*SMART MOTOR DEVICES*

https://smd.ee

**PROGRAMMABLE STEP MOTOR
CONTROLLER SMSD-4.2RS**

***Manual***
*v.1.0*

**2021**

# 1. Product designation

Programmable step motor controller SMSD-4.2RS (the unit) is designed to operate with stepper motor with maximum current per phase up to 4.2 Amp. There are three control modes provided: programmable, analog speed control or driver (position pulse control).

# 2. Functions and possibilities

- Recording the operation algorithm from a computer to EEPROM of the unit as a sequence of ASCII commands.
- Reading the saved algorithm from the EEPROM to a computer;
- Operation in the programmable mode: control the stepper motor as per a user program, which is saved in the unit's memory. Speed, acceleration/deceleration, displacement, direction of the stepper motor are set as a sequence of execution commands and is storaged in the nonvolatile memory of the unit.
- Operation in the simple driver mode: the unit receives logic signals «PULS» and «DIR» – 0VDC low level and 5-24VDC high level.
- Operation in the analog speed control mode: the unit receives analog signal «Speed» (internal potentiometer) and digital signals «Reverse» and «Enable».
- For synchronized operation of several controllers SMSD-4.2RS or other devices there are 3 digital inputs and one output. These inputs and output are used in the program mode.
- The controller can operate and be controlled by a computer or in a standalone mode.
- There is zero positioning function provided: start zero search by a command or by a signal on the digital input. Stop zero searching movement as input signal is received. The function provides homing by an individual input.
- The unit stops motor motion as receives a signal on the «Enable» input.
- The unit changes rotation direction as receives a signal on the «Reverse» input.
- The microstepping can be changed on-fly.

# 3. Technical characteristic

Table1

| Common characteristic: | |
|---|---|
| Maximum output current per phase, Amp | 4.2 |
| Minimum output current per phase, Amp | 0.2 |
| Microstepping modes | 1/1, 1/2, 1/4, 1/16 |
| Voltage input, VDC | 12 – 48 |
| Dimensions,   mm   no more | 116x100x23 |
| **Inputs DIR, STEP (simple driver mode):** | |
| High voltage level, VDC<br>(Please, connect current-limiting resistance when use high level voltage more, than 5VDC: 1 KOhm for 12VDC, 2 KOhm for 24VDC) | 5-24 |
| Low voltage level, VDC | 0-1 |
| Input resistance, KOhm  no less | 3.0 |
| **Digital inputs EN, Reverse, IN1, IN2:** | |
| Contact to GND | |
| **Communication interface -  USB, virtual COM RS232:** | |
| - baud | 9600 |
| - bits | 8 |
| - parity | even |
| - stop bit | 1 |
| **Additional output «5VDC»:** | |
| Voltage, VDC | 5 |
| Maximum load current, mAmp | 20 |
| Resistance, Ohm | 27 |

**Environmental Conditions:**

Ambient Temperature: 0…+40℃

Humidity: 90% RH or less upon condition +25℃

Condensation and freezing: none

Pressure: 650…800 mm of mercury

Tallinn Science Park Tehnopol,
Akadeemia tee 21/6, 12618, Tallinn, Estonia.

Phone: + 372 6559914,
e-mail: sale@smd.ee
url: https://smd.ee
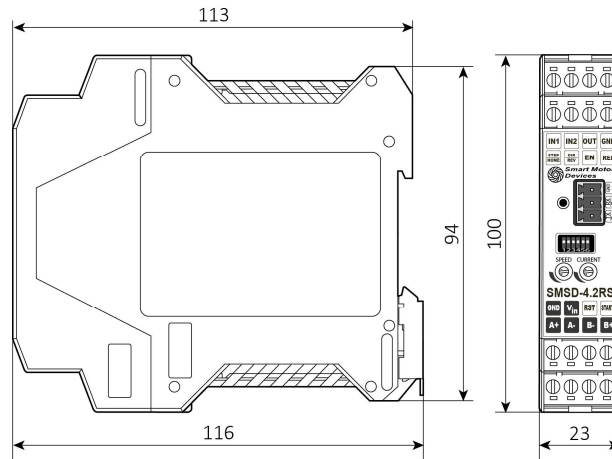
# 4. Construction



Fig.1. Overall dimensions.

SMSD-4.2RS is designed as a circuit plate with electronics components, LED indicator and control elements, terminals and connectors. The plastic case of the driver is designed to be mounted on a DIN rail. There are graphical symbols of the control elements and the terminals assignment on the case.

