CersysAnalytic Ltd.

STEPPER MOTOR & DC MOTOR CONTROLLERS

5SMDCV2

Manual

Table of contents

1. Connection

The device is manually connected to a USB connector (version 1.1 and higher) or UART. The operating system presented as a virtual COM port.

COM port settings: speed - 115200, parity - no, stop bits - 1, data bit - 8.

The recommended timeout value is not less than 20 ms.

2. USB communication protocol

All data are sent as messages. For each message, the device sends a response with a delay of no more than 20 ms. Therefore, the command execution flow is {[message, reply], [message, reply],...}.

The message and response are a packet with the following format.

Header (4 bytes)	Packet data size (1 byte)	Data (0-255 bytes)	CRC (2 bytes)
PC-> device message: 0x4E, 0xB1, 0xB7, 0x18 Device-> PC response: 0x18, 0xB7, 0xB1, 0x4E			CRC-16-CCITT, Polynomial x ^ 16 + x ^ 12 + x ^ 5 + 1

CRC is calculated by the “data size” and “data” fields. The “header” field is not considered.

The command code and related parameters are placed in the “data” field.

Multibit (16 bit, 32-bit) values are transmitted in a «little-endian» format (least significant at

low address). For example, the value 0xAABB will be transmitted by the sequence of bytes 0xBB,

0xAA, the value 0xAABBCCDD will be transmitted in sequence of bytes 0xDD, 0xCC, 0xBB, 0xAA.

If the device receives a packet with an invalid CRC or invalid header, then no reply will be sent. Such data is interpreted as garbage / noise on a communication lines.

The rate of sending messages should not exceed 100 commands per second. Recommended value is 50 commands per second.

2.1. Commands

The size of the command code is 1 byte.

2.2.1. Get the firmware version.

[0x00] No additional parameters set.

Answer:

Command execution code	Version Major	Minor version value

1 byte	2 bytes	2 bytes

2.2.2. Get board ID

[0x01] No additional parameters set.

Reply:

Command execution code	Board id in ASCII encoding
1 byte	24 bytes

2.2.3. Move forward for a given number of microsteps

[0x06, Channel number - 1 byte, Number of steps - unsigned 32-bit value]

The channel number can be in the range from 0 to 4.

Reply:

Command execution code
1 byte

2.2.4. Move backward for a given number of microsteps

[0x05, Channel number - 1 byte, Number of steps - unsigned 32-bit value]

The channel number can be in the range from 0 to 4.

Reply:

Command execution code
1 byte

2.2.5. Get channel status.

[0x0A, Channel number - 1 byte]

The channel number can be in the range from 0 to 4.

Reply:

Command execution code	Status flags	Current position	Reserved
1 byte	4 bytes	4 bytes (unsigned 32-bit)	4 bytes

2.2.6. Stop command

[0x0B, Channel number - 1 byte]

The channel number can be in the range from 0 to 4.

Reply:

Command execution code
1 byte

2.2.7. Find your «home» position.

[0x0F, Channel number - 1 byte]

The channel number can be in the range from 0 to 4.

Reply:

Command execution code
1 byte

2.2.8. Set GPIO mode

[0x12, Mode Mask (1 byte)

Mask with each bit we set the mode of operation of the pins. Where 1 means output mode and 0 means input.

The least significant bit corresponds to pin 3 on the (GPIO N 1) connector. For example, the mask 10000011 means that pin 3 (GPIO N1), pin 4 (GPIO N 2), pin 10 (GPIO N 8) are configured as outputs, the rest as inputs.

Attention: the GPIO mode mask is not saved in ROM and when the board is turned on, it is reset to

0, that is all GPIOs are set as inputs by default.

Reply:

Command execution code
1 byte

2.2.9. Get the mask of the GPIO mode.

[0x13] No additional parameters set.

Returns the GPIO mode mask byte back. For a description of the mask, see the command set GPIO mode.

Reply:

Command execution code	GPIO Mode Mask
1 byte	1 byte

2.2.10. Set the values of the GPIO outputs.

[0x14, value mask (1 byte), output values (1 byte)]

The value mask determines which values will be set and which will not. If mask bit is set to 1, then the corresponding bit of values will be written, 0 will not.

Every bit the values of the outputs set the state of output 1 or 0. The least significant bit corresponds to pin 3 (GPIO N1), the most significant bit corresponds to pin 10 (GPIO N 8). If the pins are configured as inputs, then the corresponding mask bits will be ignored.

Reply:

Command execution code
1 byte

2.2.11. Get the values of the GPIO pins.

[0x15] No additional parameters set.

Reply:

Command execution code	GPIO Mode Mask
1 byte	1 byte

Each bit of the mask corresponds to the logical state of the pin. The least significant bit corresponds to pin 3.

