



Электропривод

<https://electroprivod.ru>

Блок управления шаговым двигателем

Модели SMSD-4.2LAN и SMSD-8.0LAN
Протокол обмена данными – версия Modbus TCP/IP
Ver. 01

1. Основные сведения	3
2. Передача данных	3
3. Авторизация	4
4. Настройки параметров управления	4
4.1. Конфигурация двигателя	4
4.2. Настройки параметров работы привода	7
4.3. Контроль параметров работы	10
4.4. Входные/выходные сигналы	11
5. Чтение и запись пользовательской программы	11
6. Изменения в протоколе передачи при подключении по USB	14
Приложение А. Таблица регистров	15
Приложение Б. Исполнительные команды контроллера	19

1. Основные сведения

Блоки управления SMSD-4.2LAN и SMSD-8.0LAN, далее по тексту - контроллер, предназначены для управления шаговыми двигателями. Контроллер представляет собой электронное устройство, состоящее из корпуса, электронной платы, разъемов, передней панели на которой находятся органы управления и индикации.

В режиме работы по локальной сети Ethernet (на индикаторе «LA»), контроллер создаёт сокет для подключения к нему управляющей пользовательской программы или устройства, далее по тексту - пользователь. Данные передаются по физической линии Ethernet (протокол TCP).

Данное руководство относится к контроллерам со специальной прошивкой (доступна с мая 2023), обеспечивающей управление по протоколу Modbus TCP/IP. Протокол обмена с использованием USB остается неизменным за исключением структуры параметров передачи по LAN (см. описание протокола стандартной прошивки SMSD-4.2LAN и SMSD-8.0LAN).

2. Передача данных

Параметры подключения по сети Ethernet, установленные в контроллере по умолчанию:

- Node ID: 1
- MAC адрес: 0x01 0xF8 0xDC 0x3F 0x00 0x00
- IP адрес: 192.168.1.2
- Порт: 502
- Маска подсети: 255.255.0.0
- Основной шлюз: 192.168.1.1

Данные параметры в дальнейшем можно поменять как по сети Ethernet, так и через USB интерфейс.

Передача данных по LAN осуществляется в соответствии со стандартным протоколом Modbus TCP/IP (<https://modbus.org/>). Команды записи и чтения регистров передаются в соответствии с таблицей регистров – приложение А.

Структура пакета передачи:

MBAP Header				PDU	
Идентификатор транзакции	Идентификатор протокола (всегда 0 для Modbus)	Длина пакета	Node ID Адрес устройства	Код функции	Данные
2 байта	2 байта	2 байта	1 байт	1 байт	

Типы регистров и коды функций

Тип регистра	Размер регистра	Код функции		
		Чтение группы регистров	Запись одного регистра	Запись группы регистров
Discretes Input	1 бит	0x02	-	-
Coils	1 бит	0x01	0x05	0x0F
Input Registers	16 бит (слово)	0x04	-	-

Holding Registers	16 бит (слово)	0x03	0x06	0x10
-------------------	----------------	------	------	------

3. Авторизация

Доступ к данным контроллера защищен 64-битным паролем с таймаутом авторизации 1 секунда.

Перед началом работы с контроллером необходимо ввести и подтвердить пароль. Авторизация будет действительна для текущего подключения.

Порядок авторизации:

1. Записать значение пароля в регистры Holding Registers 0x2100 и 0x2102.
2. Записать значение 0 в регистр 0x2104
3. Считать результат авторизации из регистра Discrete Inputs 0x2200.

После успешной авторизации (значение регистра 0x2200 = TRUE) контроллер открывает доступ ко всем регистрам управления.

Адрес HEX	Тип	Размер бит	Тип данных	Имя	Описание
Авторизация					
2100	HR	32	UINT_HEX	Password_LOW32	Младшие 32 бита пароля (значение по умолчанию 0x89ABCDEF)
2102	HR	32	UINT_HEX	Password_HIGH32	Старшие 32 бита пароля (значение по умолчанию 0x01234567)
2104	HR	16	UINT_DEC	Password_CMD	Запись значения: 0 - попытка авторизации 1 - смена пароля
2200	DI	1	BOOL	Access	Отображение состояния авторизации FALSE - нет доступа TRUE - доступ разрешён

4. Настройки параметров управления

4.1. Конфигурация двигателя

К настройкам конфигурации двигателя относятся рабочий ток и ток удержания, режим дробления и тип управления (токовый или вольтовый). Параметры конфигурации можно изменять только при обесточенных фазах двигателя. Перед изменением параметров конфигурации необходимо убедиться, что двигатель находится в обесточенном (Hiz) состоянии (Discrete inputs 0x1200).

Адрес HEX	Тип	Размер бит	Тип данных	Имя	Описание
Конфигурация двигателя					
110A	HR	16	UINT_DEC	CURRENT_OR_VOLTAGE	Тип управления
110B	HR	16	UINT_DEC	MOTOR_TYPE	Модель двигателя для вольтового режима
110C	HR	16	UINT_HEX	MICROSTEPPING	Дробление шага
110D	HR	16	UINT_DEC	WORK_CURRENT	Рабочий ток для токового режима
110E	HR	16	UINT_DEC	STOP_CURRENT	Ток удержания

CURRENT_OR_VOLTAGE - тип управления двигателем:

0 – напряжение (вольтовый режим),

1 – ток (токовый режим)

MOTOR TYPE – модель двигателя для вольтового режима управления:

Значение		Макс. ток фазы, А	Сопротивление фазы, Ом	Индуктивность фазы, мГн	Угловой шаг	Модель
SMSD-4.2LAN	SMSD-8.0LAN					
0	0	-	-	-	-	нет
1	1	1.33	2.1	2.5	1.8	FL42STH33-1334 1.8 deg
2	2	1.33	2.1	4.2	0.9	FL42STH33-1334 0.9 deg
3	3	1.2	3.3	3.4	0.9	FL42STH38-1206 0.9 deg
4	4	1.68	1.65	3.2	1.8	FL42STH38-1684 1.8 deg
5	5	1.68	1.64	3.2	0.9	FL42STH38-1684 0.9 deg
6	6	1.2	3.3	2.8	0.8	FL42STH47-1206 1.8 deg
7	7	1.68	1.65	2.8	1.8	FL42STH47-1684 1.8 deg
8	8	1.68	1.65	4.1	0.9	FL42STH47-1684 0.9 deg
9	9	1.2	6	7	1.8	FL42STH60-1206 1.8 deg
10	10	1.2	12.1	36.7	0.9	FL42STH60-1206 0.9 deg
11	11	1.56	1.8	3.6	1.8	FL57ST41-1564 1.8 deg
12	12	1.0	16.7	46.5	1.8	FL57ST76-1006 1.8 deg
13	13	1.5	3.6	6	1.8	FL57ST76-1506 1.8 deg
14	14	1.0	5.7	5.4	1.8	FL57STH41-1006 1.8 deg
15	15	1.0	5.7	8	0.9	FL57STH41-1006 0.9 deg
16	16	2.8	0.7	1.4	1.8	FL57STH41-2804 1.8 deg
17	17	2.8	0.7	2.2	0.9	FL57STH41-2804 0.9 deg
18	18	1.0	6.6	8.6	1.8	FL57STH51-1006 1.8 deg
19	19	2.8	0.83	2.2	1.8	FL57STH51-2804 1.8 deg
20	20	2.8	0.9	3.7	0.9	FL57STH51-2804 0.9 deg
21	21	1.0	7.4	10	1.8	FL57STH56-1006 1.8 deg
22	22	2.0	1.8	2.5	1.8	FL57STH56-2006 1.8 deg
23	23	2.8	0.9	2.5	1.8	FL57STH56-2804 1.8 deg
24	24	1.0	8.6	14	1.8	FL57STH76-1006 1.8 deg
25	25	2.8	1.13	3.6	1.8	FL57STH76-2804 1.8 deg
26	26	2.8	1.13	5.6	0.9	FL57STH76-2804 0.9 deg
27	27	2.0	1.2	4.6	1.8	FL60STH65-2008 1.8 deg параллельное соединение обмоток
28	28	2.0	4.8	18.4	1.8	FL60STH65-2008 1.8 deg последовательное соединение обмоток
29	29	2.0	1.5	6.8	1.8	FL60STH86-2008 1.8 deg параллельное соединение обмоток
30	30	2.0	6	7.2	1.8	FL60STH86-2008 1.8 deg последовательное соединение обмоток
31	31	2.8	0.7	3.9	1.8	FL86STH65-2808 1.8 deg параллельное соединение обмоток
32	32	2.8	2.8	15.6	1.8	FL86STH65-2808 1.8 deg последовательное соединение обмоток
33	33	4.2	0,375	3.4	1.8	FL86STH80-4208 1.8 deg параллельное соединение обмоток
34	34	4.2	1.5	13.6	1.8	FL86STH80-4208 1.8 deg