Besides the control elements at the driver frame there are:

- terminal screws for power supply, stepmotor windings and control circuit connection;
- RS-485/RS-232 terminal for a computer or PLC connection;
- internal potentiometer «SPEED» to control the motor speed in analog control mode;
- switches SW1 – SW2 to set the operation mode;
- switches SW3 – SW4 to set the microstepping mode;
- switch SW5 – to set the communication interface (RS-485 – ON, RS-232 – OFF);
- switch SW6 – to connect the built-in terminal resistor;
- internal potentiometer «CURRENT» to adjust the motor phase current;
- LED for indication of the controller status.

# 5. Assembly and connection

Please, learn this manual carefully before connection and assembly.
Please, wire just when power is off. Do not attempt to change wiring while the power is ON.
Please, provide a reliable contact in connection terminals. During wiring, please, observe the polarity and wire management.

The STEP/HOME and DIR/REV inputs have dual assignments depending on the selected operation mode. Examples of connection diagrams are shown in figures 2 - 5.
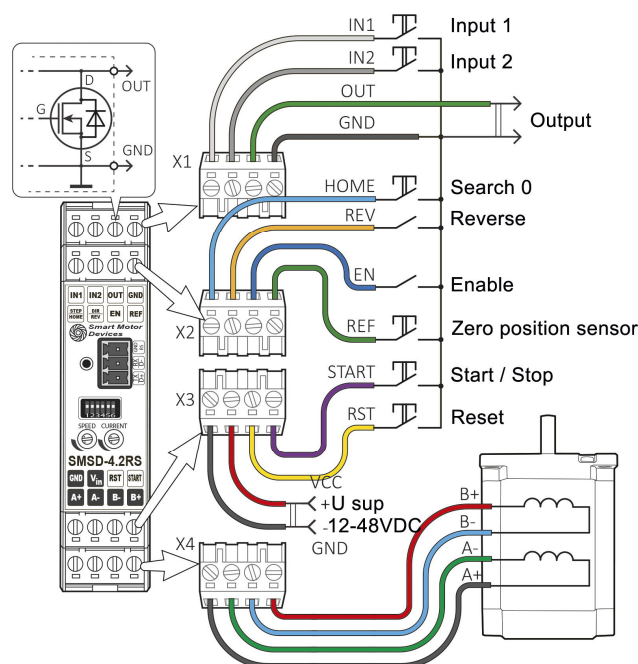


Fig.2. Example of connection in controller operation mode.

Tallinn Science Park Tehnopol,
Akadeemia tee 21/6, 12618, Tallinn, Estonia.

Phone: + 372 6559914,
e-mail: sale@smd.ee
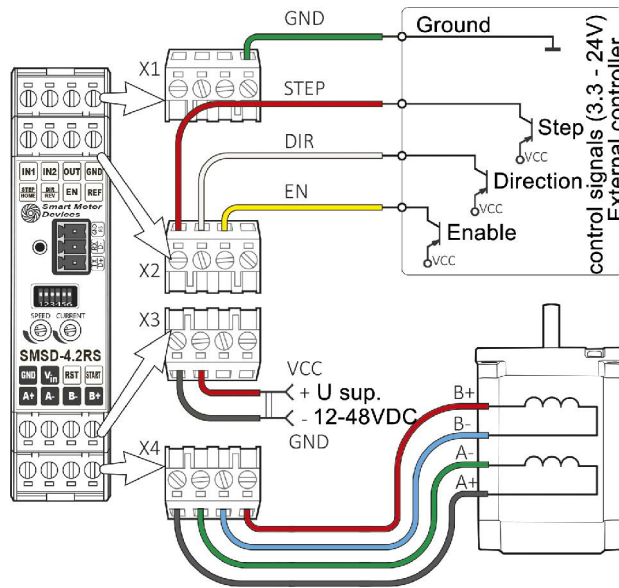url: https://smd.ee

