

**ТИРИСТОРНЫЕ ПРЕОБРАЗОВАТЕЛИ  
ТИПА 4XXX RS ДЛЯ УПРАВЛЕНИЯ  
ДВИГАТЕЛЕЙ ПОСТОЯННОГО ТОКА С  
НЕЗАВИСИМЫМ ВОЗБУЖДЕНИЕМ**

**ПОСЛЕДОВАТЕЛЬНАЯ КОММУНИКАЦИЯ RS485**

**ПРОТОКОЛ ELL\_RS485\_N1**



## СОДЕРЖАНИЕ

<b>1. ПОСЛЕДОВАТЕЛЬНАЯ КОММУНИКАЦИИ RS 485</b>	<b>4</b>
1.1. Общие положения	4
1.3. Параметры серийного интерфейса RS485	6
<b>2. ПРОТОКОЛ ELL_RS485_N1</b>	<b>8</b>
2.1. Общие положения	8
2.2. Передача данных от master устройства к преобразователям	8
2.3. Передача данных от преобразователей к master устройству	10

## 1. ПОСЛЕДОВАТЕЛЬНАЯ КОММУНИКАЦИИ RS 485

### 1.1. Общие положения

В каждом последовательном интерфейсе RS485 есть только одно устройство **master**. Подчиненные устройства **slave** не могут занимать шину по своей инициативе.

**Master** устройство инициирует коммуникацию с подчиненными устройствами **slave** с **командным байтом**, передает **данные** к ним (если есть такие), формирует и передает **контрольную сумму данных**. **Master** устройство ожидает **АСК** или **NAK** от подчиненных устройств или **данные** из них. Все **данные** сопровождаются с **контрольной суммой**. **Контрольная сумма** всегда формируется **master** устройством.

Преобразователи являются для протокола **подчиненными** устройствами, а **master** устройство это контроллер или ЧПУ, которое отправляет и принимает данные к и от преобразователей, посредством данного протокола.

Настройка серийной коммуникации **RS485** реализована с применением 1 стартового бита, 8 битов данных, 1 стоп бита и без бита для четности.

### КОМАНДНЫЙ БАЙТ

Командный байт со старшей частью **0x0..** и младшей частью, отличной от нуля. Из этого байта подчиненным устройствам указывается, что тоже должны принять участие в протоколе, если командный байт относится к ним. Т.е. подчиненные устройства следят впервые найти по линии их командный байт и после этого принять участие в протоколе до конца одного полного их протокольного цикла или до появления другого командного байта.

### Резервированные командные байты:

**0x04** – командный байт для отправления данных от **master** устройства к преобразователю;

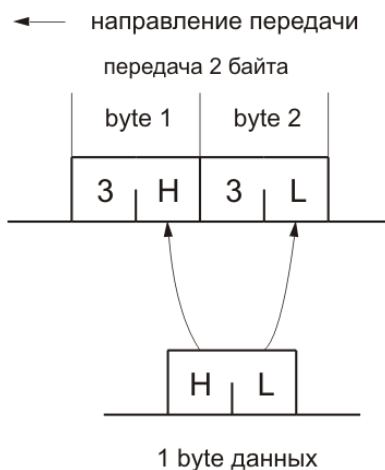
**0x05** – командный байт для отправления данных от преобразователя к **master** устройству;

**0x06** – командный байт для тайминга в ms;

### ДАННЫЕ

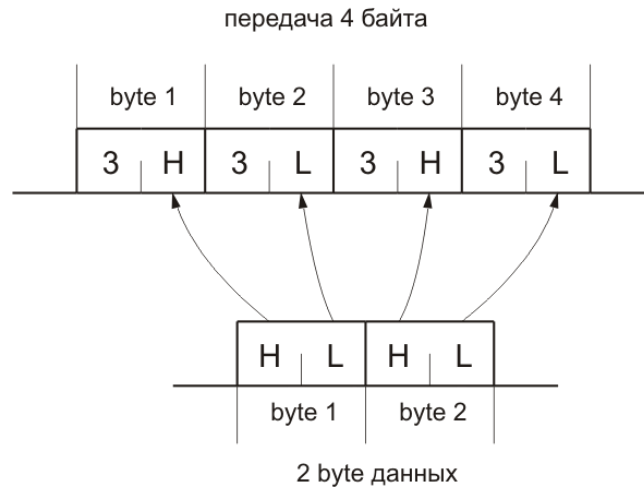
Передаваемые байты, содержащие данные, имеют старшую часть **0x3..** и младшую часть 4 бита данных. Для отправления одного байта данных, отправляются два байта, как это показано на **рисуне 1**. В первом байте находится старшая половина байта данных.

