



ООО «ЦерсисАналитик»

КОНТРОЛЛЕР ШАГОВЫХ ДВИГАТЕЛЕЙ 5SMDCV2

5SMDCV2

Руководство по программированию

Оглавление:

1. Подключение	2
2. Протокол взаимодействия <u>USB</u>	2
2.1. Команды	2
2.2.1. Получить версию прошивки	2
2.2.2. Получить идентификатор платы	3
2.2.3. Переместиться вперед на заданное количество микрошагов	3
2.2.4. Переместиться назад на заданное количество микрошагов	3
2.2.5. Получить статус канала	3
2.2.6. Команда Стоп	3
2.2.7. Найти «домашнюю» позицию	4
2.2.8. Установить режим GPIO	4
2.2.9. Получить маску режима GPIO	4
2.2.10. Установить значение выходов GPIO	4
2.2.11. Получить значение пинов GPIO	5
2.2. Коды результатов выполнения команды	5
2.3. Алгоритм расчета CRC	5
3. Протокол взаимодействия Modbus RTU	7
3.1. Подключение	7
3.2. Карта "INPUTS" регистров	7
3.3. Карта "HOLDING" регистров	11
3.4. Команды	11
3.5. GPIO	12
4. Флаги состояния оси	12

1. Подключение

Устройство физически подключается в USB разъем (версия 1.1 и выше) либо UART. Логически представлено в операционной системе как виртуальный COM порт.

Настройки COM порта: скорость – 115 200, четность – нет, стоповых бит – 1, бит данных – 8.

Рекомендуемое значение таймаута не менее 20 мс.

2. Протокол взаимодействия USB

Все данные пересылаются в виде сообщений. На каждое сообщение устройство посылает ответ с задержкой не более 20 мс. Т.е. поток выполнения команд представляет собой {[сообщение, ответ], [сообщение, ответ], ...}. Сообщение и ответ представляют из себя пакет следующего формата.

Заголовок (4 байта)	Размер данных в пакете (1 байт)	Данные (0-255 байт)	CRC (2 байт)
Сообщение PC->device: 0x4E, 0xB1, 0xB7, 0x18 Ответ device->PC: 0x18, 0xB7, 0xB1, 0x4E			CRC-16-CCITT, Полином $x^{16} + x^{12} + x^5 + 1$

CRC вычисляется по полям “размер данных” и “данные”, поле “заголовок” не учитывается.

Код команды и сопутствующие параметры помещаются в поле “данные”.

Многобайтные (16 bit, 32-bit) значения передаются в little-endian формате (младшее значение по младшему адресу). Например, значение 0xAABB будет передано последовательностью байт 0xBB, 0xAA, значение 0xAABBCCDD будет передано последовательностью байт 0xDD, 0xCC, 0xBB, 0xAA.

В случае если устройство получит пакет с неверным CRC или неверным заголовком, то ответ посылаться не будет т.к. такие данные интерпретируются как мусор/помеха на линии связи.

Темп посылки сообщений не должен превышать 100 команд в секунду. Рекомендуются 50 команд в секунду.

2.1. Команды

Размер кода команды 1 байт.

2.2.1. Получить версию прошивки

[0x00] Дополнительных параметров нет.

Ответ:

Код выполнения команды	Старшее значение версии	Младшее значение версии
1 байт	2 байта	2 байта

2.2.2. Получить идентификатор платы

[0x01] Дополнительных параметров нет.

Ответ:

Код выполнения команды	Id платы в ASCII кодировке
1 байт	24 байта

2.2.3. Переместиться вперед на заданное количество микрошагов.

[0x05, Номер канала – 1 байт, Количество шагов – без знаковое 32битное значение]

Номер канала может быть в пределах 0...4.

Ответ:

Код выполнения команды
1 байт

2.2.4. Переместиться назад на заданное количество микрошагов.

[0x06, Номер канала – 1 байт, Количество шагов – без знаковое 32битное значение]

Номер канала может быть в пределах 0...4.

Ответ:

Код выполнения команды
1 байт

2.2.5. Получить статус канала.

[0x0A, Номер канала – 1 байт]

Номер канала может быть в пределах 0...4

Ответ:

Код выполнения команды	Флаги состояния	Текущая позиция	Зарезервировано
1 байт	4 байта	4 байта (без знаковое 32-битное)	4 байта

2.2.6. Команда Стоп

[0x0B, Номер канала – 1 байт]

Номер канала может быть в пределах 0...4

Ответ:

Код выполнения команды
1 байт

Код выполнения команды
1 байт

2.2.11. Получить значение пинов GPIO.

