# PROGRAMMABLE STEP MOTOR CONTROLLER R272-42-RS

Manual v.1.0



### 1. Product designation

Programmable step motor controller R272-42-RS (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 R272-42-RS 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	5-24
(Please, connect current-limiting resistance when use high level voltage more, than 5VDC: 1 KOhm for 12VDC, 2 KOhm for 24VDC)	
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°C Humidity: 90% RH or less upon condition +25°C Condensation and freezing: none Pressure: 650...800 mm of mercury

### 4. Construction



Fig.1. Overall dimensions.

R272-42-RS 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.



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  $(1-10 \text{ k}\Omega)$  between the input and the ground of the X1 terminal.

The R272-42-RS 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.



### 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.



- Make sure the power supply is turned off. 1.
- 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
On senting mode	Microswitch		Control
Operation mode	SW1	SW2	Conuor
Drogrommoble	ON	ON	As per the preset motion algorithm, saved in the controller's memory.
Tiogrammable			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.

ii beieet suitueite iiiteit	set the meros are	s s i s und s i i ideeording to the		
				Table 4. Microstepping
	1	1/2	1/4	1/16
SW3	ON	ON	OFF	OFF
SW4	ON	OFF	ON	OFF

Set suitable for the stepper motor current per phase using the potentiometer «Current». Adjust current value according to the figure. The output current, 5. 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

- If necessary, connect the R272-42-RS controller to a computer or PLC. 6.
- 7. Check wiring once again and turn on the power supply unit.
- 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). 8.
- 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)	$\langle CR \rangle + \langle LF \rangle (\langle r \rangle n \ 0x0D \ 0x0A)$

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:  $\langle CR \rangle$  +  $\langle LF \rangle$  ( $\langle r \rangle n \text{ or } 0x0D \ 0x0A$ ) - end of the frame

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

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).

No	Instruction	User program mode	«Direct control» mode		
• 1-	mstruction	Entering the program uploading mode,			
1	LD (LD1)	the user program executing is terminated, the motor phases are	-		
		de-energized.			
2	RD	Read the user program from the controller memory	-		
2	<b>ST (ST1</b> )	Start or stop user program executing. (If the program is being exec	cuted, it is stopped by the command ST. If the user program is not being		
3	51 (511)	executed, the controller starts operation according to the user prog	gram).		
			Start loading to the operational buffer.		
			Executing of a previous command is stopped. All executing commands,		
4	LB	-	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	DD		Read the command sequence from the operational buffer.		
3	KD	-	Executing of a previous command is stopped.		
			Start the commands sequence of the operational buffer executing ddd		
			times. ddd is the commands parameters, sets the value how many times		
6	SBddd		the program from the operational buffer should be executed (from 1 to		
0		-	255 repetitions).		
			If it is necessary to stop executing commands from the buffer, use ST		
			command.		
		Complete executing commands sequence. After accepting this	Complete executing commands sequence. After accepting this command		
7	ED	command the controller records all transferred executing	the controller record all commands to operational buffer and turns to the		
,		commands into the memory and turns to the standby mode.	standby mode		
		Sat new ID address, "dd" is a new ID address in decimal format (f	(rom 1 to 255)		
		As an example, the frame <i>(</i> :001D03/r/n), will set address 03 (0x51	(as the controller. The controller won't renly this command (as the starting		
8	IDdd	As an example, the name " $(00175)$ (1) will set ductess 55 (0x51) to the controller. The controller work reply this controller will as the starting address is 00). If then send the frame "(5DID1) who has not address = 1 (0x01) will be assigned to this controller and the controller will can d			
		address is 00). If then send the frame $\ll$ :5D1D1/r/n» the new address the reply: $\ll$ 01E10/r/n»	ss = 1 (0x01) will be assigned to this controller and the controller will send		
		Turne on data transfor to a client ofter receiving of a breadcast inc	terration		
0	EM	As an axample if and broadcast instruction "ST" (start/star area	uucuon.		
9	EM	As an example, it send broadcast instruction S1 (start/stop oper inform about finish of the operation (massage "E14" is sent from	ation to an controllers, send then EW command if need the controllers to		
		morm about missi of the operation (message E14 is sent from a	a controller as the operation is ministed).		

Table 7. List of executing commands.