Двубайтовые числа данных передаются с 4 байтами, как это показано на **рисунке 2**. После передачи данных **master** отправляет обязательно **контрольную сумму** из двух байтов. Число байтов данных не фиксировано и их конец отсчитывается от получения **контрольной суммы**.



**Рисунок 1** Передача 1 byte данных

← направление передачи



**Рисунок 2** Передача 2 byte данных

### КОНТРОЛЬНАЯ СУММА

Контрольная сумма **CRC** состоит из одного байта и получается обычной суммой всех байтов данных. Она передается двумя байтами со старшей частью **0x4..** и младшей частью ее двух полубайтов. Сначала отправляется старший полубайт, при отправлении контрольной суммы возможны два случая:

- контрольная сумма при передаче данных от **master** устройства к подчиненным устройствам. В этом случае после отправления контрольной суммы **master** ожидает получить для каждого отправленного байта **ACK** или **NAK** от данного подчиненного устройства, для которого адресированы данные;
- контрольная сумма при передаче данных от подчиненных устройств к **master** устройству. В этом случае данные получаются последовательно от разных подчиненных устройств и после окончания передачи **master** устройство вычисляет и передает контрольную сумму. **Master** ожидает только получить ли из некоторого устройства **NAK**.

### ACK

**ACK** имеет старшую часть **0x5..** и младшую часть 0, 1 или 2. **ACK** принимается для подтверждения, что цикл протокола прошел успешно. Имеются следующие случаи для **ACK**:

- подчиненное устройство отвечает с **0x50**, если направленная главным устройством контрольная сумма совпадает с вычисленной им;
- подчиненное устройство отвечает с **0x51**, если оно прошло ресет (выключено) из-за **TIMEOUT**;
- подчиненное устройство отвечает с **0x52**, если оно прошло ресет (выключено) из-за отпадания напряжения питания.

### NAK

**NAK** имеет вид **0x00**, когда ожидается из одного или нескольких устройств одновременно, или отличается по некоторому из младших битов от **ACK**, когда ожидается только из одного устройства. **NAK** указывает, что текущий цикл протокола неуспешный.

**Подчиненные** устройства отправляют по одному **ACK** или **NAK** для каждого байта данных, последовательно в порядке их номеров. Если данное подчиненное устройство получает два байта данных, оно должно ответить двумя **ACK** или **NAK**.

**Подчиненные** устройства следят за всеми другими устройствами отправили ли они **АСК**, и если все ответили, после последнего **АСК** синхронно вводят информацию для выполнения. Даже если только одно устройство ответило с **НАК** или не ответило, команды не выполняются и текущий цикл протокола считается недействительным. **Подчиненные** устройства определяют сколько **АСК** или **НАК** следует ожидать, по числу байтов данных ( $N = \text{число байтов} / 2$ ).

**ПРЕРЫВАНИЕ В КОММУНИКАЦИИ / TIMEOUT /** : в случае прерывания в коммуникации для определенного времени, подчиненное устройство должно ресетоваться (выключится). Время интервала стандартно настроено на 100 ms для бод рейда 9600 или на 10 ms для бод рейда 115200. **Timeout** можно настраивается **master** устройством со следующей последовательностью:

**master** - 0x06, 0x3(ms H), 0x3(ms L), 0x4(CRC H), 0x4(CRC L)

**slave** -

После этого **master** устройство выжидает, получится ли **НАК** из какого то устройство.

#### ЗАМЕЧАНИЯ:

1. Каждое подчиненное устройство должно не интересоваться байтами 0xFF или байтами со ошибочным стартовым битом (т.е должно их не учитывать);
2. Каждое подчиненное устройство должно следит занимает ли оно шину непосредственно после **master** устройства. Если оно первое после **master**, то оно должно выжидать время для одного байта протокола RS485 и только после этого может занимать шину и передавать;
3. Каждое подчиненное устройство должно работать согласно Замечанию 2, но эму можно разрешить с параметром **P14.07** не выжидать времени для одного байта. В этом случае **master** должен иметь возможность непосредственно после отправления шифт регистра RS485 менять направление коммуникации.