						последовательное соединение обмоток
35	35	4.2	0.45	6	1.8	FL86STH118-4208 1.8 deg параллельное соединение обмоток
36	36	4.2	1.8	24	1.8	FL86STH118-4208 1.8 deg последовательное соединение обмоток
37	37	4.2	0,625	8	1.8	FL86STH156-4208 1.8 deg параллельное соединение обмоток
38	38	4.2	2.5	32	1.8	FL86STH156-4208 1.8 deg последовательное соединение обмоток
-	39	6.0	0.6	6.5	1.8	FL86STH118-6004 1.8 deg
-	40	6.2	0.75	9	1.8	FL86STH156-6204 1.8 deg
-	41	5.5	0.9	12	1.8	FL110STH99-5504 1.8 deg
-	42	6.5	0.8	15	1.8	FL110STH150-6504 1.8 deg
-	43	8	0.67	12	1.8	FL110STH201-8004 1.8 deg
39	44	0.3	32	40	1.8	ДШ3934-0,3А
40	45	0.67	8.5	7.5	1.8	ДШ2851-0,7А
41	46	1.68	2.3	3.4	1.8	ДШ4248-1,7А
42	47	3.0	1.0	3.4	1.8	ДШ5776-3,0А
43	48	3.0	1.45	6.5	1.8	ДШ57112-3,0А
44	49	3.0	1.2	6.4	1.8	ДШ8665-3,0А
45	50	4.5	0.36	3.0	1.8	ДШ8682-4,5А
-	51	6.0	0.6	5.7	1.8	ДШ86118-6,0А
-	52	6.2	0.7	8.5	1.8	ДШ86156-6,2А
-	53	8.0	0.8	16	1.8	ДШ110201-8,0А
-	54	6.0	0.8	8.7	1.8	ДШ130280-6,0А

MICROSTEPPING – дробление шага:

- 0 - 1
- 1 - 1/2
- 2 - 1/4
- 3 - 1/8
- 4 - 1/16
- 5 - 1/32
- 6 - 1/64
- 7 - 1/128

WORK_CURRENT – рабочий ток (используется в токовом режиме управления двигателем). Значение рабочего тока определяется по формуле: $0.1A \cdot \text{Значение параметра}$; $1 \leq \text{Значение} \leq 80$. Значения тока могут быть следующие:

1 - 0.1A	15 - 1.5A	29 - 2.9A	43 - 4.3A	57 - 5.7A	71 - 7.1A
2 - 0.2A	16 - 1.6A	30 - 3.0A	44 - 4.4A	58 - 5.8A	72 - 7.2A
3 - 0.3A	17 - 1.7A	31 - 3.1A	45 - 4.5A	59 - 5.9A	73 - 7.3A
4 - 0.4A	18 - 1.8A	32 - 3.2A	46 - 4.6A	60 - 6.0A	74 - 7.4A
5 - 0.5A	19 - 1.9A	33 - 3.3A	47 - 4.7A	61 - 6.1A	75 - 7.5A
6 - 0.6A	20 - 2.0A	34 - 3.4A	48 - 4.8A	62 - 6.2A	76 - 7.6A
7 - 0.7A	21 - 2.1A	35 - 3.5A	49 - 4.9A	63 - 6.3A	77 - 7.7A
8 - 0.8A	22 - 2.2A	36 - 3.6A	50 - 5.0A	64 - 6.4A	78 - 7.8A
9 - 0.9A	23 - 2.3A	37 - 3.7A	51 - 5.1A	65 - 6.5A	79 - 7.9A
10 - 1.0A	24 - 2.4A	38 - 3.8A	52 - 5.2A	66 - 6.6A	80 - 8.0A
11 - 1.1A	25 - 2.5A	39 - 3.9A	53 - 5.3A	67 - 6.7A	
12 - 1.2A	26 - 2.6A	40 - 4.0A	54 - 5.4A	68 - 6.8A	
13 - 1.3A	27 - 2.7A	41 - 4.1A	55 - 5.5A	69 - 6.9A	
14 - 1.4A	28 - 2.8A	42 - 4.2A	56 - 5.6A	70 - 7.0A	

Допустимый диапазон значений для блоков SMSD-4.2LAN: 1 – 42; для блоков SMSD-8.0LAN: 1 – 80.

STOP CURRENT – ток удержания – устанавливается в процентах от значения рабочего тока:

- 0 - 25%
- 1 - 50%
- 2 - 75%
- 3 - 100%

4.2. Настройки параметров работы привода

Параметры работы привода могут быть считаны или изменены как с помощью соответствующих регистров командами по Modbus, так и программно в режиме выполнения пользовательской программы.

Адрес HEX	Тип	Размер бит	Тип данных	Имя	Описание
Управление					
1100	HR	16	UINT_DEC	MIN_SPEED	Минимальная скорость двигателя
1101	HR	16	UINT_DEC	MAX_SPEED	Максимальная скорость двигателя
1102	HR	16	UINT_DEC	ACC	Ускорение
1103	HR	16	UINT_DEC	DEC	Замедление
1104	HR	16	UINT_DEC	FS_SPEED	Скорость перехода в полношаговый режим
1105	HR	16	UINT_DEC	TARGET_SPEED	Заданная скорость
1106	HR	32	INT_DEC	TARGET_POS	Заданное положение
1108	HR	16	UINT_DEC	TARGET_INPUT	Номер входа
1109	HR	16	UINT_HEX	CMD	Код команды
1400	DO	1	BOOL	CLR	Сброс флагов ошибок
1401	DO	1	BOOL	Reset Pos	Обнуление счетчика текущего положения
1402	DO	1	BOOL	Reset powerSTEP01	Сброс модуля управления шаговым двигателем
1403	DO	1	BOOL	Soft STOP	Плавная остановка с заданным замедлением и переход в режим удержания
1404	DO	1	BOOL	Hard STOP	Резкая остановка и переход в режим удержания
1405	DO	1	BOOL	Soft HiZ	Плавная остановка с заданным замедлением и обесточивание двигателя
1406	DO	1	BOOL	Hard HiZ	Резкая остановка и обесточивание двигателя

MIN_SPEED – заданное значение минимальной скорости шагового двигателя. Допустимый диапазон установки от 0 до 950 шагов/сек. Важно: все значения скоростей задаются в полных шагах с секунду независимо от установленного режима дробления шага.

MAX_SPEED – заданное значение максимальной скорости шагового двигателя. Допустимый диапазон установки от 16 до 15600 шагов/сек. Важно: все значения скоростей задаются в полных шагах с секунду независимо от установленного режима дробления шага.