(GPIO N1), the most significant bit corresponds to pin 10 (GPIO N 8). If the pin is configured as an input, then the value reflects the logical state of the input. If the pin is configured as an output, then the value corresponds to the previously written value (see the command to set the value of the outputs GPIO)

2.2. Codes of the command execution result

0x00 - successful

0x01 - invalid command

0x03 - invalid channel number

0x04 - not executed (for example, an attempt to perform a move while the engine is already moving or searching for home position)

2.3. Algorithm for calculating 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, 0xedae, 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);
}
Call ushort crc = CRCCCITT (buf, 0, buffer_length, 0xFFFF, 0);
```

3. Modbus RTU communication protocol

3.1. Connection

Speed, parity, address is set in the configuration software. Default settings:

- Baud rate 115200 baud
- 1 stop bit
- No parity
- Address 1

3.2. Map "INPUTS" registers

Base address 1000. Total registers 160

№	Name	Description
0	MAJOR VERSION	The high word of the firmware version.
1	MINOR VERSION	The low word of the firmware version.
2	BOARD TYPE	Hardware version.
3	AXIS COUNT	Number of axes supported by the controller.
4 - 15	BOARD ID	Controller ID. 24 ASCII characters. Contains the serial number of the controller by default.
16 - 27	BOARD NAME	Board name. 24 ASCII characters. Default Contains the serial number of the controller.

28	POWER VOLTAGE	Supply voltage. Stored packaged. The most significant byte is the value to the comma, the least significant byte after the comma. Algorithm conversions: PowerVoltage = (regValue >> 8) + (regValue & 0xFF) / 100.0
29	USB VOLTAGE	USB voltage. Stored packaged. The most significant byte is the value to the comma, the least significant byte after the comma. Conversion algorithm: UsbVoltage = (regValue >> 8) + (regValue & 0xFF) / 100.0
Next are banks of state registers for each axis. 4 register per axis.		
30	STATUS AXIS1 HIGH	The high and low word of the status flags.
31	STATUS AXIS1 LOW	
32	POS AXIS1 HIGH	High and low word of the current axis position in microsteps. Position = (POS AXIS1 HIGH << 16) | POS AXIS1 LOW
33	POS AXIS1 LOW	
34	STATUS AXIS2 HIGH	Same as registers 30, 31
35	STATUS AXIS2 LOW	
36	POS AXIS2 HIGH	Same as registers 32, 33
37	POS AXIS2 LOW	
38	STATUS AXIS3 HIGH	Same as registers 30, 31
39	STATUS AXIS3 LOW	
40	POS AXIS3 HIGH	Same as registers 32, 33
41	POS AXIS3 LOW	
42	STATUS AXIS4 HIGH	Same as registers 30, 31
43	STATUS AXIS4 LOW	
44	POS AXIS4 HIGH	Same as registers 32, 33
45	POS AXIS4 LOW	
46	STATUS AXIS5 HIGH	Same as registers 30, 31
47	STATUS AXIS5 LOW	
48	POS AXIS5 HIGH	Same as registers 32, 33
49	POS AXIS5 LOW	
50 - 59	DUMMY 50 - 59	Reserved.
Next are banks of registers of settings for each axis. 20 registers per axis.		
60	CFG_FLAGS_AXIS1	Setting flags for axis # 1
61	CFG_DUMMY_AXIS1	Reserved.
62	CFG_MAX_POS_AXIS1_H	The maximum allowed position for the steps. In case you try to move the axis to a position that is greater than this, the command will be ignored.
63	CFG_MAX_POS_AXIS1_L	
64	CFG_DEC_AXIS1	Deceleration in full steps / s ^ 2
65	CFG_ACC_AXIS1	Acceleration in full steps / s ^ 2
66	CFG_MIN_SPEED_AXIS1	Initial speed of moving.
67	CFG_SPEED_AXIS1	Target (maximum) speed at moving

68	CFG_HOLD_RUN_CURRENT_AXIS1	Holding current value (high byte), moving current value (low byte). Each of them can take values 0 - 31. Algorithm for converting to milliamperes mA = (val + 1) * 51; For example, a value of 1 corresponds to a current of 103 mA
69	CFG_REF_ROLLOUT_STEPS_AXIS1_H	The distance from the “home” position sensor in steps.
70	CFG_REF_ROLLOUT_STEPS_AXIS1_L	
71	CFG_REF_SPEED_AXIS1	Departure speed from the "home" position sensor and / or limit sensor
72	CFG_REF_CUR_PWM_AXIS1	Current when leaving the home position sensor and / or limit sensor. Can take values 0 - 31. Algorithm for converting to milliamps mA = (val + 1) * 51; For example, a value of 1 corresponds to a current of 103 mA
73	CFG_REF_DELAY_ON_STOP_AXIS1	The delay between the arrival of the sensor and the start of the exit in milliseconds. If the value is 0 - no delay
74	DC_POWER_AXIS1	Power supplied to DC motor / solenoid, etc. Specified as a percentage.
75	DUMMY0_AXIS1	Reserved.
76	DUMMY1_AXIS1	
77	DUMMY2_AXIS1	
78	DUMMY3_AXIS1	
79	DUMMY4_AXIS1	
80	CFG_FLAGS_AXIS2	Bank of registers of settings for axis No. 2. See description of registers 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	Bank of registers of settings for axis No. 3. See description of registers 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	Bank of registers of settings for axis No. 4. See description of registers 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	Bank of registers of settings for axis No. 5. See description of registers 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. "HOLDING" registers

