User Manual

Stepper Motor Controller
SMC2242/SMC4242

Notices

Edition

April 27, 2016

Warranty

The material contained in this document is provided „as is“, and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, LK-Instruments disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.

LK-Instruments shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should LK-Instruments and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

Technical Assistance

If you need product assistance or if you have suggestions, contact LK-Instruments. You will find the contact information on the LK-Instruments homepage at

`http://www.lk-instruments.com`

Representatives of LK-Instruments are available during standard German business hours. Before you contact LK-Instruments, please note the actions you took before you experienced the problem. Then describe those actions and the problem to the technical support engineer.

Found a mistake?

We encourage comments about this publication. Please report any mistakes to

`support@lk-instruments.com`

Contents

List of Figures

List of Tables

Thank you for purchasing the SMCx242 Stepper Motor Controller. This user manual will explain you how to operate the Stepper Motor Controller. Please take some time to read it carefully before operating the device. Keep the instructions in a safe place for future reference.

1 Introduction

The SMCx242 Stepper Motor Controller is a universal controller for bipolar stepper motors. It can be used to drive two (SMC2242) or four (SMC4242) stepper motors. It is designed to work perfectly together with our assortment of stepper motor driven stages, but it might be also used to drive 3rd party motors or stages.

1.1 Package contents

Please check that the package contains all the following items:

- SMCx242 Stepper Motor Controller
- USB cable
- User Manual

1.2 Intended Use

The SMCx242 Stepper Motor Controller is intended to be used only by qualified experts inside scientific research laboratories. Do not operate the device, if you are not qualified.
Further the Stepper Motor Controller is designed for the inside use. Therefore it should not be used outside and kept away from water and moisture. Do not operate the unit near any heat sources or in direct sun light. When installing the device ensure to put it on a flat and leveled surface. Also ensure that there is adequate space around the device for ventilation.
After moving the unit to a different location, condensation inside the unit may occur. In this case please wait for some hours before you connect the Stepper Motor Controller to the power supply.
Connect the device only to power sources that meet the specifications written on its rear panel.

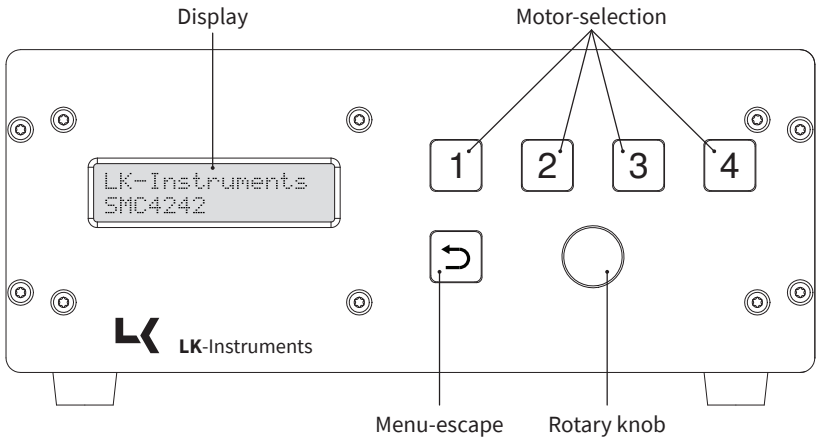


Figure 1.1: Front view of the SMCx242 Stepper Motor Controller.

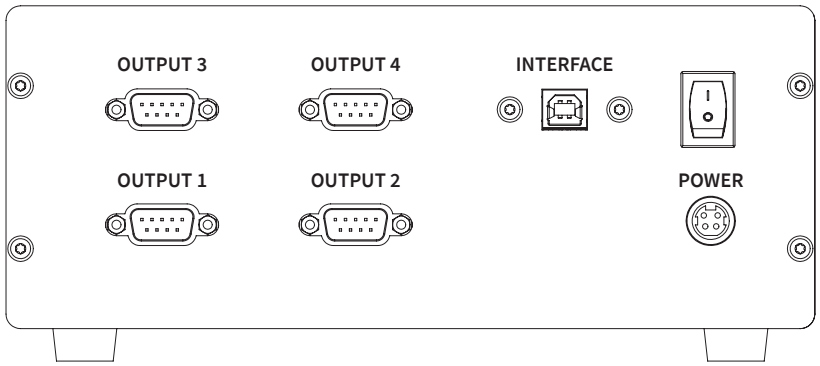


Figure 1.2: Rear view of the SMCx242 Stepper Motor Controller.