### 1.3. Параметры серийного интерфейса RS485

Параметры группы **P14**, с которыми настраивается работа преобразователей серии 4XXX по протоколу ELL\_RS485\_N1 указаны в **таблице 1**. Полная версия таблицы со всеми параметрами группы **P14** можно увидеть в Техническом описании преобразователей 4XXX, версия документа UGrSPD88\_306 RS, **таблица 10**.

№	Наименование параметра	Текст	Пределы	По умолчанию
<b>Группа 14 – параметры серийного интерфейса RS485</b>				
P14.01	Тип коммуникации	Type communic	0 ÷ 1	0
P14.02	Скорость коммуникации	Baud rate	0 ÷ 5	0
P14.03	Начальный адрес данного подчиненного устройства на входе <b>master</b> устройства	Addr INputs slv	0 ÷ 64	0
P14.04	Начальный адрес данного подчиненного устройства на выходе <b>master</b> устройства	Addr OUTputs slv	0 ÷ 64	0
P14.05	Число байтов данных данного подчиненного устройства на входе <b>master</b> устройства	Num INputs slv	0 ÷ 32	5
P14.06	Число байтов данных для данного подчиненного устройства на выходе <b>master</b> устройства	Num OUTputs slv	0 ÷ 32	3
P14.07	Выжидание главного устройства	Wait Master	0, 1	1
P14.08	Источник времени прерывания в коммуникации подчиненного устройства – защита <b>CSF</b>	Src timeout slv	0, 1	0

№	Наименование параметра	Текст	Пределы	По умолчанию
P14.09	Максимальное время прерывания в коммуникации подчиненного устройства – защита CSF	Timeout slave	5 ÷ 1000	
P14.14	Максимальное зарегистрированное время прерывания в коммуникации	Max timeout slv	-	-

**Таблица 1** Параметры преобразователя для настройки протокола

- параметр **P14.01** – тип коммуникации преобразователя:
  - **P14.01 = 0** – без коммуникации;
  - **P14.01 = 1** – RS485 slave. Преобразователь будет работать в режиме **slave** устройства;
- параметр **P14.02** – скорость коммуникации в bps. Принимает следующие значения:
  - **P14.02 = 0** – 115 200 bps;
  - **P14.02 = 1** – 57 600 bps;
  - **P14.02 = 2** – 38 400 bps;
  - **P14.02 = 3** – 19 200 bps;
  - **P14.02 = 4** – 14 400 bps;
  - **P14.02 = 5** – 9 600 bps.
- параметр **P14.03** – начальный адрес байтов данных на выходе подчиненного устройства. Определяет адрес байтов данного подчиненного устройства в общей последовательности байтов поступающих на входе **master** устройства;
- параметр **P14.04** – начальный адрес байтов данных для данного подчиненного устройства. Определяет адрес байтов для данного подчиненного устройства в общей последовательности байтов на выходе **master** устройства;
- параметр **P14.05** – число байтов данных на выходе подчиненного устройства. Определяет число байтов из данного подчиненного устройства на входе **master** устройства;
- параметр **P14.06** – число байтов данных для данного подчиненного устройства на выходе **master** устройства;
- параметр **P14.07** – выжидание после окончания передачи **master** устройства. Параметр **P14.07** принимает две значения:
  - **P14.07 = 0** – подчиненное устройство не выжидает времени одного байта после окончания передачи **master** устройства;
  - **P14.07 = 1** – подчиненное устройство выжидает время одного байта после окончания передачи **master** устройства.
- параметр **P14.08** – источник максимального времени прерывания в коммуникации подчиненного устройства. При превышении этого времени срабатывает защита **CSF**. Параметр **P14.08** принимает две значения:
  - **P14.08 = 0** – максимальное время прерывания в коммуникации определяется значением параметра **P14.08**;
  - **P14.08 = 1** – максимальное время прерывания передается по последовательной коммуникации.
- параметр **P14.09** – максимальное время прерывания / timeout / в коммуникации подчиненного устройства. При превышении этого время срабатывает защита **CSF**;
- параметр **P14.14** – максимальное зарегистрированное время прерывания в коммуникации подчиненного устройства.

