Manual

3-axle trajectory generator MI 389

Programmable trajectory generator with USB interface for stepper motor drivers

P.P.H. WObit E.K.J. Ober s.c.
62-045 Pniewy, Dęborzyce 16
tel.48 61 22 27 422, fax.48 61 22 27 439
e-mail: wobit@wobit.com.pl www.wobit.com.pl

Contains

Manual 1
Contains 2
1. Safety and assembly rules 4
1.1. Safety rules 4
1.2. Assembly recommendation 4
2. Device description 5
2.1. Application 5
2.2. Features 5
2.3. Connectors description 6
2.4. Detailed description of connectors 7
USB connector 7
Power supply 8
Output OC type for common use 8
Outputs for stepper motor drivers 9
Relay outputs 9
Inputs for proximity sensors 10
Inputs for common use 10
Single-purpose inputs 11
2.5. Parameters 11
3. Example of connection of trajectory generator MI389 12
4. PC program 14
4.1 Program description 14
4.2 Toolbar 15
4.3 Emergency stop button 15
4.4 Indicator buttons 16
4.5 Message panel 16
4.6 Hint panel 16
4.7 Direct control 17
4.8 Configuration panel of direct inputs 22
4.9 Preview of current parameters 22
4.10 Programming window 23
4.11 Motor selection panel 24
4.12 Programming panel – program commands 24
4.13 Description of program operation 33
4.14 Program compilation and programming 34
4.15 Program readout from internal memory 34
4.16 Software update 34
5. Program example 36
6. Safety and assembly rules 37

Thank you for selecting our product!

This instruction will help you at correct service and accurate exploitation of described device.

Information included in this instruction were prepared with high attention by our specialists and is description of the product without any responsibilities within the meaning of the commercial law. Based on the information should not be inferred a certain features or suitability for a particular application. This information does not release the user from the obligation of own judgment and verification. P.P.H. WObit E.K.J. Ober S.C. reserves the right to make changes without prior notice.

- Please read instructions below carefully and adhere to its recommendation
- Please pay special attention to the following characters:

CAUTION!
Not adhere to instruction can cause damage or impede the use of hardware or software.

1. Safety and assembly rules

1.1. Safety rules

- Prior to first start-up of the device please refer to this manual;
- Prior to first start-up of the device please make sure all cables are correctly connected,
- Provide appropriate working conditions, in compliance with the device specifications (e.g.: power supply voltage, temperature, maximum current consumption).
- Before making any modifications to wiring connections, disconnect the power supply voltage.

1.2. Assembly recommendation

In environments with unknown noise levels, it is recommended to follow measures described below to prevent any possible interruptions of the device operation:

- Ground or reset metal rails, on which are mounted instruments,
- Do not power devices on the same line as the device without a corresponding high power line filters;
- Please use screening of the supply, sensor and signal cables, with the ground for the screen should be connected only on one side, as close to the device;
- For motor power supply please use twisted pair cables, and if possible use a ferrite bead assumed on the wire;
- Please avoid of leading control cables (Signal) parallel or in close to electrical and power wires;
- Please avoid proximity to devices that generate high levels of electromagnetic interferences and/or pulse (high-power loads, the burden of the phase or power control group).

2. Device description

2.1. Application

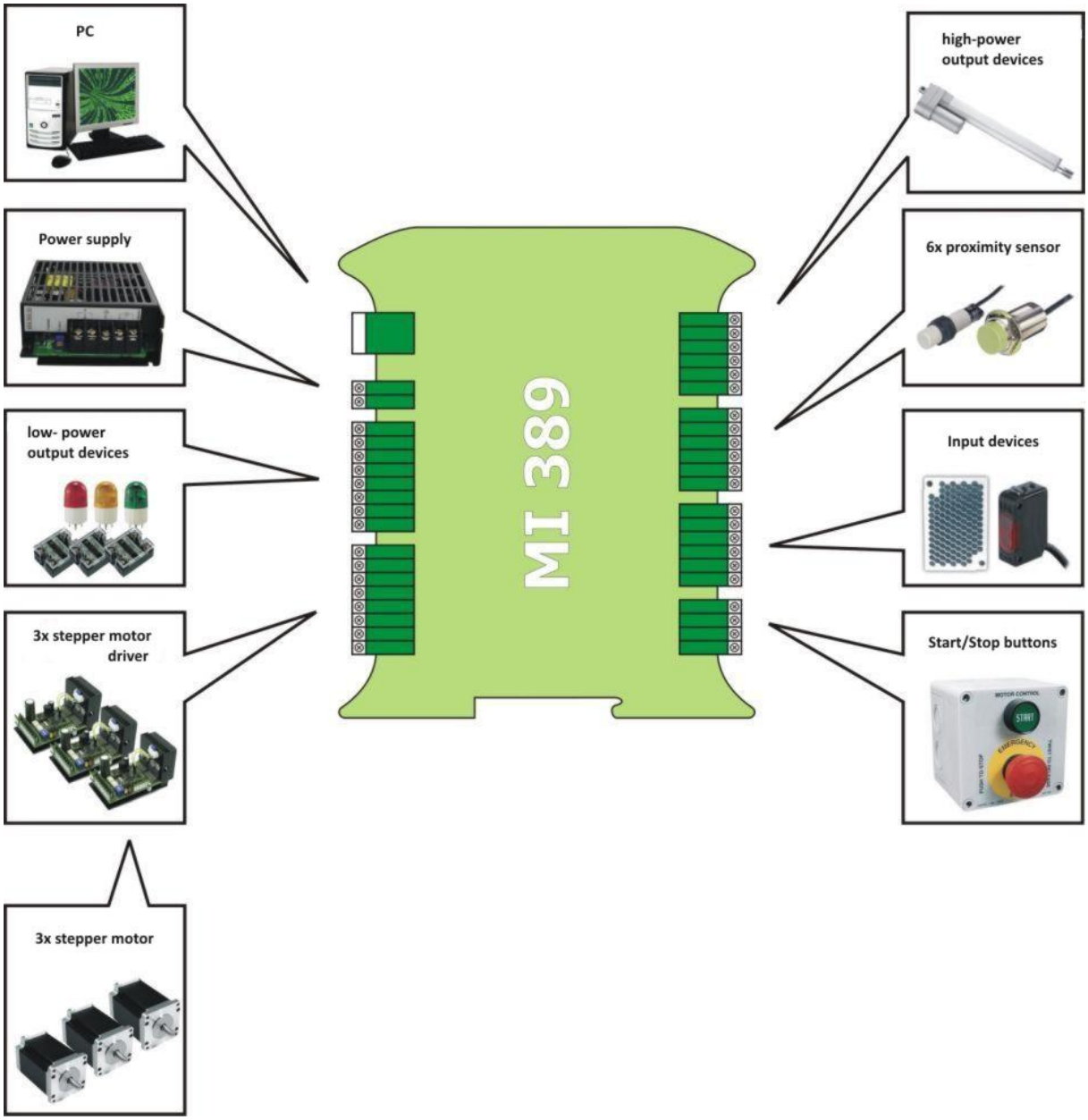
MI 3.8.9. trajectory generator is a versatile device use to generate a motion trajectory for three stepper motors by any force controller. MI 3.8.9 fulfill role of small PLC driver, dedicated for motion control of stepper motors.