2 General Operation

The SMCx242 Stepper Motor Controller features two operation modes. It can be either operated using the manual user interface or remote controlled by a computer. In this section the manual operation of the device is explained.

After turning on the Stepper Motor Controller it comes up with its start screen. Turning the rotary knob serves to scroll through the menus (see figure 2.1). Pressing the rotary knob enters the selected menu. Pressing the menu-escape-button leaves the menu again.

2.1 Display structure

In almost every menu four values are displayed. Depending on the previously selected menu the values correspond to different quantities. Where

- the upper left value accompanies to Motor 1
- the upper right value accompanies to Motor 2
- the lower left value accompanies to Motor 3
- the lower right value accompanies to Motor 4

2.2 Motor selection

To select a motor there are four buttons. Motor selection can solely be done in an entered menu. To select a motor, press the respective motor-selection button. A selected motor is signed with an arrow on the display and by a color change of the corresponding motor-selection button. Pressing the selection-button again deselects the motor. Once a motor is selected its appropriate value can be changed by turning the rotary knob. If multiple motors are selected at the same time, there values will be changed simultaneously. By leaving a menu without motor deselection the selected motor(s) stay selected in any other menu.

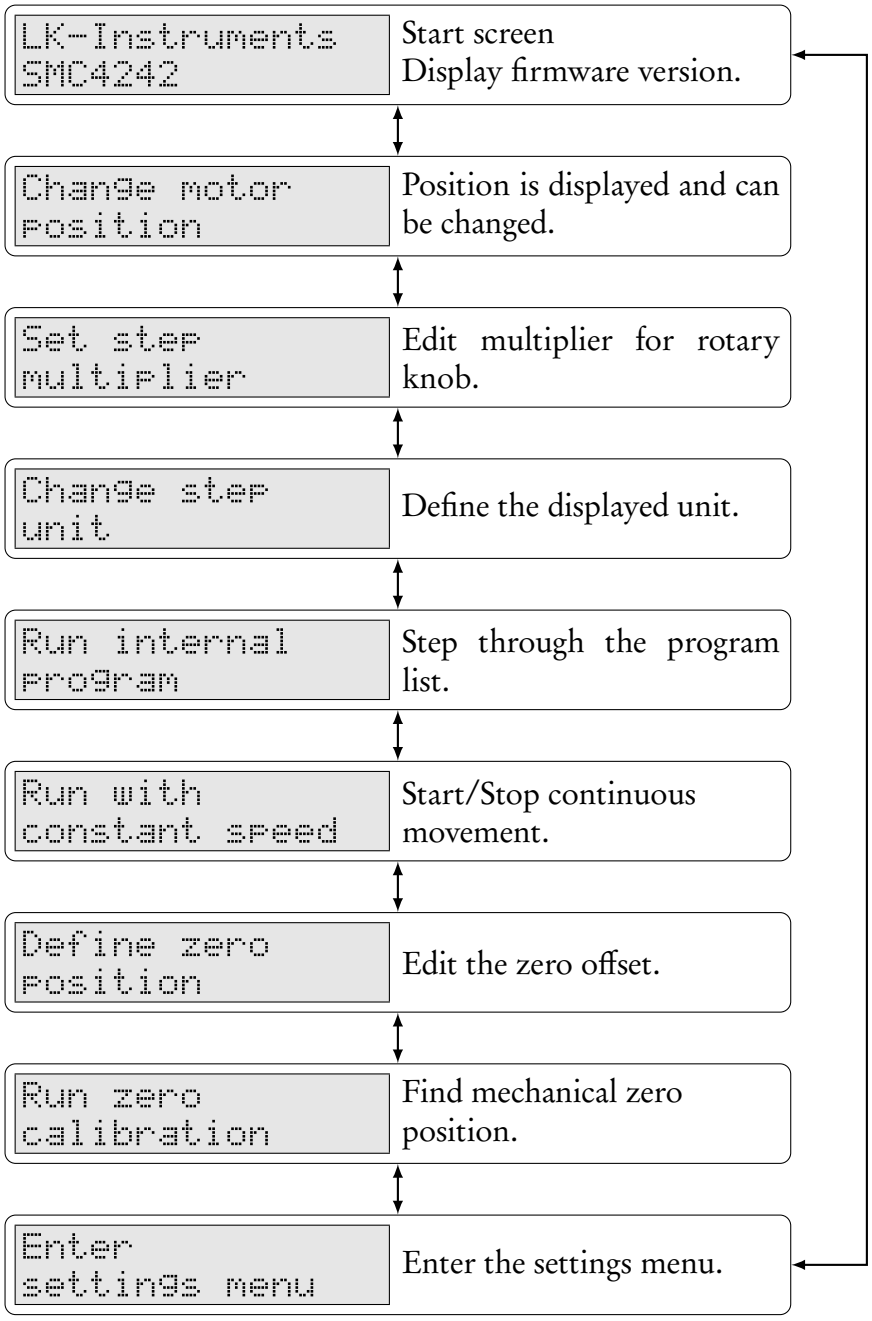


Figure 2.1: Overview of the available menus. By turning the rotary knob one can navigate through the menus as indicated by the arrows.