## 2. ПРОТОКОЛ ELL\_RS485\_N1

### 2.1. Общие положения

Дальше будет рассмотрен стандартный протокол ELL\_RS485\_N1 для последовательного интерфейса RS485 для тиристорных преобразователей типа 4XXX.

Основные моменты в осуществлении настоящего Протокола:

- для **master** устройства в этом протоколе применяется внешний контроллер;
- **master** устройство передает к всем преобразователям;
- тиристорные преобразователи 4XXX в этом протоколе являются подчиненными устройствами;
- **master** устройство передает к каждому преобразователю 1 byte команды: **ON, RESET, ORCM, SF, SR, TLL**;
- **master** устройство передает к каждому преобразователю 2 byte задания для скорости;
- каждый преобразователь передает к **master** устройству 1 byte для статуса: **TL, ZS, SA, RD, INPOS**;
- каждый преобразователь передает к **master** устройству 2 byte текущего значения действительной скорости;
- каждый преобразователь передает к **master** устройству 2 byte текущего значения тока якоря;

С этим протоколом можно решать широкий круг задач, связанный с управлением преобразователями по последовательной коммуникации RS485.

Для решения некоторых специфических задач этот протокол можно модифицировать по требованиям клиентам.

### 2.2. Передача данных от master устройства к преобразователям

Дальше будет рассмотрена передача данных от **master** устройства к преобразователям 4XXX по протоколу ELL\_RS485\_N1.

**Master** устройство инициирует коммуникацию с подчиненными устройствами с **командным байтом 0x04**, после чего **master** отправляет  $2 * N$  байтов данных. Число байтов данных  $N \geq 1$  и  $N \leq 65$ . По этому способу в одном канале RS485 можно включать до 65 байтов данных. Для каждого устройства задается адрес от 0 до 64, после которого начинается его последовательность байтов.

От **master** устройства к каждому преобразователю передаются один байт команд и два байта данных и параметр **P14.06** = 3. Предназначение каждого байта рассмотрено ниже.

Параметром **P14.04** настраиваются адреса байтов для соответного преобразователя в протоколе при передаче данных от **master** устройства к преобразователям, связанным по последовательному интерфейсу **RS485**.

**Замечание:** не допускается прерывание в последовательности адресов. Например если адрес **byte 0** второго преобразователя зададим с номером **P14.04** = 4 или больше, протокол не будет функционировать.

#### ➤ команды к преобразователю

Команды к преобразователю отправляются байтом 0 из последовательности байтов к нему. Адрес **byte 0** в последовательности байтов к преобразователю определяется параметром **P04.04**.

Соответствие между битами командного байта и коммуникационных цифровых входов указано в **таблице 2**.

С параметрами группы 15 можно запрограммировать функцию и логику каждого коммуникационного цифрового входа. В **таблице 2** указано соответствие между битами байта



0 и номерами коммуникационных цифровых входов, а также и запрограммированных функций по умолчанию.

Номера функций коммуникационных цифровых входов указаны в **таблице 3** Технического описания преобразователей 4XXX, версия документа UGbSPD88\_306 RS.

Номер на коммуникационный вход	bit	Номер на функцията	Функция по подразбиране	Логическо ниво на входа
IN1	bit 0	P15.01 = 1	ON	P15.02 = 0
IN2	bit 1	P15.03 = 2	RESET	P15.04 = 0
IN3	bit 2	P15.05 = 3	ORCM	P15.06 = 0
IN4	bit 3	P15.07 = 9	SF (FORWARD)	P15.08 = 0
IN5	bit 4	P15.09 = 10	SR (REVERSE)	P15.10 = 0
IN6	bit 5	P15.11 = 11	TLL	P15.12 = 0
IN7	bit 6	P15.13 = 13	Nmax1	P15.14 = 0
IN8	bit 7	P15.15 = 14	Nmax2	P15.16 = 0

**Таблица 2** Соответствие между коммуникационными цифровыми входами и запрограммированными функциями по умолчанию

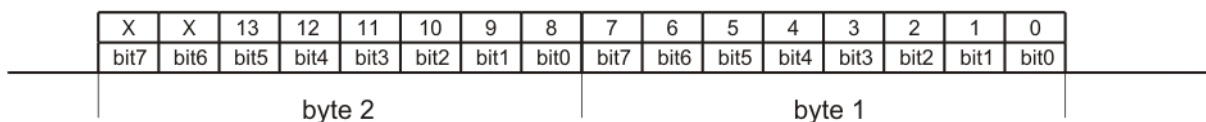
➤ **задание для скорости [VELOCITY REFERENCE]**

Задание для скорости выражается 14 битовым целым числом без знака и передается **байтом 1** и **байтом 2**.