[0x15] Дополнительных параметров нет.

Ответ:

Код выполнения команды	Маска состояния GPIO
1 байт	1 байт

Каждый бит маски соответствует логическому состоянию пина. Младший бит соответствует пину 3 (GPIO N1), старший бит соответствует пину 10 (GPIO N 8). Если пин сконфигурирован как вход, то значение отражает логическое состояние входа. Если пин сконфигурирован как выход, то значение соответствует ранее записанному значению (см. команду установить значение выходов GPIO))

2.2. Коды результата выполнения команды

0x00 – выполнена успешно

0x01 – неверная команда

0x03 – неверный номер канала

0x04 – не выполнено (например, попытка выполнить перемещение в то время как двигатель уже выполняет перемещение или идет поиск домашней позиции)

2.3. Алгоритм расчета CRC.

```
static ushort[] crc_table = {
0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50a5,
0x60c6, 0x70e7, 0x8108, 0x9129, 0xa14a, 0xb16b,
0xc18c, 0xd1ad, 0xe1ce, 0xf1ef, 0x1231, 0x0210,
0x3273, 0x2252, 0x52b5, 0x4294, 0x72f7, 0x62d6,
0x9339, 0x8318, 0xb37b, 0xa35a, 0xd3bd, 0xc39c,
0xf3ff, 0xe3de, 0x2462, 0x3443, 0x0420, 0x1401,
0x64e6, 0x74c7, 0x44a4, 0x5485, 0xa56a, 0xb54b,
0x8528, 0x9509, 0xe5ee, 0xf5cf, 0xc5ac, 0xd58d,
0x3653, 0x2672, 0x1611, 0x0630, 0x76d7, 0x66f6,
0x5695, 0x46b4, 0xb75b, 0xa77a, 0x9719, 0x8738,
0xf7df, 0xe7fe, 0xd79d, 0xc7bc, 0x48c4, 0x58e5,
0x6886, 0x78a7, 0x0840, 0x1861, 0x2802, 0x3823,
```

0xc9cc, 0xd9ed, 0xe98e, 0xf9af, 0x8948, 0x9969,
0xa90a, 0xb92b, 0x5af5, 0x4ad4, 0x7ab7, 0x6a96,
0x1a71, 0x0a50, 0x3a33, 0x2a12, 0xdbfd, 0xcbdc,
0xfbbf, 0xeb9e, 0x9b79, 0x8b58, 0xbb3b, 0xab1a,
0x6ca6, 0x7c87, 0x4ce4, 0x5cc5, 0x2c22, 0x3c03,
0x0c60, 0x1c41, 0xedaе, 0xfd8f, 0xcdec, 0xddcd,
0xad2a, 0xbd0b, 0x8d68, 0x9d49, 0x7e97, 0x6eb6,
0x5ed5, 0x4ef4, 0x3e13, 0x2e32, 0x1e51, 0x0e70,
0xff9f, 0xefbe, 0xdfdd, 0xcffc, 0xbf1b, 0xaf3a,
0x9f59, 0x8f78, 0x9188, 0x81a9, 0xb1ca, 0xa1eb,
0xd10c, 0xc12d, 0xf14e, 0xe16f, 0x1080, 0x00a1,
0x30c2, 0x20e3, 0x5004, 0x4025, 0x7046, 0x6067,
0x83b9, 0x9398, 0xa3fb, 0xb3da, 0xc33d, 0xd31c,
0xe37f, 0xf35e, 0x02b1, 0x1290, 0x22f3, 0x32d2,
0x4235, 0x5214, 0x6277, 0x7256, 0xb5ea, 0xa5cb,
0x95a8, 0x8589, 0xf56e, 0xe54f, 0xd52c, 0xc50d,
0x34e2, 0x24c3, 0x14a0, 0x0481, 0x7466, 0x6447,
0x5424, 0x4405, 0xa7db, 0xb7fa, 0x8799, 0x97b8,
0xe75f, 0xf77e, 0xc71d, 0xd73c, 0x26d3, 0x36f2,
0x0691, 0x16b0, 0x6657, 0x7676, 0x4615, 0x5634,
0xd94c, 0xc96d, 0xf90e, 0xe92f, 0x99c8, 0x89e9,
0xb98a, 0xa9ab, 0x5844, 0x4865, 0x7806, 0x6827,
0x18c0, 0x08e1, 0x3882, 0x28a3, 0xcb7d, 0xdb5c,
0xeb3f, 0xfb1e, 0x8bf9, 0x9bd8, 0xabbb, 0xbb9a,
0x4a75, 0x5a54, 0x6a37, 0x7a16, 0x0af1, 0x1ad0,
0x2ab3, 0x3a92, 0xfd2e, 0xed0f, 0xdd6c, 0xcd4d,
0xbdaa, 0xad8b, 0x9de8, 0x8dc9, 0x7c26, 0x6c07,
0x5c64, 0x4c45, 0x3ca2, 0x2c83, 0x1ce0, 0x0cc1,
0xef1f, 0xff3e, 0xcf5d, 0xdf7c, 0xaf9b, 0xbfba,
0x8fd9, 0x9ff8, 0x6e17, 0x7e36, 0x4e55, 0x5e74,
0x2e93, 0x3eb2, 0x0ed1, 0x1ef0