2.3 Start screen

By pressing the rotary knob while the start screen is active the firmware version will be displayed.

Firmware updates are available at

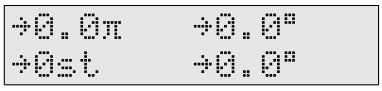
`http://www.lk-instruments.com`

on the corresponding product website.

2.4 Motor position

This menu displays the current motor positions and allows them to be changed. The default display unit is degree, which can be changed to the users preferred unit, see 2.6. The position of a selected motor can be changed by turning the rotary knob.
Default steps for the available units are:

- 1° if unit is degree
- $\frac{\pi}{8}$ if unit is radian
- 1 step if unit is steps



Fast moving mode When pressing the rotary knob inside the change-motor-position-menu one enters the fast moving mode. Pressing the rotary knob again disables the fast moving mode. The fast moving mode is indicated by another marking arrow for the corresponding motor.
Default steps in this mode are:

- 10° if unit is degree
- $\frac{\pi}{8}$ if unit is radian
- 100 steps if unit is steps

The snapped display shows the different indicating arrows.

2.5 Step multiplier

In this menu the step multipliers can be adjusted. If the step multiplier differs from 1.0 the corresponding motor will rotate more or less steps with each click of the rotary knob. It is also possible to turn the motors in different directions by applying a negative step multiplier to a motor. A step multiplier can only be applied if the step unit is degree or radians. The factory default value for all motors is 1.0.

Example: If the step multiplier for motor 1 is 1.0, the step multiplier for motor 2 is 4.0 and the step multiplier for motor 3 is -2.0, when changing all motor positions, motor 2 will move four times more steps than motor 1 and motor 3 will move twice as many steps as motor 1, but in the opposite direction.

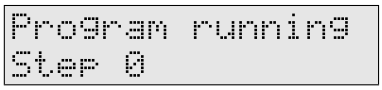
2.6 Step unit

In this menu one can choose the unit of the displayed position. Note, that changing the unit will also affect the step width, see 2.4. There are three possible choices for each motor:

- degree
- radian
- step

2.7 Internal program

This menu allows the user to step through the internal program list with the rotary knob. This function is only available if a program has previously been defined (see 4.2). Internal programs are also saved to the device when the current configuration is saved, see 2.11.4.



2.8 Constant movement

Here the motors can be set into an infinite moving state in clockwise (CW) or counter clockwise (CCW) direction. To get the motors moving with different velocities one needs to change the wait times between two steps (see 2.11.3). STOP means that the motor is not moving. Constant speed for a certain motor can not be activated if a forbidden zone is configured for this motor.

2.9 Zero position

In this menu one can define an offset for the zero position. This is necessary due to a mostly unknown placement of the load mounted to a stage. The zero position is always defined in steps. In this menu there is also a fast mode available (please refer to 2.4 for details about the fast mode). After adjustment it is recommended to save the configuration (see 2.11.4). When performing a zero calibration, as explained in 2.10, the zero position will have the defined offset form the mechanical zero position.

2.10 Zero calibration

Here one can calibrate the motor zero position for each motor. To perform a zero calibration select the motors to be calibrated and turn the rotary knob. Note, that during zero calibration no actions can be done on the device, even serial commands will not be accepted. The zero calibration will automatically deselect a motor when its calibration is finished. The zero calibration menu will be left automatically when calibration is finished.

2.11 Settings menu

The settings menu contains several submenus that allow the user to adjust some of the settings.
Turning the rotary knob serves to scroll through the settings menus (see figure 2.2). Pressing the rotary knob enters the selected settings menu. Pressing the menu-escape-button leaves the settings menu.

