|---------------------|-----|---------------------|

, где:

config – настройки модуля (см. описание модуля),

n – размер настроек модуля в байтах.

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x08 | ack/nack |
|---|---|------|----------|

**Запрос текущих настроек активного подмодуля (0x09)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x09 |
|---|---|------|

*Ответ:*

|       |   |      |                     |     |                     |
|-------|---|------|---------------------|-----|---------------------|
| 3 + n | I | 0x09 | config <sub>1</sub> | ... | config <sub>n</sub> |
|-------|---|------|---------------------|-----|---------------------|

, где:

config – текущие настройки подмодуля (см. описание подмодуля),

n – размер настроек в байтах.

**Установить настройки активного подмодуля (0x0A):***Запрос:*

|       |   |      |                     |     |                     |
|-------|---|------|---------------------|-----|---------------------|
| 3 + n | I | 0x0A | config <sub>1</sub> | ... | config <sub>n</sub> |
|-------|---|------|---------------------|-----|---------------------|

, где:

config – настройки активного подмодуля (см. описание подмодуля),

n – размер настроек подмодуля в байтах.

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x0A | ack/nack |
|---|---|------|----------|

При установке настроек, модуль проверяет соответствие поля данных размеру настроек, а также не выходят ли значения настроек за допустимые диапазоны.

**Сохранить настройки модуля и подмодуля в энергонезависимую память (0x0B)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x0B |
|---|---|------|

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x0B | ack/nack |
|---|---|------|----------|

При выполнении данной команды сохраняются только настройки выбранного модуля. Состояние и режим работы сохраняется отдельно с помощью соответствующей команды системного модуля.

**Загрузить настройки модуля и подмодуля из энергонезависимой памяти (0x0C)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x0C |
|---|---|------|

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x0C | ack/nack |
|---|---|------|----------|

**Запрос списка конфликтующих модулей (0x0D)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x0D |
|---|---|------|

*Ответ:*

|       |   |      |   |                     |     |                     |
|-------|---|------|---|---------------------|-----|---------------------|
| 4 + n | I | 0x0D | n | module <sub>1</sub> | ... | module <sub>n</sub> |
|-------|---|------|---|---------------------|-----|---------------------|

, где:

n – количество конфликтующих модулей,

module – порядковый номер конфликтующего модуля.

## Системный модуль

### Описание

Модуль позволяет пользователю:

- узнать количество модулей в устройстве,
- производить сохранение и загрузку всех настроек системы,
- запрашивать состояние настроек всех модулей системы,
- запрашивать состояние всех модулей системы,
- переводить систему в режим обновления встроенного ПО,
- узнать что за устройство, программную и аппаратную версию устройства, а также, серийный номер,
- имеет буфер значений координаты с возможностью переполнения (количество позиций зависит от устройства),
- задавать временной интервал цикла управления.

Буфер координат организован по принципу fifo, каждая позиция сохраняет статус датчика и преобразованную координату.

Для работы буфера координат необходимо:

- включить модули датчика, виртуальных входов/выходов и перенаправления сигналов.
- если запись в буфер происходит по таймеру, то необходимо сконфигурировать один из таймеров в модуле виртуальных входов/выходов, и назначить на его выход сигнал записи в буфер с помощью модуля перенаправления сигналов. Если необходим разрешающий сигнал, то используется логическая ячейка, на вход которой подается сигналы с таймера и разрешающего входа, а к выходу подключается сигнал записи в буфер.
- если запись в буфер происходит по внешнему сигналу или используется разрешающий внешний сигнал, то еще должен быть включен модуль входов/выходов.
- буфер также имеет выходной сигнал “переполнение”, который может быть перенаправлен на выходы устройства.

**id модуля**

Модуль имеет id 0.

Модуль всегда имеет нулевой порядковый номер.

**Режим работы:**

Модуль имеет только один режим работы 0, смена режима работы недоступна.

**Настройки модуля:**

| Размер(байт) | Описание                               |
|--------------|----------------------------------------|
| 2            | Интервал цикла управления системы (мс) |

**Подмодули**

Модуль не имеет подмодулей.

**Входные функциональные сигналы**

0 – запись показаний датчиков в буфер – захватывает текущую позицию для всех осей устройства во внутренний буфер координат по входному сигналу “1”.

**Выходные функциональные сигналы**

0 – буфер координат переполнен – устанавливается в “1” при переполнении буфера координат.

**Расширенные команды****Запрос количества модулей в устройстве (0x14)**

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | 0 | 0x14 |
|---|---|------|

*Ответ:*

|   |   |      |     |
|---|---|------|-----|
| 4 | 0 | 0x14 | num |
|---|---|------|-----|

, где:

num – число модулей в системе, беззнаковая.

**Запрос id устройства (0x15)**

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | 0 | 0x15 |
|---|---|------|

*Ответ:*

|   |   |      |    |    |
|---|---|------|----|----|
| 5 | 0 | 0x15 | id | id |
|---|---|------|----|----|

, где:

id – идентификатор устройства, беззнаковая.

**Запрос аппаратной версии устройства (0x16)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | 0 | 0x16 |
|---|---|------|

*Ответ:*

|   |   |      |        |        |
|---|---|------|--------|--------|
| 5 | 0 | 0x16 | hw_ver | hw_ver |
|---|---|------|--------|--------|

, где:

hw\_ver – аппаратная версия устройства, беззнаковая.

**Запрос программной версии устройства (0x17)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | 0 | 0x17 |
|---|---|------|

*Ответ:*

|   |   |      |        |        |
|---|---|------|--------|--------|
| 5 | 0 | 0x17 | sw_ver | sw_ver |
|---|---|------|--------|--------|

, где:

sw\_ver – программная версия устройства, беззнаковая.

**Запрос серийного номера устройства (0x18)**

В ответ на команду устройство возвращает строку в ASCII длиной в 15 символов, содержащую серийный номер устройства.

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | 0 | 0x18 |
|---|---|------|

*Ответ:*

|        |        |        |        |        |        |        |        |        |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 18     | 0      | 0x18   | serial | serial | serial | serial | serial | serial |
| serial | serial | serial | serial | serial | serial | serial | serial | serial |

, где:

serial – серийный номер устройства.

**Сохранить настройки всей системы (0x19)**

Сохраняет в энергонезависимой памяти устройства состояние и режим работы всех модулей, а также, настройки включенных модулей на момент сохранения.

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | 0 | 0x19 |
|---|---|------|

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | 0 | 0x19 | ack/nack |
|---|---|------|----------|

**Загрузить настройки всей системы (0x1A)**

Загружает из энергонезависимой памяти устройства состояние и режим работы для всех модулей системы, а также настройки для включенных модулей. Если сохраненные данные о состоянии модулей повреждены, то изменение состояния модулей не производится и настройки всех модулей считаются испорченными.

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | 0 | 0x1A |
|---|---|------|

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | 0 | 0x1A | ack/nack |
|---|---|------|----------|

**Запрос состояния настроек всех модулей устройства (0x1B)**

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | 0 | 0x1B |
|---|---|------|

*Ответ:*

|   |   |      |       |       |       |       |
|---|---|------|-------|-------|-------|-------|
| 7 | 0 | 0x1B | state | state | state | state |
|---|---|------|-------|-------|-------|-------|

, где:

state – состояние настроек всех модулей.

State представляет из себя битовое поле, каждый бит в котором соответствует состоянию настроек модуля с тем же порядковым номером. Если бит в “1” – настройки повреждены, если “0” – настройки в порядке.

**Запрос состояния всех модулей устройства (0x1C)**

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | 0 | 0x1C |
|---|---|------|

*Ответ:*

|   |   |      |       |       |       |       |
|---|---|------|-------|-------|-------|-------|
| 7 | 0 | 0x1C | state | state | state | state |
|---|---|------|-------|-------|-------|-------|

, где:

state – состояние всех модулей.

State представляет из себя битовое поле, каждый бит в котором соответствует состоянию модуля с тем же порядковым номером. Если бит в “1” – модуль включен, если “0” – выключен.

**Запрос-маркер (0x1D)**

Позволяет пометить пакет управления для облегчения сопоставления запрос – ответа.

Маркер представляет из себя любое двухбайтное число. Устройство возвращает в ответе маркер без изменений.

*Запрос:*

|   |   |      |      |      |
|---|---|------|------|------|
| 5 | 0 | 0x1D | mark | mark |
|---|---|------|------|------|

, где:

mark – маркер.

*Ответ:*

|   |   |      |      |      |
|---|---|------|------|------|
| 5 | 0 | 0x1D | mark | mark |
|---|---|------|------|------|

**Запрос чтения буфера координат (0x1E)**

Команда запроса у устройства содержимого буфера координат. Возвращает содержимое буфера и статусный байт, содержащий информацию о том, сколько координат в буфере было на момент начала чтения и было ли переполнение. При чтении буфера флаг переполнения сбрасывается.

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | 0 | 0x1E |
|---|---|------|

*Ответ:*

|                    |                    |                    |                    |          |                     |                     |                    |                    |                    |                    |
|--------------------|--------------------|--------------------|--------------------|----------|---------------------|---------------------|--------------------|--------------------|--------------------|--------------------|
| 5+10*n             | 0                  | 0x1E               | b_status           | b_status | status <sub>1</sub> | status <sub>1</sub> | coord <sub>1</sub> | coord <sub>1</sub> | coord <sub>1</sub> | coord <sub>1</sub> |
| coord <sub>1</sub> | coord <sub>1</sub> | coord <sub>1</sub> | coord <sub>1</sub> | ...      | status <sub>n</sub> | status <sub>n</sub> | coord <sub>n</sub> | coord <sub>n</sub> | coord <sub>n</sub> | coord <sub>n</sub> |
| coord <sub>n</sub> | coord <sub>n</sub> | coord <sub>n</sub> |                    |          |                     |                     |                    |                    |                    |                    |

, где:

b\_status – состояние буфера координат, старший бит – флаг переполнения, остальные – количество записей координат на момент команды,

status – статус датчика,

coord – координата датчика,

n – количество записей.

**Перевести устройство в режим обновления прошивки (0x70)**

*Запрос:*

|   |   |      |     |     |
|---|---|------|-----|-----|
| 5 | 0 | 0x70 | key | key |
|---|---|------|-----|-----|

, где:

key – 0x3A27



## Модуль датчика

### Описание

Модуль позволяет осуществлять работу с датчиком. Поддерживаемый тип датчиков зависит от подключенных подмодулей. Независимо от типа датчика, интерфейс выдает 64 битную координату и 16 битный статус. Поле статуса делится на две группы по 8 бит старшая и младшая. В младшей группе отображаются неисправности датчика (биты ошибки/предупреждения переданные датчиком, или повреждение линии). В старшей группе отображаются код операции подмодуля, индикация захваченности референтной метки и процесс захвата, контроль формата сообщения датчика.

| Бит     | Описание                                                                      |
|---------|-------------------------------------------------------------------------------|
| 15 - 13 | Код текущей операции подмодуля                                                |
| 12      | Повреждение сохраненного смещения абсолютной оси (1 – ошибка, 0 – все хорошо) |
| 11      | Процесс поиска референтной метки (1 – ищем, 0 – не ищем)                      |
| 10      | Повреждена таблица коррекции (1 – повреждена, 0 – не повреждена)              |
| 9       | Референтная метка не захвачена (1 – не захвачена, 0 – захвачена или нету)     |
| 8       | Ошибка чтения координаты (1 – ошибка, 0 – все хорошо)                         |
| 7 - 0   | Поле ошибок датчика, см описание подмодуля (1 – ошибка, 0 – все хорошо)       |

Модуль имеет возможность привести код датчика к физической величине путем задания коэффициента пересчета. Коэффициент задается с помощью числителя и знаменателя. Пользователь должен максимально упростить дробь коэффициента, это увеличит рабочий диапазон системы (не произойдет переполнение). Также модуль позволяет задать смещение координаты относительно нуля датчика и развернуть ось.

В модуле встроена функция коррекции по таблице методом линейной интерполяции. Пользователю доступны функции для просмотра содержания таблицы, добавления, удаления и редактирования точек таблицы. Таблица хранится в энергонезависимой памяти, при каждом включении устройства проверяется целостность таблицы. Результат проверки отображается с помощью соответствующего бита в статусе.

Для правильной работы функции коррекции необходимо задать в параметрах тип оси (линейная/угловая) и, в случае угловой, количество дискрет на оборот.

Точки в таблице коррекции отсортированы по возрастанию по координате. Сортировка происходит в момент добавления новой точки. Максимальный размер коррекции зависит от устройства. Незаполненные ячейки таблицы помечаются значением коррекции соответствующим установленному старшему биту.

Модуль поддерживает работу с датчиками, передающими координату по интерфейсу в коде Грея.

Модуль позволяет получать координату в трех системах координат (двух абсолютных со смещением относительно нуля датчика и одной в приращениях), а также скорректированную позицию датчика в дискретах без преобразования.

**id модуля**

Модуль имеет id 1.

**Режим работы:**

Режим работы модуля определяет какой из подмодулей (тип датчика) используется в данный момент.

**Настройки интерфейса:**

| Размер(байт) | Описание                                                                     |
|--------------|------------------------------------------------------------------------------|
| 4            | Коэффициент А (1... 4294967295)                                              |
| 4            | Коэффициент В (1... 4294967295)                                              |
| 1            | Разворот оси (0 – выкл, 1 - вкл)                                             |
| 1            | Тип оси (0 – линейная, 1 - угловая)                                          |
| 8            | Смещение ноля станка (смещение нуля оси относительно нуля датчика, знаковая) |
| 4            | Количество импульсов на оборот датчика (для угловой оси, беззнаковая)        |
| 1            | Конвертирование координаты из кода Грея (0 – выкл, 1 – вкл)                  |

**Подмодули**

Модуль может иметь следующие подмодули:

- 0 – инкрементный,
- 1 – ssi,
- 2 – biss,
- 3 – аналоговый.

**Типы систем отсчета:**

- 0 – скорректированная позиция датчика в дискретах без преобразований,
- 1 – работа в приращениях (G52),
- 2 – абсолютная1 (G53),
- 3 – абсолютная2 (G54).

Важно помнить, что позиции всех систем отсчета рассчитываются относительно СО абсолютная1, поэтому при установке смещения для нее или захвате референтной метки (инкрементные датчики) это повлияет на другие СО.

**Входные функциональные сигналы**

0 – обнуление позиции – по входному сигналу “1” обнуляет текущую координату датчика для системы координат абсолютная1 (доступно только для инкрементных датчиков);

1 – захват референтной метки – по входному сигналу “1” начинает процесс захвата референтной метки.

**Выходные функциональные сигналы**

0 – ошибка датчика – устанавливается в “1” когда есть проблемы с подключением датчика, таблица коррекции повреждена или повреждено смещение;

1 – референтная метка не захвачена – устанавливается в “1” когда у датчика есть референтная метка и она не захвачена.

## Расширенные команды

**Обновить показания датчика (0x14)**

По этой команде устанавливается флаг опроса датчика. Опрос датчика произойдет по первому запросу координаты.

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x14 |
|---|---|------|

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x14 | ack/nack |
|---|---|------|----------|

**Запрос показаний датчика (0x15)**

Команда возвращает значения, захлопнутые во внутренний регистр устройства.

*Запрос:*

|   |   |      |      |
|---|---|------|------|
| 4 | I | 0x15 | axis |
|---|---|------|------|

, где:

axis – тип системы отсчета, для которой запрашивается координата.

*Ответ:*

|    |   |      |       |       |       |       |       |       |       |       |        |        |
|----|---|------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|
| 13 | I | 0x15 | coord | coord | coord | coord | coord | coord | coord | coord | status | status |
|----|---|------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|

, где:

status – статус датчика,

coord – координата после коррекции, приведенная к единицам СИ со смещением (знаковая).