};

```

ushort CRCCCITT(byte[] data, int index, int length, ushort seed, ushort final)
{
    uint crc = seed;
    uint temp;
    for (int i = index, count = 0; count < length; i++, count++)
    {
        temp = (data[i] ^ (crc >> 8)) & 0xff;
        crc = crc_table[temp] ^ (crc << 8);
    }
    return (ushort)(crc ^ final);
}

```

Вызов `ushort crc = CRCCCITT (buf, 0, buffer_length, 0xFFFF, 0);`

3. Протокол взаимодействия Modbus RTU

3.1. Подключение

Скорость, четность, адрес задаются в конфигурационном ПО. Настройки по умолчанию:

- Скорость 115200 бод
- 1 стоповый бит
- Без четности
- Адрес 1

3.2. Карта "INPUTS" регистров

Базовый адрес 1000. Всего регистров 160.

№	Название	Назначение
0	MAJOR VERSION	Старшее слово версии прошивки.
1	MINOR VERSION	Младшее слово версии прошивки.
2	BOARD TYPE	Аппаратная версия.
3	AXIS COUNT	Количество осей поддерживаемых контроллером
4 - 15	BOARD ID	Идентификатор контроллера. 24 ASCII символа. По умолчанию содержит серийный номер контроллера.
16 - 27	BOARD NAME	Имя платы. 24 ASCII символа. По умолчанию содержит серийный номер контроллера.

28	POWER VOLTAGE	Напряжение питания. Хранится в упакованном виде. Старший байт – значение до запятой, младший после запятой. Алгоритм преобразования: $\text{PowerVoltage} = (\text{regValue} \gg 8) + (\text{regValue} \& 0xFF) / 100.0$
29	USB VOLTAGE	Напряжение USB. Хранится в упакованном виде. Старший байт – значение до запятой, младший после запятой. Алгоритм преобразования: UsbVoltage
Далее идут банки регистров состояния каждой оси. По 4 регистра на ось.		
30	STATUS AXIS1 HIGH	Старшее и младшее слово флагов состояния.
31	STATUS AXIS1 LOW	
32	POS AXIS1 HIGH	Старшее и младшее слово текущей позиции оси в микрошагах. Позиция = $(\text{POS AXIS1 HIGH} \ll 16) \text{POS AXIS1 LOW}$.
33	POS AXIS1 LOW	
34	STATUS AXIS2 HIGH	Аналогично регистрам 30, 31
35	STATUS AXIS2 LOW	
36	POS AXIS2 HIGH	Аналогично регистрам 32, 33
37	POS AXIS2 LOW	
38	STATUS AXIS3 HIGH	Аналогично регистрам 30, 31
39	STATUS AXIS3 LOW	
40	POS AXIS3 HIGH	Аналогично регистрам 32, 33
41	POS AXIS3 LOW	
42	STATUS AXIS4 HIGH	Аналогично регистрам 30, 31
43	STATUS AXIS4 LOW	
44	POS AXIS4 HIGH	Аналогично регистрам 32, 33
45	POS AXIS4 LOW	
46	STATUS AXIS5 HIGH	Аналогично регистрам 30, 31
47	STATUS AXIS5 LOW	
48	POS AXIS5 HIGH	Аналогично регистрам 32, 33
49	POS AXIS5 LOW	
50 - 59	DUMMY 50 - 59	Зарезервированы.
Далее идут банки регистров настроек каждой оси. По 20 регистров на ось.		
60	CFG_FLAGS_AXIS1	Флаги настроек для оси №1
61	CFG_DUMMY_AXIS1	Зарезервирован.
62	CFG_MAX_POS_AXIS1_H	Максимально допустимая позиция в шагах. В случае если вы попытаетесь переместить ось в положение, которое больше чем это команда будет проигнорирована.
63	CFG_MAX_POS_AXIS1_L	
64	CFG_DEC_AXIS1	Торможение в шагах/с ²
65	CFG_ACC_AXIS1	Ускорение в шагах/с ²
66	CFG_MIN_SPEED_AXIS1	Начальная скорость при перемещении.
67	CFG_SPEED_AXIS1	Целевая (максимальная) скорость при перемещении