Младшая часть задания для скорости передается **байтом 1**. Адресс **байта 1** в последовательности байтов к преобразователю определяется значением **P14.04+1**.

Старшая часть задания для скорости передается **байтом 2**. Адресс **байта 2** в последовательности байтов к преобразователю определяется значением **P14.04+2**.

**VELOCITY REFERENCE**



**Рисунок 3** Представление задания для скорости в BIN формате

Пределы изменения задания для скорости в разных форматах указаны в **таблице 3**.

Максимальное значение переданного задания для скорости соответствует 100% заданию для скорости двигателя.

Пределы	BIN формат	HEX формат	DEC формат
MIN	0000 0000 0000 0000	0x0000	0
MAX	0011 1000 0000 0000	0x3800	14336

**Таблица 3** Пределы изменения задания для скорости и форматы его представления

**Внимание: преобразователь получает задание для скорости по коммуникацию при значении параметра P02.14 = 4.**

**Пример 1**

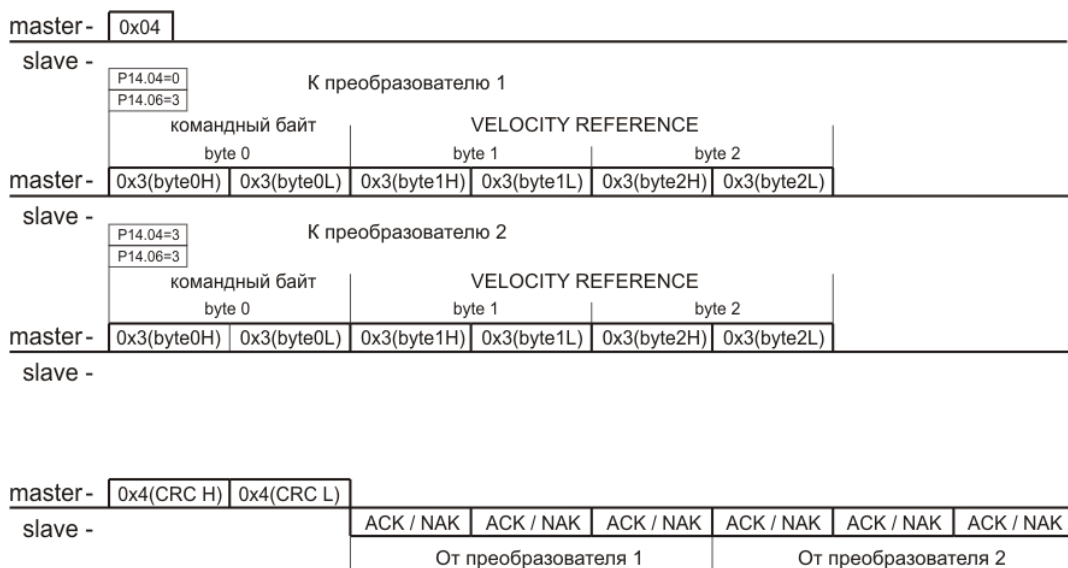
Пример относится для управления двумя преобразователями 4XXX по протоколу ELL\_RS485\_N1 от управляющего устройства **master**.

Для преобразователя 1 параметр **P14.04** = 0, а для преобразователя 2 параметр **P14.04** = 3 /таблица 1/.

**Master** устройство отправляет командный байт **0x04**, 6 байтов данных (разделены в полубайтах т.е. 12 байтов со старшей частью 0x3X) и вычисленную контрольную сумму 1 байта (разделенный в 2 полубайтах с старшей частью 0x4X).

Преобразователи отвечают своими **ACK/NAK** для каждого байта данных т.е. 6 байтов **ACK/NAK**.

Первые три байта **byte 0 ÷ byte 2** относятся для преобразователя 1, следующие три байта **byte 0 ÷ byte 2** для преобразователя 2.