ACC – заданное значение ускорения шагового двигателя. Допустимый диапазон установки от 15 до 59000 шагов/сек².

DEC – заданное значение замедления шагового двигателя. Допустимый диапазон установки от 15 до 59000 шагов/сек².

FS_SPEED – заданное значение скорости перехода на полношаговый режим работы шагового двигателя. Допустимый диапазон установки от 15 до 15600 шагов/сек. Важно: все значения скоростей задаются в полных шагах с секунду независимо от установленного режима дробления шага.



TARGET_SPEED – заданное значение скорости вращения вала шагового двигателя. Допустимый диапазон установки от 16 до 15600 шагов/сек. Важно: все значения скоростей задаются в полных шагах с секунду независимо от установленного режима дробления шага.

TARGET_POS – заданное значение величины перемещения или заданная координата для позиционирования. Допустимый диапазон установки от -2 097 152 до + 2 097 151 шагов (микрошагов).

TARGET_INPUT – Заданный номер входа для команд перемещения с условиями по входным сигналам.

CMD – код команды. Запись в регистр значения (см. ниже) приведёт к началу выполнения команды, параметризованной в регистрах 0x1100...0x1110.

Допустимые значения для записи в регистр CMD:

0x0e - RUN_F – непрерывное движение в прямом направлении с постоянной заданной скоростью (TARGET_SPEED).

0x0f - RUN_R – непрерывное движение в реверсном направлении с постоянной заданной скоростью (TARGET_SPEED).

0x10 - MOVE_F – смещение на заданную величину (TARGET_POS) в прямом направлении. Скорость перемещения определяется предварительно заданными параметрами минимальной и максимальной скорости, ускорения и замедления. Перед отправкой данной команды двигатель должен быть остановлен.

0x11 - MOVE_R – смещение на заданную величину (TARGET_POS) в реверсном направлении. Скорость перемещения определяется предварительно заданными параметрами минимальной и максимальной скорости, ускорения и замедления. Перед отправкой данной команды двигатель должен быть остановлен.

0x12 - GO_TO_F – перемещение в заданную координату (TARGET_POS) в прямом направлении. Скорость перемещения определяется предварительно заданными параметрами минимальной и максимальной скорости, ускорения и замедления.

0x13 - GO_TO_R – перемещение в заданную координату (TARGET_POS) в реверсном направлении. Скорость перемещения определяется предварительно заданными параметрами минимальной и максимальной скорости, ускорения и замедления.

0x14 - GO_UNTIL_F – непрерывное движение в прямом направлении на максимальной скорости до поступления сигнала на заданный вход (TARGET_INPUT). После получения сигнала двигатель останавливается с заданным замедлением. При отработке команды учитывается заданная маска сигнала.

0x15 - GO_UNTIL_R – непрерывное движение в реверсном направлении на максимальной скорости до поступления сигнала на заданный вход (TARGET_INPUT). После получения сигнала двигатель останавливается с заданным замедлением. При отработке команды учитывается заданная маска сигнала.

0x16 - SCAN_ZERO_F – поиск нулевого положения в прямом направлении с заданной скоростью (TARGET_SPEED). Движение продолжается до поступления сигнала на вход SET_ZERO. При поступлении сигнала двигатель останавливается, текущее положение принимается за нулевое.

0x17 - SCAN_ZERO_R – поиск нулевого положения в реверсном направлении с заданной скоростью (TARGET_SPEED). Движение продолжается до поступления сигнала на вход SET_ZERO. При поступлении сигнала двигатель останавливается, текущее положение принимается за нулевое.



0x18 - SCAN_MARK_F – поиск положения метки в прямом направлении с заданной скоростью (TARGET_SPEED). Движение продолжается до поступления сигнала на вход IN1. При поступлении сигнала двигатель останавливается, текущее положение запоминается как метка.

0x19 - SCAN_MARK_R – поиск положения метки в реверсном направлении с заданной скоростью (TARGET_SPEED). Движение продолжается до поступления сигнала на вход IN1. При поступлении сигнала двигатель останавливается, текущее положение запоминается как метка.

0x1a - GO_ZERO – перемещение в нулевое положение.

0x1b - GO_MARK – перемещение в положение, которое было отмечено как метка.

0x1c - GO_TO – перемещение в заданное положение (TARGET_POS) по кратчайшему пути.

0x1f - SOFT_STOP – плавная остановка шагового двигателя с заданным замедлением (DEC). После остановки двигатель удерживает положение с заданным током удержания (STOP_CURRENT).

0x20 - HARD_STOP – резкая остановка шагового двигателя. После остановки двигатель удерживает положение с заданным током удержания (STOP_CURRENT).

0x21 - SOFT_HI_Z – плавная остановка шагового двигателя с заданным замедлением (DEC). После остановки питание с обмоток двигателя снимается.

0x22 - HARD_HI_Z – резкая остановка и обесточивание обмоток шагового двигателя.

0x2f - START_PROGRAM_MEM0 – старт выполнения программы, записанной в область памяти 0 контроллера.

0x30 - START_PROGRAM_MEM1 – старт выполнения программы, записанной в область памяти 1 контроллера

0x31 - START_PROGRAM_MEM2 – старт выполнения программы, записанной в область памяти 2 контроллера

0x32 - START_PROGRAM_MEM3 – старт выполнения программы, записанной в область памяти 3 контроллера

0x33 - STOP_PROGRAM_MEM – остановка выполнения программы, записанной память контроллера.

0x3D - SCAN_MARK2_F – поиск положения метки в прямом направлении. Движение продолжается до поступления сигнала на вход IN1. При поступлении сигнала двигатель останавливается с заданным значением торможения, текущее положение запоминается как метка.

0x3E - SCAN_MARK2_R – – поиск положения метки в реверсном направлении. Движение продолжается до поступления сигнала на вход IN1. При поступлении сигнала двигатель останавливается с заданным значением торможения, текущее положение запоминается как метка.

CLR – Coil – запись TRUE в регистр - сброс всех флагов ошибок

Reset Pos – Coil – запись TRUE в регистр – обнуление счетчика текущего положения. После выполнения команды текущее положение принимается как нулевое.

Reset powerSTEP01 – Coil – запись TRUE в регистр - полный аппаратный и программный сброс модуля управления шаговым двигателем, но не контроллера в целом.

Soft STOP – Coil – запись TRUE в регистр – плавная остановка шагового двигателя с заданным замедлением (DEC). После остановки двигатель удерживает положение с заданным током удержания (STOP_CURRENT)

Hard STOP – Coil – запись TRUE в регистр - резкая остановка шагового двигателя. После остановки двигатель удерживает положение с заданным током удержания (STOP_CURRENT).

Soft HiZ – Coil – запись TRUE в регистр - плавная остановка шагового двигателя с заданным замедлением (DEC). После остановки питание с обмоток двигателя снимается.

Hard HiZ – Coil – запись TRUE в регистр - резкая остановка и обесточивание обмоток шагового двигателя.

4.3. Контроль параметров работы

Параметры работы могут быть считаны с помощью соответствующих регистров командами по Modbus.