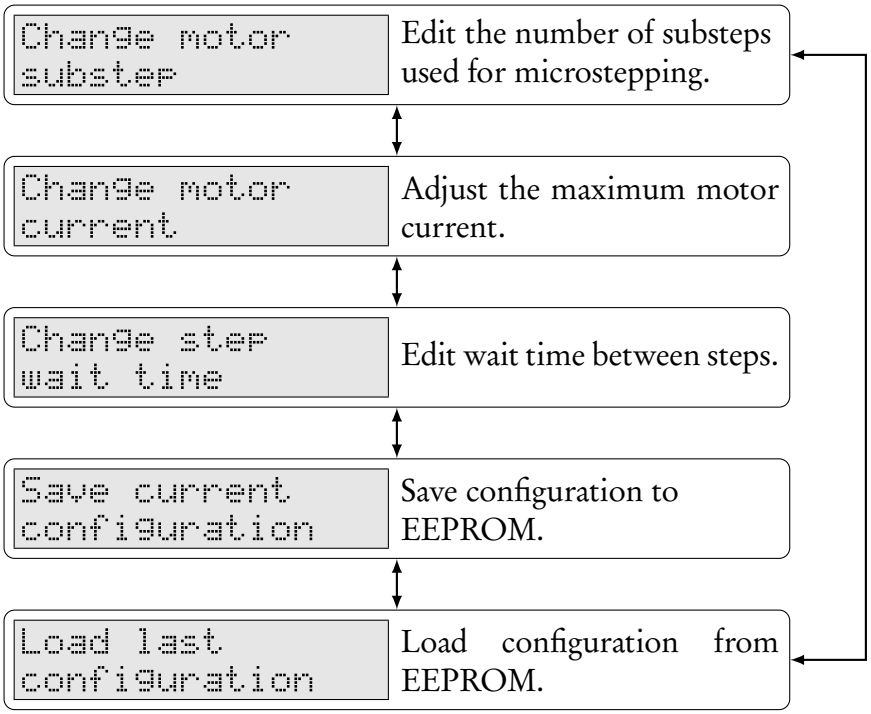


Figure 2.2: Overview of the settings menus. By turning the rotary knob one can navigate through the settings menus as indicated by the arrows.

2.11.1 Motor substeps

In this menu one can change the number of substeps used for microstepping. Possible values are 1, 2, 4, 8, 16 or 32.
Note, that a value of 1 is equivalent to full step operation. So no microstepping is used to drive the stepper motor. If the adjusted value differs from 1, microstepping is used to obtain finer positioning resolution. Values above 8 are not recommended because of the decreasing motor torque.

A note on microstepping. When increasing the number of substeps (microsteps per full step) the incremental torque per microstep decreases heavily. The expression for calculating the incremental tourque τ_{inc} is

$$\tau_{\text{inc}} = \tau_{\text{H}} \cdot \sin\left(\frac{90}{\mu}\right),$$

where τ_{H} is the holding torque per full step (without microstepping) and μ is the number of microsteps per full step.

The incremental torque τ_N for N microsteps is

$$\tau_N = \tau_H \cdot \sin\left(\frac{90 \cdot N}{\mu}\right).$$

So, the holding torque per microstep decreases as shown in table 2.1.

Table 2.1: Decrease of motor holding torque in dependence of the number of adjusted microsteps.

Microsteps per full step	Holding torque per microstep
1	100 %
2	70.7 %
4	38.3 %
8	19.5 %
16	9.8 %
32	4.9 %

2.11.2 Motor current

In this menu the user can change the maximum current that is used to drive the corresponding stepper motor. Please ensure, that the value set here does not exceed the maximum ratings of your motor.

2.11.3 Step wait time

Here the wait time between two steps can be changed. This results in faster or slower movements of the stepper motor. The default value is 3 milliseconds.
Note, that a very short wait time and therefore fast movement speed can lead to stalling of the stepper motor.

2.11.4 Save current configuration

To save the current Stepper Motor Controller configuration enter this menu and turn the rotary knob in any direction.

Note, that always the configuration for all motors will be saved. It is not possible to save the configuration for a single motor only.

2.11.5 Load configuration

To load the last saved Stepper Motor Controller configuration enter this menu and turn the rotary knob in any direction.
Note: The last saved configuration is loaded automatically when powering on or resetting the Stepper Motor Controller.
Note: There is just one memory space for a configuration.

3 Basic setup

3.1 Stepper motor setup

Before a stepper motor is connected to the Stepper Motor Controller, please ensure, that the following parameters are set according to the specifications of you stepper motor.

- maximum motor current
- current decay mode
- gear ratio
- steps per full rotation
- number of substeps
- wait time between steps

Warning: Take especially care, that the maximum motor current does not exceed the maximum ratings of your stepper motor. A high current might damage the stepper motor or the Stepper Motor Controller.

The parameters can be set to the desired value either using the manual user interface (see chapter 2) or via remote control through a computer (see chapter 4).