The device is useful whether it is necessary to cyclic perform complicated and repetitive motion sequences of many parameters (e.g. position, velocity) with acceleration and controlled stop of motor.

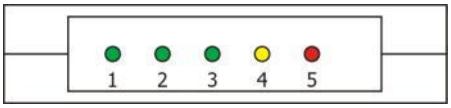
Attached PC software to the device, communicates by USB connector, enables simple and intuitive programming of motion trajectory for three stepper motors. This program also allows to preview of current motor parameters, status of inputs and outputs and controlling in real time. After programming device can work independently – without PC.

2.2. Features

- Power supply 10…36 VDC;
- Power supply by USB connector
- Operation with stepper motor drivers –CKL and DIR outputs for each driver and common ENABLE output in TTL standard;
- 2 special opt insulated outputs: RUN and STOP;
- 6 opt insulated inputs for common use;
- 3 x 2 inputs of proximity sensors;
- 7 transistor outputs OC type and 2 relay outputs;
- LED indicators, indicate power supply, and device operations status;
- No violate memory – up to 999 commands;
- USB transmission;
- Pluggable terminal block;
- Housing adapter for mounting on DIN rail.



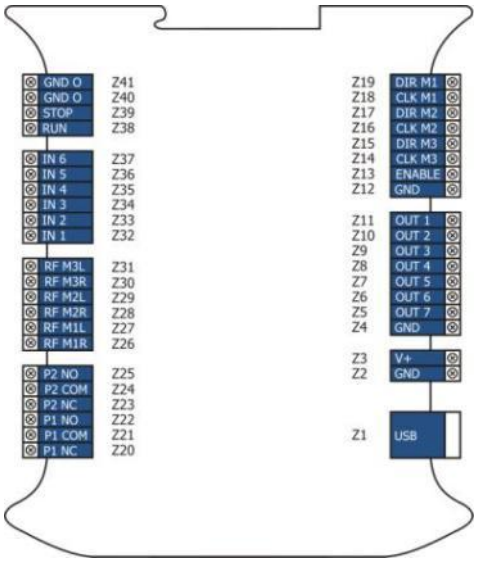
2.3. Connectors description



Drawing1. Front panel

	Name	Color	Description
1	PROGRAM	Green	Diode is active at program playback from internal memory
2	ENABLE	Green	Diode is active at ENABLE signal

3	TRANSMISSION	Green	Diode is active at data transmission to PC
4	UPLOAD/ERROR	Yellow	Diode is active at program updating Diode is active at activated proximity sensor – it blinks
5	POWER	Red	Power supply signalization diode



Drawing 2. Connectors description

	Name	Description
Z1	USB	USB connector for communication with PC
Z2, Z4, Z12	GND	Power supply ground
Z3	V+	Power supply
Z5 - Z11	OUT 1 … 7	Common output OC type
Z13	ENABLE	Drive ENABLE for stepper motor drivers in TTL standard; active – high level (+5V)
Z14, Z16, Z18	CLK 1 … 3	CLK signal for stepper motor drivers in TTL standard
Z15, Z17, Z19	DIR 1 … 3	Direction signal for stepper motor drivers in TTL standard; Left – high level(+5V)
Z20, Z23	P1 … 2 NC	1 and 2 relay output – normally closed
Z21, Z24	P1 … 2 COM	1 and 2 relay output – common
Z22, Z25	P1 … 2 NO	1 and 2 relay output – normally open
Z26, Z28, Z30	RF M1 … 3R	Input of right proximity sensor for 1,2 and 3 motor
Z27, Z29, Z31	RF M1 … 3L	Input of left proximity sensor for 1,2 and 3 motor
Z32 – Z37	IN 1 … 6	Opt insulated inputs for common use; active – high level
Z38	RUN	Special opt insulated input– program activation from internal memory; active – high level
Z39	STOP	Special opt insulated input – program deactivation from internal memory; active – high level
Z40, Z41	GND O	Opt insulation ground

2.4. Detailed description of connectors

USB connector

Trajectory generator MI 389 is equipped in USB connector. Device connection to driver is made by standard A – B cable. Due connection with PC it is possible to program trajectory, control of motors and outputs in real time, also speed view, view of motors location, I/O state.

After programming trajectory the device can operate without connection with PC, i.e. as independent controlling device.

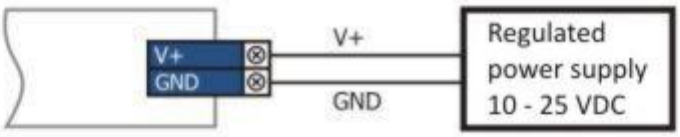
Drawing 3. Device connection to PC

CAUTION!
To connect trajectory generator with PC it is recommended to use cable no longer than 2 [m]. Use of longer cable can cause transmission errors and make impossible correct operation of the device.

Power supply

Trajectory generator should be powered from regulated power supply with output voltage in range 10 VDC up to 36 VDC.

Device can be powered directly by USB from PC. Due to limited current efficiency of USB connector it is recommended to use this possibility only at tests and programming of the device.

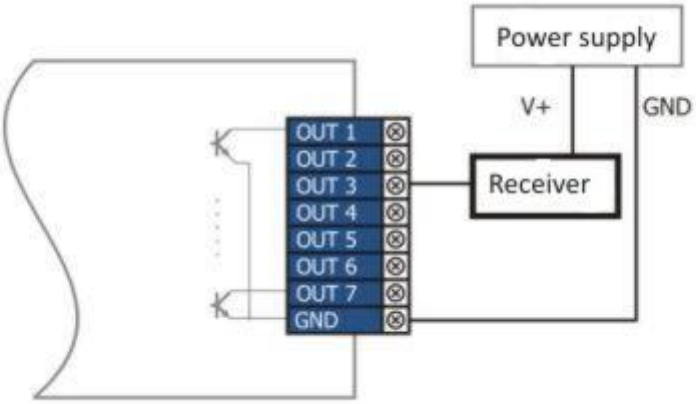


Drawing 4. Connection of power supply to the device

CAUTION!
Reverse polarization or exceeding of maximal voltage supply can cause damage of the device.