Адрес HEX	Тип	Размер бит	Тип данных	Имя	Описание
Информация о текущем состоянии двигателя					
1000	IR	16	UINT_DEC	SPEED	Текущая скорость двигателя
1001	IR	32	INT_DEC	ABS_POS	Текущее положение
1003	IR	16	UINT_HEX	EL_POS	Электрическое положение ротора
1004	IR	16	UINT_HEX	STATUS	Текущий статус контроллера
1200	DI	1	BOOL	HiZ	Питание фаз двигателя
1201	DI	1	BOOL	STOP	Двигатель остановлен
1202	DI	1	BOOL	CONST_SPEED	Двигатель вращается с постоянной скоростью
1203	DI	1	BOOL	ACC	Разгон двигателя
1204	DI	1	BOOL	DEC	Торможение двигателя
1205	DI	1	BOOL	READY	Готовность выполнения следующей задачи
1206	DI	1	BOOL	SW_F	Функция SW
1207	DI	1	BOOL	SW_EVN	флаг события SW
1208	DI	1	BOOL	DIR	направление вращения
1209	DI	1	BOOL	CMD_ERROR	ошибка выполнения команды
Информация о ходе выполнения программы					
3000	IR	16	UINT_DEC	MODE_N_PROGRAMS	Номер программы, выполнение которой задал пользователь
3200	DI	1	BOOL	PROGRAM_RUN	Флаг выполнения программы
3001	IR	16	UINT_DEC	N_PROGRAM	Номер программы, которая выполняется в данный момент
3002	IR	16	UINT_DEC	N_COMMAND	Номер строки программы, которая выполняется в данный момент

SPEED – текущая скорость шагового двигателя

ABS_POS – Текущее положение шагового двигателя

EL_POS – Информацию о текущем электрическом положении ротора: биты 8,7 – текущий шаг, биты 6..0 – текущий микрошаг в пределах полного шага (измеряется как 1/128 от величина полного шага).

HiZ – информация о состоянии фаз шагового двигателя (запитаны или обесточены).

STOP – флаг остановки шагового двигателя

CONST_SPEED – флаг движения с постоянной скоростью

ACC – флаг движения с ускорением

DEC – флаг замедления шагового двигателя

READY – флаг готовности к выполнению следующей задачи.

SW_F – Флаг функции SW: 1- функция SW включена, 0 – функция SW выключена

SW_EVN – флаг события SW если 1- событие наступило, 0 – событие не наступило

DIR – Флаг направления вращения шагового двигателя

CMD_ERROR – флаг ошибки выполнения команды: 1- ошибка выполнения команды, 0 – без ошибок

MODE_N_PROGRAMS – Номер программы, выполнение которой задал пользователь для режима автономной работы (bF) через переднюю панель контроллера.

PROGRAM_RUN – Флаг выполнения программы

N_PROGRAM – Номер программы, которая выполняется в данный момент

N_COMMAND – Номер строки программы, которая выполняется в данный момент

4.4. Входные/выходные сигналы

Адрес HEX	Тип	Размер бит	Тип данных	Имя	Описание
Входы/выходы					
1005	IR	16	UINT_HEX	Inputs	Состояние входов, состояние и установка маски и ожидания появления сигнала по входам
110F	HR	16	UINT_HEX	Mask	
1110	HR	16	UINT_HEX	Wait	
1407	DO	1	BOOL	Relay	Управление релейным выходом

Inputs – состояние входов

Mask – маска входных сигналов. При выполнении команд с условиями по входам учитывается маска сигналов. Если маска соответствующего сигнала равна 1, команда выполняется.

Wait - ожидание появления сигнала по входам

Relay – coil – управление реле.

5. Чтение и запись пользовательской программы

Контроллер позволяет создавать пользовательские программы и записывать их в энергонезависимую память. Контроллер имеет 4 области памяти для хранения программ, по 255 команд каждая. Чтение и запись программы возможны блоками – до 32 строк программы за один раз. Каждая команда при чтении и записи занимает 2 регистра Modbus (4 байта памяти) и имеет вид, соответствующий основному (стандартному) протоколу передачи контроллеров SMSD-LAN):

```
typedef struct
{
    uint32_t RESERVE :4;
    uint32_t COMMAND :6;
    uint32_t DATA :22;
} SMSD_CMD_Type;
```

Первые 4 бита являются зарезервированными и содержат 0 при чтении/записи строк программы. Следующие 6 бит содержат код команды в соответствии с приложением Б. Последние 22 бита содержат данные команды.

Команда установки максимальной скорости SET_MAX_SPEED в соответствии с приложением Б имеет код 0x06, в двоичном виде b1100. Данные команды 100 в двоичном виде b1100100. При записи в структуру SMSD_CMD_Type получается значение b11001000001100000 = d102496 = 0x00019060. Таким образом, команда установки максимальной скорости 100 шагов/сек при чтении/записи программы имеет вид 0x00019060.

Байт 3										Байт 2										Байт 1										Байт 0														
Биты 0..7										Биты 0..7										Биты 2..7										Биты 0..1					Биты 4..7					Биты 0..3				
Данные команды = 100 = b1100100																				Код команды = 0x06 = b110										0 (резерв)														
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0												

Адрес HEX	Тип	Размер бит	Тип данных	Имя	Описание
Программирование					
3100	HR	32	UINT_HEX	CMD 00	Буфер команд для записи/чтения программы
3102	HR	32	UINT_HEX	CMD 01	
3104	HR	32	UINT_HEX	CMD 02	
...	HR	32	UINT_HEX	...	
313E	HR	32	UINT_HEX	CMD 31	
3180	HR	16	UINT_DEC	N_PROG	Номер программы (от 0 до 3).
3181	HR	16	UINT_DEC	N_STR	Стартовый адрес строки чтения/записи памяти программы
3182	HR	16	UINT_DEC	SECTOR_SIZE	Количество команд, которые будут записаны в память из буфера или считаны из памяти в буфер
3183	HR	16	UINT_DEC	PROG_SIZE	Итоговый размер программы
3184	HR	16	UINT_DEC	MEM_CMD	Запись значения в регистр запустит процедуру: 0 - чтение 1 - запись 2 - стирание (PROG_SIZE в том числе) 3 - чтение размера программы в PROG_SIZE 4 - запись размера программы из PROG_SIZE
3400	DO	1	BOOL	B_COMPLETE	Флаг завершения выполнения процедуры. Сбрасывается вручную.

Порядок действий для записи программы из N команд в память контроллера:

1. Записать номер области памяти для записи программы – регистр N_PROG
2. Сбросить флаг B_COMPLETE
3. Выполнить команду стирания памяти (MEM_CMD = 2)
4. Дождаться установки флага завершения выполнения команды B_COMPLETE



5. Сбросить флаг B_COMPLETE
6. Разделить N команд для записи на n групп, в каждой по X команд ($x \leq 32$, так как размер буфера для записи = 32 строки). Для каждой группы из X команд выполнить следующие действия:
 - 6.1. Установить номер строки программы для начала записи (регистр N_STR). Первая группа команд записывается в строку 0 (если предполагается запись программы в начало памяти). Для каждой следующей группы это значение увеличивается на X.
 - 6.2. Установить размер сектора для записи (регистр SECTOR_SIZE) – количество команд для записи за один раз ($SECTOR_SIZE = X$).
 - 6.3. Записать X команд в буфер передачи (регистры CMD_00...CMD_31) – одна строка программы занимает 2 слова данных.
 - 6.4. Выполнить команду записи буфера команд в память ($MEM_CMD = 1$)
 - 6.5. Дождаться установки флага завершения выполнения команды B_COMPLETE
 - 6.6. Сбросить флаг B_COMPLETE
7. Записать значение полного размера программы в регистр PROG_SIZE = N
8. Выполнить команду записи размера программы ($MEM_CMD = 4$)
9. Дождаться установки флага завершения выполнения команды B_COMPLETE
10. Сбросить флаг B_COMPLETE

Порядок действий для чтения программы из памяти контроллера:

1. Записать номер области памяти для чтения программы – регистр N_PROG
2. Сбросить флаг B_COMPLETE
3. Выполнить команду чтения полного размера программы ($MEM_CMD = 3$)
4. Дождаться установки флага завершения выполнения команды B_COMPLETE
5. Сбросить флаг B_COMPLETE
6. Считать значение N - полный размер программы из регистра PROG_SIZE
7. Общее количество команд N разделить на n групп по X команд ($x \leq 32$, так как размер буфера для записи = 32 строки). Для каждой группы из X команд выполнить следующие действия:
 - 7.1. Установить номер строки программы для начала чтения (регистр N_STR). Первая группа команд начинается со строки 0 (если предполагается чтение с начала программы). Для каждой следующей группы это значение увеличивается на X.
 - 7.2. Установить размер сектора для чтения (регистр SECTOR_SIZE) – количество команд для чтения за один раз ($SECTOR_SIZE = X$).
 - 7.3. Выполнить команду чтения в буфер ($MEM_CMD = 0$)
 - 7.4. Дождаться установки флага завершения выполнения команды B_COMPLETE
 - 7.5. Сбросить флаг B_COMPLETE
 - 7.6. Считать X строк программы из буфера (регистры CMD_00...CMD_31) – одна строка программы занимает 2 слова данных.