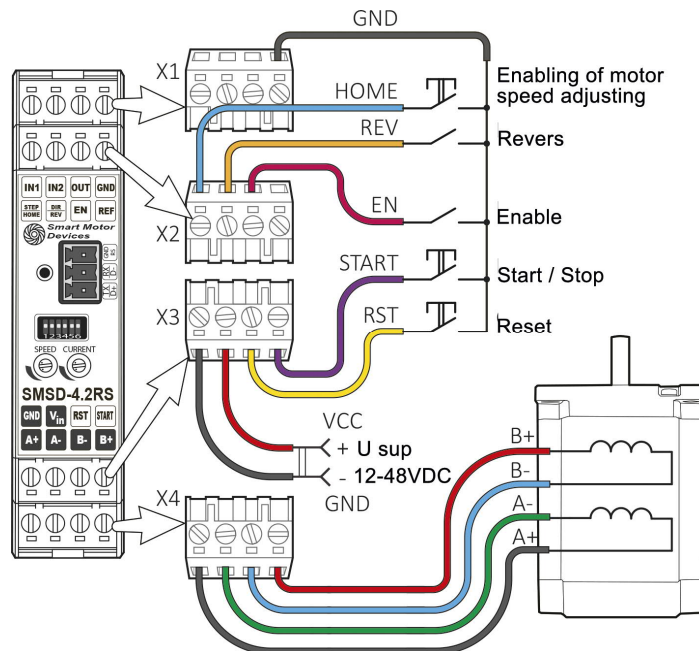Fig.3. Example of connection in driver operation mode.



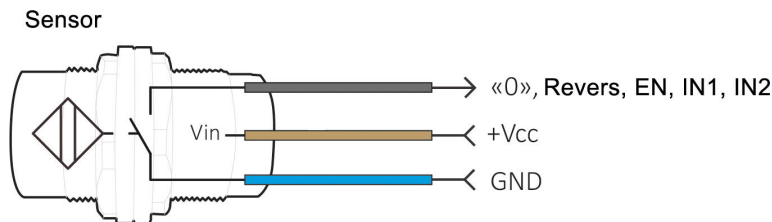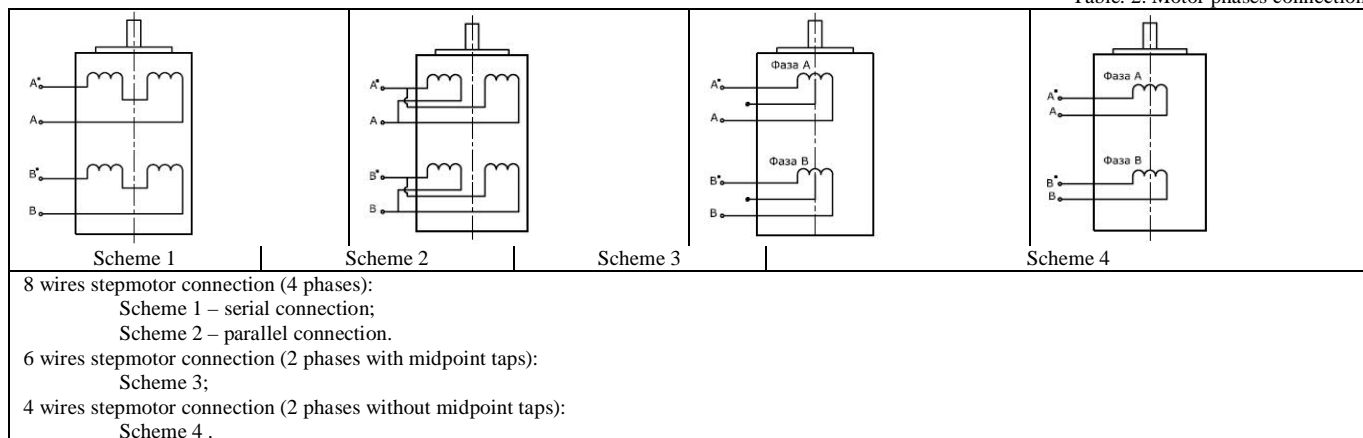Fig.4. Example of connection in analog speed control mode.



Fig.5. Example of NPN sensor connection.

When using the controller in driver mode, the STEP and DIR inputs have high input impedance. Noises can appear at the inputs, if a low signal level is provided only by the absence of a control voltage (not by applying a low signal level to the input). To eliminate input noise, it is recommended to connect a shunting resistor (1-10 kΩ) between the input and the ground of the X1 terminal.

The SMSD-4.2RS controller provides operation with 2 or 4-phase stepper motors, 4, 6 or 8 wires. Winding connection examples are in the table 2. Connect step motor wires to A, A*, B and B* terminals of SMSD-4.2.

Table. 2. Motor phases connection

| Scheme 1 | Scheme 2 | Scheme 3 | Scheme 4 |
|---|---|---|---|

8 wires stepmotor connection (4 phases):
     Scheme 1 – serial connection;
     Scheme 2 – parallel connection.
6 wires stepmotor connection (2 phases with midpoint taps):
     Scheme 3;
4 wires stepmotor connection (2 phases without midpoint taps):
     Scheme 4 .