3.2 Example setup for M101A

In the following configuration example it is assumed, that four LK-Instruments M101A rotation stages shall be connected to the Stepper Motor Controller. In case different stepper motors shall be connected to the Stepper Motor Controller, the same procedure applies, but you need to take care to change the values according to the data sheet of the stepper motor used.

3.2.1 Motor current

Before any of the stepper motors or rotation stages are connected to the Stepper Motor Controller the maximum motor current needs to be adjusted. In the data sheet of the M101A rotation stage a typical motor current of 1.3 A is specified.
In order to adjust the settings of the SMCx242 Stepper Motor Controller to this value, connect it to an appropriate power source and turn it on using the switch on its back panel.

There are two possible ways of adjusting the current setting either using the manual user interface or via remote control.
To adjust the setting using the manual user interface, navigate to

the settings menu by rotating the rotary knob until the settings menu shows up (see 2). Enter this menu by pressing the rotary knob and navigate to the motor current menu by rotating the rotary knob (see 2.11). Enter the current menu by pressing the rotary knob (see 2.11.2). Now select all motors by pressing the corresponding motor-selection buttons and adjust the the motor current to the desired value of 1.3 A by turning the rotary knob. The next step is to save the settings to the EEPROM. To do so, exit the current menu by pressing the menu-escape-button once. Navigate to the save current configuration menu and enter it by pressing the rotary knob (see 2.11.4). To save the configuration turn the rotary knob in any direction. If successful the word "saved" will show up on the display.

To adjust the settings using the remote interface, connect the Stepper Motor Controller to a free USB port of a computer and connect to it, as described in chapter 4, using a serial terminal. To set the motor current of all four motors to 1.3 A and save the settings to the EEPROM issue the following commands:

```
SETCURR 0 1.3
SETCURR 1 1.3
SETCURR 2 1.3
SETCURR 3 1.3
SAVECONF
```

Now one may already connect the stepper motors to the Stepper Motor Controller. To do so, first turn of the Stepper Motor Controller, connect the stepper motors to its outputs and once connected, turn it back on.

3.2.2 Current decay mode

The SMCx242 Stepper Motor Controller allows the user to select between three different current decay modes, namely fast, slow and mixed decay. In this example we would like to set to decay mode to slow. Therefore the following commands need to be send:

```
SETDECAY 0 0
SETDECAY 1 0
SETDECAY 2 0
SETDECAY 3 0
```

3.2.3 Gear ratio

The M101A rotation stage uses a gear with 20 teeth on the motor shaft and a gear with 60 teeth for the load. Therefore this

stage has a gear ratio of $n_{\text{gear}} = \frac{60}{20} = 3.0$. To set this gear ratio for all four motor channels send the following commands to the Stepper Motor Controller:

```
SETGEARRATIO 0 3.0
SETGEARRATIO 1 3.0
SETGEARRATIO 2 3.0
SETGEARRATIO 3 3.0
```

3.2.4 Steps per full rotation

The stepper motor used in the M101A rotation stage has a step angle of 0.9 ° and therefore 400 steps per full rotation. To set this value for all four motor channels send the following commands to the Stepper Motor Controller:

```
SETFULLROT 0 400
SETFULLROT 1 400
SETFULLROT 2 400
SETFULLROT 3 400
```

3.2.5 Number of substeps

In this example we would like to benefit from the higher resolution, that microstepping can provide to us. Therefore the number of substeps shall be set to 4. To do so, navigate to the motor substeps menu inside the settings menu and adjust the values using the rotary knob (see 2.11.1) or send the following commands to the Stepper Motor Controller:

```
SETSUBSTEPS 0 4
SETSUBSTEPS 1 4
SETSUBSTEPS 2 4
SETSUBSTEPS 3 4
```

3.2.6 Wait time between steps

In order to rotate the M101A rotation stage with a desired speed of $\omega = 25\,^{\circ}/\text{s}$, according to the previously specified parameters, the wait time between steps τ is given by:

$$\tau = \frac{360^{\circ}}{n_{\text{gear}} \cdot n_{\text{fullrot}} \cdot n_{\text{substeps}}} \frac{1}{\omega} = \frac{360^{\circ}}{3 \cdot 400 \cdot 4} \frac{1}{25\,^{\circ}/\text{s}} = 3\,\text{ms}$$

with the gear ratio n_{gear}, the number of steps per full rotation n_{fullrot} and the number of substeps n_{substeps}.
In order to set this value, navigate to the step wait time menu inside the settings menu and adjust the values using the rotary

knob (see 2.11.3) or send the following commands to the Stepper Motor Controller:

```
SETWAITTIME 0 3
SETWAITTIME 1 3
SETWAITTIME 2 3
SETWAITTIME 3 3
```