6. Изменения в протоколе передачи при подключении по USB

При подключении по USB сохраняется стандартная версия протокола обмена за исключением чтения и установки параметров передачи по LAN.

В структуру SMSD_LAN_Config_Type добавлено поле unitID.

```
typedef struct
{
    uint8_t mac[6];
    uint8_t ip[4];
    uint8_t sn[4];
    uint8_t gw[4];
    uint8_t dns[4];
    uint16_t Port;
    dhcp_mode dhcp;
    uint8_t unitID;
} SMSD_LAN_Config_Type;
```

Значения по умолчанию:

```
{
    .mac= {0x00, 0xf8, 0xdc, 0x3f, 0x00, 0x00},
    .ip = {192, 168, 1, 2},
    .sn = {255, 255, 0, 0},
    .gw = {192, 168, 1, 1},
    .dns= {0, 0, 0, 0},
    .Port = 502,
    .dhcp = 1
    .unitID = 1
};
```

Приложение А. Таблица регистров

Карта регистров контроллеров SMSD-4.2LAN и SMSD-8.0LAN – версия Modbus TCP:

Адрес HEX	Тип	Размер бит	Тип данных	Имя	Описание
Авторизация					
2100	HR	32	UINT_HEX	Password_LOW32	Младшие 32 бита пароля (значение по умолчанию 0x89ABCDEF)
2102	HR	32	UINT_HEX	Password_HIGH32	Старшие 32 бита пароля (значение по умолчанию 0x01234567)
2104	HR	16	UINT_DEC	Password_CMD	Запись значения: 0 - попытка авторизации 1 - смена пароля
2200	DI	1	BOOL	Access	Отображение состояния авторизации FALSE - нет доступа TRUE - доступ разрешён
Конфигурация двигателя					
110A	HR	16	UINT_DEC	CURRENT_OR_VOLTAGE	Тип управления
110B	HR	16	UINT_DEC	MOTOR_TYPE	Модель двигателя для вольтового режима
110C	HR	16	UINT_HEX	MICROSTEPPING	Дробление шага
110D	HR	16	UINT_DEC	WORK_CURRENT	Рабочий ток для токового режима
110E	HR	16	UINT_DEC	STOP_CURRENT	Ток удержания
Управление					
1000	IR	16	UINT_DEC	SPEED	Текущая скорость двигателя
1001	IR	32	INT_DEC	ABS_POS	Текущее положение
1003	IR	16	UINT_HEX	EL_POS	Электрическое положение ротора
1100	HR	16	UINT_DEC	MIN_SPEED	Минимальная скорость двигателя
1101	HR	16	UINT_DEC	MAX_SPEED	Максимальная скорость двигателя
1102	HR	16	UINT_DEC	ACC	Ускорение
1103	HR	16	UINT_DEC	DEC	Замедление
1104	HR	16	UINT_DEC	FS_SPEED	Скорость перехода в полношаговый режим
1105	HR	16	UINT_DEC	TARGET_SPEED	Заданная скорость
1106	HR	32	INT_DEC	TARGET_POS	Заданное положение
1108	HR	16	UINT_DEC	TARGET_INPUT	Номер входа
1109	HR	16	UINT_HEX	CMD	Код команды
1004	IR	16	UINT_HEX	STATUS	Текущий статус контроллера
1200	DI	1	BOOL	HiZ	Питание фаз двигателя
1201	DI	1	BOOL	STOP	Двигатель остановлен
1202	DI	1	BOOL	CONST_SPEED	Двигатель вращается с постоянной скоростью
1203	DI	1	BOOL	ACC	Разгон двигателя
1204	DI	1	BOOL	DEC	Торможение двигателя
1205	DI	1	BOOL	READY	Готовность выполнения следующей задачи
1206	DI	1	BOOL	SW_F	Функция SW
1207	DI	1	BOOL	SW_EVN	Флаг события SW
1208	DI	1	BOOL	DIR	Направление вращения
1209	DI	1	BOOL	CMD_ERROR	Флаг ошибки выполнения команды
1400	DO	1	BOOL	CLR	Сброс флагов ошибок
1401	DO	1	BOOL	Reset Pos	Обнуление счетчика текущего положения
1402	DO	1	BOOL	Reset powerSTEP01	Сброс модуля управления шаговым двигателем
1403	DO	1	BOOL	Soft STOP	Плавная остановка с заданным

Адрес HEX	Тип	Размер бит	Тип данных	Имя	Описание
					замедлением и переход в режим удержания
1404	DO	1	BOOL	Hard STOP	Резкая остановка и переход в режим удержания
1405	DO	1	BOOL	Soft HiZ	Плавная остановка с заданным замедлением и обесточивание двигателя
1406	DO	1	BOOL	Hard HiZ	Резкая остановка и обесточивание двигателя
Входы/выходы					
1005	IR	16	UINT_HEX	Inputs	Состояние входов, состояние и установка маски и ожидания появления сигнала по входам
110F	HR	16	UINT_HEX	Mask	
1110	HR	16	UINT_HEX	Wait	
1407	DO	1	BOOL	Relay	Управление релейным выходом
Информация о ходе выполнения программы					
3000	IR	16	UINT_DEC	MODE_N_PROGRAMS	Номер программы, выполнение которой задал пользователь
3200	DI	1	BOOL	PROGRAM_RUN	Флаг выполнения программы
3001	IR	16	UINT_DEC	N_PROGRAM	Номер программы, которая выполняется в данный момент
3002	IR	16	UINT_DEC	N_COMMAND	Номер строки программы, которая выполняется в данный момент
Программирование					
3100	HR	32	UINT_HEX	CMD 00	Буфер команд для записи/чтения программы
3102	HR	32	UINT_HEX	CMD 01	
3104	HR	32	UINT_HEX	CMD 02	
3106	HR	32	UINT_HEX	CMD 03	
3108	HR	32	UINT_HEX	CMD 04	
310A	HR	32	UINT_HEX	CMD 05	
310C	HR	32	UINT_HEX	CMD 06	
310E	HR	32	UINT_HEX	CMD 07	
3110	HR	32	UINT_HEX	CMD 08	
3112	HR	32	UINT_HEX	CMD 09	
3114	HR	32	UINT_HEX	CMD 10	
3116	HR	32	UINT_HEX	CMD 11	
3118	HR	32	UINT_HEX	CMD 12	
311A	HR	32	UINT_HEX	CMD 13	
311C	HR	32	UINT_HEX	CMD 14	
311E	HR	32	UINT_HEX	CMD 15	
3120	HR	32	UINT_HEX	CMD 16	
3122	HR	32	UINT_HEX	CMD 17	
3124	HR	32	UINT_HEX	CMD 18	
3126	HR	32	UINT_HEX	CMD 19	
3128	HR	32	UINT_HEX	CMD 20	
312A	HR	32	UINT_HEX	CMD 21	
312C	HR	32	UINT_HEX	CMD 22	
312E	HR	32	UINT_HEX	CMD 23	
3130	HR	32	UINT_HEX	CMD 24	
3132	HR	32	UINT_HEX	CMD 25	
3134	HR	32	UINT_HEX	CMD 26	
3136	HR	32	UINT_HEX	CMD 27	
3138	HR	32	UINT_HEX	CMD 28	
313A	HR	32	UINT_HEX	CMD 29	
313C	HR	32	UINT_HEX	CMD 30	
313E	HR	32	UINT_HEX	CMD 31	