**Установить текущую систему отсчета (0x16)**

*Запрос:*

|   |   |      |      |
|---|---|------|------|
| 4 | I | 0x16 | axis |
|---|---|------|------|

, где:

axis – система отсчета.

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x16 | ack/nack |
|---|---|------|----------|

**Запросить текущую систему отсчета (0x17)**

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x17 |
|---|---|------|

*Ответ:*

|   |   |      |      |
|---|---|------|------|
| 4 | I | 0x17 | axis |
|---|---|------|------|

, где:

axis – система отсчета.

**Установить смещение по датчику (0x18)**

Команда устанавливает смещение равное текущей позиции с противоположным знаком для заданной системы отсчета. Для относительных систем, смещение хранится в энергозависимой памяти и при выключении устройства - теряется.

Для G53, смещение записывается и хранится вместе с настройками датчика (смещение будет сохранено вместе с сохранением настроек). Запись остальных абсолютных систем координат не регламентируется и происходит согласно описанию устройства.

Результат выполнения команды может быть отрицательным, если во время установки смещения произошла ошибка датчика, референтная метка была не захвачена (для G52).

*Запрос:*

|   |   |      |      |
|---|---|------|------|
| 4 | I | 0x18 | axis |
|---|---|------|------|

, где:

axis – система отсчета.

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x18 | ack/nack |
|---|---|------|----------|

**Установить смещение вручную (0x19)**

Команда записывает заданное смещение в поле настроек. Настройки при этом не сохраняются. Чтобы смещение не потерялось после выключения устройства, пользователь должен выполнить команду записи настроек.

*Запрос:*

|    |   |      |      |        |        |        |        |        |        |        |        |
|----|---|------|------|--------|--------|--------|--------|--------|--------|--------|--------|
| 12 | I | 0x19 | axis | offset | offset | offset | offset | offset | offset | offset | offset |
|----|---|------|------|--------|--------|--------|--------|--------|--------|--------|--------|

, где:

axis – система отсчета,

offset – смещение (знаковая).

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x19 | ack/nack |
|---|---|------|----------|

**Запрос текущего смещения (0x1A)**

Возвращает текущее смещение нуля оси относительно нуля датчика.

*Запрос:*

|   |   |      |      |
|---|---|------|------|
| 3 | I | 0x1A | axis |
|---|---|------|------|

, где:

axis – система отсчета.

*Ответ:*

|    |   |      |        |        |        |        |        |        |        |        |
|----|---|------|--------|--------|--------|--------|--------|--------|--------|--------|
| 11 | I | 0x1A | offset | offset | offset | offset | offset | offset | offset | offset |
|----|---|------|--------|--------|--------|--------|--------|--------|--------|--------|

, где:

offset – текущее смещение (знаковая).

**Запрос количества ошибок датчика (0x1B)**

Возвращает количество ошибок считывания позиции датчика, произошедших с момента последнего обнуления.

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x1B |
|---|---|------|

*Ответ:*

|   |   |      |     |     |
|---|---|------|-----|-----|
| 5 | I | 0x1B | err | err |
|---|---|------|-----|-----|

, где:

err – текущее количество ошибок датчика (беззнаковая).

**Сбросить счетчик ошибок датчика (0x1C)**

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x1C |
|---|---|------|

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x1C | ack/nack |
|---|---|------|----------|

**Запрос последней ошибки датчика (0x1D)**

Возвращает статус последнего ошибочного считывания координаты датчика.

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x1D |
|---|---|------|

*Ответ:*

|   |   |      |        |        |
|---|---|------|--------|--------|
| 5 | I | 0x1D | status | status |
|---|---|------|--------|--------|

, где:

status – последняя ошибка датчика

**Инициализация процесса коррекции (0x32)**

Команда создает в памяти устройства пустую таблицу коррекции и переключает на нее функцию коррекции.

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x32 |
|---|---|------|

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x32 | ack/nack |
|---|---|------|----------|

**Получить информацию о точке коррекции N (0x33)**

Команда возвращает значение координаты и размера коррекции для выбранной точки из таблицы. Если в таблице меньше точек, устройство отвечает отказом.

*Запрос:*

|   |   |      |   |
|---|---|------|---|
| 3 | I | 0x33 | N |
|---|---|------|---|

, где N – номер точки в таблице коррекции.

*Ответ:*

|    |   |      |   |   |   |   |   |   |   |   |   |   |
|----|---|------|---|---|---|---|---|---|---|---|---|---|
| 19 | I | 0x33 | X | X | X | X | X | X | X | X | C | C |
| R  | R | R    | R | R | R |   |   |   |   |   |   |   |

, где:

X – координата (знаковая),

C – коррекция (знаковая),

R – зарезервировано.

Количество байт зарезервированной области и коррекции зависит от размера коррекции конкретного устройства. Может быть запрошено соответствующей командой.

**Получить количество точек в таблице коррекции (0x34)**

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x34 |
|---|---|------|

*Ответ:*

|   |   |      |   |
|---|---|------|---|
| 4 | I | 0x34 | N |
|---|---|------|---|

, где:

N – количество точек в таблице, беззнаковая.

**Добавить точку коррекции в таблицу (0x35)**

Команда добавляет в таблицу точку с текущей координатой и заданной коррекцией.

Если во время выполнения команды произошла ошибка датчика или не захвачена референтная метка – устройство отвечает отказом.

*Запрос:*

|    |   |      |   |   |   |   |   |   |   |   |
|----|---|------|---|---|---|---|---|---|---|---|
| 11 | I | 0x35 | C | C | R | R | R | R | R | R |
|----|---|------|---|---|---|---|---|---|---|---|

, где:

R – зарезервировано,

C – значение коррекции для текущей координаты, знаковая.

Количество байт зарезервированной области и коррекции зависит от размера коррекции конкретного устройства. Может быть запрошено соответствующей командой.

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x35 | ack/nack |
|---|---|------|----------|

**Сохранить таблицу коррекции (0x36)**

Сохраняет текущую таблицу коррекции в энергонезависимую память устройства.

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x36 |
|---|---|------|

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x36 | ack/nack |
|---|---|------|----------|

**Прервать процесс коррекции (0x37)**

Перезаписывает текущую таблицу коррекции сохраненной в энергонезависимую память.

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x37 |
|---|---|------|

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x37 | ack/nack |
|---|---|------|----------|

**Удалить точку из таблицы коррекции (0x38)**

*Запрос:*

|   |   |      |   |
|---|---|------|---|
| 4 | I | 0x38 | N |
|---|---|------|---|

, где:

N – номер точки, которую надо удалить, беззнаковая.

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x38 | ack/nack |
|---|---|------|----------|

**Отредактировать точку в таблице коррекции (0x39)**

Заменяет значение коррекции для выбранной точки в таблице на заданную. Устройство отвечает отказом если в таблице нету точки с соответствующим номером.

*Запрос:*

|    |   |      |   |   |   |   |   |   |   |   |   |
|----|---|------|---|---|---|---|---|---|---|---|---|
| 12 | I | 0x39 | N | C | C | R | R | R | R | R | R |
|----|---|------|---|---|---|---|---|---|---|---|---|

, где:

N – номер точки в таблице, которую надо отредактировать, беззнаковая,

R – зарезервировано,

C – значение коррекции, знаковая.

Количество байт зарезервированной области и коррекции зависит от размера коррекции конкретного устройства. Может быть запрошено соответствующей командой.

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x39 | ack/nack |
|---|---|------|----------|

**Добавить вручную точку коррекции в таблицу (0x3A)***Запрос:*

|    |   |      |   |   |   |   |   |   |   |
|----|---|------|---|---|---|---|---|---|---|
| 19 | I | 0x3A | X | X | X | X | X | X | X |
| X  | C | C    | R | R | R | R | R | R |   |

, где:

X – координата, знаковая,

R – зарезервировано,

C – коррекция, знаковая.

Количество байт зарезервированной области и коррекции зависит от размера коррекции конкретного устройства. Может быть запрошено соответствующей командой.

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x3A | ack/nack |
|---|---|------|----------|

**Запрос размера поля данных коррекции (0x3B)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x3B |
|---|---|------|

*Ответ:*

|   |   |      |   |
|---|---|------|---|
| 4 | I | 0x3B | N |
|---|---|------|---|

, где:

N – размер в байтах поля данных коррекции.



## Подмодули

### Подмодуль инкрементный

#### Описание

Инкрементный подмодуль позволяет подключать к устройству инкрементные датчики. Модуль имеет 64 битный счетчик координаты. Захват референтной метки происходит только по команде пользователя. Подмодуль обеспечивает контроль дифференциальных пар линии датчика и устанавливает соответствующие биты ошибки в поле статуса. При отсутствии у датчика референтной метки (согласно настройкам) контроль сигнала референтной метки не производится.

#### id подмодуля

Подмодуль имеет id 0.

#### Настройки подмодуля:

| Размер(байт) | Описание                                                 |
|--------------|----------------------------------------------------------|
| 1            | Наличие у датчика референтной метки (0 – нету, 1 - есть) |
| 1            | Поменять местами сигналы А и В (0 – нет, 1 - да)         |
| 1            | Использовать фильтр дребезга (0 – нет, 1 - да)           |
| 1            | Уровень фильтрации (0...63)                              |

#### Статус датчика

|          |    |    |    |            |    |     |     |          |   |   |   |   |       |       |       |
|----------|----|----|----|------------|----|-----|-----|----------|---|---|---|---|-------|-------|-------|
| 15       | 14 | 13 | 12 | 11         | 10 | 9   | 8   | 7        | 6 | 5 | 4 | 3 | 2     | 1     | 0     |
| reserved |    |    |    | ref search |    | ref | err | reserved |   |   |   |   | R err | B err | A err |

, где:

reserved – зарезервировано,

ref search – процесс поиска референтной метки,

ref – референтная метка (1 – не захвачена, 0 - захвачена или у датчика нет референтной метки),

err – флаг ошибки,

A err – линия А рассогласована,

B err – линия В рассогласована,

R err – линия R рассогласована.

## Команды

### Начать поиск референтной метки (0x3C)

Устройство переходит в режим поиска референтной метки если согласно настройкам она есть, иначе, значение счетчика обнуляется и устройство выходит из режима поиска референтной метки. Во время поиска референтной метки счетчик продолжает работать. Из режима поиска референтной метки можно выйти:

- захватив референтную метку,
- выполнив команду остановки поиска референтной метки.

#### Запрос:

|   |   |      |
|---|---|------|
| 3 | I | 0x3C |
|---|---|------|

#### Ответ:

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x3C | ack/nack |
|---|---|------|----------|

**Сбросить захват референтной метки (0x3D)**

При выполнении команды устанавливается статус «референтная метка не захвачена». Значение счетчика остается без изменений.

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x3D |
|---|---|------|

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x3D | ack/nack |
|---|---|------|----------|

**Отменить поиск референтной метки (0x3E)**

Команда отменяет режим поиска референтной метки.

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x3E |
|---|---|------|

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x3E | ack/nack |
|---|---|------|----------|

**Обнулить показания счетчика (0x3F)**

Команда обнуляет счетчик дискрет. Состояние захвата референтной метки сохраняется.

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x3F |
|---|---|------|

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x3F | ack/nack |
|---|---|------|----------|

**Подмодуль SSI****Описание**

Данный подмодуль позволяет подключать к устройству абсолютные датчики с интерфейсом SSI. Подмодуль поддерживает датчики, длина посылок которых не превышает 64 бит (сумма количества бит данных и статуса не должна превышать 64). Подмодуль позволяет задавать начальный интервал ожидания готовности датчика после первого фронта, частоту тактирования, производить двойное чтение в цикле для контроля целостности посылки. Внутренний счетчик подмодуля имеет разрядность 64 бита. При использовании датчика, поле данных которого имеет меньшую разрядность, чем внутренний счетчик, старшие биты счетчика могут быть использованы для подсчета оборотов. Число оборотов не сохраняется в энергонезависимую память, поэтому при отключении питания будет утеряно. Также данную функцию не следует использовать при большом количестве ошибок чтения координаты с датчика. Также подмодуль поддерживает линейные датчики со смещенным нулем (имеется отрицательная область).

Подмодуль поддерживает до 16 бит статуса датчика. Статусные биты могут быть расположены как перед, так и после поля данных (или одновременно до и после). Для каждого бита статуса может быть индивидуально задана инверсия. Биты статуса датчика отображаются в соответствующем поле регистра статуса. Если их число превышает 8, то биты 15:8 объединяются с битами 7:0 через логическое “или” после наложения маски инверсии.

Подмодуль производит контроль формата сообщения с датчика:

- проверка уровня “1” между опросами,
- проверка уровня “0” таймаута,
- проверка уровня “0” по первому спаду для датчиков СКБИС со статус битами,
- проверка двойным чтением на достоверность полученной информации.

**id подмодуля**

Подмодуль имеет id 1.

**Настройки подмодуля:**

| Размер(байт) | Описание                                                                                                      |
|--------------|---------------------------------------------------------------------------------------------------------------|
| 2            | Маска инверсии битов ошибки (0...65535)                                                                       |
| 1            | Количество бит данных (4...64 – (количество бит ошибки))                                                      |
| 1            | Количество бит ошибки (0...16)                                                                                |
| 1            | Начало слова данных (0...(количество бит ошибки))                                                             |
| 1            | Частота тактирования (0 – 2 МГц, 1 – 1 МГц, 2 – 500 КГц, 4 – 250 КГц, 5 – 200 КГц, 8 – 125 КГц, 10 – 100 КГц) |
| 1            | Пауза чтения (0...255) мкс                                                                                    |
| 1            | Двойное чтение (0 – выкл, 1 - вкл)                                                                            |
| 1            | Считать обороты датчика (0 – выкл, 1 - вкл)                                                                   |
| 1            | Датчик имеет смещенный ноль (0 – нет, 1 - да)                                                                 |
| 1            | Проверка индикатора бита ошибки для датчиков СКБИС (0 – выкл, 1 - вкл)                                        |

**Статус датчика**

|          |    |    |    |    |    |   |     |       |   |   |   |   |   |   |   |
|----------|----|----|----|----|----|---|-----|-------|---|---|---|---|---|---|---|
| 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8   | 7     | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| reserved |    |    |    |    |    |   | err | alarm |   |   |   |   |   |   |   |

, где:

reserved – зарезервировано,

err – ошибка датчика,

alarm – биты ошибки.

**Команды****Обнулить счетчик оборотов (0x3C)**

Сбрасывается счетчик оборотов. Координата соответствует коду датчика.

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x3C |
|---|---|------|

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x3C | ack/nack |
|---|---|------|----------|

### **Подмодуль BISS**

#### **Описание**

Данный подмодуль позволяет подключать к устройству абсолютные датчики с интерфейсом BISS-C. Поддерживаются датчики, размер сообщения которых не превышает 64 бит (сумма бит данных, статуса и контрольной суммы).

Модуль поддерживает работу с датчиками имеющими до 8 битов статуса и позволяет задать индивидуально для каждого статусного бита инверсию.

Позволяет задать частоту тактирования, выбрать полином (0x43 / 0x25) для расчета контрольной суммы, а также начальное значение.

При использовании датчика, поле данных которого имеет меньшую разрядность, чем внутренний счетчик (64 бита), старшие биты счетчика могут быть использованы для подсчета оборотов. Число оборотов не сохраняется в энергонезависимую память, поэтому при отключении питания будет утеряно. Также данную функцию не следует использовать при большом количестве ошибок чтения координаты с датчика. Также модуль поддерживает линейные датчики со смещенным нулем (имеется отрицательная область).

Позволяет производить запись и чтение внутренних регистров датчика.

Процесс записи регистров:

- запись в устройство требуемых значений регистров,
- команда устройству на запись значений из внутренних регистров в датчик,
- ожидание окончания выполнения операции записи,
- запрос результата выполнения операции.