Output OC type for common use

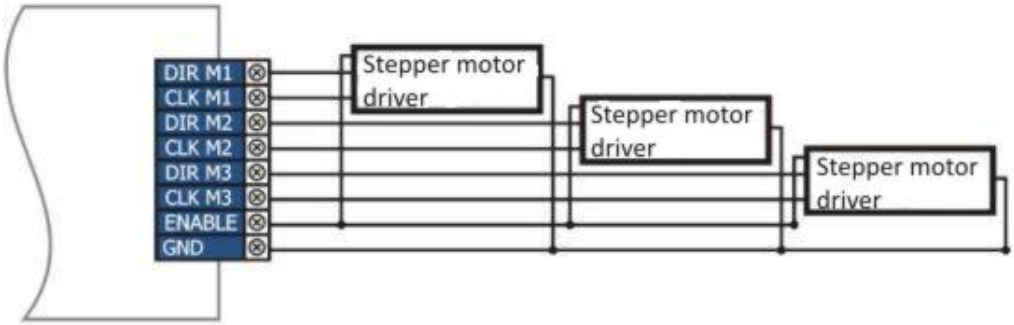
Trajectory generator has 7 outputs OC type for common use. Outputs have common ground with power supply of the device. On drawing below is shown an internal input construction.



Drawing 5. Connection OC type

Outputs for stepper motor drivers

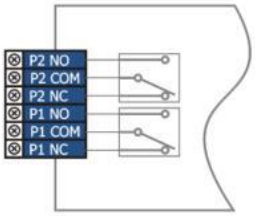
Device enables independent control of three stepper motor by use of proper drivers. To each driver is subordinate a proper controlling line – step (CLK) and direction (DIR). Enable output is common for all three drivers. For correct operations is necessary to connect signal ground of all devices.



Drawing 6. Connection of stepper motor drivers

Relay outputs

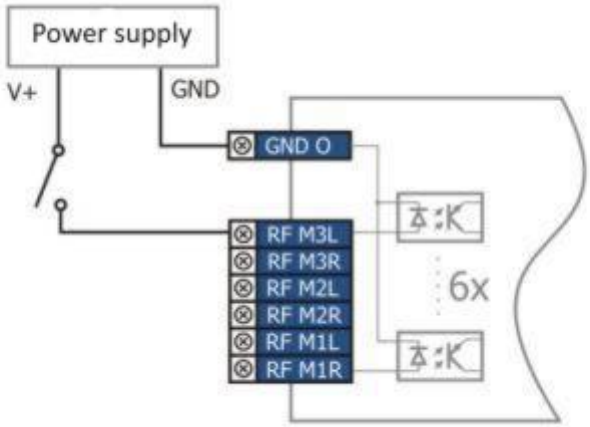
Trajectory generator has two independent relay outputs P1 and P2 for common use, with low fuse load capacity, controlled by program. On drawing below is shown an internal relay output construction.



Drawing 7. Relay outputs

Inputs for proximity sensors

To each motor is subordinate two inputs of proximity sensors, activate by high level on connector.



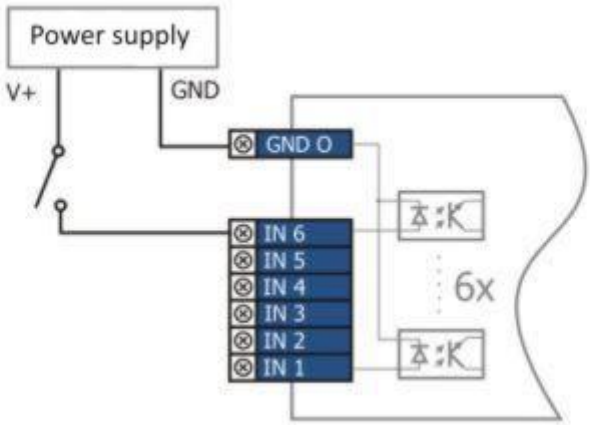
Drawing 8. Connection of proximity sensors

Name	Description	Comments
RF M1L RF M2L RF M3L	Input of left proximity sensor	Activation of left sensor in case of motor motion into the left cause stop of the motor. Motion can be continued after change of motor motion direction or after taking off high level from sensor state input Input is used also for detection of HOME position.
RF M1R RF M2R RF M3R	Input of right proximity sensor	Activation of right sensor in case of motor motion into the right cause stop of the motor. Motion can be continued only after change of motor motion direction or taking off high level from sensor state input.

CAUTION!
Operation of proximity sensor cause immediate stop of the motor, without braking, which can cause loss of step, that means, loosing of current position. Thereby program is stopped, unless sensor is used for detection of HOME position. Then it is necessary to restart the program.

Inputs for common use

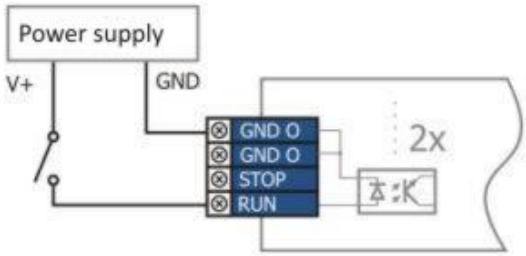
Trajectory generator has 6 opt insulated inputs for common use. By use of this inputs you can easily get complex sequence with loop and conditional operations, dependent on state of selected inputs.



Drawing 9. Opt insulated inputs for common use

Single-purpose inputs

Opt- insulated single purpose inputs RUN and STOP are used property for activation and deactivation of program saved to internal memory of trajectory generator. Signals are activated by high level.



Drawing 10. Single-purpose controlling inputs

Name	Description	Comments
RUN	RUN Input	Appearing of high state on input by at least 15 [µs], cause activation of program recorded in internal memory of the program. Each time program starts with first command. Trajectory generator respond for RUN signal only in case of offline single-purpose STOP input.
STOP	STOP Input	High state on input by at least 15 [µs], cause immediate stop of program regardless on other inputs state (including RUN input). Additionally ENABLE signal for stepper motor driver is take off. Other inputs remain unaltered.