68	CFG_HOLD_RUN_CURRENT_AXIS1	Значение тока удержания (старший байт), значение тока при движении (младший байт). Каждый из них может принимать значения 0 – 31. Алгоритм пересчета в миллиамперы mA = (val + 1)*60; Например значение 1 соответствует току в 120 mA
69	CFG_REF_ROLLOUT_STEPS_AXIS1_H	Отступ от датчика “домашней” позиции в шагах.
70	CFG_REF_ROLLOUT_STEPS_AXIS1_L	
71	CFG_REF_SPEED_AXIS1	Скорость съезда с датчика “домашней” позиции и/или концевого датчика.
72	CFG_REF_CUR_PWM_AXIS1	Ток при съезде с датчика “домашней” позиции и/или концевого датчика. Может принимать значения 0 – 31. Алгоритм пересчета в миллиамперы mA = (val + 1)*60; Например значение 1 соответствует току в 120 mA.
73	CFG_REF_DELAY_ON_STOP_AXIS1	Задержка между наездом на датчик и началом съезда в миллисекундах. Если значение 0 – без задержки.
74	DC_POWER_AXIS1	Мощность подаваемая в двигатель постоянного тока/соленоид и т.п. Задается в процентах.
75	DUMMY0_AXIS1	Зарезервировано.
76	DUMMY1_AXIS1	
77	DUMMY2_AXIS1	
78	DUMMY3_AXIS1	
79	DUMMY4_AXIS1	
80	CFG_FLAGS_AXIS2	Банк регистров настроек для оси №2. См. описание регистров 60-79.
81	CFG_DUMMY_AXIS2	
82	CFG_MAX_POS_AXIS2_H	
83	CFG_MAX_POS_AXIS2_L	
84	CFG_DEC_AXIS2	
85	CFG_ACC_AXIS2	
86	CFG_MIN_SPEED_AXIS2	
87	CFG_SPEED_AXIS2	
88	CFG_HOLD_RUN_CURRENT_AXIS2	
89	CFG_REF_ROLLOUT_STEPS_AXIS2_H	
90	CFG_REF_ROLLOUT_STEPS_AXIS2_L	
91	CFG_REF_SPEED_AXIS2	
92	CFG_REF_CUR_PWM_AXIS2	
93	CFG_REF_DELAY_ON_STOP_AXIS2	
94	DC_POWER_AXIS2	
95	DUMMY0_AXIS2	
96	DUMMY1_AXIS2	
97	DUMMY2_AXIS2	
98	DUMMY3_AXIS2	
99	DUMMY4_AXIS2	