№	Instruction	User program mode	«Direct control» mode
		Set ad	dress pointer as 0
10	BG	Begin - start a new user program, the previous sequence is	Begin - start a new algorithm in the buffer, the previous sequence is
		cleared.	cleared.
11	EN	Set «enable» –	turn on the motor phases
12	DS	Set «disable» –	turn off the motor phases
			Forward motion
			- if the motor is stopped, the start of the next motion will be in this
13	DL	Forward motion	direction;
			- if this command was received while driving, the remaining steps are
			processed in the direction according this command
			<u>Backward motion</u>
1.4	DD		- if the motor is stopped, the start of the next motion will be in this
14	DR	Backward motion	direction;
			- If this command was received while driving, the remaining steps are
			<u>Reverse</u> if the motor is stonned the start of the payt motion will be carried out in
15	RS	Reverse	the opposite direction:
15	KJ	Ke vei se	if this command was received while driving the remaining steps are
			- If this command was received while driving, the remaining steps are
			Acceleration
			- 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.
16	AL (-)ddd	Acceleration ddd: $\min = -1000 \max = 1000$	- if this command was received while driving after receiving the next SD
10	/ IE( )uuu		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).
			Default value $AL = 0$
			Speed:
	00.111	0 1111	- if the motor is stopped, the start of the next motion will be carried out
17	SDddd	Speed ddd	with this speed; if acceleration is used this command sets the final speed;
17	(1 to	If start speed (SS) and acceleration (AL) are not specified, the	if there is no acceleration this speed will be used as start speed;
	10000)	value SD is used as a constant speed	- if this command was received while driving, the remaining steps are
			processed with this speed
		Start speed ddd:	
	SSddd	This command is used only for starting motion and only when ac	celeration is used ( $AL\neq 0$ ).
18	(1  to  2000)	When accelerating $AL \neq 0$ , acceleration (deceleration) to SD spec	ed is performed.
	(1 to 2000)	The SS command is only taken into account when starting the mo	ovement (motor stopped).
10	65	Default value $SS = 1$	
19	SF	1Ui	rn the relay on
20	Cr	1 ui	Continuous motion
			if the motor is stopped this command starts perpetual motion:
21	MV	Start continuous movement	- if this command was received while driving continuous motion with
			current sneed begins
			Motion on number of steps (min 1nul – 1000000nul)
	MVddd		- if this command was received when the motor is stopped the motor will
22	(1 to	Move to ddd steps	make <i>ddd</i> steps and stop:
	10000000)	<u>I</u> .	- if this command was received while driving, the motor will make <i>ddd</i>
1	,		steps and stop, previous command will be considered complete
			Continuous movement, till signal to input IN2
1			- if this command is received when the motor is stopped, continuous
22	3.411	To definite mean with the last to the DIO	motion will begin till signal to input IN2;
23	MH	indefinite movement, till signal to input IN2	- if this command is received while driving, from this moment the motor
1			will start continuous movement with the current speed till signal to input
			IN2
			Continuous movement, till signal to input IN1
1			- if this command is received when the motor is stopped, continuous
24	МІ	Indefinite movement till signal to input IN1	motion will begin till signal to input IN1;
24	1111	incomme movement, an signar to input fivi	- if this command is received while driving, from this moment the motor
1			will start continuous movement with the current speed till signal to input
1			continuous movement, till signal to input «0» (zero limit switch)
1			- IT THIS COMMAND IS RECEIVED WHEN THE MOTOR IS STOPPED, CONTINUOUS
25	НМ	Indefinite movement, till signal to input «0» (zero limit	motion will begin till signal to input «0»;
		switch)	- if this command is received while driving from this moment the motor
1			will start continuous movement with the current speed till signal to
			input «0»
			Pause: min 1 ms – max 10000000 ms
24	CD444	Dauss for dddd ma m 10000000	After receiving this command the motor will stop and wait for ddd
20	SPUUU	rause 101 uuuuu 1115, 111ax = 100000000	ms and then will continue to execute the remaining steps.
			If in hause time the commands MV_MVddd_MH_MI_HM_a

			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, max=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	Elndefinite 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 R272-42-RS 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.

The controller turns to this submode during program executing, as per the command wWH or wWL – 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.