**Рисунок 4** Передача данных от master устройства к преобразователям

### 2.3. Передача данных от преобразователей к master устройству

Дальше будет рассмотрена передача данных от преобразователей к **master** устройству по протоколу ELL\_RS485\_N1.

**Master** устройство иницирует коммуникацию с подчиненными устройствами с **командным байтом 0x05**. Подчиненные устройства читают синхронно свои данные (статус) в междинных буферах, после чего каждое подчиненное устройство отправляет два байта данных для каждого байта своих данных. Общее число байтов данных одного RS канала поступающих на вход **master** устройства  $2*N$ . Число байтов данных  $N \geq 1$  и  $N \leq 65$ . По этому способу в одном канале **RS485** можно передать до 65 байтов. Для каждого устройства задается адрес от 0 до 64, от которого начинается последовательность его байтов.

**Подчиненные устройства** следят и вычисляют контрольную сумму всех байтов данных и отправляют **NAK** если она не совпадает с контрольной суммой, поданной **master** устройством. Если контрольная сумма верна, не отправляют ничего. Подчиненные устройства узнают когда по линии передаются данные или контрольные суммы по старшим полубайтам.

**NAK** имеет значение **0x00**. По этому способу, даже если несколько подчиненных устройств его отправят одновременно, возможно **master** устройство получит ошибку в стоп битах или **break** по линии, что для него тоже является **NAK**.

В данном протоколе от каждого преобразователя к **master** устройству передаются 5 байтов данных.

От преобразователя, после получения командного байта **0x05**, к **master** устройству отправляются 5 байтов данных и параметр **P14.05 = 5**. Предназначение каждого байта рассмотрено ниже.

Параметром **P14.03** настраиваются адреса байтов соответного преобразователя в протоколе при передаче даннн от преобразователя к **master** устройству.

**Замечание:** не допускается прерывание в последовательности номеров, например если адрес **byte 0** второго преобразователя зададим с номером **P14.03 = 6** или больше, протокол не будет функционировать.

От преобразователя к **master** устройству передаются следующие байты:

➤ **статус преобразователя**

Статус преобразователя передается **байтом 0**.

Адресс байта 0 в последовательности байтов к master устройству определяется параметром **P14.03**.

Соответствие между битами статусного байта и коммуникационными цифровыми выходами указано в **таблице 4**.

С параметрами группы 16 преобразователя можно запрограммировать функцию и активный логический уровень каждого коммуникационного цифрового выхода. В **таблице 4** указано соответствие между битами байта 0 и номерами коммуникационных цифровых выходов, а также и запрограммированные функции по умолчанию.

В **таблице 4** Технического описания преобразователей 4XXX, версия документа UGьSPD88\_306 RS, указаны номера функций коммуникационных цифровых выходов.

Номер на коммуникационный изход	bit	Номер на функцията	Функция по подразбиране	Активно логическо ниво на изхода
OUT 1	bit 0	P16.01 = 1	TL	P16.02 = 0
OUT 2	bit 1	P16.03 = 11	ZS	P16.04 = 0
OUT 3	bit 2	P16.05 = 12	SA	P16.06 = 0
OUT 4	bit 3	P16.07 = 4	RD	P16.08 = 0
OUT 5	bit 4	P16.09 = 6	INPOS	P16.10 = 0
OUT 6	bit 5	P16.11 = 0	DISABLED	P16.12 = 0
OUT 7	bit 6	P16.13 = 0	DISABLED	P16.14 = 0
OUT 8	bit 7	P16.15 = 0	DISABLED	P16.16 = 0

**Таблица 4** Соответствие между коммуникационными цифровыми выходами и запрограммированными функциями по умолчанию

➤ **текущее значение скорости [VELOCITY FEEDBACK]**

Текущее значение скорости двигателя является 14 битовым целым числом со знаком и передается двумя байтами.

Младшая часть текущего значения скорости передается **байтом 1**. Адресс **байта 1** в последовательности байтов к **master** определяется значением **P14.03+1**.

Старшая часть текущего значения скорости и знак передаются **байтом 2**. Адресс **байта 2** в последовательности байтов к **master** определяется значением **P14.03+2**.

В **таблице 5** указаны пределы текущей скорости двигателя и их представление в HEX и BIN форматах. Пределы соответствуют максимальной скорости двигателя.