Адрес HEX	Тип	Размер бит	Тип данных	Имя	Описание
3180	HR	16	UINT_DEC	N_PROG	Номер программы
3181	HR	16	UINT_DEC	N_STR	Стартовый адрес строки чтения/записи памяти программы
3182	HR	16	UINT_DEC	SECTOR_SIZE	Количество команд, которые будут записаны в память из буфера или считаны из памяти в буфер
3183	HR	16	UINT_DEC	PROG_SIZE	Итоговый размер программы
3184	HR	16	UINT_DEC	MEM_CMD	Запись значения в регистр запустит процедуру: 0 - чтение 1 - запись 2 - стирание (PROG_SIZE в том числе) 3 - чтение размера программы в PROG_SIZE 4 - запись размера программы из PROG_SIZE
3400	DO	1	BOOL	B_COMPLETE	Флаг завершения выполнения процедуры. Сбрасывается вручную.
Настройка сети					
7000	HR	16	UINT_DEC	ID	Идентификатор устройства в сети Modbus TCP
7001	HR	16	UINT_HEX	MAC 0	MAC адрес устройства По умолчанию: 0x00 0xF8 0xDC 0x3F 0x00 0x00
7002	HR	16	UINT_HEX	MAC 1	
7003	HR	16	UINT_HEX	MAC 2	
7004	HR	16	UINT_HEX	MAC 3	
7005	HR	16	UINT_HEX	MAC 4	
7006	HR	16	UINT_HEX	MAC 5	IP адрес устройства По умолчанию: 192.168.1.2
7007	HR	16	UINT_DEC	IP 0	
7008	HR	16	UINT_DEC	IP 1	
7009	HR	16	UINT_DEC	IP 2	
700A	HR	16	UINT_DEC	IP 3	Маска подсети По умолчанию: 255.255.0.0
700B	HR	16	UINT_DEC	Subnet Mask 0	
700C	HR	16	UINT_DEC	Subnet Mask 1	
700D	HR	16	UINT_DEC	Subnet Mask 2	
700E	HR	16	UINT_DEC	Subnet Mask 3	Основной шлюз По умолчанию: 192.168.1.1
700F	HR	16	UINT_DEC	Gateway IP 0	
7010	HR	16	UINT_DEC	Gateway IP 1	
7011	HR	16	UINT_DEC	Gateway IP 2	
7012	HR	16	UINT_DEC	Gateway IP 3	DNS сервер По умолчанию: 0.0.0.0
7013	HR	16	UINT_DEC	DNS server IP Address 0	
7014	HR	16	UINT_DEC	DNS server IP Address 1	
7015	HR	16	UINT_DEC	DNS server IP Address 2	
7016	HR	16	UINT_DEC	DNS server IP Address 3	Порт По умолчанию: 502
7017	HR	16	UINT_DEC	Port	
7018	HR	16	UINT_DEC	DHCP	Динамическая настройка адреса 1 – Статический (по умолчанию) 2 – DHCP
7400	DO	1	BOOL	Apply Network Configuration	Применить и сохранить настройки
Регистры версии и идентификации					
8001	IR	16	UINT_DEC	HW_MAJOR	Идентификатор устройства: 6 - SMSD-4.2ModbusTCP 7 - SMSD-8.0ModbusTCP
8002	IR	16	UINT_DEC	HW_VER	Версия аппаратной части: 1



Адрес HEX	Тип	Размер бит	Тип данных	Имя	Описание
8003	IR	16	UINT_DEC	FW_MAJOR	Версия прошивки: 1.3
8004	IR	16	UINT_DEC	FW_MINOR	
8007	IR	16	UINT_DEC	RM_MAJOR	Версия карты регистров: 4.3
8008	IR	16	UINT_DEC	RM_MINOR	

Исполнительная команда **CMD_PowerSTEP01_SET_FS_SPEED**

Исполнительная команда **CMD_PowerSTEP01_SET_FS_SPEED** = 0x09 предназначена для установки скорости перехода на полношаговый режим работы. Поле DATA должно содержать скорость перехода в полношаговый режим в диапазоне от 15 до 15600 шагов/сек. Внимание: в командах установки скорости единицы измерения параметра - полные шаги в секунду.

Исполнительная команда **CMD_PowerSTEP01_SET_MASK_EVENT**

Исполнительная команда **CMD_PowerSTEP01_SET_MASK_EVENT** = 0x0A предназначена для маскирования входных сигналов. В случае, если значение маски сигнала установлено 1 – контроллер обрабатывает сигналы на соответствующем физическом входе. Если значение маски сигнала установлено 0 – контроллер не обрабатывает сигналы на соответствующем физическом входе.

Битовая раскладка поля данных:

21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	Mask_7	Mask_6	Mask_5	Mask_4	Mask_3	Mask_2	Mask_1	Mask_0

Mask_X – Маскирование сигнала на входе X.

Исполнительная команда **CMD_PowerSTEP01_RUN_F**

Исполнительная команда **CMD_PowerSTEP01_RUN_F** = 0x0E предназначена для старта непрерывного вращения двигателя в прямом направлении с указанной скоростью. Поле DATA содержит значение скорости вращения в диапазоне от 15 до 15600 шагов/сек. Внимание: в командах установки скорости единицы измерения параметра - полные шаги в секунду.

Исполнительная команда **CMD_PowerSTEP01_RUN_R**

Исполнительная команда **CMD_PowerSTEP01_RUN_R** = 0x0E предназначена для старта непрерывного вращения двигателя в обратном направлении с указанной скоростью. Поле DATA содержит значение скорости вращения в диапазоне от 15 до 15600 шагов/сек. Внимание: в командах установки скорости единицы измерения параметра - полные шаги в секунду.

Исполнительная команда **CMD_PowerSTEP01_MOVE_F**

Исполнительная команда **CMD_PowerSTEP01_MOVE_F** = 0x10 предназначена для перемещения двигателя в прямом направлении на указанную величину. Поле DATA должно содержать величину перемещения в диапазоне $-(2^{21}) \dots + (2^{21}-1)$. Скорость перемещения определяется предварительно заданными параметрами минимальной скорости, максимальной скорости, ускорения и замедления. Перед отправкой данной команды двигатель должен быть остановлен (поле **Mot_Status** структуры **powerSTEP_STATUS_Type** = 0).

Исполнительная команда **CMD_PowerSTEP01_MOVE_R**

Исполнительная команда **CMD_PowerSTEP01_MOVE_R** = 0x10 предназначена для перемещения двигателя в обратном направлении на указанную величину. Поле DATA должно содержать величину перемещения в диапазоне $-(2^{21}) \dots + (2^{21}-1)$. Скорость перемещения определяется предварительно заданными параметрами минимальной скорости, максимальной скорости, ускорения и замедления. Перед отправкой данной команды двигатель должен быть остановлен (поле **Mot_Status** структуры **powerSTEP_STATUS_Type** = 0).