# 6. Before starting

**Attention:** When working with a high inertia load, there is a possibility of forced rotation of the motor shaft when it is suddenly decelerated. In this case, as well as during forced rotation of the shaft, the motor produces EMF, which can damage the output stages of the controller. To avoid damaging, it is necessary to exclude abrupt decelerations, forced rotation, disconnection of the motor phases or disconnection of the power supply unit during operation.
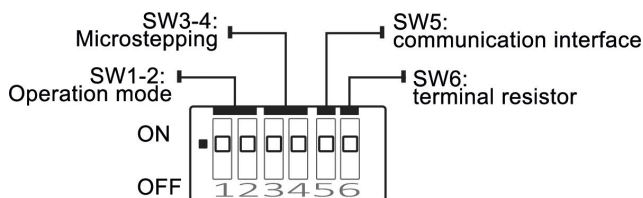


Fig.6. Switches SW1 – SW6.

1. Make sure the power supply is turned off.
2. Make assembly and connection according to the section 5.
3. Select the suitable operation mode and set microswitches SW1 and SW2 according to the table 3.

Table .3. Operation mode

| Operation mode | Microswitch | | Control |
|---|---|---|---|
| | SW1 | SW2 | |
| Programmable | ON | ON | As per the preset motion algorithm, saved in the controller's memory. Algorithm consists of a sequence of commands – simple instructions. |
| Simple driver | OFF | ON | Pulse and Dir logic signals – low and high level voltage. |
| Analog speed control | ON | OFF | Speed is adjusted by a potentiometer or analog signal 0-5VDC, direction is changed by a logic signal (pulse). |

4. Select suitable microstepping - set the microswitches SW3 and SW4according to the table 4.

Table 4. Microstepping

| | 1 | 1/2 | 1/4 | 1/16 |
|---|---|---|---|---|
| SW3 | ON | ON | OFF | OFF |
| SW4 | ON | OFF | ON | OFF |

5. Set suitable for the stepper motor current per phase using the potentiometer «Current». Adjust current value according to the figure. The output current, set by the potentiometer, should be set according to the motor's description. Low current leads to a weak torque of the motor, high current leads to the motor heating and can damage the motor.
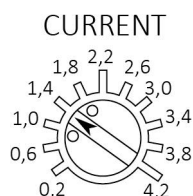


Fig. 7. Motor phase current

6. If necessary, connect the SMSD-4.2RS controller to a computer or PLC.
7. Check wiring once again and turn on the power supply unit.
8. If the operation mode should be changed after power is on, set SW1 and SW2 according to the table 3 and RESET the controller (use input Reset).
9. Set the communication interface using microswitch SW5: ON – RS485, OFF – RS232;
10. If need, connect the built-in terminal resistor using the microswitch SW6: ON – connected, OFF – disconnected;
11. Control the stepper motor according to the selected operation mode:

**The simple driver mode** (connection example – figure 3):

Set the required sequence of logic signals «STEP» and «DIR» according to the diagram below (fig. 8). One step (or microstep) executes as the front edge of the voltage pulse on the «STEP» input. Direction switches by changing voltage level on the «DIR» input. The motor can be stopped by the active signal on the «EN» input (clean contact of EN and GND).
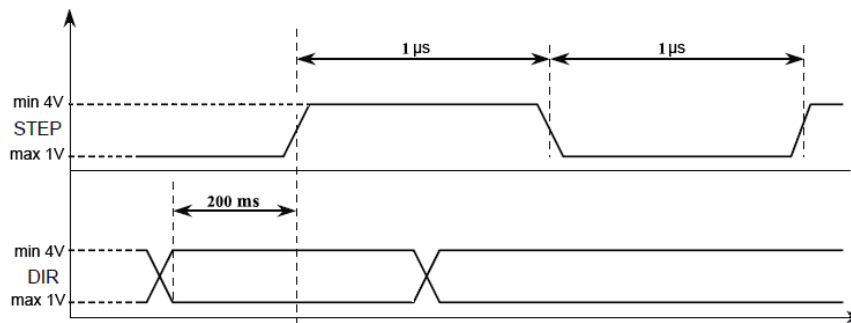