Процесс чтения регистров:

- команда устройству на чтение значений регистров в память устройства,
- ожидание окончания выполнения операции,
- запрос результата выполнения операции,
- при успешном выполнении – считать значения регистров из устройства.

**id подмодуля**

Подмодуль имеет id 2.

**Настройки подмодуля:**

| Размер(байт) | Описание                                                                                           |
|--------------|----------------------------------------------------------------------------------------------------|
| 1            | Количество бит данных (4...64 – (размер поля CRC + кол-во статусных битов))                        |
| 1            | Количество статусных бит (0...8)                                                                   |
| 1            | Частота тактирования (1 – 1 МГц, 2 – 500 КГц, 4 – 250 КГц, 5 – 200 КГц, 8 – 125 КГц, 10 – 100 КГц) |
| 1            | Размер поля CRC (0...6)                                                                            |
| 1            | CRC полином (0x43, 0x25)                                                                           |
| 1            | Начальное значение CRC (0...15)                                                                    |
| 1            | Маска инверсии статусных бит (0...255)                                                             |
| 1            | Считать обороты датчика (0 – выкл, 1 - вкл)                                                        |
| 1            | Датчик имеет смещенный ноль (0 – нет, 1 - да)                                                      |

**Статус датчика**

|         |    |    |    |    |    |   |     |        |   |   |   |   |   |   |   |
|---------|----|----|----|----|----|---|-----|--------|---|---|---|---|---|---|---|
| 15      | 14 | 13 | 12 | 11 | 10 | 9 | 8   | 7      | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| op_code |    |    |    |    |    |   | err | status |   |   |   |   |   |   |   |

, где:

err – ошибка датчика,

status –статусные биты,

op\_code – код выполняемого модулем действия.

**Список кодов операции модуля:**

| Код  | Операция                                                      |
|------|---------------------------------------------------------------|
| 0x01 | Синхронизация значений регистров между датчиком и устройством |

**Команды****Обнулить счетчик оборотов (0x3C)**

Сбрасывается счетчик оборотов. Координата соответствует коду датчика.

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x3C |
|---|---|------|

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x3C | ack/nack |
|---|---|------|----------|

**Запись регистров EDS в память устройства (0x3E)**

*Запрос:*

|         |   |      |      |     |                   |     |                     |
|---------|---|------|------|-----|-------------------|-----|---------------------|
| 5 + num | I | 0x3E | addr | num | data <sub>0</sub> | ... | data <sub>num</sub> |
|---------|---|------|------|-----|-------------------|-----|---------------------|

, где:

addr – адрес первого регистра,

num – количество регистров для записи,

data – данные регистров.

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x3E | ack/nack |
|---|---|------|----------|

**Чтение регистров EDS из памяти устройства (0x3F)**

*Запрос:*

|   |   |      |      |     |
|---|---|------|------|-----|
| 5 | I | 0x3F | addr | num |
|---|---|------|------|-----|

, где:

addr – адрес первого регистра,

num – количество регистров для чтения.

*Ответ:*

|         |   |      |                   |     |                     |
|---------|---|------|-------------------|-----|---------------------|
| 3 + num | I | 0x3F | data <sub>0</sub> | ... | data <sub>num</sub> |
|---------|---|------|-------------------|-----|---------------------|

, где:

num – количество регистров для чтения,

data – данные регистров.

**Запустить процесс синхронизации значений регистров EDS в памяти устройства и датчике (0x40)**

*Запрос:*

|   |   |      |    |    |      |     |
|---|---|------|----|----|------|-----|
| 7 | I | 0x40 | id | op | addr | num |
|---|---|------|----|----|------|-----|

, где:

id – id датчика,

op – направление синхронизации (1 – с устройством, 2 – с датчиком),

addr – адрес первого регистра,

num – количество регистров для синхронизации.

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x40 | ack/nack |
|---|---|------|----------|

Прочитать результат процесса обмена (0x41)

*Запрос:*

|   |   |      |  |
|---|---|------|--|
| 3 | I | 0x41 |  |
|---|---|------|--|

*Ответ:*

|   |   |      |     |
|---|---|------|-----|
| 4 | I | 0x41 | res |
|---|---|------|-----|

, где:

res – результат синхронизации (0 – не было процесса обмена, 1 – пересылка команды, 2 – пересылка данных, 3 – успешно, 4 – неудачно).



**Подмодуль аналоговый****Описание**

Данный подмодуль позволяет устройству работать с аналоговыми датчиками. Тип подключаемого датчика определяется кодом заказа устройства.

Пользователю доступно задание:

- уровней сигнала, выход за которые воспринимается интерфейсом как неподключенный датчик,
- уровней сигнала, выход за которые воспринимается как ошибка датчика (статус ошибки),
- коэффициент фильтрации.

**id подмодуля**

Подмодуль имеет id 3.

**Настройки подмодуля:**

| Размер(байт) | Описание                                              |
|--------------|-------------------------------------------------------|
| 2            | Нижняя граница рабочего диапазона датчика (дискреты)  |
| 2            | Верхняя граница рабочего диапазона датчика (дискреты) |
| 2            | Нижняя граница ошибки датчика (дискреты)              |
| 2            | Верхняя граница ошибки датчика (дискреты)             |
| 2            | Временной коэффициент фильтрации                      |

**Статус датчика**

|         |    |    |    |    |    |   |     |          |   |   |   |   |   |        |   |
|---------|----|----|----|----|----|---|-----|----------|---|---|---|---|---|--------|---|
| 15      | 14 | 13 | 12 | 11 | 10 | 9 | 8   | 7        | 6 | 5 | 4 | 3 | 2 | 1      | 0 |
| op_code |    |    |    |    |    |   | err | reserved |   |   |   |   |   | status |   |

, где:

err – ошибка датчика,

status – статусные биты,

op\_code – код выполняемого модулем действия,

reserved - зарезервировано.

**Команды**

Подмодуль не имеет команд.

## Модуль RS485

### Описание

Данный модуль обеспечивает управление параметрами, а также работу интерфейса RS485.

#### id модуля

Модуль имеет id 2.

#### Режим работы:

Режим работы модуля переключает используемый в данный момент подмодуль.

#### Настройки модуля:

Модуль не имеет настроек

#### Подмодули

Модуль может иметь следующие подмодули:

0 – modbus RTU,

1 – сппу,

2 – протокол ЛИР-915/6.

#### Входные функциональные сигналы

Модуль не имеет входных функциональных сигналов.

#### Выходные функциональные сигналы

0 – ошибка связи – устанавливает выход в “1” когда: в режиме мастера достигнута критическая серия последовательных ошибок связи с ведомым; в режиме ведомого – превышено время опроса мастером.

#### Расширенные команды

Модуль не имеет расширенных команд.

## Подмодули

### Подмодуль Modbus RTU

#### Описание

Подмодуль позволяет осуществлять связь с другими устройствами согласно протоколу modbus rtu. В режиме ведущего – позволяет проводить опрос ведомых устройств, контролируя формат пакета, контрольную сумму и таймаут ответа, в режиме ведомого – предоставляет доступ к регистрам устройства через стандартные средства протокола modbus rtu.

Процедура опроса ведомого устройства в режиме ведущего:

- переключение в режим ведущего,
- задание параметров соединения (адрес, скорость передачи, стоп биты и биты четности),
- отправка команды устройству, содержащую modbus сообщение без CRC,
- ожидание конца modbus транзакции, путем запроса ошибки связи,
- при успешной транзакции – запрос modbus ответа.

Работа в режиме ведомого устройства:

- задание параметров соединения (адрес, скорость передачи, стоп биты и биты четности),
- устройство доступно через стандартные команды протокола modbus rtu согласно описанию регистров на конкретное устройство.

#### id подмодуля

Подмодуль имеет id 0.

#### Настройки подмодуля:

| Размер(байт) | Описание                                                  |
|--------------|-----------------------------------------------------------|
| 1            | Режим работы (0 – ведомый, 1 - ведущий)                   |
| 1            | Адрес (1...247)                                           |
| 2            | Таймаут, мс (0...1000)                                    |
| 4            | Скорость передачи, бод/с (0...3750000)                    |
| 1            | Контроль четности (0 – even, 1 – odd, 4 – нет)            |
| 1            | Количество стоп-бит (0 – 1 бит, 1 – 1,5 бита, 2 – 2 бита) |

В режиме ведущего с помощью параметра “таймаут” задается время ожидания ответа от ведомого. В режиме ведомого – этот параметр задает максимальное время опроса устройства, при превышении которого, будет считаться, что соединение потеряно и установится выходной сигнал “ошибка связи”.

**Команды****Запрос количества ошибок связи (0x3C)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x3C |
|---|---|------|

*Ответ:*

|   |   |      |     |     |     |     |
|---|---|------|-----|-----|-----|-----|
| 7 | I | 0x3C | err | err | err | err |
|---|---|------|-----|-----|-----|-----|

, где:

err – количество ошибок передачи.

**Сбросить счетчик ошибок связи (0x3D)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x3D |
|---|---|------|

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x3D | ack/nack |
|---|---|------|----------|

**Запрос кода последней ошибки связи (0x3E)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x3E |
|---|---|------|

*Ответ:*

|   |   |      |     |
|---|---|------|-----|
| 4 | I | 0x3E | err |
|---|---|------|-----|

, где:

err – код последней ошибки передачи.

*Коды ошибок передачи:*

0x01 – 0x08 – стандартные коды ошибок Modbus протокола,

0x09 – таймаут ответа ведомого,

0x0A – ошибка CRC,

0x0B – нет ошибок,

0x0C – ошибка линии (контроль четности, формат пакета, переполнение).

Во время процесса передачи, в ответ на данную команду возвращается nack.

**Инициализация начала передачи (0x3F)***Запрос:*

|          |   |      |                 |     |                 |
|----------|---|------|-----------------|-----|-----------------|
| 3 + size | I | 0x3F | mp <sub>0</sub> | ... | mp <sub>n</sub> |
|----------|---|------|-----------------|-----|-----------------|

, где:

mp – сообщение modbus без CRC,

size – размер сообщения modbus в байтах.

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x3F | ack/nack |
|---|---|------|----------|

**Запрос результата передачи (0x40)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x40 |
|---|---|------|

*Ответ:*

|          |   |      |                 |     |                 |
|----------|---|------|-----------------|-----|-----------------|
| 3 + size | I | 0x40 | mp <sub>0</sub> | ... | mp <sub>n</sub> |
|----------|---|------|-----------------|-----|-----------------|

, где:

mp – сообщение modbus без CRC,

size – размер сообщения modbus в байтах.

Во время передачи, а также, если передача закончилась ошибкой – ответ на команду nack.

**Подмодуль СППУ****Описание**

Данный подмодуль позволяет производить синхронизацию между устройствами ЛИР, содержащими такой же модуль, а также подключать внешний контроллер входов/выходов ЛИР-986. Одно из устройств выступает в роли ведущего, остальные – ведомого. Синхронизация осуществляется с помощью специального протокола, который позволяет производить опрос группы устройств за меньший промежуток времени, чем через стандартный протокол Modbus RTU, а так же обеспечивает одновременное защелкивание интерфейсами координат датчиков.

При необходимости, данный протокол может быть реализован пользователем на собственном устройстве, выступающем в роли ведущего.

Настройки соединения:

- скорость: 250кбод/с,
- кол-во бит данных: 8,
- кол-во стоп бит: 1,
- контроль четности: 1 бит, multidrop.

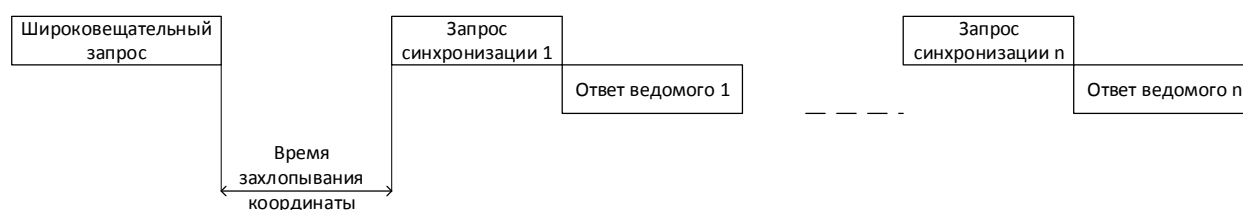
**Описание протокола**

Каждый сеанс обмена информацией начинается с широковещательного запроса ведущего устройства. Он содержит команду с параметрами для выполнения, а также, информацию для синхронизации виртуальных входов/выходов.

Получив этот запрос, ведомые устройства захлопывают текущую координату (согласно настройкам) и выполняют требуемую команду.

Ведущее устройство выжидает время, необходимое для обновления координаты самым медленным устройством в сети и начинает посылать адресуемые ведомым устройствам запросы синхронизации.

При получении такого запроса ведомое устройство дает ответ на предыдущую команду. Ответ содержит захлопнутую координату, результат выполнения операции позиционирования, состояние регистров виртуальных входов/выходов ведомого устройства. Если после широковещательного запроса устройству пришло несколько запросов синхронизации, то оно ответит только на первый.



Формат широковещательного запроса:

| Поле:          | Адрес | Чужие входы | Свои выходы | Команда | Параметр Команды         | CRC |
|----------------|-------|-------------|-------------|---------|--------------------------|-----|
| Размер (байт): | 1     | 4           | 4           | 1       | в зависимости от команды | 2   |

В качестве широковещательного используется адрес 100.

Чужие входы – ведущий передает состояние входов ЛИР986 другим устройствам ЛИР.

Свои выходы – ведущий передает состояние регистра своих выходов (первые три ведомых берут по байту согласно своему адресу и копируют в начало своих регистров выходов).

CRC – контрольная сумма, используется полином 0x8005, начальное значение 0xFFFF.

Команда – запрос на выполнение ведомыми передаваемой команды. Доступны следующие команды:

- Синхронизация (0) – показывает, что ничего кроме синхронизации координаты и входов/выходов от ведомого не требуется. Размер поля параметров команды – 0.
- Стоп (4) – остановить процесс позиционирования. Все ведомые останавливают процесс позиционирования. Размер поля параметров команды – 0.
- Поиск референтной метки (1) – все ведомые запускают процесс поиска референтной метки. Размер поля параметров команды – 0.
- Переместиться в координату со скоростью (3) – запуск процесса позиционирования для выбранных ведомых в заданную позицию с заданной скоростью. Поле параметров команды имеет следующий вид:

| Поле:          | Скорость | Ведомый 0 | Позиция 0 | ... | Ведомый n | Позиция n |
|----------------|----------|-----------|-----------|-----|-----------|-----------|
| Размер (байт): | 1        | 1         | 8         |     | 1         | 8         |

В поле “ведомый” указывается адрес ведомого, которому необходимо выполнить перемещение, а в соответствующем поле “позиция” – координата, в которую требуется переместиться. Задание для ведомых может следовать в любом порядке.

Формат запроса синхронизации:

| Поле:          | Адрес | Команда | CRC |
|----------------|-------|---------|-----|
| Размер (байт): | 1     | 1       | 2   |

В качестве адреса используется адрес ведомого + 101. Адреса ведомых могут принимать значения 0..9.

Доступна только одна команда – Синхронизация (0).

CRC – контрольная сумма

Формат ответа ведомого:

| Поле:          | Адрес | Результат позиционирования | Статус датчика | Координата датчика | Чужие выходы | Свои входы | CRC |
|----------------|-------|----------------------------|----------------|--------------------|--------------|------------|-----|
| Размер (байт): | 1     | 1                          | 2              | 8                  | 4            | 1          | 2   |

Адрес – адрес ведомого + 101.

В поле результата позиционирования отображается текущий результат работы модуля позиционирования ведомого.

В полях статус и координата датчика отображаются соответствующие значения модуля датчика ведомого устройства для первой абсолютной системы координат.

Чужие выходы содержат состояние выходных сигналов, которые необходимо перенаправить на выходы ЛИР986.

Свои входы – состояние первых восьми входов ведомого.