2.5. Parameters

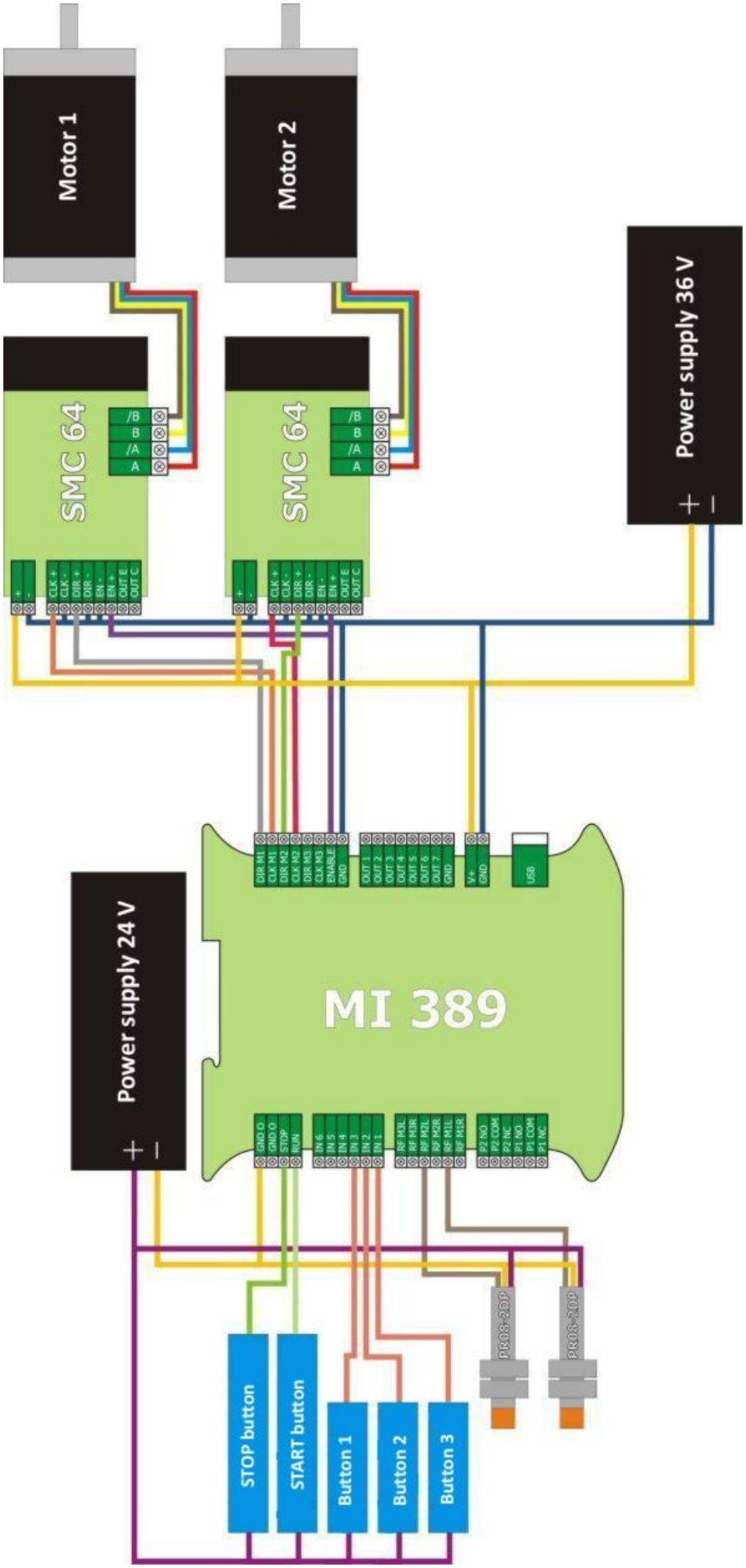
Name	Description	Parameters
V+	Power supply	10 … 36VDC
GND	Supply ground	
OUT 1 … 7	Outputs OC type	0,5 A / 40 VDC
P1 … 2	Relay outputs	2 A / 30 VDC 1 A / 125 VAC
CLK DIR ENABLE	Single-purpose outputs for stepper motor drivers (clock, direction, enable)	TTL Standard
RF 1 … 3 L RF 1 … 3 R	Opt insulated inputs for proximity sensors	5 … 30 VDC (low level – max 1 V)
IN 1 … 6	Opt insulated input for common use	5 … 30 VDC (low level – max 1 V)
RUN STOP	Single-purpose inputs	5 … 30 VDC (low level – max 1 V)

GND O	Ground of opt insulation	

Caution!
Connection supply ground **GND** with ground of opt insulation **GND O** cause loss of input opt insulation properties. It can cause trajectory generator damage in case of overvoltage on inputs.

3. Example of connection of trajectory generator MI389

On picture below is shown a connection between trajectory generator MI389 and two stepper motors through SMC64 drivers. As proximity sensor were used PR08-20DP inductive sensors.


Motor 1
Motor 2
SMC 64
SMC 64
/B
B
/A
A
Power supply 36 V
+ -
Power supply 24 V
+ -
MI 389
STOP button
START button
Button 1
Button 2
Button 3
PR08-20P
PR08-20P

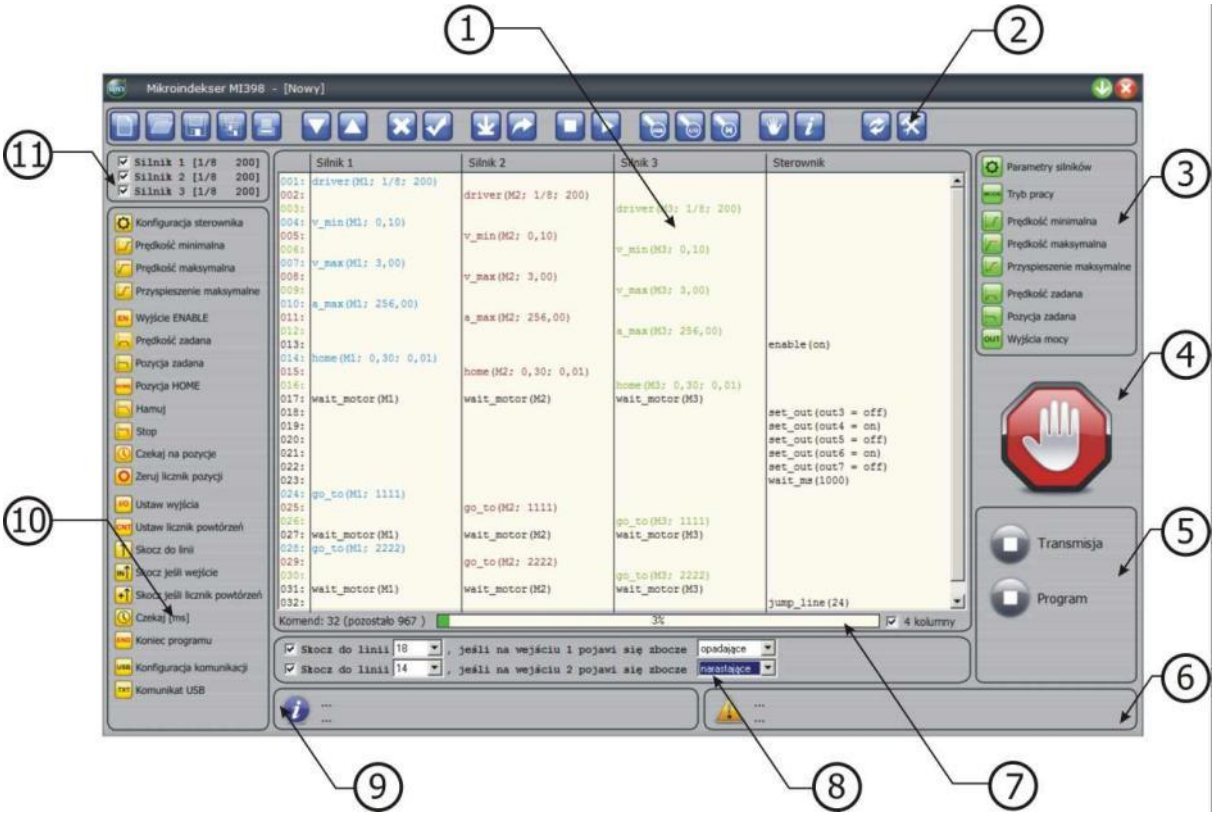
4. PC program

MI389-PC software is dedicated for operation with trajectory generator through USB connector. This application is used for support programming of motors motion trajectory. Additionally this program is equipped in such functions like controlling and preview of motor operations in real time.