Base address 2000. Total registers 17. Five banks of three registers, plus 2 GPIO registers.

№	Name	Description
0	TARGET_1_H	High-order word of the command parameter for axis # 1.
1	TARGET_1_L	The least significant word of the command parameter for axis # 1.
2	CMD_1	Command code for axis # 1.
3	TARGET_2_H	Similarly, to registers 0 - 2. Axis # 2.
4	TARGET_2_L	
5	CMD_2	
6	TARGET_3_H	Similarly, to registers 0 - 2. Axis # 3.
7	TARGET_3_L	
8	CMD_3	
9	TARGET_4_H	Similarly, to registers 0 - 2. Axis # 4.
10	TARGET_4_L	
11	CMD_4	
12	TARGET_5_H	Similarly, to registers 0 - 2. Axis # 5.
13	TARGET_5_L	
14	CMD_5	
15	GPIO MODE MASK	GPIO operating mode
16	GPIO VALUE	Meaning of GPIO pins

3.4. Commands

For instance, command has been given to move to a given point and we want to move axis No. 1 to position 1000. In order to do this, we need to put to the “holding”

registers 0-1 value 0x000003E8 (0x0000 in register 0, 0x03E8 in register 1), and in register 2 the command code MoveAbs (0x0008).

Name	The code	Parameter	Description
MoveFw	1	The number of microsteps .	Move forward by the specified number of microsteps. It can execute the command while moving. If there is a search for a home position, the team will be ignored.
MoveBw	2	The number of microsteps .	Move back the specified number of microsteps. It can execute the command while moving. If there is a search for a home position, the command will be ignored.
Stop	3	Not	Stop moving or searching for home position.
MotorPower	4	On/off	Turn power on or off to the motor windings. If the parameter register contains 0 then must be turn off, otherwise turn on. It can perform the command during movement or while looking for a '"home" position.
SetCurSpeed	5	Speed value	Set maximum / current value speed. The value must be in the range [1- 32765]. It can perform the command during movement, in this case the controller will smoothly raise / decrease the speed to the specified one using the acceleration / deceleration settings.
FindHome	6	Not	Search for home position. Execution of the command is allowed only if the motor is stopped at the moment, otherwise the command will be ignored.
SetDcPower	7	Percentage of DC motor power.	Sets the power value of the DC motor or solenoid as a percentage of the maximum. Valid values are 1 - 100.
MoveAbs	8	Target position in microsteps.	Move the axis to the specified position. It can execute the command while moving. If you are searching for your home position command will be ignored.

3.5. GPIO

Two registers N 15 (GPIO MODE MASK), N 16 (GPIO VALUE) are available to control GPIO.

Only the least significant 8 bits in each register are used.

GPIO MODE MASK - contains the operating mode (input or output) for each GPIO pin. For Description of masks go to the command “set the GPIO mode” or “get the GPIO mode mask”.

GPIO VALUE - contains the logical state of the GPIO. See commands “set values GPIO outputs”, “get the values of the GPIO pins”. In the case of Modbus, a simplified option for writing values is used. Only all bits of the values can be written at the same time.

4. Axis status flags

Status flags are a 32-bit bit field.

In the case of the USB protocol, you can get the status with the 0x0A command (look at the Description).

For Modbus RTU see registers 30 - 31, 34 - 35, 38 - 39, 42 - 43, 46 - 47.

Pseudocode for getting values (flags is the value received by the command or from the registers):

```
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;
```

Description of fields:

IsOnline - true -> power supply of the channel is on, the channel driver is OK.

IsOverCurrent - true -> maximum current exceeded, possible short circuit in the circuit engine.

IsUnderVoltage - true -> power supply voltage too low (less than 6 volts) or the driver is defective.

IsOverHeat - true -> motor driver overheating.

IsMoving - true -> moving is in progress.

IsMotorOn - true -> power supply is applied to the motor windings.

IsSignalA - true -> the sensor connected to input “A” has triggered. For example, terminal

switch.

IsSignalB– true -> the sensor connected to input “B” has triggered. For example, terminal

switch.

IsSignalC– true -> the sensor connected to input “C” has triggered. For example, terminal

switch.

IsRefSearchRequired - true -> you need to search for the home position because current physical position may not correspond to the position value in microsteps. It may happen if there was a run over the limit switch or the power supply disappeared in the process of moving. See the command “Search home position” [USB], [Modbus RTU].

IsStopTriggered - true -> the end sensor has been run over.

LastMoveDirection - 0 -> the last of the move commands was backward, 1 - forward.

IsRefSearch - true -> home position is searched.