100	CFG_FLAGS_AXIS3	Банк регистров настроек для оси №3. См. описание регистров 60-79.
101	CFG_DUMMY_AXIS3	
102	CFG_MAX_POS_AXIS3_H	
103	CFG_MAX_POS_AXIS3_L	
104	CFG_DEC_AXIS3	
105	CFG_ACC_AXIS3	
106	CFG_MIN_SPEED_AXIS3	
107	CFG_SPEED_AXIS3	
108	CFG_HOLD_RUN_CURRENT_AXIS3	
109	CFG_REF_ROLLOUT_STEPS_AXIS3_H	
110	CFG_REF_ROLLOUT_STEPS_AXIS3_L	
111	CFG_REF_SPEED_AXIS3	
112	CFG_REF_CUR_PWM_AXIS3	
113	CFG_REF_DELAY_ON_STOP_AXIS3	
114	DC_POWER_AXIS3	
115	DUMMY0_AXIS3	
116	DUMMY1_AXIS3	
117	DUMMY2_AXIS3	
118	DUMMY3_AXIS3	
119	DUMMY4_AXIS3	
120	CFG_FLAGS_AXIS4	Банк регистров настроек для оси №4. См. описание регистров 60-79
121	CFG_DUMMY_AXIS4	
122	CFG_MAX_POS_AXIS4_H	
123	CFG_MAX_POS_AXIS4_L	
124	CFG_DEC_AXIS4	
125	CFG_ACC_AXIS4	
126	CFG_MIN_SPEED_AXIS4	
127	CFG_SPEED_AXIS4	
128	CFG_HOLD_RUN_CURRENT_AXIS4	
129	CFG_REF_ROLLOUT_STEPS_AXIS4_H	
130	CFG_REF_ROLLOUT_STEPS_AXIS4_L	
131	CFG_REF_SPEED_AXIS4	
132	CFG_REF_CUR_PWM_AXIS4	
133	CFG_REF_DELAY_ON_STOP_AXIS4	
134	DC_POWER_AXIS4	
135	DUMMY0_AXIS4	
136	DUMMY1_AXIS4	
137	DUMMY2_AXIS4	
138	DUMMY3_AXIS4	
139	DUMMY4_AXIS4	
140	CFG_FLAGS_AXIS5	Банк регистров настроек для оси №5. См. описание регистров 60-79
141	CFG_DUMMY_AXIS5	
142	CFG_MAX_POS_AXIS5_H	
143	CFG_MAX_POS_AXIS5_L	
144	CFG_DEC_AXIS5	

145	CFG_ACC_AXIS5
146	CFG_MIN_SPEED_AXIS5
147	CFG_SPEED_AXIS5
148	CFG_HOLD_RUN_CURRENT_AXIS5
149	CFG_REF_ROLLOUT_STEPS_AXIS5_H
150	CFG_REF_ROLLOUT_STEPS_AXIS5_L
151	CFG_REF_SPEED_AXIS5
152	CFG_REF_CUR_PWM_AXIS5
153	CFG_REF_DELAY_ON_STOP_AXIS5
154	DC_POWER_AXIS5
155	DUMMY0_AXIS5
156	DUMMY1_AXIS5
157	DUMMY2_AXIS5
158	DUMMY3_AXIS5
159	DUMMY4_AXIS5

3.3. Карта "HOLDING" регистров

Базовый адрес 2000. Всего регистров 17. Пять банков по три регистра, плюс 2 регистра GPIO.

№	Название	Назначение
0	TARGET_1_H	Старшее слово параметра команды для оси №1.
1	TARGET_1_L	Младшее слово параметра команды для оси №1.
2	CMD_1	Код команды для оси №1.
3	TARGET_2_H	Аналогично регистрам 0 – 2. Ось №2.
4	TARGET_2_L	
5	CMD_2	
6	TARGET_3_H	Аналогично регистрам 0 – 2. Ось № 3.
7	TARGET_3_L	
8	CMD_3	
9	TARGET_4_H	Аналогично регистрам 0 – 2. Ось № 4.
10	TARGET_4_L	
11	CMD_4	
12	TARGET_5_H	Аналогично регистрам 0 – 2. Ось № 5.
13	TARGET_5_L	
14	CMD_5	
15	GPIO MODE MASK	Режим работы GPIO
16	GPIO VALUE	Значение пинов GPIO

3.4. Команды

Рассмотрим на примере выполнение команды перемещения в заданную точку. Допустим мы хотим переместить ось № 1 в позицию 1000. Для этого необходимо записать в "holding" регистры 0-1 значение 0x000003E8 (0x0000 в регистр 0, 0x03E8 в регистр 1), а в регистр 2 код команды MoveAbs (0x0008).