Before activation of the program it is necessary to install appropriate drivers.

Caution!
Latest software version and up to date drivers, necessary for correct application operations can be downloaded from manufacturer's website: www.wobit.com.pl

4.1 Program description



Drawing 11. Main program window

Number	Name
1	Program window
2	Toolbar
3	Panel for direct control of motors and outputs
4	Emergency stop of motors
5	Indicator buttons
6	Message panel
7	Progress bar
8	Configuration panel for direct inputs
9	Help
10	Programming panel
11	Motor selection panel

4.2 Toolbar

Drawing 12. Toolbar

Nr	Name	Description
1	New file	This command creates a new program file. In case, when other program is currently open, there is a possibility to save it.
2	Open a file	This command opens previously recorded file with program. In case, when other program is currently open, there is a possibility to save it.
3	Save a file	This command save currently open file with its name.
4	Save file as…	This command saves a file with any name.
5	Print	This command prints program code.
6	Scroll down	This command scroll program command one position down.
7	Scroll up	This command scroll program command one position up.
8	Disconnect USB	This command closes connection with trajectory generator through USB connector
9	Connect USB	This command make a connection with device through USB connector.
10	Save EEPROM	This command saves program to internal trajectory generator memory.
11	Read out EEPROM	This command read out program from device internal memory. In case, when other program is currently open, there is a possibility to save it.
12	Stop program	This command stops program saved in internal trajectory generator memory. Additionally ENABLE signal is set as inactive. State of other outputs remain unaltered.
13	Start program	This command starts program saved in internal trajectory generator memory.
14	USB Preview	This command is active at connection – opens USB message window.
15	I/O Preview	This command is active at connection – opens I/O preview.
16	Motor Preview	This command is active at connection – opens motor state preview
17	Help	This command opens help window
18	About program	Information about the program – manufacturer, software version
19	Reset	This command restart state after reset
20	Update	This command is available after giving password – enables MI389 software update

4.3 Emergency stop button

During correct connection with MI389 trajectory generator there is a possibility of emergency stop of motors. Pressing the button cause immediate stop of all motors without braking and setting ENABLE signal as inactive. Digital and relay output state remain unaltered.

Drawing 13. Emergency stop button

Caution!
Emergency stop of motors cause lost of current position. To continue operation it is necessary to make homing to find start position.

4.4 Indicator buttons

Program has two indicator buttons, which informs about operations status of the device. First indicator informs about communication state of PC program with MI389 trajectory generator, second indicator mirrors state of program made on internal memory of the device.

Drawing 14. Message panel

Name	Description	
USB transmission		USB communication inactive – connection is not established.
		USB communication active – device connection has been made properly.
Program		Playing program from internal trajectory generator stopped.
		Playing program from internal trajectory generator in train.

4.5 Message panel

This window is used for displaying two latest message with information about program operations status.

Drawing 15. Message panel

4.6 Hint panel

Hint panel is used for displaying information about button or indicator function. After moving the cursor over selected target, in hint panel appears an information about its function.

Drawing 16. Hint panel

4.7 Direct control

Direct control panel is used for setting motor parameters and outputs in real time. This functions can't be used during program replaying from trajectory generator internal memory. All changes are immediately set to the device.

Caution!
Change of any parameter (operations mode, minimal velocity, maximal velocity, maximal acceleration, set velocity, set position) cause setting ENABLE signal in active state.

Before setting of position or speed it is necessary to configure other parameters (operation mode, minimal velocity, maximal velocity, maximal acceleration). At turning on this parameters have zero-balance, which make impossible any movement.

Drawing 17. Direct control

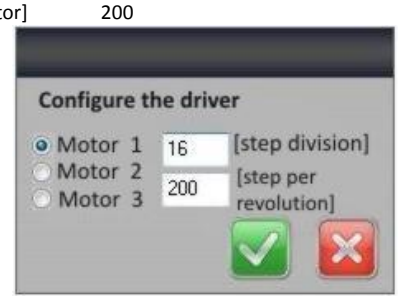
Nr	Name	Description
1	Motor parameters	This command opens motors & drivers configuration window
2	Operation mode	This command set motor operations mode (velocity regulation, position)
3	Minimal velocity	This command configures minimal velocity for motors
4	Maximal velocity	This command configures maximal velocity for motors
5	Maximal acceleration	This command configures maximal acceleration for motors
6	Set velocity	This command sets set velocity (only at Velocity mode)
7	Set position	This command sets set position (only at Ramp and Soft mode)
8	Output	This command enables controlling of outputs

Motors and drivers parameters

Program allows independent configuration each of three motors. First parameter defines step division made by stepper motor driver. Second parameter defines number of full step accrue to one motor resolution.

For example for driver with set step division 1/16 and motor with 1,8° step should be set parameters as below:

- Step division [pertain to driver] 16
- Steps per revolution[pertain to motor] 200

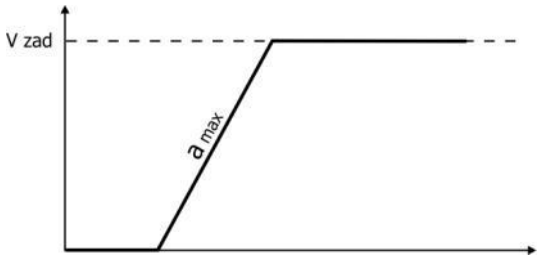
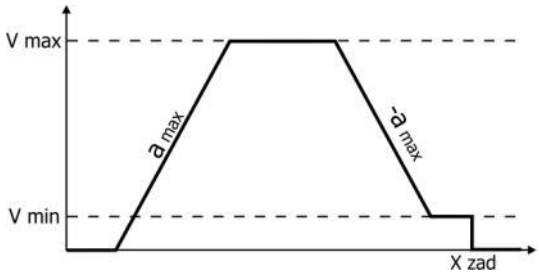
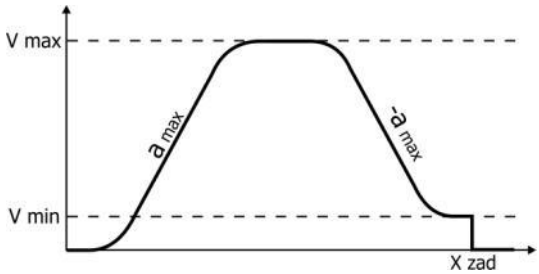


Drawing 18. Motors and drivers configuration panel

Operation mode

MI389 trajectory generator enables motor operations in three modes, two position regulation modes and one velocity regulation mode.