**id подмодуля**

Подмодуль имеет id 1.

**Настройки подмодуля:**

| Размер(байт) | Описание                                                                                             |
|--------------|------------------------------------------------------------------------------------------------------|
| 1            | Режим работы (0 – ведомый, 1 - ведущий)                                                              |
| 1            | Адрес (0...10)                                                                                       |
| 1            | Синхронизации с ЛИР-986 (0 – нет, 1 – да)                                                            |
| 1            | Адрес ЛИР-986 (1...247)                                                                              |
| 1            | Число последовательных ошибок, считаемое потерей связи (0...255)                                     |
| 1            | В режиме ведомого, при запросе координаты производить принудительный опрос датчика (0 – нет, 1 – да) |

**Команды**

Подмодуль не имеет команд.



**Подмодуль протокола ЛИР-915/6****Описание**

Данный подмодуль осуществляет совместимость устройства с протоколами обмена ЛИР-915 и ЛИР-916 в двоично-десятичном и ASCII формате.

Оба формата протокола обмена поддерживают работу устройства в сети в качестве ведомого. Для этого необходимо, чтобы все устройства имели одинаковые настройки сетевого соединения и различные адреса. Нулевой адрес использовать запрещено.

Для отделения сообщений друг от друга используются паузы в 3,5 символа (соответственно, пауза зависит от скорости передачи). Все входящие команды проверяются на корректность.

**Двоично-десятичный формат**

*Формат запроса выполнения команды:*

|     |     |
|-----|-----|
| cmd | adr |
|-----|-----|

, где:

cmd – команда,

adr – адрес интерфейса в сети.

Размер запроса составляет два байта.

*Ответ:*

|      |                   |     |                   |      |
|------|-------------------|-----|-------------------|------|
| 0x0A | data <sub>0</sub> | ... | data <sub>n</sub> | 0x0B |
|------|-------------------|-----|-------------------|------|

, где:

0x0A – синхронизирующий байт начала передачи,

0x0B – синхронизирующий байт конца передачи,

data – поле данных.

Поле данных имеет фиксированную длину для каждого ответа. Значение разделено по байтам и передается в обратном порядке (little endian).

**Пример:** значение 12345678 передается как 78 56 34 12.

Возможно два режима работы с протоколом: режим полной совместимости и расширенный.

В режиме совместимости возможна передача значений счетчика от –9999999 до 9999999. Значение позиции передается четырьмя байтами.

В этом режиме возможна настройка передачи битов статуса датчика. Выбранные биты статуса объединяются через логическое “или” и добавляются к позиции в разряд, с номером равным разрядности датчика. Для инкрементных датчиков передача бита статуса невозможна.

**Пример:** позиция датчика 436, разрядность датчика 10, тогда:

без бита ошибки: 01 1011 0100 (двоичный), 36 04 00 00 (двоично-десятичный);

с битом ошибки: 101 1011 0100 (двоичный), 60 14 00 00 (двоично-десятичный).

Отрицательные значения передаются в двоично-десятичном дополнительном коде. При этом старший двоично-десятичный разряд позиции содержит число “9”. Для преобразования в позицию из полученного значения необходимо вычесть число 100000000.

**Пример:** позиция датчика -395, тогда: передаваемое значение в двоично-десятичном формате будет иметь вид: 05 96 99 99.

Расширенный режим работы позволяет передавать полностью 64-разрядный счетчик позиции и состояние датчика. Диапазон передаваемой позиции от -9223372036854775807 до 9223372036854775807. В данном режиме статус датчика передается полностью. Он располагается перед позицией в первых трех байтах данных и может принимать значения от 0 до 65535 (расшифровку статуса датчика см. в разделе модуль датчика). Значение позиции передается в следующих десяти байтах. Для отрицательных чисел, для преобразования в позицию из полученного значения необходимо вычесть число 9223372036854775808.

**Пример:** статус датчика 256, позиция 734283634, тогда полученное значение будет иметь вид: 56 02 00 34 36 28 34 07 00 00 00 00 00

#### Список поддерживаемых команд:

| Команда                             |      | Ответ                               |                                                              |
|-------------------------------------|------|-------------------------------------|--------------------------------------------------------------|
| Описание                            | Код  | Режим совместимости                 | Расширенный режим                                            |
| Установка смещения относительной СО | 0x30 | Нет ответа                          | Нет ответа                                                   |
| Захват референтной метки            | 0x31 | Нет ответа                          | Нет ответа                                                   |
| Координата референтной метки        | 0x32 | Если референтная метка не захвачена |                                                              |
|                                     |      | 0x0A 0xDD 0xDD 0xDD 0xDD 0x0B       | 0x0A 0xDD 0xDD 0xDD 0xDD 0xDD 0xDD 0xDD 0x0DD 0xDD 0xDD 0x0B |
|                                     |      | Если референтная метка захвачена    |                                                              |
|                                     |      | 0x0A 0x00 0x00 0x00 0x00 0x0B       | 0x0A 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x0B  |
| Запрос позиции относительной СО     | 0x33 | 0x0A 0x** 0x** 0x** 0x** 0x0B       | 0x0A 0x** 0x** 0x** 0x** 0x** 0x** 0x** 0x** 0x** 0x** 0x0B  |
| Запрос позиции абсолютной СО        | 0x34 | Если референтная метка не захвачена |                                                              |
|                                     |      | 0x0A 0xDD 0xDD 0xDD 0xDD 0x0B       | 0x0A 0x** 0x** 0x** 0xDD 0xDD 0xDD 0xDD 0xDD 0xDD 0xDD 0x0B  |
|                                     |      | Если референтная метка захвачена    |                                                              |
|                                     |      | 0x0A 0x** 0x** 0x** 0x** 0x0B       | 0x0A 0x** 0x** 0x** 0x** 0x** 0x** 0x** 0x** 0x** 0x** 0x0B  |

**ASCII формат***Формат запроса выполнения команды:*

|      |     |     |
|------|-----|-----|
| 0x23 | adr | cmd |
|------|-----|-----|

, где:

0x23– символ начала команды “#”,

adr – адрес интерфейса в сети,

cmd – команда.

Размер запроса составляет три байта.

*Ответ:*

|      |                   |     |                   |      |
|------|-------------------|-----|-------------------|------|
| 0x3E | data <sub>0</sub> | ... | data <sub>n</sub> | 0x0D |
|------|-------------------|-----|-------------------|------|

, где:

0x3E – символ начала ответа “&gt;”,

0x0D – символ конца ответа / возврат каретки,

data – поле данных.

Размер поля данных зависит от передаваемой позиции. Данные передаются в ASCII символах, начиная со старшего разряда. Для отрицательного значения первым символом будет “-”.

**Пример:** значение 0123456789 передается как 123456789.

Возможно два режима работы с протоколом: режим полной совместимости и расширенный.

В режиме совместимости возможна передача значений счетчика от –4294967295 до 4294967295. Размер поля данных позиции датчика может быть от 1 до 11 байт.

В этом режиме возможна настройка передачи битов статуса датчика. Выбранные биты статуса объединяются через логическое “или” и добавляются к позиции в разряд, равный разрядности датчика. Для инкрементных датчиков передача бита ошибки невозможна.

**Пример:** позиция датчика 436, разрядность датчика 10, тогда:

без бита ошибки: 01 1011 0100 (двоичный), 436 (переданное значение);

с битом ошибки: 11 1011 0100 (двоичный), 948 (переданное значение).

Расширенный режим работы позволяет передавать полностью 64-разрядный счетчик позиции и состояние датчика. Диапазон передаваемой позиции от –9223372036854775807 до 9223372036854775807. В данном режиме статус датчика передается полностью. Он располагается перед позицией и отделен от нее символом “|”. Он может принимать значения от 0 до 65535 (расшифровку статуса датчика см. в разделе модуль датчика).

Размер поля данных статуса может быть от 1 до 5 байт, поля позиции датчика – от 1 до 20 байт.

**Пример:** статус датчика 256, позиция 734283634, тогда полученное значение будет иметь вид: 256|734283634.

**Список поддерживаемых команд:**

| Команда                            |             | Ответ                               |                                               |
|------------------------------------|-------------|-------------------------------------|-----------------------------------------------|
| Описание                           | Код         | Режим совместимости                 | Расширенный режим                             |
| Установка смещения относительно СО | 0x7A<br>“Z” | Нет ответа                          | Нет ответа                                    |
| Захват референтной метки           | 0x5A<br>“Z” | Нет ответа                          | Нет ответа                                    |
| Координата референтной метки       | 0x72<br>“r” | Если референтная метка не захвачена |                                               |
|                                    |             | 0x3E 0x0D                           | 0x3E 0x0D                                     |
|                                    |             | Если референтная метка захвачена    |                                               |
|                                    |             | 0x3E 0x00 0x0D                      | 0x3E 0x00 0x0D                                |
| Запрос позиции относительно СО     | 0x6F<br>“o” | 0x3E 0x** ... 0x** 0x0D             | 0x3E 0x** ... 0x** 0x7C<br>0x** ... 0x** 0x0D |
| Запрос позиции абсолютной СО       | 0x61<br>“a” | Если референтная метка не захвачена |                                               |
|                                    |             | 0x3E 0x0D                           | 0x3E 0x0D                                     |
|                                    |             | Если референтная метка захвачена    |                                               |
|                                    |             | 0x3E 0x** ... 0x** 0x0D             | 0x3E 0x** ... 0x** 0x7C<br>0x** ... 0x** 0x0D |

**Оба формата**

Для инкрементных датчиков с референтной меткой по команде захвата референтной метки запускается процесс захвата референтной метки. Обнуление счетчика происходит только по приходу сигнала референтной метки, до этого счет позиции сохраняется.

Для инкрементных датчиков без референтной метки по команде захвата референтной метки позиция обнуляется.

Для абсолютных датчиков команда захвата референтной метки не выполняется.

Поскольку модуль датчика не поддерживает счет от референтной до референтной метки, то после захвата референтной метки на запрос координаты референтной метки выдается всегда нулевое значение.

**id подмодуля**

Подмодуль имеет id 2.

**Настройки подмодуля:**

| Размер(байт) | Описание                                         |
|--------------|--------------------------------------------------|
| 4            | Скорость передачи, бод/с (0...3750000)           |
| 2            | Таймаут запросов, мс (0...65535)                 |
| 1            | Режим работы (0 – ASCII, 1 - BCD)                |
| 1            | Адрес (1...255)                                  |
| 1            | Статус датчика (0 – выкл, 1 - вкл)               |
| 1            | Ошибка связи с датчиком (0 – выкл, 1 - вкл)      |
| 1            | Повреждена таблица коррекции (0 – выкл, 1 - вкл) |
| 1            | Повреждение смещения (0 – выкл, 1 - вкл)         |
| 1            | Режим совместимости (0 – выкл, 1 - вкл)          |

Поля статусов включают отображение соответствующих ошибок в бите статуса передаваемого сообщения. В расширенном режиме значение этих полей ни на что не влияет.

Если время между запросами к устройству превышает заданное в таймауте, устанавливается сигнал “ошибка связи”.

Параметр “таймаут запросов” позволяет задать максимальное время опроса устройства, при превышении которого, будет считаться, что соединение потеряно и установится выходной сигнал “ошибка связи”.

**Команды**

Подмодуль не имеет команд.

## Модуль входов/выходов

### Описание

Модуль входов/выходов позволяет вручную управлять физическими выходами устройства и считывать значения входов. Количество входов/выходов зависит от устройства.

#### id интерфейса

Интерфейс имеет id 3.

#### Режим работы:

Модуль имеет только один режим работы 0, смена режима работы недоступна.

#### Настройки модуля:

Модуль не имеет настроек.

#### Подмодули

Модуль не имеет подмодулей.

#### Входные функциональные сигналы

Модуль не имеет функциональных входных сигналов

#### Выходные функциональные сигналы

Модуль не имеет функциональных выходных сигналов

## Расширенные команды

### Установка выходов (0x14)

С помощью команды задается состояние всех выходов устройства.

*Запрос:*

|   |   |      |         |         |
|---|---|------|---------|---------|
| 5 | I | 0x14 | outputs | outputs |
|---|---|------|---------|---------|

*Ответ:*

|   |   |      |          |  |
|---|---|------|----------|--|
| 4 | I | 0x14 | ack/nack |  |
|---|---|------|----------|--|

, где:

outputs – состояние выходов, представляет из себя битовую область, каждый бит которой соответствует физическому выходу устройства с тем же номером.

### Получить состояние выходов (0x15)

С помощью команды запрашивается состояние всех выходов устройства.

*Запрос:*

|   |   |      |  |  |
|---|---|------|--|--|
| 3 | I | 0x15 |  |  |
|---|---|------|--|--|

*Ответ:*

|   |   |      |         |         |
|---|---|------|---------|---------|
| 5 | I | 0x15 | outputs | outputs |
|---|---|------|---------|---------|

, где:

outputs – состояние физических выходов устройства.

**Получить состояние входов (0x16)**

С помощью команды запрашивается состояние всех входов устройства.

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x16 |
|---|---|------|

*Ответ:*

|   |   |      |        |        |
|---|---|------|--------|--------|
| 5 | I | 0x14 | inputs | inputs |
|---|---|------|--------|--------|

, где:

inputs – состояние физических входов устройства.

**Установить один выход (0x17)**

С помощью команды устанавливается состояние одного выхода устройства.

*Запрос:*

|   |   |      |        |       |
|---|---|------|--------|-------|
| 5 | I | 0x17 | output | state |
|---|---|------|--------|-------|

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x17 | ack/nack |
|---|---|------|----------|

, где:

output – номер физического выхода устройства,

state – состояние.

**Запрос количества входов/выходов (0x18)**

С помощью команды можно узнать количество входов/выходов устройства.

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x18 |
|---|---|------|

*Ответ:*

|   |   |      |    |    |
|---|---|------|----|----|
| 5 | I | 0x18 | NI | NO |
|---|---|------|----|----|

, где:

NI – количество входов,

NO – количество выходов.

## Модуль виртуальных входов/выходов

### Описание

Данный модуль расширяет возможности физических входов/выходов устройства. Для правильного функционирования должен быть включен модуль ИО и/или подмодуль синхронизации сппу модуля RS-485.

### Входы

Входы делятся на 4 группы, каждая из которых состоит из 32 входов:

- *входы*

| Индекс  | Вход                        |
|---------|-----------------------------|
| 0...n   | Физические входы устройства |
| n...15  | Зарезервировано             |
| 16...31 | modbus coils                |

, где n – количество физических входов устройства.

Данная группа содержит физические входы устройства, расположенные начиная с индекса 0. Так же в конце регистра располагаются coils, которые доступны в режиме ведомого устройства modbus. Они позволяют использовать внешние устройства, следящие за состоянием подключенных к ним линий и устанавливающих эти состояния в соответствующие coil устройства.

- *входы ЛИР-986*

| Индекс  | Вход                    |
|---------|-------------------------|
| 0...16  | Входы ЛИР-986           |
| 16...32 | Входные маркеры ЛИР-986 |

Данная группа содержит состояние входов и входных маркеров подключенного ЛИР-986. Для взаимодействия с ЛИР-986 интерфейс RS-485 должен быть сконфигурирован в режим сппу и выбран режим работы с ЛИР-986.

- *логические входы*

| Индекс | Вход                           |
|--------|--------------------------------|
| 0...n  | Выход логической ячейки 0...15 |
| n...31 | Зарезервировано                |

, где n – количество логических входов устройства.

Входные логические ячейки позволяют произвести бинарную операцию с двумя сигналами из любой входной группы. Результат записывается в регистр логических входов с индексом, соответствующим номеру логического входа.

Доступные операции:

- ✓ или
- ✓ и
- ✓ исключающее или
- ✓ равенство



Важно помнить, что выполнение логических действий ячейками выполняется строго в соответствии с их порядковым номером.