Пределы	HEX формат	BIN формат	DEC формат
<b>NEGATIVE MAX</b>	<b>0xC800</b>	<b>1100 1000 0000 0000</b>	<b>- 14336</b>
<b>POSITIVE MAX</b>	<b>0x3800</b>	<b>0011 1000 0000 0000</b>	<b>+ 14336</b>

**Таблица 5** Пределы текущей скорости и их представление в HEX и BIN формат

➤ **текущее значение тока якоря [CURR ARM ACTUAL]**

Текущее значение скорости двигателя является 14 битовым целым числом без знака и передается двумя байтами. Это значение соответствует значению параметра **P01.04** и в DEC формате имеет одна цифра после запятой.

Младшая часть текущего значения тока передается **байтом 3**. Адрес **байта 3** в последовательности байтов к **master** определяется значением **P14.03+3**.

Старшая часть текущего значения тока передаются **байтом 4**. Адрес **байта 4** в последовательности байтов к **master** определяется значением **P14.03+4**.

### Пример 2

Пример относится для управления двумя преобразователями по протоколу ELL\_RS485\_N1 от управляющего устройства **master**.

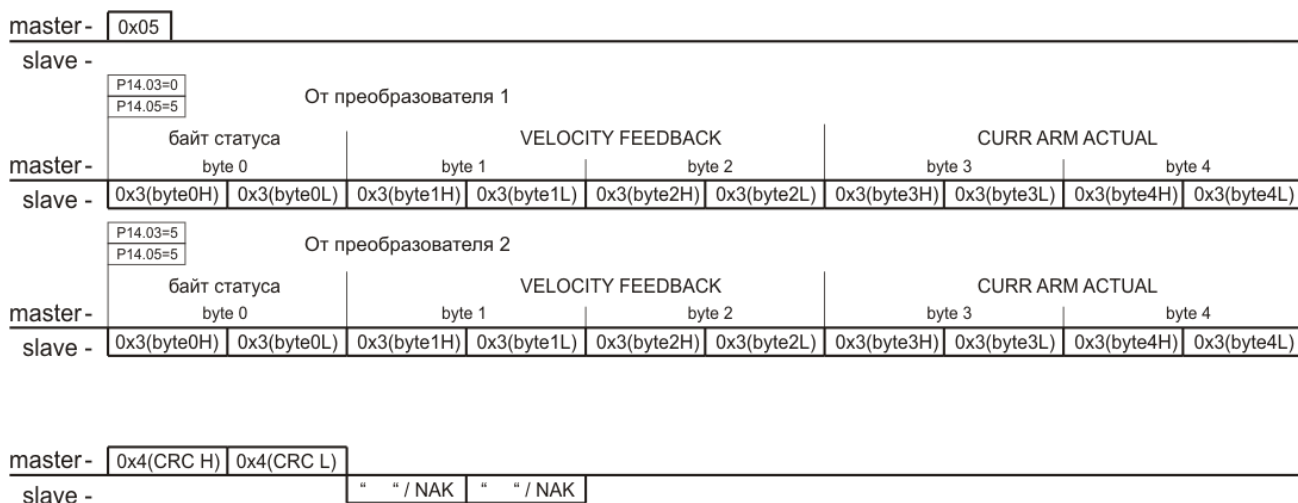
Настройка преобразователей для передачи данных по последовательной коммуникации к **master** устройству следующая:

- преобразователь 1: **P14.03 = 0** и **P14.05 = 5**;
- преобразователь 2: **P14.03 = 5** и **P14.05 = 5**.

**Master** устройство отправляет командный байт **0x05**. Первый преобразователь высылает 5 байтов данных (разделены в полубайтах т.е. 10 байтов со старшей частью 0x3X). После этого второй преобразователь высылает свои данные. **Master** устройство отправляет вычисленную контрольную сумму 1 байтом (разделенным в 2 полубайтах со старшей частью 0x4X). После этого преобразователи проверяют контрольную сумму с вычисленной ими и при несовпадении отправляют **NAK (0x00)** или ничего.

Преобразователи узнают когда по линии отправляются данные или контрольная сумма по старшим полубайтам.

На рисунке 5 показано отправление данных от преобразователей к **master** устройству. Первые 5 байта byte 0 ÷ byte 4 относятся для преобразователя 1, следующие 5 байта byte 0 ÷ byte 4 для преобразователя 2.



**Рисунок 5** Отправление данных от преобразователей к **master** устройству