Drawing 19. Motors and drivers configuration panel

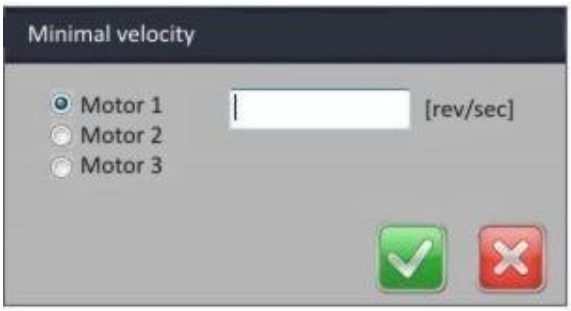
Name	Description	Comments
Ramp	**Position regulation mode with trapezoid profile of motor acceleration and deceleration.** 	Motor accelerate to max. velocity according to set motion parameters (trapezoid profile). Near to set position motor starts braking and stops exactly in set position, without configuration. When total time of acceleration and deceleration exceed time need for achieve set position, motor don't reach max. velocity.
Soft	**Position regulation mode with smoother motor acceleration and deceleration.** 	Principle of operation is the same as in Ramp mode, with one distinction: change of velocity is more smooth than in trapezoid al profile
Velocity	**Velocity regulation mode with trapezoid motor acceleration and deceleration.** 	Motor reaches set velocity according to set acceleration parameters. When set velocity is bigger than max. velocity, final motor velocity will not exceed max. speed. Each change of velocity is made with set motion parameters.

Minimal velocity

This parameter defines the smallest velocity, which will be used for accomplishing motion tasks. Minimal velocity, defined in revolution per second, means safe velocity of getting motor to set position. It is necessary to define this parameter for each motor. This parameter should always take values above zero.

Parameter settings:

- max +31234.74 / (motor step division * driver step division) [rev/sec]
- min bigger than zero



Drawing 20. Minimal velocity configuration panel

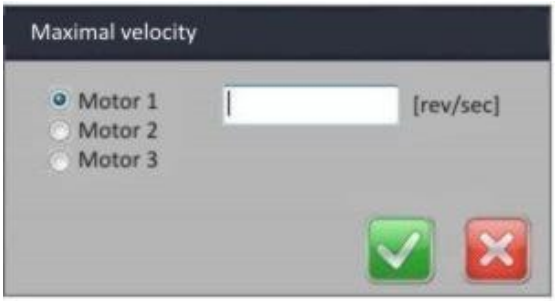
Caution!
During setting of parameter application automatically take account step division. For correct operation of the program it is necessary to define this parameter for selected motor –driver pair.

Maximal velocity

This parameter defines the biggest velocity, which can be implemented to accomplish motion tasks. Maximal velocity defined in revolution per second means max velocity used for of getting motor to set position. It is necessary to define this parameter for each motor. This parameter should always take values above zero.

Parameter settings:

- max +31234.74 / (motor step division * driver step division) [rev/sec]
- min bigger than zero



Drawing 21. Maximal velocity configuration panel

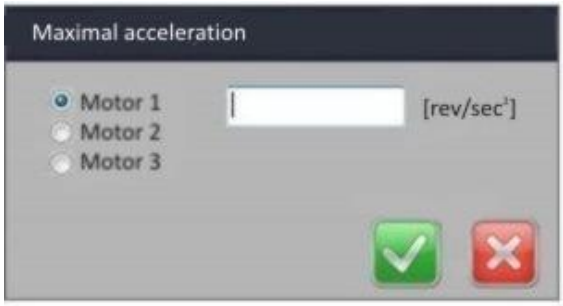
Caution!
During setting of parameter application automatically take account step division. For correct operation of the program it is necessary to define this parameter for selected motor –driver pair.

Maximal acceleration

This parameter is defined in resolution per square second. It is a value of acceleration, which trajectory generator will use to reach maximal velocity, set velocity and at motor braking. This parameter must take values above zero.

Parameter settings:

- max +31234.74 / (motor step division * driver step division) [rev/sec^2]
- min bigger than zero



Drawing 22. Max acceleration configuration panel

Caution!
During setting of parameter application automatically take account step division. For correct operation of the program it is necessary to define this parameter for selected motor –driver pair.

Set velocity

This parameter is defined in revolution per second, means set value for selected motor. It is available only in velocity regulation mode. This parameter can take values below zero. Sign defines motion direction. Zero velocity cause stopping the motor.

Parameter settings:

- max +31234.74 / (motor step division * driver step division) [rev/sec]
- min -31234.74 / (motor step division * driver step division) [rev/sec]

Drawing 23. Set velocity setting panel

Caution!
Change of parameter cause motor motion as long as other settings were made correctly.

Caution!
During setting of parameter application automatically take account step division. For correct operation of the program it is necessary to define this parameter for selected motor –driver pair.

Set position

Parameter is defined in steps, is used for setting final motor position. Additionally program enables change of motor position through decreasing (**<**) and increasing (**>**) of position by set number of steps.
It is available in position regulation mode.

Parameter settings:

- max +8388607
- min +8388607

Drawing 24. Set position setting panel

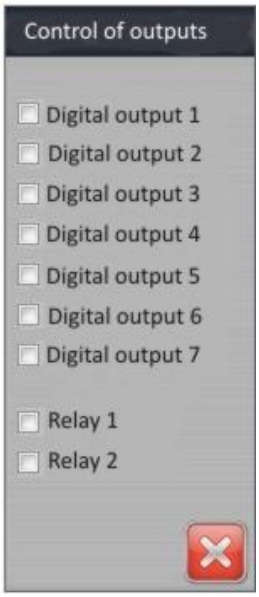
Caution!
Change of parameter cause motor motion as long as other settings were made correctly.

Caution!
To each motor is assigned a separate pair of parameters – number of steps per full motor revolution and driver step division.
Parameters can be read out in window – motor selection panel.

All settings are counted based on this parameters, that's why it is import to configure it before setting other parameters.

Control of outputs

Program allows to control seven digital outputs (OUT 1 … 7) and two relay outputs (P1 and P2). This window is used only for controlling outputs, not for preview. Thence at opening window may occur discrepancy between real output status and controlling window.

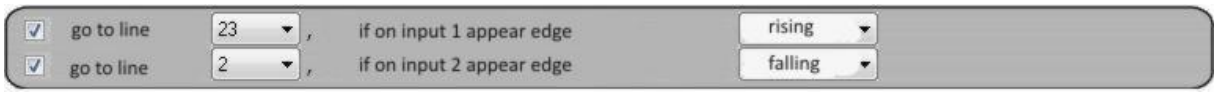


Drawing 25. Control of outputs panel

Caution!
Controlling of outputs in real time doesn't account output state during program execution from device internal memory.