- **таймеры**

| Индекс | Вход            |
|--------|-----------------|
| 0...n  | Выходы таймеров |
| n...31 | Зарезервировано |

, где n – количество таймеров устройства.

Пользователю доступно четыре таймера. Делитель частоты работы задается для каждого таймера независимо в настройках. Итоговая частота работы получается по формуле  $F = \frac{1}{2 * T_{sys} * DIV}$ , где  $T_{sys}$  – время системного цикла управления (в секундах),  $DIV$  – делитель частоты для каждого таймера. Максимальная возможная частота – 500 Гц.

Таймеры имеют два режима работы:

- ✓ импульсный – через заданный интервал времени таймер выдает импульс длительностью от 1 мс,
- ✓ меандр – через заданный интервал времени выходной логический уровень таймера инвертируется.

Для всех входных групп доступны настройки, позволяющие установить любой вход в логическую “1” и задать инверсию. Зарезервированные части входных регистров могут быть использованы как источник постоянных логических уровней “1” или “0”.

Также для всех групп работает механизм выделения фронтов, поэтому при выборе входного сигнала пользователь может выбрать какое событие считать за “1”:

- ✓ логическая “1”,
- ✓ логический “0”,
- ✓ нарастающий фронт,
- ✓ убывающий фронт,
- ✓ любой фронт,
- ✓ предыдущее значение логического уровня

### Выходы

Выходы делятся на 3 группы:

- **свои выходы**

| Индекс  | Выход                        |
|---------|------------------------------|
| 0...n   | Физические выходы устройства |
| n...15  | Зарезервировано              |
| 16...31 | modbus discrete inputs       |

, где n – количество физических выходов устройства.

Начиная с нулевого индекса идут физические выходы устройства. Конец группы занят виртуальными выходами, доступными через Modbus в режиме ведомого.

- **выходы ЛИР-986**

| Индекс  | Выход                    |
|---------|--------------------------|
| 0...16  | Выходы ЛИР-986           |
| 16...32 | Выходные маркеры ЛИР-986 |

При подключении ЛИР-986 состояние данного регистра соответствует состоянию выходов и выходных маркеров ЛИР-986. Устройство должно быть сконфигурировано в режим ведущего Modbus.

- **логические выходы**

| Индекс  | Вход выходной логической ячейки |
|---------|---------------------------------|
| 0...n   | Первый вход лог. ячейки 0...7   |
| n...2n  | Второй вход лог. ячейки 0...7   |
| 2n...31 | Зарезервировано                 |

, где n – количество логических выходов устройства.

Выходная логическая ячейка производит выбранную бинарную операцию с двумя поданными на нее выходными сигналами и записывает результат в указанный выходной регистр с заданным индексом.

Доступные операции:

- ✓ или,
- ✓ и,
- ✓ исключающее или,
- ✓ равенство.

Важно помнить, что порядок выполнения операций выходных логических ячеек строго соответствует их порядковому номеру.

Для пользователя доступны настройки, позволяющие задать инверсию для всех выходных групп. При назначении нескольких функций на один выход – результатом будет бинарное “или” их сигналов.

**id модуля**

Модуль имеет id 4.

**Режим работы:**

Модуль имеет только один режим работы 0, смена режима работы недоступна.

**Настройки модуля:**

| Кол-во                              | Размер(байт) | Описание                                                                                |
|-------------------------------------|--------------|-----------------------------------------------------------------------------------------|
| <i>Настройки входов</i>             |              |                                                                                         |
| 4                                   | 4            | Инвертировать вход                                                                      |
|                                     | 4            | Установить вход в "1"                                                                   |
| <i>Настройки логических входов</i>  |              |                                                                                         |
| NLI                                 | 1            | Состояние (0 – выкл, 1 – вкл)                                                           |
|                                     | 1            | Источник (0 – входы, 1 – ЛИР-986, 2 – логические, 3 – таймеры)                          |
|                                     | 1            | Триггер (0 – "1", 1 – "0", 2 – фронты, 3 – нарастающий, 4 – спадающий, 5 – прошлая "1") |
|                                     | 1            | Номер входа (0...31)                                                                    |
|                                     | 1            | Логическая операция (0 – "и", 1 – "или", 2 – "искл или", 3 – "равно")                   |
|                                     | 1            | Источник (0 – входы, 1 – ЛИР-986, 2 – логические, 3 – таймеры)                          |
|                                     | 1            | Триггер (0 – "1", 1 – "0", 2 – фронты, 3 – нарастающий, 4 – спадающий, 5 – прошлая "1") |
|                                     | 1            | Номер входа (0...31)                                                                    |
| <i>Настройки таймеров</i>           |              |                                                                                         |
| NT                                  | 4            | Делитель (0...4294967295)                                                               |
|                                     | 1            | Режим работы (0 – меандр, 1 – импульс)                                                  |
|                                     | 3            | Зарезервировано                                                                         |
| <i>Настройки выходов</i>            |              |                                                                                         |
| 3                                   | 4            | Инвертировать выход                                                                     |
| <i>Настройки логических выходов</i> |              |                                                                                         |
| NLO                                 | 1            | Состояние (0 – выкл, 1 – вкл)                                                           |
|                                     | 1            | Логическая операция (0 – "и", 1 – "или", 2 – "искл или", 3 – "равно")                   |
|                                     | 1            | Получатель (0 – выходы, 1 – ЛИР-986, 2 – логические)                                    |
|                                     | 1            | Номер выхода (0...31)                                                                   |

, где NLI – количество логических входов,

NLO – количество логических выходов,

NT – количество таймеров.

Количество NLI, NLO и NT задается в описании устройства, а так же может быть получено с помощью соответствующей команды.

Настройки входов располагаются в следующем порядке (начиная с младшего):

0 – входы,

1 – ЛИР-986,

2 – логические,

3 – таймеры.

Настройки выходов располагаются в следующем порядке:

0 – выходы,

1 – ЛИР-986,

2 – логические.

Для входов и выходов каждая настройка представляет из себя битовое поле, в котором порту соответствует бит с тем же порядковым номером.

Настройки таймеров, логических входов и выходов задаются структурами данных и располагаются согласно их порядковому номеру.

**Подмодули**

Модуль не имеет подмодулей.

**Входные функциональные сигналы**

Модуль не имеет входных функциональных сигналов

**Выходные функциональные сигналы**

Модуль не имеет выходных функциональных сигналов

## Расширенные команды

**Запрос параметров модуля (0x14)**

Возвращает число логических входов/выходов и таймеров.

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x14 |
|---|---|------|

*Ответ:*

|   |   |      |    |    |    |
|---|---|------|----|----|----|
| 6 | I | 0x14 | LI | TI | LO |
|---|---|------|----|----|----|

, где:

LI – количество логических выходов,

TI – количество таймеров,

LO – количество логических входов.

**Запрос состояния всех входов (0x15)**

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x15 |
|---|---|------|

*Ответ:*

|    |    |      |    |    |    |    |    |    |    |    |    |
|----|----|------|----|----|----|----|----|----|----|----|----|
| 19 | I  | 0x15 | SI | SI | SI | SI | EI | EI | EI | EI | LI |
| LI | LI | LI   | TI | TI | TI | TI |    |    |    |    |    |

, где:

SI – состояние своих выходов и modbus coils,

EI – состояние чужих выходов,

LI – состояние логических входов,

TI – состояние таймеров.

**Запрос состояния всех выходов (0x16)**

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x16 |
|---|---|------|

*Ответ:*

|    |    |      |    |    |    |    |    |    |    |    |    |
|----|----|------|----|----|----|----|----|----|----|----|----|
| 15 | I  | 0x16 | SO | SO | SO | SO | EO | EO | EO | EO | LO |
| LO | LO | LO   |    |    |    |    |    |    |    |    |    |

, где:

SO – свои выхода и Modbus discrete inputs,

EO – внешние выхода,

LO – входы логических выходов.

## Модуль функциональных сигналов

### Описание

Данный модуль предназначен для настройки и организации механизма сигналов между виртуальными входами/выходами устройства и функциональными сигналами модулей. Для функционирования данного модуля необходимо, чтобы модуль виртуальных входов выходов был включен.

Входные и выходные функциональные сигналы для каждого модуля перечислены в их описании. Данный модуль предоставляет агрегированный список всех функциональных сигналов модулей устройства и возможность их настройки.

Элементы данного модуля делятся на три группы:

- входные сигналы,
- выходные сигналы,
- сигналы байпаса.

Для каждого входного сигнала задается состояние (вкл/выкл), выбирается входная группа, индекс входа и триггер, состояние которого будет передано в функцию устройства.

Для каждого выходного сигнала задается состояние (вкл/выкл), выбирается выходная группа и индекс выхода, куда будет передан результат внутренней функции.

Сигналы байпаса позволяют скопировать состояние виртуального входа на виртуальный выход. Для каждого сигнала байпаса задается состояние (вкл/выкл), выбирается входная группа, индекс входа и триггер, откуда будет взято состояние, а также, выходная группа и индекс выхода, куда это состояние будет передано.

Для каждого устройства доступен свой набор функций. Список задействованных функций и их количество можно получить с помощью соответствующей команды.

**id модуля**

Модуль имеет id 5.

**Режим работы:**

Модуль имеет только один режим работы 0, смена режима работы недоступна.

**Настройки модуля:**

| Кол-во                      | Размер(байт) | Описание                                                                                                      |
|-----------------------------|--------------|---------------------------------------------------------------------------------------------------------------|
| Настройки входных сигналов  |              |                                                                                                               |
| ISN                         | 1            | Состояние (0 – выкл, 1 – вкл)                                                                                 |
|                             | 1            | Входная группа<br>(0 – входы, 1 – входы ЛИР-986, 2 – логические, 3 - таймеры)                                 |
|                             | 1            | Триггер (0 – “1”, 1 – “0”, 2 – оба фронта, 3 – нарастающий, 4 – убывающий, 5 – предыдущий логический уровень) |
|                             | 1            | Индекс входа                                                                                                  |
| Настройки выходных сигналов |              |                                                                                                               |
| OSN                         | 1            | Состояние (0 – выкл, 1 – вкл)                                                                                 |
|                             | 1            | Выходная группа (0 – выходы, 1 – выходы ЛИР-986, 2 - логические)                                              |
|                             | 1            | Индекс выхода                                                                                                 |
| Настройки сигналов байпаса  |              |                                                                                                               |
| ICN                         | 1            | Состояние (0 – выкл, 1 – вкл)                                                                                 |
|                             | 1            | Входная группа<br>(0 – входы, 1 – входы ЛИР-986, 2 – логические, 3 - таймеры)                                 |
|                             | 1            | Триггер (0 – “1”, 1 – “0”, 2 – оба фронта, 3 – нарастающий, 4 – убывающий, 5 – предыдущий логический уровень) |
|                             | 1            | Индекс входа                                                                                                  |
|                             | 1            | Выходная группа (0 – выходы, 1 – выходы ЛИР-986, 2 - логические)                                              |
|                             | 1            | Индекс выхода                                                                                                 |

, где:

ISN – количество функций входного сигнала,

OSN – количество функций выходного сигнала,

ICN – количество функций байпаса.

Настройки сигналов располагаются в последовательности согласно их порядковому номеру.

**Подмодули**

Модуль не имеет подмодулей.

**Входные функциональные сигналы**

Модуль не имеет входных функциональных сигналов

**Выходные функциональные сигналы**

Модуль не имеет выходных функциональных сигналов

**Расширенные команды****Команда получения количества входных функций (0x14)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x14 |
|---|---|------|

*Ответ:*

|   |   |      |   |
|---|---|------|---|
| 4 | I | 0x14 | n |
|---|---|------|---|

, где:

n – количество входных функций.

**Команда получения количества выходных функций (0x15)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x15 |
|---|---|------|

*Ответ:*

|   |   |      |   |
|---|---|------|---|
| 4 | I | 0x15 | n |
|---|---|------|---|

, где:

n – количество выходных функций.

**Команда получения количества функций байпаса (0x16)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x16 |
|---|---|------|

*Ответ:*

|   |   |      |   |
|---|---|------|---|
| 4 | I | 0x16 | n |
|---|---|------|---|

, где:

n – количество функций байпаса.

**Команда получения списка входных функций (0x17)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x17 |
|---|---|------|

*Ответ:*

|       |   |      |                    |                    |     |                    |                    |
|-------|---|------|--------------------|--------------------|-----|--------------------|--------------------|
| 3+n*2 | I | 0x17 | m_idx <sub>1</sub> | m_sig <sub>1</sub> | ... | m_idx <sub>n</sub> | m_sig <sub>n</sub> |
|-------|---|------|--------------------|--------------------|-----|--------------------|--------------------|

, где:

m\_idx – порядковый номер модуля,

m\_sig – идентификатор сигнала,

n – количество входных функций.



**Команда получения списка выходных функций (0x18)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x18 |
|---|---|------|

*Ответ:*

|         |   |      |            |            |     |            |            |
|---------|---|------|------------|------------|-----|------------|------------|
| $3+n*2$ | I | 0x17 | $m\_idx_1$ | $m\_sig_1$ | ... | $m\_idx_n$ | $m\_sig_n$ |
|---------|---|------|------------|------------|-----|------------|------------|

, где:

 $m\_idx$  – порядковый номер модуля, $m\_sig$  – идентификатор сигнала, $n$  – количество выходных функций.

## Модуль позиционирования

### Описание

Данный модуль позволяет осуществлять контроль движения в заданную точку согласно команде. Проверяет по окончании движения нахождение в допуске позиционирования.

Во время перемещения при превышении заданного порога последовательных ошибок чтения координаты с датчика происходит аварийная остановка.

Также модуль позволяет задать последовательность действий для автоматического поиска референтной метки. Последовательность состоит из двух действий настраиваемых пользователем. Для каждого действия можно задать:

- действие:
  - 0 – двигаться вправо,
  - 1 – двигаться влево,
  - 2 – обнулить координату,
  - 3 – ничего не делать.
- захват референтной метки:
  - 0 – захватывать,
  - 1 – захватывать по сигналу зоны референтной метки,
  - 2 – не захватывать.
- максимальная скорость (0...65535)

Модуль позволяет использовать сигнал выбора оси. При этом, перед началом движения он выдает выходной сигнал выбора оси, и ожидает появление входного сигнала. После появления входного сигнала выбора оси начинает выполняться алгоритм движения. Если в процессе движения входной сигнал выбора оси пропадает – движение прекращается. Выходной сигнал выбора оси остается в установленном состоянии до конца перемещения.

При начале движения, модуль анализирует расстояние между текущей координатой и координатой назначения. Если разница между ними меньше допуска позиционирования, то перемещение не осуществляется, команда считается выполненной.

Если включена компенсация люфта, то при первом движении после включения, а также после смены направления движения будет запущен алгоритм компенсации люфта. При этом двигатель запустится на скорости компенсации люфта (согласно настройкам) и устройство будет ожидать изменение координаты на значение, большее, чем задано в настройках (изменение координаты, считаемое перемещением) за установленный промежуток времени. Как только значение будет превышено, люфт считается скомпенсированным и запустится алгоритм позиционирования с заданной скоростью.

Если компенсация люфта не включена, то сразу запускается алгоритм позиционирования.

После окончания движения модуль ожидает полной остановки – изменение координаты на значение, меньшее, чем задано в настройках (изменение координаты, считаемое перемещением) за установленный промежуток времени. После этого будет рассчитана разница текущей координаты с координатой назначения. Если эта разница превышает заданный в настройках допуск позиционирования, то модуль выставляет соответствующий результат.

Модуль контролирует параметры движения – направление и скорость. Если во время движения координата не изменяется или движение осуществляется в противоположном направлении, происходит остановка перемещения и выставляется соответствующий результат.

Модуль поддерживает два входных сигнала с ограничителей перемещения. При поступлении такого сигнала происходит остановка движения и остановка соответствующего результата.