3.2.7 Save configuration

Once all the parameters are set to the desired values, it is recommended to save the configuration to the EEPROM, because the SMCx242 Stepper Motor Controller will load the last saved configuration when powering on or resetting the device. To save the configuration either navigate to the save current configuration menu inside the settings menu and turn the rotary knob while in this menu (see 2.11.4) or send the following command to the Stepper Motor Controller:

```
SAVECONF
```

4 Remote Programming

4.1 Communication Settings

In order to remote control the Stepper Motor Controller from a computer, connect it to a free USB port of the computer. The Stepper Motor Controller will show up as a new Virtual COM Port (VCP). In some cases it might be necessary to install drivers, which can be found at

`http://www.lk-instruments.com`

on the corresponding product website. Once the Virtual COM Port has been installed successfully, the Stepper Motor Controller can be controlled by sending commands via a serial terminal. A list of all available commands can be found in section 4.2. The serial terminal needs to be configured as

- 57600 Baud
- 8 bit character size
- no parity bit
- 1 stop bit
- no flow control

4.2 Instruction set

The following commands are available on the Stepper Motor Controller. Note, that the command parser is case sensitive. The command parameters, denoted by `<xxx>`, must be separated by either `SPACE` or “,” or “;” or `TAB`. The command is completed by sending a Carriage Return + Line Feed (`CRLF`) or Line Feed (`LF`).
Note, that in order to simplify writing programs for the Stepper Motor Controller, the remote interface uses a different numbering for the motors. On the device itself the motor channels are labeled with "Motor 1" to "Motor 4", but the remote interface starts counting at `0`. Therefore in the following we will refer to the stepper motor connected to the first channel as "motor `0`", to the motor connected to the second channel as "motor 1" and so on.

Command: `*IDN?`
Function: Returnes the identification name of the Stepper Motor Controller.
Example: `*IDN?`

Command: `*RST`
Function: Resets the Stepper Motor Controller to the initial state and loads the last saved configuration.
Example: `*RST`

Command: `FACTORYRESET`
Function: Resets the Stepper Motor Controller to factory state.
Example: `FACTORYRESET`

Command: `GETMOTSTATE <mot>`
Function: Returns whether motor `<mot>` is turned on or off.
Example: `GETMOTSTATE 3`
Returns 1 if motor 3 is turned on or `0` if motor 3 is turned off.

Command: `ENABLE <mot> <on/off>`
Function: Turns motor `<mot>` on (1) or off (`0`).
Note: Both for enabeling and disabeling of a motor the same command is used.
Example: `ENABLE 2 1`
Turns motor 2 on.
`ENABLE 3 0`
Turns motor 3 off.

Command: `ISCON <mot>`
Function: Returns if a motor is connected to the output.
1: a motor `<mot>` is connected.
`0`: no motor `<mot>` is connected.
Example: `ISCON 2`
Returns 1 if a motor is connected to the motor 2 output and `0` if not.

Command: `MOVEABS <mot> <pos> <unit>`
Function: Moves motor `<mot>` to the absolute position `<pos> <unit>`.
The units can be steps, degree or radians.
Note: The unit must be written in lower case letters.
Example:
```
MOVEABS 1 0 deg
MOVEABS 1 0 pi
MOVEABS 1 0 steps
```
All three examples move motor 1 to the zero position, but in different units.

Command: `MOVEREL <mot> <pos> <unit>`
Function: Moves motor `<mot>` relative to the current position. The units can be steps, degree or radians.
Note: The unit must be written in lower case letters.
Example:
```
MOVEREL 2 22.5 deg
MOVEREL 2 0.125 pi
```
Both examples move motor 2 by the same angle, but in different units.

Command: `ZERORUN <mot>`
Function: Finds the mechanical zero position of the motor.
Note: During motor zero run no communication or usage of the Stepper Motor Controller is allowed.
Example:
```
ZERORUN 1
```
Finds the mechanical zero position of motor 1.

Command: `GETPOS <mot> <unit>`
Function: Returns the actual motor position in the given unit.
Example:
```
GETPOS 1 deg
```
Returns the current position of motor 1 in degree.

Command: `ISMOVING <mot>`
Function: Returns the motor moving state.
1: motor `<mot>` is moving.
0: motor `<mot>` is not moving.
Example:
```
ISMOVING 0
```
Returns 1 if motor 0 is moving and 0 if it is currently not moving.