Исполнительная команда **CMD_PowerSTEP01_GO_TO_F**

Исполнительная команда **CMD_PowerSTEP01_GO_TO_F** = 0x12 предназначена для перемещения в заданную позицию в прямом направлении. Поле DATA должно содержать требуемое положение двигателя в диапазоне $-(2^{21}) \dots +(2^{21}-1)$. Скорость перемещения определяется предварительно заданными параметрами минимальной скорости, максимальной скорости, ускорения и замедления.

Исполнительная команда **CMD_PowerSTEP01_GO_TO_R**

Исполнительная команда **CMD_PowerSTEP01_GO_TO_R** = 0x13 для перемещения в заданную позицию в обратном направлении. Поле DATA должно содержать требуемое положение двигателя в диапазоне $-(2^{21}) \dots +(2^{21}-1)$. Скорость перемещения определяется предварительно заданными параметрами минимальной скорости, максимальной скорости, ускорения и замедления.

Исполнительная команда **CMD_PowerSTEP01_GO_UNTIL_F**

Исполнительная команда **CMD_PowerSTEP01_GO_UNTIL_F** = 0x14 предназначена для старта вращения двигателя в прямом направлении на максимальной скорости до получения сигнала на вход, номер которого задан в поле DATA. После получения сигнала двигатель останавливается с заданным замедлением. При отработке команды учитывается заданная маска сигнала. Маскирование сигналов может быть изменено командой **CMD_PowerSTEP01_SET_MASK_EVENT**.

Исполнительная команда **CMD_PowerSTEP01_GO_UNTIL_R**

Исполнительная команда **CMD_PowerSTEP01_GO_UNTIL_R** = 0x15 предназначена для старта вращения двигателя в обратном направлении на максимальной скорости до получения сигнала на вход, номер которого задан в поле DATA. После получения сигнала двигатель останавливается с заданным замедлением. При отработке команды учитывается заданная маска сигнала. Маскирование сигналов может быть изменено командой **CMD_PowerSTEP01_SET_MASK_EVENT**.

Исполнительная команда **CMD_PowerSTEP01_SCAN_ZERO_F**

Исполнительная команда **CMD_PowerSTEP01_SCAN_ZERO_F** = 0x16 предназначена для поиска нулевого положения в прямом направлении с заданной скоростью. Движение продолжается до поступления сигнала на вход SET_ZERO. При поступлении сигнала двигатель останавливается, текущее положение принимается за нулевое. Поле DATA должно содержать скорость движения при поиске нулевого положения. Внимание: в командах установки скорости единицы измерения параметра - полные шаги в секунду.

Исполнительная команда **CMD_PowerSTEP01_SCAN_ZERO_R**

Исполнительная команда **CMD_PowerSTEP01_SCAN_ZERO_R** = 0x17 предназначена для поиска нулевого положения в обратном направлении с заданной скоростью. Движение продолжается до поступления сигнала на вход SET_ZERO. При поступлении сигнала двигатель останавливается, текущее положение принимается за нулевое. Поле DATA должно содержать скорость движения при поиске нулевого положения. Внимание: в командах установки скорости единицы измерения параметра - полные шаги в секунду.

Исполнительная команда **CMD_PowerSTEP01_SCAN_MARK_F**

Исполнительная команда **CMD_PowerSTEP01_SCAN_MARK_F** = 0x18 предназначена для поиска метки положения в прямом направлении. Движение продолжается до поступления сигнала на вход IN1. При поступлении сигнала двигатель останавливается, текущее положение запоминается как метка. Поле DATA определяет скорость движения при поиске метки. Внимание: в командах установки скорости единицы измерения параметра - полные шаги в секунду.



Исполнительная команда CMD_PowerSTEP01_SCAN_MARK_R

Исполнительная команда CMD_PowerSTEP01_SCAN_MARK_R = 0x19 предназначена для поиска метки положения в обратном направлении. Движение продолжается до поступления сигнала на вход IN1. При поступлении сигнала двигатель останавливается, текущее положение запоминается как метка. Поле DATA определяет скорость движения при поиске метки. Внимание: в командах установки скорости единицы измерения параметра - полные шаги в секунду.

Исполнительная команда CMD_PowerSTEP01_GO_ZERO

Исполнительная команда CMD_PowerSTEP01_GO_ZERO = 0x1A предназначена для перемещения в нулевое положение. Поле данных игнорируется.

Исполнительная команда CMD_PowerSTEP01_GO_LABEL

Исполнительная команда CMD_PowerSTEP01_GO_LABEL = 0x1B предназначена для перемещения в положение, которое было отмечено как метка. Поле данных игнорируется.

Исполнительная команда CMD_PowerSTEP01_GO_TO

Исполнительная команда CMD_PowerSTEP01_GO_TO = 0x1C предназначена для перемещения в заданное в поле Data положение по кратчайшему пути.

Исполнительная команда CMD_PowerSTEP01_RESET_POS

Исполнительная команда CMD_PowerSTEP01_RESET_POS = 0x1D предназначена для обнуления счетчика текущего положения. После выполнения команды текущее положение принимается как нулевое. Поле данных игнорируется.

Исполнительная команда CMD_PowerSTEP01_RESET_POWERSTEP01

Исполнительная команда CMD_PowerSTEP01_RESET_POWERSTEP01 = 0x1E используется для осуществления полного аппаратного и программного сброса модуля управления шаговым двигателем, но не контроллера в целом. Поле данных игнорируется.

Исполнительная команда CMD_PowerSTEP01_SOFT_STOP

Исполнительная команда CMD_PowerSTEP01_SOFT_STOP = 0x1F используется для плавной остановки двигателя с заданным ускорением. После остановки двигатель удерживает положение с заданным током удержания. Поле данных игнорируется.

Исполнительная команда CMD_PowerSTEP01_HARD_STOP

Исполнительная команда CMD_PowerSTEP01_HARD_STOP = 0x20 используется для резкой остановки шагового двигателя. После остановки двигатель удерживает положение с заданным током удержания. Поле данных игнорируется.

Исполнительная команда CMD_PowerSTEP01_SOFT_HI_Z

Исполнительная команда CMD_PowerSTEP01_SOFT_HI_Z = 0x21 используется для плавной остановки шагового двигателя с заданным ускорением. После остановки питания с обмоток двигателя снимается. Поле данных игнорируется.

Исполнительная команда CMD_PowerSTEP01_HARD_HI_Z

Исполнительная команда CMD_PowerSTEP01_HARD_HI_Z = 0x22 используется для резкой остановки и обесточивания обмоток двигателя. Поле данных игнорируется.

Исполнительная команда **CMD_PowerSTEP01_SET_WAIT**

Исполнительная команда CMD_PowerSTEP01_SET_WAIT = 0x23 предназначена для задания паузы. Поле DATA должно содержать время ожидания паузы в диапазоне от 0 до 3600000мс.

Исполнительная команда **CMD_PowerSTEP01_SET_RELE**

Исполнительная команда CMD_PowerSTEP01_SET_RELE = 0x24 предназначена для включения реле контроллера. Поле данных игнорируется.

Исполнительная команда **CMD_PowerSTEP01_CLR_RELE**

Исполнительная команда CMD_PowerSTEP01_CLR_RELE = 0x25 предназначена для выключения реле контроллера. Поле данных игнорируется.

Исполнительная команда **CMD_PowerSTEP01_WAIT_IN0**

Исполнительная команда CMD_PowerSTEP01_WAIT_IN0 = 0x27 предназначена для ожидания поступления сигнала на вход IN0. Поле данных игнорируется.

Исполнительная команда **CMD_PowerSTEP01_WAIT_IN1**

Исполнительная команда CMD_PowerSTEP01_WAIT_IN1 = 0x28 предназначена для ожидания поступления сигнала на вход IN1. Поле данных игнорируется.