Fig. 8 - STEP and DIR input signals

The motor can also be controlled with a two-phase quadrature signal with a phase different of 90°. Such a signal can be obtained, for example, at the output of an optical selsyn. The connection diagram of the optical selsyn is shown in Fig. 9.
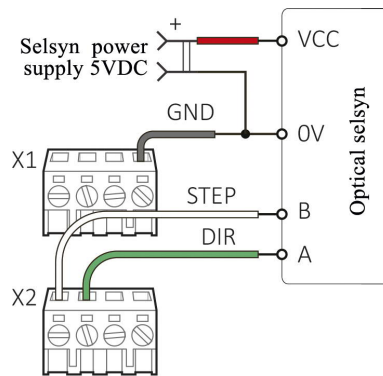


Fig. 9. Optical selsyn connection example

**Analog speed control mode** (connection example - figure 4):

Close contacts «START» and «GND» to run the motor, adjust speed by the built-in potentiometer «SPEED».  When exiting this control mode, the controller memorizes the current speed value. When re-entering the mode, the speed value is restored. The information of current speed as a decimal value is available via communication interface as an ASCII string. To change direction set signal to the «REVERSE» input – close contacts «REVERS» and «GND». Direction changes as the front edge of the signal.

When the "RESET" input is closed to the "GND" terminal, the motor stops, phases are de-energized.

When the power supply is applied to the controller and the motor is stopped in this mode, the green LED  is on, while the motor is rotating, the green LED blinks  (frequency is 1 time per second). When the contacts "EN" and "GND" are closed, the emergency mode is activated, the motor stops, the orange LED is blinking.

**Program mode** (connection example - figure 2):

Configure the connection port according to the parameters specified in Table 1. To control the drive via a computer or PLC the connection through RS-232 or RS-485 should be provided. For a standalone operation the executing program should be saved to the controller in advance via RS-232 or RS-485, after that the communication cable can be disconnected.
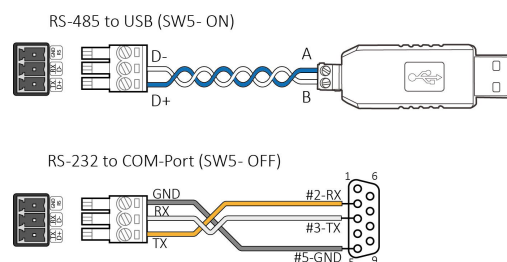


Fig. 10. Communication interfaces connection

The RS-232 data transmission interface allows 1 controller to be connected to the network.

The RS-485 data transmission interface allows connecting up to 32 controllers to the network. A terminal resistor must be connected to the end of the RS-485 data line on the controller using switch SW6 (ON).
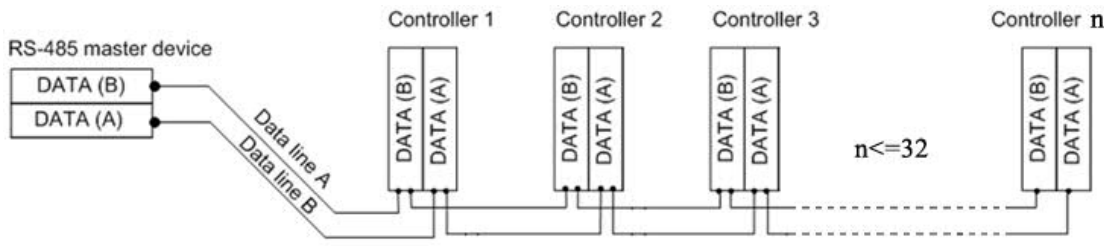
Fig. 11. RS-485 connection

After uploading the user program into the controller memory, it is possible either to continue to work using the RS-485/RS-232 interface, or disconnect the controller from the master device (PC or PLC) and work as a standalone unit according to the user program. To start the program executing, it is necessary to close the «START» and «GND» pins in the controller standby mode.

# 7. Data communication protocol

Commands are sent to the controller as frames. A frame is a string of ASCII characters. The frame format is shown in the Table 5.

Table 5. Data transmission frame format

| Start of a frame | ID address | Command | Parameter | End of frame |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
| : | XX (hex) | CC | NN (Dec) | <CR> + <LF> (\r\n или 0x0D 0x0A) |

1: ":" Start of the frame
2: "xx" - address of the controller in the network (from 01 to FF), hexadecimal. Address 00 is used for broadcast instructions.
3: "CC" - command code. Please, refer to the tables 6 and 7l for the list instructions..
4: Command parameter, decimal.
5: <CR> + <LF> (\r\n or 0x0D 0x0A) - end of the frame

Frame example: the ASCII string ": 01SD100 \ r \ n" means the command to the controller with the ID address 0x01 to set the motor rotation speed 100 steps per second.