Command: `SAVECONF`
Function: Saves all current configurations for all motors.
Note: The driver configuration is stored in an EEPROM. Maximum write cycles are 100000.
Example: `SAVECONF`

Command: `LOADCONF`
Function: Load saved configuration for all motors.
Example: `LOADCONF`

Command: `GETZEROPOS <mot>`
Function: Returns the zero position of motor `<mot>`.
Note: Zero positions are only available in steps.
Example: `GETZEROPOS 3`
Returns the zero position of motor 3.

Command: `SETZEROPOS <mot>`
Function: Set the zero position for motor `<mot>`.
Note: For the zero position the unit is always steps.
Example: `SETZEROPOS 3 574`
Sets the zero position of motor 3 to 574 steps.

Command: `GETGEARRATIO <mot>`
Function: Returns the gear ratio of a motor.
Example: `GETGEARRATIO 0`
Returns the gear ratio of motor `0`.

Command: `SETGEARRATIO <mot> <val>`
Function: Sets the gear ratio of motor `<mot>` to a gear ratio of `<val>`.
Example: `SETGEARRATIO 2 3.0`
Sets the gear ratio of motor 2 to a value of 3.

Command: `GETFULLROT <mot>`
Function: Returns the number of steps per full rotation without microsteps of a motor. Typical values for stepper motors are 200 or 400 steps per full rotation.
Example: `GETFULLROT 0`
Returns the number of steps per full rotation of motor 0.

Command: `SETFULLROT <mot> <val>`
Function: Sets the number of steps per full rotation of motor `<mot>` to `<val>` steps.
Example: `SETFULLROT 1 400`
Sets the number of steps per full rotation of motor 1 to 400 steps.

Command: `GETSUBSTEPS <mot>`
Function: Returns the number of substeps of a motor.
Example: `GETSUBSTEPS 0`
Returns the number of substeps of motor 0.

Command: `SETSUBSTEPS <mot> <val>`
Function: Sets the number of substeps of motor `<mot>` to `<val>` substeps.
Note: Possible values for substeps are 1, 2, 4, 8, 16, 32.
Example: `SETSUBSTEPS 1 4`
Sets the number of substeps of motor 1 to 4 substeps.

Command: `GETCURR <mot>`
Function: Returns the motor current of a motor in ampere.
Example: `GETCURR 0`
Returns the motor current of motor 0.

Command: `SETCURR <mot> <val>`
Function: Sets the motor current of motor `<mot>` to `<val>` ampere.
Note: The current can be adjusted between 0 and 2.5 A with 8-bit resolution. The unit for the current is always ampere.
Example: `SETCURR 0 1.0`
Sets the motor current of motor 0 to 1 A.

Command: `GETDECAY <mot>`
Function: Returns the setting of the decay mode of motor `<mot>`.
Example: `GETDECAY 0`
Returns `0` if decay mode of motor `0` is set to slow, returns 1 if set to fast and returns 2 if set to mixed.

Command: `SETDECAY <mot> <val>`
Function: Sets the decay mode of motor `<mot>` to slow `<val=0>`, fast `<val=1>` or mixed `<val=2>`.
Note: For microstepping fast decay should be preferred. However if audible noise is an issue, one might consider using slow decay mode to reduce the noise.
Example: `SETDECAY 1 0`
Sets decay mode of motor 1 to slow.
`SETDECAY 1 1`
Sets decay mode of motor 1 to fast.
`SETDECAY 1 2`
Sets decay mode of motor 1 to mixed.

Command: `GETWAITTIME <mot>`
Function: Returns the wait time between two steps of a motor.
Example: `GETWAITTIME 0`
Returns the wait time between two steps of motor `0`.

Command: `SETWAITTIME <mot> <time>`
Function: Sets the wait time between two steps to `<time>` milliseconds for motor `<mot>`
Note: The wait time must be an integer. The unit for the wait time is always milliseconds.
Example: `SETWAITTIME 1 5`
Sets the wait time of motor 1 to 5 milliseconds.

Command: `STOPALL`
Function: Stops all motor movements immediately.
Example: `STOPALL`

Command: `SETCONSTSPEED <mot> <dir> <time>`
Function: Enables motor `<mot>` to move infinite in direction `<dir>`. Possible values for `<dir>` are clock wise `CW` or counter clock wise `CCW`.
Note: That the wait time between two steps is overwritten in order to move with the desired speed.
Example: `SETCONSTSPEED 1 CW 10.0`
Moves motor 1 infinite in clockwise direction. One full rotation takes 10 seconds.