Исполнительная команда **CMD_PowerSTEP01_GOTO_PROGRAM**

Исполнительная команда CMD_PowerSTEP01_GOTO_PROGRAM = 0x29 предназначена для безусловного перехода к заданной команде заданной программы. Поле DATA должно содержать информацию о требуемом номере программы и порядковом номере команды в программе: биты 0..7 поля Data определяют номер команды, биты 8,9 поля DATA определяют номер программы.

Битовая раскладка поля данных:

21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	Номер программы		Номер команды							

Исполнительная команда **CMD_PowerSTEP01_GOTO_PROGRAM_IF_IN0**

Исполнительная команда CMD_PowerSTEP01_GOTO_PROGRAM_IF_IN0 = 0x2A предназначена для перехода к заданной команде заданной программы, если на входе IN0 присутствует сигнал. Поле DATA должно содержать информацию о требуемом номере программы и порядковом номере команды в программе: биты 0..7 поля Data определяют номер команды, биты 8,9 поля DATA определяют номер программы.

Битовая раскладка поля данных:

21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	Номер программы		Номер команды							

Исполнительная команда **CMD_PowerSTEP01_GOTO_PROGRAM_IF_IN1**

Исполнительная команда CMD_PowerSTEP01_GOTO_PROGRAM_IF_IN1 = 0x2B предназначена для перехода к заданной команде заданной программы, если на входе IN1 присутствует сигнал. Поле DATA должно содержать информацию о требуемом номере программы и порядковом номере команды в программе: биты 0..7 поля Data определяют номер команды, биты 8,9 поля DATA определяют номер программы.

Битовая раскладка поля данных:

21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	Номер программы		Номер команды							

Исполнительная команда **CMD_PowerSTEP01_LOOP_PROGRAM**

Исполнительная команда **CMD_PowerSTEP01_LOOP_PROGRAM** = 0x2C используется для создания циклов – контроллер повторяет в цикле заданное число раз заданное количество команд (начиная с команды, следующей за **CMD_PowerSTEP01_LOOP_PROGRAM**). Поле DATA содержит количество циклов (от 1 до 1023) и количество команд в цикле (от 1 до 1023): 0..9 поля Data – количество команд, биты 10..19 поля Data – количество циклов.

Битовая раскладка поля данных:

21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	Количество циклов										Количество команд в цикле									

Исполнительная команда **CMD_PowerSTEP01_CALL_PROGRAM**

Исполнительная команда **CMD_PowerSTEP01_CALL_PROGRAM** = 0x2D предназначена для вызова подпрограммы. Поле DATA содержит информацию о номере программы и порядковом номере команды в программе, с которой начинается подпрограмма: биты 0..7 поля DATA – порядковый номер команды, биты 8,9 поля DATA – номер программы. Для возврата в основную программу подпрограмма должна содержать команду возврата - **CMD_PowerSTEP01_RETURN_PROGRAM**. Подпрограмма выполняется до тех пор, пока не дойдет до команды **CMD_PowerSTEP01_RETURN_PROGRAM**, затем возвращает управление основной вызвавшей ее программе.

Битовая раскладка поля данных:

21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	Номер программы		Номер команды							

Исполнительная команда **CMD_PowerSTEP01_RETURN_PROGRAM**

Исполнительная команда **CMD_PowerSTEP01_RETURN_PROGRAM** = 0x2E используется для возврата из подпрограммы в основную программу. Поле данных игнорируется. Вызов подпрограммы осуществляется командой **CMD_PowerSTEP01_CALL_PROGRAM**. В случае, если перед вызовом команды возврата **CMD_PowerSTEP01_RETURN_PROGRAM** не была вызвана подпрограмма, контроллер сгенерирует ошибку.

Исполнительная команда **CMD_PowerSTEP01_START_PROGRAM_MEM0**

Исполнительная команда **CMD_PowerSTEP01_START_PROGRAM_MEM0** = 0x2F используется для старта программы, записанной в область памяти 0 контроллера. Поле данных игнорируется.

Команды **CMD_PowerSTEP01_START_PROGRAM_MEM1** = 0x30, **CMD_PowerSTEP01_START_PROGRAM_MEM2** = 0x31, **CMD_PowerSTEP01_START_PROGRAM_MEM3** = 0x32 используются для старта программ, записанных области памяти 1, 2 и 3 соответственно.

Исполнительная команда **CMD_PowerSTEP01_STOP_PROGRAM_MEM**

Исполнительная команда **CMD_PowerSTEP01_STOP_PROGRAM_MEM** = 0x33 используется для остановки выполнения программы. Поле данных игнорируется.

Исполнительная команда **CMD_PowerSTEP01_GOTO_PROGRAM_IF_ZERO**

Исполнительная команда **CMD_PowerSTEP01_GOTO_PROGRAM_IF_ZERO** = 0x39 предназначена для перехода к заданной команде заданной программы, если значение текущей позиции равно 0. Поле DATA должно содержать информацию о требуемом номере программы и порядковом номере команды в программе: биты 0..7 поля Data определяют номер команды, биты 8,9 поля DATA определяют номер программы.

Битовая раскладка поля данных:

21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	Номер программы		Номер команды							

Исполнительная команда **CMD_PowerSTEP01_GOTO_PROGRAM_IF_IN_ZERO**

Исполнительная команда **CMD_PowerSTEP01_GOTO_PROGRAM_IF_IN_ZERO** = 0x3A предназначена для перехода к заданной команде заданной программы, в случае, если на входе SET_ZERO присутствует сигнал. Поле DATA должно содержать информацию о требуемом номере программы и порядковом номере команды в программе: биты 0..7 поля Data определяют номер команды, биты 8,9 поля DATA определяют номер программы.

Битовая раскладка поля данных:

21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	Номер программы		Номер команды							

Исполнительная команда **CMD_PowerSTEP01_WAIT_CONTINUE**

Исполнительная команда **CMD_PowerSTEP01_WAIT_CONTINUE** = 0x3B предназначена для ожидания прихода синхросигнала на вход CONTINUE, необходимого для синхронизации программ в нескольких блоках. Поле данных игнорируется.

Исполнительная команда **CMD_PowerSTEP01_SET_WAIT_2**

Исполнительная команда **CMD_PowerSTEP01_SET_WAIT_2** = 0x3C предназначена для задания паузы. Поле DATA должно содержать время ожидания паузы в диапазоне от 0 до 3600000мс. В отличие от аналогичной команды **CMD_PowerSTEP01_SET_WAIT**, выполнение данной команды может быть прервано поступлением сигнала на вход IN0, IN1 или SET_ZERO.

Исполнительная команда **CMD_PowerSTEP01_SCAN_MARK2_F**

Исполнительная команда **CMD_PowerSTEP01_SCAN_MARK2_F** = 0x3D предназначена для поиска метки положения в прямом направлении. Движение продолжается до поступления сигнала на вход IN1. При поступлении сигнала двигатель останавливается с заданным значением торможения, текущее положение запоминается как метка. Поле DATA определяет скорость движения при поиске метки. Внимание: в командах установки скорости единицы измерения параметра - полные шаги в секунду.

Исполнительная команда **CMD_PowerSTEP01_SCAN_MARK2_R**

Исполнительная команда **CMD_PowerSTEP01_SCAN_MARK2_R** = 0x3E предназначена для поиска метки положения в обратном направлении. Движение продолжается до поступления сигнала на вход IN1. При поступлении сигнала двигатель останавливается с заданным значением торможения, текущее положение запоминается как метка. Поле DATA определяет скорость движения при поиске метки. Внимание: в командах установки скорости единицы измерения параметра - полные шаги в секунду.

Редакция от 31.05.2024