The instructions in the Tables 6 and 7 are presented for user program operation mode (the program is uploaded to the controller, afterwards the controller can be either disconnected from the master device or continue operation as a network slave) and for the "Direct control" mode (real time control is performed by a master device).

Table 6. List of control commands.

| № | Instruction | User program mode | «Direct control» mode |
|---|---|---|---|
| 1 | LD (LD1) | Entering the program uploading mode, the user program executing is terminated, the motor phases are de-energized. | - |
| 2 | RD | Read the user program from the controller memory | - |
| 3 | ST (ST1) | Start or stop user program executing. (If the program is being executed, it is stopped by the command ST. If the user program is not being executed, the controller starts operation according to the user program). | |
| 4 | LB | - | Start loading to the operational buffer. Executing of a previous command is stopped. All executing commands, which are sent to the controller after LB instruction, are saved as a buffered commands sequence. The buffer is completed after receiving ED instructions. |
| 5 | RB | - | Read the command sequence from the operational buffer. Executing of a previous command is stopped. |
| 6 | SBddd | - | Start the commands sequence of the operational buffer executing ddd times. ddd is the commands parameters, sets the value how many times the program from the operational buffer should be executed (from 1 to 255 repetitions). If it is necessary to stop executing commands from the buffer, use ST command. |
| 7 | ED | Complete executing commands sequence. After accepting this command the controller records all transferred executing commands into the memory and turns to the standby mode. | Complete executing commands sequence. After accepting this command the controller record all commands to operational buffer and turns to the standby mode |
| 8 | IDdd | Set new ID address. "dd" is a new ID address in decimal format (from 1 to 255). As an example, the frame «:00ID93\r\n» will set address 93 (0x5D) to the controller. The controller won't reply this command (as the starting address is 00). If then send the frame «:5DID1\r\n» the new address – 1 (0x01) will be assigned to this controller and the controller will send the reply: «:01E10\r\n». | |
| 9 | EM | Turns on data transfer to a client after receiving of a broadcast instruction. As an example, if send broadcast instruction "ST" (start/stop operation) to all controllers, send then "EM" command if need the controllers to inform about finish of the operation (message "E14" is sent from a controller as the operation is finished). | |

Table 7. List of executing commands.

| № | Instruction | User program mode | «Direct control» mode |
|---|---|---|---|
| 10 | BG | Set address pointer as 0 | |
| | | Begin - start a new user program, the previous sequence is cleared. | Begin - start a new algorithm in the buffer, the previous sequence is cleared. |
| 11 | EN | Set «enable» – turn on the motor phases | |
| 12 | DS | Set «disable» – turn off the motor phases | |
| 13 | DL | Forward motion | *Forward motion*<br>- if the motor is stopped, the start of the next motion will be in this direction;<br>- if this command was received while driving, the remaining steps are processed in the direction according this command |
| 14 | DR | Backward motion | *Backward motion*<br>- if the motor is stopped, the start of the next motion will be in this direction;<br>- if this command was received while driving, the remaining steps are processed in the direction according this command |
| 15 | RS | Reverse | *Reverse*<br>- if the motor is stopped, the start of the next motion will be carried out in the opposite direction;<br>- if this command was received while driving, the remaining steps are processed in the opposite direction |
| 16 | AL(-)ddd | Acceleration ddd: min = -1000, max =1000 | *Acceleration*<br>- if the motor is stopped, the start of the next motion will be executed according to the new acceleration value, start speed SS and final speed SD;<br>- if this command was received while driving, after receiving the next SD value, the remaining steps are processed according to the new parameters (accelerating or decelerating will be executed to reach the new value of speed SD).<br>Default value AL = 0 |
| 17 | SDddd (1 to 10000) | Speed ddd<br>If start speed (SS) and acceleration (AL) are not specified, the value SD is used as a constant speed | *Speed*:<br>- if the motor is stopped, the start of the next motion will be carried out with this speed; if acceleration is used this command sets the final speed; if there is no acceleration this speed will be used as start speed;<br>- if this command was received while driving, the remaining steps are processed with this speed |
| 18 | SSddd (1 to 2000) | Start speed ddd:<br>This command is used only for starting motion and only when acceleration is used (*AL≠0*).<br>When accelerating AL ≠ 0, acceleration (deceleration) to SD speed is performed.<br>The SS command is only taken into account when starting the movement (motor stopped).<br>Default value SS = 1 | |
| 19 | SF | Turn the relay on | |
| 20 | CF | Turn the relay off | |
| 21 | MV | Start continuous movement | *Continuous motion*<br>- if the motor is stopped, this command starts perpetual motion;<br>- if this command was received while driving, continuous motion with current speed begins |
| 22 | MVddd (1 to 10000000) | Move to ddd steps | *Motion on number of steps* (min 1pul – 10000000pul)<br>- if this command was received when the motor is stopped, the motor will make *ddd* steps and stop;<br>- if this command was received while driving, the motor will make *ddd* steps and stop, previous command will be considered complete |
| 23 | MH | Indefinite movement, till signal to input IN2 | *Continuous movement, till signal to input IN2*<br>- if this command is received when the motor is stopped, continuous motion will begin till signal to input IN2;<br>- if this command is received while driving, from this moment the motor will start continuous movement with the current speed till signal to input IN2 |
| 24 | ML | Indefinite movement, till signal to input IN1 | *Continuous movement, till signal to input IN1*<br>- if this command is received when the motor is stopped, continuous motion will begin till signal to input IN1;<br>- if this command is received while driving, from this moment the motor will start continuous movement with the current speed till signal to input IN1 |
| 25 | HM | Indefinite movement, till signal to input «0» (zero limit switch) | Continuous movement, till signal to input «0» (zero limit switch)<br>- if this command is received when the motor is stopped, continuous motion will begin till signal to input «0»;<br>- if this command is received while driving from this moment the motor will start continuous movement with the current speed till signal to input «0» |
| 26 | SPddd | Pause for ddddd ms, max = 100000000 | Pause: min 1 ms – max 10000000 ms<br>After receiving this command the motor will stop and wait for ddd ms and then will continue to execute the remaining steps.<br>If in pause time the commands MV, MVddd, MH, ML, HM a |