Command: `SETFORBZONE <mot> <start> <stop>`
Function: Defines a forbidden zone for motor `<mot>`. The motor will not move into this zone. `<start>` must be always smaller than `<stop>`. The unit for `<start>` and `<stop>` is always steps.
Example: `SETFORBZONE 0 148 1333`
Defines a forbidden zone for motor `0` between step 148 and step 1333.

Command: `ENABFORBZONE <mot> <val>`
Function: Enables `<val=1>` or disables `<val=0>` the defined forbidden zone for motor `<mot>`.
Example: `ENABFORBZONE 0 1`
Enables the forbidden zone for motor `0`.
`ENABFORBZONE 3 0`
Disables the forbidden zone for motor 3.

Command: `SETPROGSTEP <step> <M0> <M1> <M2> <M3> <mode>`
Function: Defines an internal program step for all motors. `<step>` is the program sequence number. The position for all motors `<M0...4>` must be given in steps. Mode defines if the movement is to an absolute position `<mode=ABS>` or a movement relative to the current position `<mode=REL>`.
Example: `SETPROGSTEP 0 112 294 0 12 ABS`
Defines the program step `0`. Motor `0` moves to 112, motor 1 moves to 294, motor 2 to `0` and motor 3 to 12.
`SETPROGSTEP 1 10 10 -10 -10 REL`
Defines the internal program step 1 so that motor `0` and motor 1 move 10 steps forward and motor 2 and 3 move 10 steps backwards from the current position.

Command: `LED <mode> <red> <green> <blue>`
Function: Using this command the color and brightness of the buttons can be adjusted. The first parameter indicates the mode, that should be changed. The available modes correspond to the following buttons:
`<mode=0>` menu-escape button,
`<mode=1>` deselected motor-selection button,
`<mode=2>` selected motor-selection button,
`<mode=3>` disabled and deselected motor-selection button and
`<mode=4>` disabled and selected motor-selection button.
The color and brightness are controlled by the parameters `<red> <green> <blue>`. Each of the parameters accepts a value in the range of `0` to 255. Where larger values represent a higher brightness of the respective color.
Use the `SAVECONF` command to save the new color pattern to the EEPROM.
Example: `LED 0 15 0 0`
Sets the color of the menu-escape button to red with medium brightness.
`LED 1 0 15 0`
Sets the color of a deselected motor-selection button to green with medium brightness.
`LED 2 8 15 15`
Sets the color of a selected motor-selection button to white with medium brightness.

Command: `DISPLAY <val>`
Function: Sets the brightness of the OLED display to the value `<val>`. Possible values are:
`<val=0>` display is turned off,
`<val=1>` reduced display brightness,
`<val=2>` normal display brightness.
Example: `DISPLAY 0`
Turns off the OLED display.
`DISPLAY 1`
OLED display set to reduced brightness.
`DISPLAY 2`
OLED display set to normal brightness.

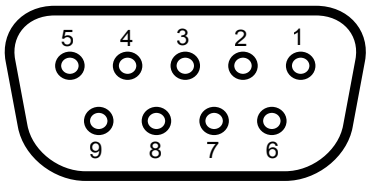
5 Specifications

Table 5.1: Specifications of the SMCx242 Stepper Motor Controller. With $V_{Supply} = 24\,VDC$, $T_A = 25\,°C$ and 50% RH unless otherwise noted.

Parameter	SMC2242			SMC4242			Unit
	Min	Typ	Max	Min	Typ	Max	
Number of outputs			2			4	
Output current	0		2.5	0		2.5	A/Phase
PWM frequency		30			30		kHz
Microsteps	1		32	1		32	
Step frequency			1			1	kHz
Limit/Home switches per motor			3			3	
Temperature range	0		40	0		40	°C
Weight		1.7			1.8		kg
Supply voltage	10	24	36	10	24	36	VDC
Power consumption			100			200	W

6 Pinout

The connectors for the stepper motors are 9-Pin D-Type, female connectors. Please refer to figure 6.1 for their pinout.

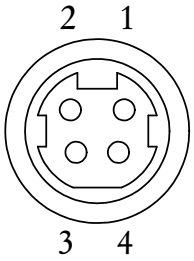


Front view

Pin	Function
1	Phase B1
2	Phase B2
3	Phase A2
4	Phase A1
5	Ground
6	+5V
7	Sens 1
8	Sens 2
9	Sens 3
Shield	NC

Figure 6.1: Pinout of the motor connectors.

The power connector is a R7B female connector. Please refer to figure 6.2 for its pinout.



Front view

Pin	Function
1	+ 24V
2	Ground
3	Ground
4	+ 24V
Shield	NC

Figure 6.2: Pinout of the power connector.

Index