Название	Код	Параметр	Описание
MoveFw	1	Количество микрошагов.	Сместиться вперед на заданное количество микрошагов. Допускается выполнение команды во время перемещения. Если идет поиск домашней позиции команда будет проигнорирована.
MoveBw	2	Количество микрошагов.	Сместиться назад на заданное количество микрошагов. Допускается выполнение команды во время перемещения. Если идет поиск домашней позиции команда будет проигнорирована.
Stop	3	Нет	Остановить перемещение или поиск домашней позиции.
MotorPower	4	Включить/выключить	Включить или выключить питание на обмотках двигателя. Если регистр параметра содержит 0 то выключить, в противном случае включить. Допускается выполнение команды во время перемещения или поиска домашней позиции.
SetCurSpeed	5	Значение скорости	Установить значение максимальной/текущей скорости. Значение должно быть в пределах [1-32765]. Допускается выполнение команды во время перемещения, в таком случае контроллер плавно поднимет/уменьшит скорость до заданной используя настройки ускорения/торможения.
FindHome	6	Нет	Выполнить поиск домашней позиции. Выполнение команды допустимо только если в данный момент двигатель остановлен, в противном случае команда будет проигнорирована.
SetDcPower	7	Значение мощности DC двигателя в процентах.	Задаёт значение мощности двигателя постоянного тока или соленоида в процентах от максимальной. Допустимые значения 1 – 100.
MoveAbs	8	Целевая позиция в микрошагах.	Переместить ось в заданную позицию. Допускается выполнение команды во время перемещения. Если идет поиск домашней позиции команда будет проигнорирована.

3.5. GPIO

Для управления GPIO доступны два регистра N 15 (GPIO MODE MASK), N 16 (GPIO VALUE). Используются только младшие 8 битов в каждом регистре.

GPIO MODE MASK – содержит режим работы (вход или выход) для каждого пина GPIO. Описание маски смотри в команде «установить режим GPIO», «получить маску режима GPIO».

GPIO VALUE – содержит логическое состояние GPIO. Смотри команды установить значения выходов GPIO, получить значения пинов GPIO. В случае Modbus используется упрощенный вариант записи значений. Можно записать только все биты значений одновременно.

4. Флаги состояния оси

Флаги состояния представляет собой битовое поле размером 32 бита. В случае USB протокола вы можете получить состояние командой 0x0A. См . описание.

В случае Modbus RTU см. регистры 30 – 31, 34 – 35, 38 – 39, 42 – 43, 46 – 47. Псевдокод получения значений (flags – значение полученное командой или из регистров):

```
bool IsOnline = (flags & 0x01) != 0;  
bool IsOverCurrent = (flags & 0x02) != 0;  
bool IsUnderVoltage = (flags & 0x04) != 0;  
bool IsOverHeat = (flags & 0x08) != 0;  
bool IsMoving = (flags & 0x10) != 0;  
bool IsMotorOn = (flags & 0x20) != 0;  
bool IsSignalA = (flags & 0x40) != 0;  
bool IsSignalB = (flags & 0x80) != 0;  
bool IsSignalC = (flags & 0x100) != 0;  
bool IsRefSearchRequired = (flags & 0x200) != 0;  
bool IsStopTriggered = (flags & 0x400) != 0;  
byte LastMoveDirection = (flags & 0x800) != 0 ? Forward : Backward;  
byte SwitchRollDirection = (flags & 0x1000) != 0 ? Forward : Backward;  
bool IsRefSearch = (flags & 0x2000) != 0;
```

Описание полей:

IsOnline – true -> силовое питание канала включено, драйвер канала исправен.

IsOverCurrent – true -> превышен максимальный ток, возможно короткое замыкание в цепи двигателя.

IsUnderVoltage – true -> слишком низкое напряжение силового питания (менее 6 вольт) либо драйвер неисправен.

IsOverHeat – true -> перегрев драйвера двигателя.

IsMoving – true -> выполняется перемещение.

IsMotorOn – true -> силовое питание подано на обмотки двигателя.

IsSignalA – true -> сработал датчик, подключенный ко входу “А”. Например, концевой переключатель.

IsSignalB – true -> сработал датчик, подключенный ко входу “В”. Например, концевой переключатель.

IsSignalC – true -> сработал датчик, подключенный ко входу “С”. Например, концевой переключатель.

IsRefSearchRequired – true -> необходимо выполнить поиск домашней позиции т.к. текущее физическое положение может не соответствовать значению позиции в микрошагах. Это может случиться если произошел наезд на концевой переключатель либо исчезло силовое питание в процессе перемещения. См. команду “Поиск домашней позиции” [USB], [Modbus RTU].

IsStopTriggered – true -> произошел наезд на концевой датчик.

LastMoveDirection – 0 -> последняя из команд на перемещение была команда назад, 1 – вперед.

IsRefSearch – true -> выполняется поиск домашней позиции.

ООО «ЦерсисАналитик»