Tallinn Science Park Tehnopol,
Akadeemia tee 21/6, 12618, Tallinn, Estonia.

Phone: + 372 6559914,
e-mail: sale@smd.ee
url: https://smd.ee

| | | | received, they are executed immediately and previous command is considered complete |
|---|---|---|---|
| 27 | LL | Set label for cycle operation | Set label for cycle operation – for program in buffer |
| 28 | JPddd (1 to 255) | Repeat from label ddd times, мax=255. Set 2 commands JPddd to create endless cycle. | Repeat from label ddd times, max 255. Set 2 commands JPddd to create endless cycle. The command is available in direct control mode to assemble a sequence in the operational buffer. |
| 29 | WL | Indefinite pause, wait for a signal to input IN1 | Indefinite pause, wait for a signal to input IN1 - if this command is received when the motor is stopped, next command will be executed after signal on the input IN1; - if this command is received while driving, from this moment the motor is stopped waiting for a signal to input IN1 the previous command is considered complete |
| 30 | WH | Indefinite pause, wait for a signal to input IN2 | EIndefinite pause, wait for a signal to input IN2 - if this command is received when the motor is stopped, next command will be executed after signal on the input IN2; - if this command is received while driving, from this moment the motor is stopped waiting for a signal to input IN2, the previous command is considered complete |

**Attention:** RS-485 interface doesn't mechanisms for dealing with collisions, which can be caused by simultaneous data transfer. The customer should provide the data not to be sent simultaneously. Collisions could be caused in case if 2 or more controllers with the same address operate in one RS-485 network. In case of use "EM" command, the customer should keep a close watch on the controllers, which can send reply "E14" simultaneously (message about the end of operation).

The value of accelerating is checked when the drive operates in the «Direct control»mode:
1. If AL=0, the speed is stepping incremented to SD value, start speed SS is not taken into account.
2. If the motor is not rotating: if at the motion start moment the value $(SD-SS)*AL < 0$, the acceleration and start speed values are ignored, the motor starts motion at SD speed.
3. If the motor is rotating: if the value $(SD-Scurrent)*AL < 0$ at the moment of receiving the AL command, acceleration value is ignored, the motor stepping increments the speed to SD value, the remain steps are executed according to the new SD speed.

Execution of the user program by the controller consists in reading the next command from the non-volatile memory and controlling the motor correspondingly. At the same time the controller checks and handles the signals at the inputs "IN1", "IN2", "REV" and "EN". It is possible to change the direction of rotation by closing the inputs "GND" and "REV", in this mode the signal operates on the edge.