4.8 Configuration panel of direct inputs

First and second input (IN1 and IN2) may be configured in fast operation mode. While this function is active, during program execution from device internal memory, selected input is sampled. While detecting change of state go to earlier selected line.
While using IN1 or IN2 input in fast operations mode, it can't be used during program operations, e.g. at command „go to if input".



Drawing 26. Configuration panel of direct inputs

4.9 Preview of current parameters

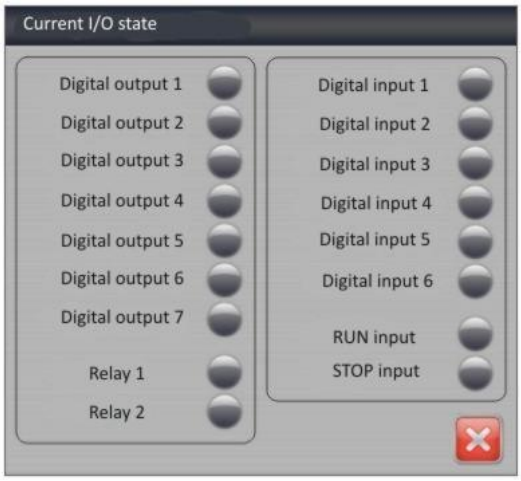
Program provides tracking in real time such parameters like velocity, motor position, current I/O state and preview of send messages by the device if such message appears in program saved in trajectory generator internal memory.

USB message preview

Drawing 27. USB message preview window

I/O state preview

This window gives current state preview of 6 digital inputs and 7 digital outputs, two relay outputs and single-purpose RUN and STOP inputs.



Drawing 28. I/O state preview window

Indicator	Description
	Input / Output inactive.
	Input/ Output active.

Motor state preview

For each motor is possible to preview its current velocity (in resolution per second) and current position (in steps).

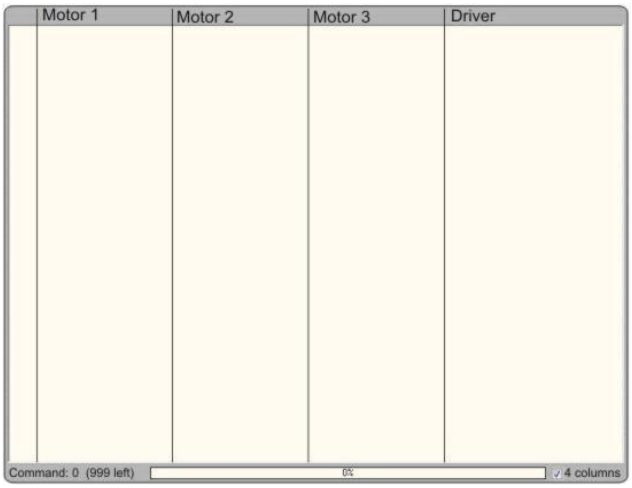
Drawing 29. Motor state preview window

Caution!
Displayed parameters may insignificantly diverge from real.

4.10 Programming window

The main part of application is a window, which enables creating own controlling program. It is divided into four columns. Respectively in columns are displayed commands for first,

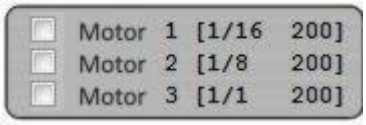
second and third motor. In fourth column are placed other commands, which concerns to trajectory generator.
In program there is placed a program progress bar and command counter. 4 columns Indicator – provides change of program displaying.



Drawing 30. Motor state preview window

4.11 Motor selection panel

All programming commands, concerns as well controlling as motor configuration, requires at least one motor, with assigned program command. It is possible to mark more than one motor when selected command should concern several motors.



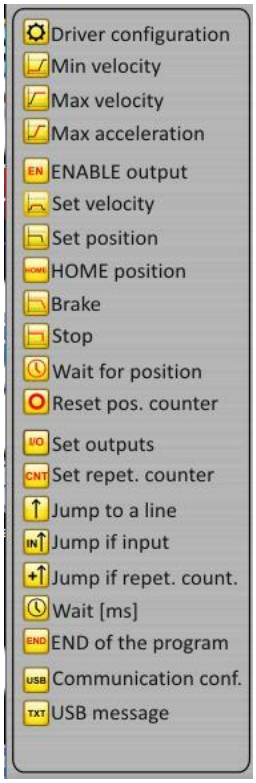
Drawing 31. Motor selection panel

Additionally in this panel is enclosed information about current step division settings of motor and driver. This parameters are very important, because other motion parameters are calculated on its base.

Caution!
Incorrect step division settings for driver and motor have influence for further program operations.

4.12 Programming panel – program commands

Commands, on base whose, are accomplished all functions are divided on four groups. In first group are located commands for initial device configuration. All four commands of this group have to be placed in the beginning of each program. Second group is responsible for motion control, third is responsible for jumps inside the program and controlling of outputs.



Drawing 32. Programming panel – program commands

Name	Description
Driver configuration	This command configures step division for motor and driver
Min velocity	This command configures minimal velocity of the motor
Max velocity	This command configures max velocity of the motor
Max acceleration	This command configures max acceleration of the motor
ENABLE output	This command set ENABLE output in active or inactive state
Set velocity	This command turns into velocity control mode and sets set velocity
Set position	This command turns into position control mode and sets set position
HOME position	This command starts searching for HOME position for selected motor
Brake	This command starts braking of the motor according to set motion parameters
Stop	This command cause immediate stop of the motor
Reset position counter	This command cause reset of the selected motor position
Wait for position	This command waits for reaching position, end of braking, stop and homing
Set outputs	This command set outputs in active or inactive state
Set repetition counter	This command set repetition counter
Jump to a line	This command jump to set program line
Jump if input	This command jump to in dependence on input state
Jump if repetition counter	This command jump in dependence on repetition counter state
Wait [ms]	This command cause setting of delay
END of the program	This command finishes operations of the program
Communication configuration	This command turns on/off sending messages
USB message	This command sends a message through USB

Driver configuration

Program provides independent configuration each of three motors. First parameter is defining step division made by stepper motor driver. Second parameter is defining number of full steps accrue to one motor revolution.

For ex ample for driver with set step division 1/16 of step and motor with 1,8° step should be set parameters as below:

- Step division [pertain to driver] 16
- Step per revolution [pertain to motor] 200

Command	Parameter	Description
drv(M1; 1/16; 200; 1; 1)	M1	Motor number

	1/16 200 1 1	Driver step division Number of motor full step per revolution Proximity sensor left (1 – active, 0 - inactive) Proximity sensor right (1 – active, 0 - inactive)

Drawing 33. Programming window – driver configuration

When Sensor XXX checkbox is check off, program automatically ignores activation of proximity sensor during movement. It is never ignored during searching of HOME position.