**id модуля**

Модуль имеет id 6.

**Режим работы:**

Режим работы модуля переключает действующий подмодуль. Подмодули дают возможность модулю управлять различными двигателями.

**Настройки модуля:**

| Размер(байт) | Описание                                                                                       |
|--------------|------------------------------------------------------------------------------------------------|
| 4            | Допуск позиционирования, мкм (0...4294967295)                                                  |
| 2            | Скорость выборки люфта (0...65535)                                                             |
| 2            | Ограничение максимальной скорости перемещения (0...65535)                                      |
| 2            | Скорость перемещения при ручном управлении (0...65535)                                         |
| 2            | Изменение координаты, считаемое перемещением (0...65535)                                       |
| 2            | Интервал времени, за который берется изменение кода, для выявления перемещения, мс (0...65535) |
| 2            | Аварийный порог последовательности ошибок датчика, шт (0...65535)                              |
| 2            | Таймаут операции выборки люфта, м (0...30000)                                                  |
| 1            | Прм к1: действие                                                                               |
| 1            | Прм к1: захват метки                                                                           |
| 1            | Прм к1: макс скорость                                                                          |
| 1            | Зарезервировано                                                                                |
| 1            | Прм к2: действие                                                                               |
| 1            | Прм к2: захват метки                                                                           |
| 1            | Прм к2: макс скорость                                                                          |
| 1            | Зарезервировано                                                                                |
| 1            | Выбирать люфт (0 – нет, 1 - да)                                                                |
| 1            | Использовать сигнал выбора оси (0 – нет, 1 - да)                                               |
| 2            | Зарезервировано                                                                                |

, где:

Прм к – команда поиска референтной метки.

Заданная максимальная скорость в настройках имеет максимальный приоритет и не может быть превышена.

**Подмодули**

0 – подмодуль управления двигателем с помощью дискретных выходов.

**Входные функциональные сигналы**

- 0 – зона референтной метки;
- 1 – правый концевой выключатель;
- 2 – левый концевой выключатель;
- 3 – выбор оси;
- 4 – поиск референтной метки;
- 5 – ручное перемещение влево;
- 6 – ручное перемещение вправо.

**Выходные функциональные сигналы**

- 0 – мотор включен;
- 1 – торможение мотором;
- 2 – направление вращения мотора (0 – влево, 1 - вправо);
- 3 – мотор вращается вправо;

- 4 – мотор вращается влево;
- 5 – мотор вращается на первой скорости;
- 6 – мотор вращается на второй скорости;
- 7 – мотор вращается на третьей скорости;
- 8 – ошибка позиционирования;
- 9 – выбор оси.

Сигналы выбора оси используются в системах с переключаемой муфтой.

### Расширенные команды

#### Запуск процесса поиска референтной метки (0x14)

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x14 |
|---|---|------|

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x14 | ack/nack |
|---|---|------|----------|

#### Запуск процесса перемещения в координату (0x15)

Посылает координату, в которую требуется совершить перемещение. Позиционирование в новую координату возможно, если на данный момент не выполняется другое задание.

*Запрос:*

|    |   |      |       |       |       |       |       |       |       |       |
|----|---|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 11 | I | 0x15 | coord | coord | coord | coord | coord | coord | coord | coord |
|----|---|------|-------|-------|-------|-------|-------|-------|-------|-------|

, где:

coord – координата, в которую требуется совершить перемещение, знаковая

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x15 | ack/nack |
|---|---|------|----------|

#### Запуск процесса перемещения в координату с ограничением по скорости (0x16)

Аналогично предыдущей команде, только дополнительно задается скорость перемещения.

*Запрос:*

|       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|
| 13    | I     | 0x16  | coord | coord | coord | coord |
| coord | coord | coord | coord | speed | speed |       |

, где:

coord – координата, в которую требуется выполнить перемещение, знаковая,  
speed – скорость перемещения.

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x16 | ack/nack |
|---|---|------|----------|

**Команда остановки (0x17)**

Останавливает текущий процесс позиционирования/поиска референтной метки.

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x17 |
|---|---|------|

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x17 | ack/nack |
|---|---|------|----------|

**Команда запроса текущего состояния (0x18)**

Возвращает текущую операцию, выполняемую интерфейсом.

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x18 |
|---|---|------|

*Ответ:*

|   |   |      |    |
|---|---|------|----|
| 4 | I | 0x18 | op |
|---|---|------|----|

, где:

op - код текущей операции.

Возможные коды операций:

- 0 – остановлен,
- 1 – ожидание сигнала выбора оси,
- 2 – перемещение,
- 3 – останавливаемся,
- 4 – тормозим.

**Команда запроса результата выполнения последнего задания (0x19)**

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x19 |
|---|---|------|

*Ответ:*

|   |   |      |        |
|---|---|------|--------|
| 4 | I | 0x19 | result |
|---|---|------|--------|

, где:

result – результат выполнения последнего задания.

Возможные коды результатов:

- 0 – выполняется перемещение,
- 1 – выполняется поиск референтной метки,
- 2 – выполняется ручное перемещение,
- 3 – задание выполнено,
- 4 – ошибка позиционирования,
- 5 – ошибка датчика,
- 6 – наезд на концевик,
- 7 – референтная метка не захвачена,
- 8 – прервано командой остановки,
- 9 – аварийно остановлено из-за движения в противоположном направлении,
- 10 – аварийно остановлено из-за отсутствия движения,
- 11 – остановлено из-за внутренней ошибки.

**Команда запроса текущей скорости (0x1A)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x1A |
|---|---|------|

*Ответ:*

|   |   |      |       |       |
|---|---|------|-------|-------|
| 5 | I | 0x1A | speed | speed |
|---|---|------|-------|-------|

, где:

speed – текущая скорость.

**Команда запроса текущего направления перемещения (0x1B)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x1B |
|---|---|------|

*Ответ:*

|   |   |      |     |
|---|---|------|-----|
| 4 | I | 0x1B | dir |
|---|---|------|-----|

, где:

dir – направление перемещения (0 – влево, 1 - вправо).

Команда запроса порядкового номера модуля датчика, который используется данным модулем (0x1C)

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x1C |
|---|---|------|

*Ответ:*

|   |   |      |    |
|---|---|------|----|
| 4 | I | 0x1C | id |
|---|---|------|----|

, где:

id – порядковый номер модуля датчика.

## Подмодули

### Подмодуль управления двигателем с помощью дискретных выходов

#### Описание

Данный подмодуль позволяет управлять двигателем с помощью дискретных выходов устройства. Он позволяет использовать до трех скоростей и задать зону торможения для каждой, а также, торможение мотором и зону тормоза. Размер всех зон задаются относительно точки назначения.

#### id подмодуля

Подмодуль имеет id 0.

#### Настройки подмодуля:

| Размер(байт) | Описание                   |
|--------------|----------------------------|
| 4            | Зона торможения двигателем |
| 4            | Зона торможения 1          |
| 4            | Зона торможения 2          |
| 4            | Зона торможения 3          |

#### Команды

Подмодуль не имеет команд.

## Модуль обработки g кодов

### Описание

Данный модуль позволяет выполнять последовательность действий устройством по заранее сформированной пользователем программе g-кодов. Программа хранится в энергонезависимой памяти устройства. Также модуль имеет возможность задавать отдельные параметры выполнения программы с помощью переменных, что не требует перезаписывать программу. Переменные хранятся в энергозависимой памяти устройства. Количество переменных доступных пользователю описывается в описании устройства.

Модуль поддерживает работу с подпрограммами и допускает 4 уровня вложенности.

Модуль позволяет осуществлять управление 6 осями. В настройках для каждой оси задается тип:

- 0 – не использовать,
- 1 – ось принадлежит этому устройству,
- 2 – ось принадлежит синхронизованному устройству.

Если выбран тип – ось принадлежит этому устройству, то для перемещения используется модуль позиционирования этого устройства. Адрес задает порядковый номер модуля среди модулей позиционирования.

Если выбран тип – ось принадлежит синхронизованному устройству, то для перемещения используется модуль позиционирования другого устройства, связь с которым происходит с помощью подмодуля сппу (должен быть в режиме ведущего) модуля RS485. На ведомом устройстве также должен быть настроен и включен подмодуль сппу (в режиме ведомого). При потере связи как со стороны ведущего, так и ведомого, происходит аварийное завершение перемещения.

Если выбран тип – не используется, данная ось не используется.

Также модуль позволяет управлять входами/выходами. Для этого он использует модуль виртуальных входов/выходов.

Список id состояния модуля:

- 0 – программа выполнена,
- 1 – программа проверяется,
- 2 – начало поиска референтной метки,
- 3 – ожидание окончания поиска референтной метки,
- 4 – чтение кадра,
- 5 – обработка кадра,
- 6 – ожидание окончания позиционирования,
- 7 – ожидание окончания паузы программы,
- 8 – ожидание завершения синхронизации входов/выходов,
- 9 – ожидание установки требуемого уровня на входе,
- 10 – кадр выполнен,
- 11 – пауза программы,
- 12 – программа завершилась с ошибкой,
- 13 – программа прервана,
- 14 – процесс автоматического запуска программы,
- 15 – процесс синхронизации текущей координаты всех осей,
- 16 – установка смещения,
- 17 – ожидание пока смещение синхронизируется,
- 18 – установка текущей системы отсчета,
- 19 – ожидание пока система отсчета синхронизируется.



Список id результата выполнения:

- 0 – выполнено,
- 1 – работает,
- 2 – аварийное завершение из-за ошибки позиционирования,
- 3 – аварийное завершение из-за ошибки связи между устройствами,
- 4 – аварийное завершение из-за повреждения кода пользовательской программы,
- 5 – аварийное завершение из-за неверных настроек,
- 6 – получена команда остановки.

Модуль позволяет задать настройки поиска референтной метки перед началом программы:

- 0 – искать, если не захвачена,
- 1 – всегда искать,
- 2 – никогда не искать.

При запуске поиска референтной метки, соответствующий сигнал передается всем управляемым модулям позиционирования. Поиск референтной метки осуществляется в соответствии с их настройками.

Также модуль позволяет установить автоматический запуск программы. При этом, после включения устройства, модуль ожидает время отсрочки запуска программы, заданное в настройках. И запускает на выполнение программу 0, с заданным по умолчанию количеством циклов.

Модуль позволяет с помощью настроек задать состояние выходов, которое устанавливается при аварийном завершении программы. При успешном завершении программы, все выхода устанавливаются в начальное состояние. Необходимо обратить внимание, что к состоянию выходов установленному данным модулем применяются заданные пользователем правила инверсии в модуле виртуальных входов выходов. Возвращение из аварийного к начальному состоянию выходов происходит по команде “СТОП”. Начальное состояние выходов задается пользователем с помощью правил инверсии и далее учитывается в процессе написания программы.

**Работа с пользовательскими программами**

Область памяти, выделенной под программы пользователя, разделена на страницы. Размер этой области памяти, размер страницы, а также максимально допустимое количество программ можно узнать с помощью соответствующих команд. Начало каждой программы должно быть выровнено по началу страницы.

Первая страница области памяти содержит структуру, содержащую информацию о сохраненных программах:

| К-во | Размер(байт) | Описание                                              |
|------|--------------|-------------------------------------------------------|
| N    | 2            | Адрес страницы, с которой начинается текст программы  |
|      | 4            | Размер программы в байтах (0 – программа отсутствует) |

, где: N – максимально допустимое количество программ.

Выполнение всегда начинается с программы 0. Подпрограммы вызываются согласно очередности расположения информации о них в структуре информации, т.е. при вызове подпрограммы 3 – будет вызвана программа, данные которой хранятся во второй ячейке структуры.

Для контроля целостности программы после передачи, а также хранения, используется CRC-16, полином:  $x^{16}+x^{15}+x^2+1$ , начальное значение 0xFFFF. Контрольная сумма располагается сразу после текста программы и не учитывается при указании ее размера в структуре информации о программе.

Для записи программы в энергонезависимую память необходимо выполнить следующие действия:

1. Создать буфер соответствующего размера области программ устройства.
2. Поместить в буфер все программы (каждая программа должна начинаться с ячейки, порядковый номер которой кратен размеру страницы).
3. Рассчитать контрольную сумму для каждой программы, записав их в буфер, сразу после данных этой программы.
4. Разделить буфер на части, соответствующие размеру страницы устройства (размер страницы можно узнать с помощью соответствующей команды).
5. Передать одну страницу в внутренний буфер устройства по частям в 128 байт, т.к. используемый протокол вносит ограничения на размер передаваемой информации за раз. При передаче в соответствующем поле указывается какая часть страницы передается.
6. Когда все части страницы переданы, необходимо вызвать команду, запускающую процесс сохранения содержимого буфера в указанную страницу области программ. Номер страницы указывается в соответствующем поле команды.
7. Выполнять пока не будут переданы все страницы.
8. В структуру информации о программах записать номер начальной страницы, с которой начинается программа и размер программы в байтах.
9. Повторять пока не будут записаны все программы.
10. Записать структуру данных информации о программах в энергонезависимую память устройства с помощью соответствующей команды.
11. Выполнить команду проверки целостности программ. Устройство в соответствии с информацией о программах выполнит расчет CRC для каждой программы и сравнит с сохраненной.

**Синтаксис пользовательской программы**

Текст программы должен состоять из символов ASCII.

Программы и подпрограммы должны начинаться и заканчиваться символом “%”. Символ должен располагаться на отдельной строке:

```
%  
G01 X10  
G01 X00  
M30  
%
```

Программа состоит из кадров. Один кадр – отдельная строка. Кадр может содержать только одну команду G или M. Если в кадре будут записаны одновременно G и M команды, то будет выполнена только G. Если в одной строке записаны две G или M команды, будет выполнена последняя по тексту.

Допускается использование пробелов. Номера команд могут быть заданы любым удобным способом:

```
G 1 X 10  
G 01 Y 30  
G001 Z9
```

Изменение регистра символов не допускается. Используются только заглавные буквы.

Программа может содержать комментарии, заключенные в круглых скобках:

G01 X10 (перемещение в точку 1)

Координаты задаются в мм:

1 – 1мм,  
0.1 – 100 мкм,  
.1 – 100 мкм,  
.001 – 1 мкм.

С помощью задания коэффициента пересчета, показания датчика должны быть приведены к мкм.

Чтобы в программе задать значение с помощью переменной, используется символ “#” и порядковый номер переменной, которые располагаются на месте числового значения:

```
G1 X#3  
G4 P #05
```

Порядковый номер переменной должен быть целым числом.

**Поддерживаемые G-команды:**

- G00 – холостой ход в заданную координату. Используется максимально разрешенная скорость в настройках позиционера. Пример: G00 X0
- G01 – позиционирование в заданную координату. Скорость перемещения задается через аргумент Fxx, где xx – номер скорости. Номер скорости может быть от 0 до 65535. До момента задания скорости с помощью аргумента F используется максимальная скорость. Пример: G01 X0
- G04 – пауза программы. Задает временной интервал в секундах, на который выполнение программы будет приостановлено. Временной интервал задается с помощью аргумента P. Пример: G04 P2.
- G90 – переход к абсолютной системе координат. Пример: G90.
- G91 – переход к относительной системе координат с обнулением в данной точке. Пример: G91.