Table 8. Controller response codes.

| Reply | Description |
|---|---|
| E10* | The command is successfully accepted |
| E13* | There is the error in the executing program |
| E14* | Program executing is completed |
| E15* | Communication Error (check port parameters) |
| E16* | Command error (check controller mode or ASCII code of the command) |
| E19* | Command data error (check command data – integer, in allowed range, see table 6 and 7) |

## 8. Programmable mode - submodes

While operating in a program mode, the controller goes in one of the sub-modes:

**Standby mode** – green color LED indicator.

The controller goes to the standby mode in one of the next cases:
- After program executing completed;
- After program executing was stopped by command «ST» (refer to the table 6).
- After power is on (if microswitches SW1=On and SW2=On);
- After «RST» input is activated (if microswitches SW1=On and SW2=On).

In the standby mode the controller waits for one of the next events:
- Arrival one of the control commands from a computer or PLC (table 6 and 7);
- Start executing program by activating the input signal «START» (close «START» and «GND»);
- Turn to the other operation mode: analog control or simple driver mode (table 3).

**User program executing mode** – blink red color LED indicator.

The controller turns to the user program executing mode and starts motion algorithm from the standby mode in case of:
- «START» signal is activated (close «START» and «GND»);
- Arrival the control command «ST1;

The controller turns back from the user program executing to the standby mode in case of:
- Executing of the user program is completed;
- Command «ST» comes from the PC or PLC;
- «RST» signal is activated (close «RST» and «GND»).

The commands of the sequence (operation algorithm is saved to the controller as a user program) are executed one by one, as they were recorder to the controller. The controller commutates the motor windings according to the executing commands. At the same time SMSD-4.2RS controls the inputs state for «IN1», «IN2», «EN», «REV.

Active «EN» signal suspends and inactive signal resumes the user program executing. Active «REV» signal changes motion direction (by the front edge). «IN1» and «IN2» signals are handed according to the current executing command.

**Waiting for an external signal submode** – red color blinking (double blinking in 1 sec period) LED indicator.

Tallinn Science Park Tehnopol,
Akadeemia tee 21/6, 12618, Tallinn, Estonia.

Phone: + 372 6559914,
e-mail: sale@smd.ee
url: https://smd.ee

![SMART MOTOR DEVICES](http://smd.ee)
The controller turns to this submode during program executing, as per the command «WH» or «WL» – waiting for an external signal. The controller suspends program executing till receiving active signal to input IN1 or IN2. As the signal arrives to the input (as per the command) the controller resumes program executing.

**Commands loading to the controller memory** – orange color LED indicator.

The controller turns to this mode from the standby mode as receives the command «LD». In the commands loading mode the controller accepts the executing commands only (table 6). After arriving the command «ED» the controller records all received executing commands to the memory and turns to the standby mode.

**Reading commands from the controller** – orange color LED indicator.

The controller turns to this mode from the standby mode. The sequence of executing commands from the controller memory transfers to the computer as the controller receives the command «RD». After the commands transferred the controller turns back to the standby mode.

**Temporary stop mode** – orange color LED indicator.

The controller turns to this submode during program executing as receives the signal «EN» (close «EN» and «GND»). The controller suspends program executing while the signal is active. As the signal turns to inactive the controller resumes the program executing.

**Executing program error** – blink orange color LED indicator.

The controller turns to this submode during program executing in case of wrong command in the sequence. The controller exits this mode when «RST» input is activated (close «RST» and «GND»). The executing program should be recorded again to the controller memory.

## 9. Delivery in complete sets

Programmable stepper motor controller SMSD-4.2RS                     1 pcs.

## 10. Manufacturer information

Smart Motor Devices adheres to the line of continuous development and reserves the right to make changes and improvements in the design and software of the product without prior notice.

The information contained in this manual is subject to change at any time and without prior notice.

## 11. Warranty

**Any repair or modifications are performed by the manufacturer or an authorized company.**
The manufacturer guarantees the failure-free operation of the controller for 12 months since date of sale when the operation conditions are satisfied.

The manufacturer sales department address:
Smart Motor Devices OÜ,
Tallinn Science Park Tehnopol, Akadeemia tee 21/6, 12618, Tallinn, Estonia
Phone: + 372 6559914,
e-mail: mail@smd.ee
url: https://smd.ee

Date of sale:

*Last modified: 27.04.2023*

Tallinn Science Park Tehnopol,
Akadeemia tee 21/6, 12618, Tallinn, Estonia.

Phone: + 372 6559914,
e-mail: sale@smd.ee
url: https://smd.ee