**Поддерживаемые M-команды:**

- M02 – окончание программы. Команда завершает выполнение программы. Пример: M02.
- M17 – завершение выполнения подпрограммы. При выполнении данной команды интерпретатор либо запускает следующую итерацию цикла подпрограммы, либо возвращается к выполнению следующей команды программы, из которой была вызвана подпрограмма. Пример: M17.
- M30 – окончание программы. Команда завершает выполнение программы. Пример: M30.
- M98 – запускает выполнение подпрограммы. Через аргумент P задается номер программы, через аргумент L – число циклов выполнения. Если аргумент L не задан, подпрограмма выполняется один раз. Пример: M98 P1 L5.
- M99 – завершение выполнения подпрограммы. При выполнении данной команды интерпретатор либо запускает следующую итерацию цикла подпрограммы, либо возвращается к выполнению следующей команды программы, из которой была вызвана подпрограмма. Пример: M99.
- M110 – установить выход в 0. Устанавливает заданный через аргумент P выход в 0. Тип выхода задается в сотнях аргумента, адрес выхода – в десятках и единицах. Пример: M110 P1 – установить выход с адресом 1 своих выходов.
- M111 – установить выход в 1. Устанавливает заданный через аргумент P выход в 1. Тип выхода задается в сотнях аргумента, адрес выхода – в десятках и единицах. Пример: M111 P102 – установить чужой выход с адресом 2 в единицу.
- M120 – ожидать установку входа в 0. Выполнение программы останавливается до момента установки заданного через аргумент P входа в 0. Тип входа задается в сотнях аргумента, адрес входа – в десятках и единицах. Пример: M120 P1 – ожидать установки в 0 своего входа с адресом 1.
- M121 – ожидать установку входа в 1. Выполнение программы останавливается до момента установки заданного через аргумент P входа в 1. Тип входа задается в сотнях аргумента, адрес входа – в десятках и единицах. Пример: M121 P101 – ожидать установки в 1 чужого входа с адресом 1.

Максимально допустимая вложенность программ зависит от устройства и описана в его руководстве по эксплуатации.

**id модуля**

Модуль имеет id 7.

**Режим работы:**

Модуль имеет два режима работы:

0 – автоматический переход к следующему кадру,

1 – ручной переход к следующему кадру.

Ручное управление сменой кадра может быть использовано при отладке, а также когда необходим дополнительный контроль каждой операции.

**Настройки модуля:**

| Размер(байт) | Описание                                                                       |
|--------------|--------------------------------------------------------------------------------|
| 2            | Количество циклов по умолчанию, шт (0...65535, 0 - бесконечно)                 |
| 2            | Время задержки запуска выполнения программы при автостарте, мс (0...65535)     |
| 4            | Состояние своих выходов при аварии                                             |
| 4            | Состояние чужих выходов при аварии                                             |
| 4            | Состояние логических выходов при аварии                                        |
| 1            | Ось 1: тип оси                                                                 |
| 1            | Ось 1: адрес оси                                                               |
| 1            | Ось 2: тип оси                                                                 |
| 1            | Ось 2: адрес оси                                                               |
| 1            | Ось 3: тип оси                                                                 |
| 1            | Ось 3: адрес оси                                                               |
| 1            | Ось 4: тип оси                                                                 |
| 1            | Ось 4: адрес оси                                                               |
| 1            | Ось 5: тип оси                                                                 |
| 1            | Ось 5: адрес оси                                                               |
| 1            | Ось 6: тип оси                                                                 |
| 1            | Ось 6: адрес оси                                                               |
| 1            | Параметры поиска референтной метки                                             |
| 1            | Автозапуск выполнения программы после включения устройства (0 – выкл, 1 - вкл) |
| 2            | Зарезервировано                                                                |

**Подмодули**

Интерфейс не имеет подмодулей.

**Входные функциональные сигналы**

0 – остановить выполнение программы;

1 – запустить выполнение программы;

2 – приостановить выполнение программы.

**Выходные функциональные сигналы**

0 – ошибка выполнения программы.

**Расширенные команды****Команда запроса текущего состояния (0x14)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x14 |
|---|---|------|

*Ответ:*

|   |   |      |       |
|---|---|------|-------|
| 5 | I | 0x14 | state |
|---|---|------|-------|

, где:

state – текущее состояние.

**Команда запроса результата выполнения программы (0x15)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x15 |
|---|---|------|

*Ответ:*

|   |   |      |        |
|---|---|------|--------|
| 5 | I | 0x15 | result |
|---|---|------|--------|

, где:

result – результат выполнения программы.

**Команда запроса номера цикла выполнения программы (0x16)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x16 |
|---|---|------|

*Ответ:*

|   |   |      |   |
|---|---|------|---|
| 4 | I | 0x16 | n |
|---|---|------|---|

, где:

n – номер цикла.

**Команда запроса номера выполняемого фрейма текущей программы (0x17)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x17 |
|---|---|------|

*Ответ:*

|   |   |      |   |
|---|---|------|---|
| 4 | I | 0x17 | n |
|---|---|------|---|

, где:

n – номер выполняемого фрейма текущей программы.

**Команда запроса текста выполняемого фрейма (0x18)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x18 |
|---|---|------|

*Ответ:*

|       |   |      |                   |     |                   |
|-------|---|------|-------------------|-----|-------------------|
| 3 + n | I | 0x18 | data <sub>0</sub> | ... | data <sub>n</sub> |
|-------|---|------|-------------------|-----|-------------------|

, где:

data – строка в ASCII,

n – длина строки + 1.

**Команда запроса номера текущей программы (0x19)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x19 |
|---|---|------|

*Ответ:*

|   |   |      |   |
|---|---|------|---|
| 4 | I | 0x19 | n |
|---|---|------|---|

, где:

n – номер выполняемой программы.

**Команда запроса размера области программ (0x1A)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x1A |
|---|---|------|

*Ответ:*

|   |   |      |      |      |      |      |
|---|---|------|------|------|------|------|
| 7 | I | 0x1A | size | size | size | size |
|---|---|------|------|------|------|------|

, где:

size – размер области программ в байтах.

**Команда запроса размера страницы области программ (0x1B)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x1B |
|---|---|------|

*Ответ:*

|   |   |      |      |      |      |      |
|---|---|------|------|------|------|------|
| 7 | I | 0x1B | size | size | size | size |
|---|---|------|------|------|------|------|

, где:

size – размер страницы в байтах.

**Команда запроса максимально допустимого количества программ (0x1C)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x1C |
|---|---|------|

*Ответ:*

|   |   |      |   |
|---|---|------|---|
| 4 | I | 0x1C | n |
|---|---|------|---|

, где:

n – максимально допустимое количество программ.

**Команда запуска программы (0x1D)**

Запускает выполнение нулевой программы с заданным в настройках количеством циклов.

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x1D |
|---|---|------|

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x1D | ack/nack |
|---|---|------|----------|

**Команда запуска программы заданное количество раз (0x1E)**

Запускает выполнение нулевой программы с заданным командой количеством циклов.

*Запрос:*

|   |   |      |   |   |
|---|---|------|---|---|
| 3 | I | 0x1E | N | N |
|---|---|------|---|---|

, где:

N – количество циклов программы.

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x1E | ack/nack |
|---|---|------|----------|

, где:

size – размер области программ в байтах.

**Команда запроса остановки выполнения программы (0x1F)**

Команда останавливает выполнение программы.

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x1F |
|---|---|------|

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x1F | ack/nack |
|---|---|------|----------|

**Команда запроса приостановки выполнения программы (0x20)**

Команда ставит выполнение программы на паузу, после чего программа может быть продолжена подачей сигнала старт или завершена с помощью сигнала стоп.

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x20 |
|---|---|------|

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x20 | ack/nack |
|---|---|------|----------|

**Команда перехода к следующему кадру (0x21)**

Используется при работе интерфейса в режиме ручного переключения кадров.

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x21 |
|---|---|------|

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x21 | ack/nack |
|---|---|------|----------|



**Команда запроса записи структуры информации о программах (0x22)**

Записывает информацию о программах в энергонезависимую память устройства.

*Запрос:*

|       |                |                |                |                |                |                |                |                |
|-------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 3+6*n | I              | 0x22           | P <sub>0</sub> | P <sub>0</sub> | S <sub>0</sub> | S <sub>0</sub> | S <sub>0</sub> | S <sub>0</sub> |
| ...   | P <sub>n</sub> | P <sub>n</sub> | S <sub>n</sub> | S <sub>n</sub> | S <sub>n</sub> | S <sub>n</sub> |                |                |

, где:

P – страница,

S – размер,

n – количество программ.

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x22 | ack/nack |
|---|---|------|----------|

**Команда запроса чтения структуры информации о программах (0x23)**

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x23 |
|---|---|------|

*Ответ:*

|       |                |                |                |                |                |                |                |                |
|-------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 3+6*n | I              | 0x23           | P <sub>0</sub> | P <sub>0</sub> | S <sub>0</sub> | S <sub>0</sub> | S <sub>0</sub> | S <sub>0</sub> |
| ...   | P <sub>n</sub> | P <sub>n</sub> | S <sub>n</sub> | S <sub>n</sub> | S <sub>n</sub> | S <sub>n</sub> |                |                |

, где:

P – страница,

S – размер,

n – количество программ.

**Команда записи части программы во внутренний временный буфер страницы памяти программ устройства (0x24)**

*Запрос:*

|     |   |      |      |                   |     |                     |
|-----|---|------|------|-------------------|-----|---------------------|
| 132 | I | 0x24 | part | data <sub>0</sub> | ... | data <sub>127</sub> |
|-----|---|------|------|-------------------|-----|---------------------|

, где:

part – часть буфера, куда будет записана data,

data – часть сохраняемой программы.

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x24 | ack/nack |
|---|---|------|----------|

**Команда записи временного буфера страницы программ в энергонезависимую память (0x25)**

*Запрос:*

|   |   |      |      |
|---|---|------|------|
| 4 | I | 0x25 | page |
|---|---|------|------|

, где:

page – номер страницы энергонезависимой памяти, в которую будет сохранен буфер.

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x25 | ack/nack |
|---|---|------|----------|

**Команда проверки целостности программы (0x26)**

Команда запускает процесс проверки целостности сохраненных программ путем расчета контрольных сумм и сравнением их с сохраненными. Также проверяется целостность структуры информации о программах.

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x26 |
|---|---|------|

*Ответ:*

|   |   |      |     |
|---|---|------|-----|
| 4 | I | 0x26 | res |
|---|---|------|-----|

, где:

res – результат проверки (0 – контрольные суммы сошлись, 1 – программа повреждена).

**Команда чтения части страницы области программ (0x27)**

*Запрос:*

|   |   |      |      |      |
|---|---|------|------|------|
| 5 | I | 0x27 | page | part |
|---|---|------|------|------|

, где:

page – номер страницы области программ,

part – читаемая часть страницы.

*Ответ:*

|     |   |      |                   |     |                     |
|-----|---|------|-------------------|-----|---------------------|
| 131 | I | 0x27 | data <sub>0</sub> | ... | data <sub>127</sub> |
|-----|---|------|-------------------|-----|---------------------|

, где:

data – данные программы.

**Команда установки переменной программы (0x28)**

*Запрос:*

|   |   |      |   |       |       |       |       |
|---|---|------|---|-------|-------|-------|-------|
| 8 | I | 0x28 | n | param | param | param | param |
|---|---|------|---|-------|-------|-------|-------|

, где:

n – номер параметра,

param – значение.

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x28 | ack/nack |
|---|---|------|----------|

**Команда чтения переменной программы (0x29)**

*Запрос:*

|   |   |      |   |
|---|---|------|---|
| 4 | I | 0x29 | n |
|---|---|------|---|

, где:

n – номер параметра.

*Ответ:*

|   |   |      |       |       |       |       |
|---|---|------|-------|-------|-------|-------|
| 7 | I | 0x29 | param | param | param | param |
|---|---|------|-------|-------|-------|-------|

, где:

param – значение.

**Команда запроса количества переменных программ (0x2A)**

Запрос поддерживаемого модулем количества переменных программ.

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x2A |
|---|---|------|

*Ответ:*

|   |   |      |   |
|---|---|------|---|
| 4 | I | 0x2A | n |
|---|---|------|---|

, где:

n – количество переменных.

## Модуль зоны

### Описание

Данный модуль позволяет контролировать положение измеряемой величины в заданном диапазоне. При удовлетворении условий попадания в зону, для данной зоны устанавливается сигнал “1”. Этот сигнал с помощью модуля перенаправления функциональных сигналов может быть выведен на выход. Количество зон зависит от устройства и может быть получено с помощью соответствующей команды.

Для каждой зоны доступно три режима работы:

- 0 – выключена,
- 1 – диапазон,
- 2 – пересечение.

В выключенном режиме, выходной сигнал зоны всегда “0”.

В режиме диапазона “1” выдается, если текущая координата находится между двумя заданными координатами в настройках. При этом необходимо, чтобы координата 2 была меньше координаты 1.

В режиме пересечения “1” выдается только при изменении координаты с пересечением значения координаты 1 из настроек зоны. “1” держится до тех пор, пока текущая координата не выйдет за диапазон от (координата1 – координата2) до (координата1 + координата2).

В рабочих режимах работы используется заданная в настройках система отсчета. Если выбрана абсолютная система отсчета и используется инкрементный датчик с референтной меткой, зона не будет определяться до момента захвата референтной метки.

Если выбрано использование действующей системы отсчета, то модуль следит за сменой систем отсчета пользователем. При смене СО происходит сброс сигнала зоны.

### id модуля

Модуль имеет id 9.

### Режим работы:

Модуль имеет только один режим работы 0, смена режима работы недоступна.

### Настройки модуля:

| К-во | Размер(байт) | Описание                                  |
|------|--------------|-------------------------------------------|
| NZ   | 8            | Координата 1                              |
|      | 8            | Координата 2                              |
|      | 1            | Режим                                     |
|      | 1            | Система отсчета (согласно модулю датчика) |
|      | 2            | Зарезервировано                           |

, где:

NZ – количество зон в устройстве.

### Подмодули

Интерфейс не имеет подмодулей.

### Входные функциональные сигналы

Модуль не имеет входных функциональных сигналов

### Выходные функциональные сигналы

0 – позиция в зоне.

**Расширенные команды****Команда получения количества зон (0x14)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x14 |
|---|---|------|

*Ответ:*

|   |   |      |   |
|---|---|------|---|
| 4 | I | 0x14 | n |
|---|---|------|---|

, где:

n – количество зон.

## Модуль Ethernet

### Описание

Данный модуль обеспечивает управление параметрами, а также работу интерфейса Ethernet. Для клиентов задан фиксированный таймаут опроса: если от клиента не приходят запросы более 5 секунд, то клиент автоматически отключается.

### id модуля

Модуль имеет id 10.

### Режим работы:

Режим работы модуля переключает используемый в данный момент подмодуль.

### Настройки модуля:

Модуль не имеет настроек

### Подмодули

Модуль может иметь следующие подмодули:

0 – Modbus RTU over TCP/IP,

1 – Modbus TCP/IP.

### Входные функциональные сигналы

Модуль не имеет входных функциональных сигналов

### Выходные функциональные сигналы

0 – ошибка связи: устанавливается когда к модулю не подключен ни один клиент.

### Расширенные команды

Модуль не имеет расширенных команд.

## Подмодули

### Подмодуль Modbus RTU over TCP/IP

#### Описание

Подмодуль выступает сервером с одним открытым сокетом, предоставляя возможность производить опрос устройства с помощью стандартных запросов протокола Modbus RTU over TCP/IP. При включенной функции ретрансляции и настроенном модуле RS485 в режим Modbus RTU/Master – подмодуль перенаправляет все запросы, адрес которых, отличается от адреса устройства, а также широковещательные запросы по интерфейсу RS485. При ответе от подключенного через RS485 ведомого устройства, подмодуль пересылает ответ клиенту, подключенному к сокету.

При использовании ретрансляции не стоит посылать новый запрос раньше, чем получен ответ от ведомого устройства, подключенного к линии RS485.

#### id подмодуля

Подмодуль имеет id 0.

#### Настройки подмодуля:

| Размер(байт) | Описание                                                    |
|--------------|-------------------------------------------------------------|
| 1            | Адрес (1...247)                                             |
| 4            | IP адрес шлюза (1...255,1...255,1...255,1...255)            |
| 4            | Маска подсети (0...255,0...255,0...255,0...255)             |
| 6            | MAC адрес (0...255,0...255,0...255,0...255,0...255,0...255) |
| 4            | IP адрес устройства (1...255,1...255,1...255,1...255)       |
| 2            | Порт (0...65535)                                            |
| 1            | Ретрансляция (0 – выкл, 1 - вкл)                            |

**Команды****Запрос количества ошибок связи (0x3C)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x3C |
|---|---|------|

*Ответ:*

|   |   |      |     |     |     |     |
|---|---|------|-----|-----|-----|-----|
| 7 | I | 0x3C | err | err | err | err |
|---|---|------|-----|-----|-----|-----|

, где:

err – количество ошибок передачи.

**Сбросить счетчик ошибок связи (0x3D)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x3D |
|---|---|------|

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x3D | ack/nack |
|---|---|------|----------|

**Запрос кода последней ошибки связи (0x3E)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x3E |
|---|---|------|

*Ответ:*

|   |   |      |     |
|---|---|------|-----|
| 4 | I | 0x3E | err |
|---|---|------|-----|

, где:

err – код последней ошибки передачи.

*Коды ошибок передачи:*

0x01 – 0x08 – стандартные коды ошибок Modbus протокола,

0x09 – таймаут ответа ведомого,

0x0A – ошибка CRC,

0x0B – нет ошибок,

0x0C – ошибка линии (контроль четности, формат пакета, переполнение).

Во время процесса передачи, в ответ на данную команду возвращается nack.



**Подмодуль Modbus TCP/IP****Описание**

Подмодуль выступает сервером с одним открытым сокетом, предоставляя возможность производить опрос устройства с помощью стандартных запросов протокола Modbus TCP/IP. При включенной функции ретрансляции и настроенном модуле RS485 в режим Modbus RTU/Master – подмодуль перенаправляет все запросы, адрес которых, отличается от адреса устройства, а также широковещательные запросы по интерфейсу RS485, конвертируя их в формат протокола Modbus RTU. При ответе от подключенного через RS485 ведомого устройства, подмодуль конвертирует ответ в формат протокола Modbus TCP/IP и пересылает ответ клиенту, подключенному к сокету.

При использовании ретрансляции не стоит посылать новый запрос раньше, чем получен ответ от ведомого устройства, подключенного к линии RS485.

**id подмодуля**

Подмодуль имеет id 1.

**Настройки подмодуля:**

| Размер(байт) | Описание                                                    |
|--------------|-------------------------------------------------------------|
| 1            | Адрес (1...247)                                             |
| 4            | IP адрес шлюза (1...255,1...255,1...255,1...255)            |
| 4            | Маска подсети (0...255,0...255,0...255,0...255)             |
| 6            | MAC адрес (0...255,0...255,0...255,0...255,0...255,0...255) |
| 4            | IP адрес устройства (1...255,1...255,1...255,1...255)       |
| 2            | Порт (0...65535)                                            |
| 1            | Ретрансляция (0 – выкл, 1 - вкл)                            |

**Команды****Запрос количества ошибок связи (0x3C)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x3C |
|---|---|------|

*Ответ:*

|   |   |      |     |     |     |     |
|---|---|------|-----|-----|-----|-----|
| 7 | I | 0x3C | err | err | err | err |
|---|---|------|-----|-----|-----|-----|

, где:

err – количество ошибок передачи.

**Сбросить счетчик ошибок связи (0x3D)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x3D |
|---|---|------|

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x3D | ack/nack |
|---|---|------|----------|

**Запрос кода последней ошибки связи (0x3E)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x3E |
|---|---|------|

*Ответ:*

|   |   |      |     |
|---|---|------|-----|
| 4 | I | 0x3E | err |
|---|---|------|-----|

, где:

err – код последней ошибки передачи.

*Коды ошибок передачи:*

0x01 – 0x08 – стандартные коды ошибок Modbus протокола,

0x09 – таймаут ответа ведомого,

0x0A – ошибка CRC,

0x0B – нет ошибок,

0x0C – ошибка линии (контроль четности, формат пакета, переполнение).

Во время процесса передачи, в ответ на данную команду возвращается nack.

## Модуль графического интерфейса

### Описание

Данный модуль предоставляет доступ к настройкам графического интерфейса устройств с экраном, позволяя производить настройку с удаленного устройства. Так же данный модуль позволяет получать снимок экрана и эмулировать нажатия на клавиши устройства.

Модуль графического интерфейса индивидуален для каждого устройства и имеет различные настройки и набор команд. Однако все модули графического интерфейса поддерживают ряд стандартных команд перечисленных ниже.

### id модуля

Модуль имеет id 11.

### Режим работы:

Модуль имеет два режима работы: нормальный и тестовый. В нормальном режиме отображается информация согласно назначению устройства. В тестовом режиме экран заливается монотонным цветом для проверки на битые пиксели и, при нажатии на кнопки, возникают всплывающие сообщения с названием нажатой кнопки.

### Настройки модуля:

Индивидуальны для каждого устройства (см. в инструкции).

### Подмодули

Модуль не имеет подмодулей.

### Входные функциональные сигналы

Модуль может иметь различные функциональные сигналы в зависимости от устройства

### Выходные функциональные сигналы

Модуль может иметь различные функциональные сигналы в зависимости от устройства

**Расширенные команды****Запрос информации о дисплее (0x14)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x14 |
|---|---|------|

*Ответ:*

|         |         |         |         |         |         |         |         |      |
|---------|---------|---------|---------|---------|---------|---------|---------|------|
| 17      | I       | 0x14    | width   | width   | height  | height  | depth   | type |
| act_col | act_col | act_col | act_col | neg_col | neg_col | neg_col | neg_col |      |

, где:

width – ширина экрана в пикселях,

height – высота экрана в пикселях,

depth – глубина цвета (бит),

type – тип дисплея (0 – монохромный, 1 – цветной),

act\_col - для монохромных дисплеев, цвет зажженного пикселя,

neg\_col – для монохромных дисплеев, цвет фона.

**Запрос части графического буфера (0x15)**

Буфер передается частями по 128 байт за команду. Во время перерисовки экрана устройство отвечает на данную команду пустым пак.

*Запрос:*

|   |   |      |   |
|---|---|------|---|
| 4 | I | 0x15 | n |
|---|---|------|---|

, где:

n – номер части буфера кадра.

*Ответ:*

|     |   |      |                   |     |                     |
|-----|---|------|-------------------|-----|---------------------|
| 131 | I | 0x15 | data <sub>0</sub> | ... | data <sub>127</sub> |
|-----|---|------|-------------------|-----|---------------------|

, где:

data – данные буфера кадра.

**Команда эмуляции нажатия на клавишу (0x16)**

Важно помнить, что устройство не имеет буфера нажатий, так что посылать данную команду следует не чаще 50мс. В противном случае часть нажатий перезапишется последующими. Коды клавиш можно посмотреть в инструкции на устройство.

*Запрос:*

|   |   |      |          |
|---|---|------|----------|
| 3 | I | 0x16 | key_code |
|---|---|------|----------|

, где:

key\_code – код нажатой клавиши.

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x16 | ack/nack |
|---|---|------|----------|

## Модуль математический

### Описание

Модуль позволяет пользователю производить арифметические операции с позициями как собственных осей устройства, так и с синхронизованными через протокол сппу (только для мастера).

Результаты вычисления кэшируются в устройстве и обновляются при первом запросе после установки флага пересчета. Флаг пересчета автоматически устанавливается в начале каждого системного цикла, а также, после получения соответствующей команды от пользователя.

Формула вычисления задается пользователем с помощью последовательности арифметических команд. Каждая команда состоит из кода операции и двух операндов:

| Поле:          | Код операции | Операнд 1 | Операнд 2 |
|----------------|--------------|-----------|-----------|
| Размер (байт): | 1            | 1         | 1         |

Пользователю доступны следующие операции:

| Код: | Операция:             | Описание:                                                                                                                                                                                                           |
|------|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x00 | Сложение              | Складывает два числа, адреса которых записаны в операнды.                                                                                                                                                           |
| 0x01 | Вычитание             | Вычитает из значения ячейки памяти, адрес которой записан в операнд один, значение ячейки, адрес которой записан в операнд два.                                                                                     |
| 0x02 | Умножение             | Перемножает два значения ячеек памяти, адреса которых записаны в операнды.                                                                                                                                          |
| 0x03 | Деление               | Делит значение ячейки памяти, адрес которой записан в операнд один, на значение ячейки, адрес которой записан в операнд два. Округление в меньшую сторону.                                                          |
| 0x04 | Остаток от деления    | Остаток от деления значения ячейки памяти, адрес которой записан в операнд один, на значение ячейки, адрес которой записан в операнд два.                                                                           |
| 0x05 | Деление с округлением | Делит значение ячейки памяти, адрес которой записан в операнд один, на значение ячейки, адрес которой записан в операнд два. Округление до ближайшего целого.                                                       |
| 0x06 | Синус                 | Взятие синуса от значение ячейки памяти, адрес которой записан в операнд два. Число должно быть в градусах. Во время выполнения операции учитывается заданное в настройках положение точки (целая/дробная часть).   |
| 0x07 | Косинус               | Взятие косинуса от значение ячейки памяти, адрес которой записан в операнд два. Число должно быть в градусах. Во время выполнения операции учитывается заданное в настройках положение точки (целая/дробная часть). |
| 0x08 | Перемещение           | Приравнивает ячейку памяти, на которую указывает операнд один к значению ячейки, на которую указывает операнд два.                                                                                                  |
| 0x09 | Окончание вычисления  | Конец выполнения программы расчета.                                                                                                                                                                                 |

Операнды содержат адрес ячейки памяти, с которой необходимо произвести действие. Пользователю доступна область памяти, состоящая из 64-битных регистров, разделенная на четыре группы: регистры под переменные, регистры констант, регистры позиции осей и регистры результата.

| Группа     | Адрес               | Регистры        | Доступ          |
|------------|---------------------|-----------------|-----------------|
| Переменные | 0x00                | Переменная 0    | Чтение и запись |
|            | 0x01                | Переменная 1    |                 |
|            | ...                 |                 |                 |
|            | 0x05                | Переменная 6    |                 |
|            | 0x05 – 0x27         | Зарезервировано |                 |
| Константы  | 0x28                | Константа 0     | Чтение          |
|            | 0x29                | Константа 1     |                 |
|            | ...                 |                 |                 |
|            | 0x3B                | Константа 19    |                 |
|            | 0x3C – 0x4F         | Зарезервировано |                 |
| Позиции    | 0x50                | Позиция оси 0   | Чтение          |
|            | 0x51                | Позиция оси 1   |                 |
|            | ...                 |                 |                 |
|            | $0x50 + m - 1$      | Позиция оси m-1 |                 |
|            | $(0x50 + m) - 0x77$ | Зарезервировано |                 |
| Результаты | 0x78                | Результат 0     | Чтение и запись |
|            | 0x79                | Результат 1     |                 |
|            | ...                 |                 |                 |
|            | $0x78 + k - 1$      | Результат k-1   |                 |
|            | $(0x78 + k) - 0x9F$ | Зарезервировано |                 |

, где:

m – максимально возможное количество осей для устройства,

k – количество регистров результата.

Количество регистров можно узнать с помощью соответствующей команды модуля.

Количество регистров констант постоянно для всех устройств и равно 20. Во время работы устройства туда проецируются соответствующие поля настроек (константы).

Регистры переменных предназначены для временного сохранения промежуточных результатов вычислений. Результат арифметических операций всегда сохраняется в нулевой регистр этой группы. Модуль всегда содержит 6 таких регистров.

В регистры позиции проецируются значения осей устройства. Начиная с нулевого регистра этой группы располагаются собственные оси устройства. Следом за ними идут оси ведомых устройств, синхронизация с которыми осуществляется через протокол сппу (в режиме мастера). Для этого должен быть настроен подмодуль сппу модуля Модбас. Если синхронизация с другими устройствами не требуется, то необходимо считать соответствующие регистры зарезервированной областью.

Во время арифметических операций со значениями осей, модуль накапливает их значения статуса (через логическое “ИЛИ”) в теневом регистре статуса, после чего, при записи в регистр результата, накопленное значение статуса записывается в соответствующее поле результата. Теневой регистр статуса при этом не обнуляется.

Регистры результата содержат результат выполнения арифметических действий согласно последовательности, заданной пользователем.

Операнды не должны содержать адреса регистров, являющихся зарезервированными. Если во время вычисления будет обработана команда с операндом, указывающим на зарезервированный регистр, то вычисление будет остановлено.

**id модуля**

Модуль имеет id 12.

**Режим работы:**

Модуль имеет только один режим работы 0, смена режима работы недоступна.

**Настройки модуля:**

| Размер(байт) | Описание                                             |
|--------------|------------------------------------------------------|
| 1            | Положение точки (используется при расчете sin и cos) |
| 50* 3        | Массив команд из 50 элементов                        |
| 20 * 4       | Массив констант из 20 элементов (знаковых).          |

**Подмодули**

Модуль не имеет подмодулей.

**Входные функциональные сигналы**

Модуль не имеет входных функциональных сигналов

**Выходные функциональные сигналы**

Модуль не имеет выходных функциональных сигналов

**Расширенные команды****Обновить все результаты (0x14)**

По этой команде принудительно устанавливается флаг необходимости пересчета результатов. Значения будут рассчитаны по первому запросу результата.

*Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x14 |
|---|---|------|

*Ответ:*

|   |   |      |          |
|---|---|------|----------|
| 4 | I | 0x14 | ack/nack |
|---|---|------|----------|

**Запрос результата вычисления (0x15)**

Команда возвращает результат расчета по формуле пользователя, из заданного регистра результата.

*Запрос:*

|   |   |      |   |
|---|---|------|---|
| 4 | I | 0x15 | n |
|---|---|------|---|

, где:

n – номер регистра результата.

*Ответ:*

|    |   |      |     |     |     |     |     |     |     |     |        |        |
|----|---|------|-----|-----|-----|-----|-----|-----|-----|-----|--------|--------|
| 13 | I | 0x15 | res | res | res | res | res | res | res | res | status | status |
|----|---|------|-----|-----|-----|-----|-----|-----|-----|-----|--------|--------|

, где:

status – статус осей, позиции которых участвовали в расчете,

res – результат расчета из запрошенного регистра (знаковая).

**Информация о количестве регистров модуля (0x16)***Запрос:*

|   |   |      |
|---|---|------|
| 3 | I | 0x16 |
|---|---|------|

*Ответ:*

|   |   |      |       |             |              |
|---|---|------|-------|-------------|--------------|
| 6 | I | 0x16 | res_n | self_axis_n | slave_axis_n |
|---|---|------|-------|-------------|--------------|

, где:

res\_n – количество регистров результата,

self\_axis\_n – количество осей у устройства,

slave\_axis\_n – максимальное количество осей ведомых устройств.



## История изменений

### 22.03.2018:

- Добавлено описание форматов пакета управления для связи через Ethernet.
- Добавлено описание подмодуля аналоговых датчиков.
- Добавлено описание модуля Ethernet и его подмодулей.
- Добавлено описание протокола сппу.

### 29.06.2018:

- Добавлено описание модуля математического.

### 16.08.2018:

- Изменено поле данных команды “Запрос параметров модуля” модуля виртуальных входов/выходов.
- Подправлено описание модуля виртуальных входов/выходов.
- В поле статуса координаты модуля датчика добавлен дополнительный флаг повреждения сохраненного смещения системы координат.
- Исправлен размер поля данных команды “Команда запроса размера страницы области программ” модуля обработки g-кодов.
- Добавлена команда “Команда запроса количества переменных программ” в модуле обработки g-кодов.

### 05.09.2018

- Добавлено описание модуля графического интерфейса.

### 04.12.2018

- Добавлено описание конфликтующих модулей.
- Обновление описания модуля функциональных сигналов и описание сигналов перенесено в модули, к которым они относятся.
- Добавлено описание подмодуля протокола ЛИР-915/6.