Minimal velocity

This parameter defines the lowest velocity, which must be implemented to accomplish motion tasks. Minimal velocity is defined in revolution per second, it is a safe velocity of reaching the position by motor. It is necessary to define this parameter to each motor. This parameter can't take values below zero.

Parameter settings:

- max +31234.74 / (motor step division * driver step division) [rev/sec]
- min above zero

Command	Parameter	Description
v_min(M1; 0,10)	M1 0,10	Motor number Min velocity [rev/sec]

Drawing 34. Programming window – min velocity

Maximal velocity

This parameter defines the biggest velocity, which be implemented to accomplish motion tasks. Maximal velocity is defined in revolution per second, it is upper limit of velocity used by motor for reaching set position. It is necessary to define this parameter for each motor. This parameter can't take values below zero.

Parameter settings:

- max +31234.74 / (motor step division * driver step division) [rev/sec]
- min above zero

Command	Parameter	Description

v_max(M1; 3,00)	M1 3,00	Motor number Max velocity [rev/sec]

Drawing 35. Programming window – max velocity

Max acceleration

This parameter, defined in resolution per second square means value of acceleration, which will be implemented by trajectory generator to reach max velocity, set velocity and at motor braking. This parameter must take values above zero.

Parameter settings:

- max e +543263.78 / (motor step division * driver step division) [rev/sec^2]
- min bigger than zero

Command	Parameter	Description
a_max(M1; 50,00)	M1 50,00	Motor number Max acceleration [rev/sec^2]

Drawing 36. Programming window – max acceleration

ENABLE output

This command cause setting of ENABLE output in active state (when indicator is marked), or inactive when is checked off. This signal is necessary for most of stepper motor drivers.

Command	Parameter	Description
enable(on)	on	ENABLE output state (on – active; off – inactive)

Drawing 37. Programming window – ENABLE output

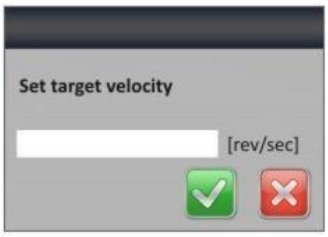
Set velocity

This parameter defined in resolution per second, determines target velocity for each motor. This command cause immediate turn into velocity control mode. This parameter can take values below zero. Sign defines motion direction. Zero velocity cause stopping the motor.

Parameter settings:

- max +31234.74 / (motor step division * driver step division) [rev/sec]
- min -31234.74 / (motor step division * driver step division) [rev/sec]

Command	Parameter	Description
v_dest(M1; 7,23)	M1 7,23	Motor number Set velocity [rev/sec]



Drawing 38. Programming window – set velocity

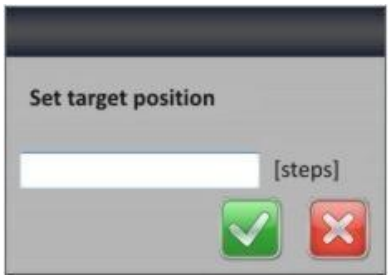
Set position

This parameter defined in steps is used for setting target motor position. This command cause automatic turn into position control mode with trapezoid acceleration profile by motor.

Parameter settings:

- max +8388607 [steps]
- min -8388607 [steps]

Command	Parameter	Description
go_to(M1; 1235)	M1 1235	Motor number Set position [step]



Drawing 39. Programming window –set position

HOME position

This command is used for searching of HOME position. First parameter defines velocity of moving away proximity sensor, second defines precise positioning velocity.

Parameter settings:

- max +31234.74 / (motor step division * driver step division) [rev/sec]
- min bigger than zero

Command	Parameter	Description
home(M1; 22; 0,11)	M1 22 0,11	Motor number homing velocity [rev/sec] Positioning velocity [rev/sec]

Drawing 40. Programming window – HOME position

Searching of Home position should take account kinematic features of controlled target. Signal of proximity sensor appears suddenly and cause rapid stop of the device. Reaching the Home position with too high speed may cause damage of the motor.
HOME signal (from proximity sensor) has hysteresis in its operations and motor stops in some distance from its appearing. As a result the trajectory generator reverse the motor with selected positioning velocity up to signal disappearance.
Velocity setting should be in inverse proportion to moved mass and should take account system rigidity. Properly selected parameters allows achieving high repeatability of reaching this point.
End of homing operation cause reset of position for selected motor.

Brake

This command starts braking of selected motor according to set motion parameters. Motor stopps is made with trapezoid velocity regulation, so it doesn't cause loss of current position.

Command	Parameter	Description
brake(M1)	M1	Motor number

Stop

This command cause immediate stop of selected motor. In some cases using of this command can cause loss of current position as a result of no velocity regulation during braking.

Command	Parameter	Description
stop(M1)	M1	Motor number

Wait for position

Orders like: Set position, HOME position, Stop, brake, to correct operations requires additional command – waits for end of movement. Order can concern three of the nearest

commands above, after one order for each motor. Incorrect use of order can cause stop of program execution.

Command	Parameter	Description
wait_motor(M1, M2)	M1, M2	Motor number

Example of command use:

Order	Description
001: go_to(M1; 12345) 002: home(M1; 5,00; 0,10) 003: wait_motor(M1) 004: set_out(out1 = on)	In this case waiting for position relates only to 002 order. 004 order will be executed only after finding Home position. 001 order –reaching 12345position – won't be executed correctly.
001: go_to(M1; 11111) 002: go_to(M2; 22222) 003: go_to(M3; 33333) 004: wait_motor(M1, M2, M3) 005: set_out(out1 = on)	In this case all orders will be executed correctly, 005 order will be executed only when all motors reach set position.
001: go_to(M1; 11111) 002: set_out(out1 = on) 003: wait_motor(M1) 004: set_out(out2 = on)	In this case 11111 position won't be reached due to 002 order. wait_motor(M1) command works correctly only when Set position, HOME position, Stop and Brake commands are directly above commands waiting for end of motion.

Reset position

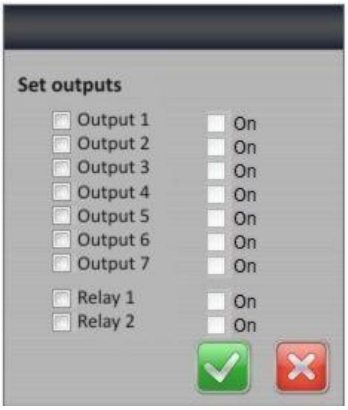
This command cause reset of current motor position. Execution of this command cause lost of current position.

Command	Parameter	Description
clear_motor(M1, M2)	M1, M2	Motor number

Set outputs

This command allows controlling of each output. Commands will set only when indicator in left column will be pointed out. Each change of output is placed in single order.

Command	Parameter	Description
set_out(out1 = off)	out1 off	Output number Output state (on – active; off – inactive)



Drawing 41. Programming window – output settings

Set repetition counter

Command of setting value of repetition counter. Counter can be used for multiple repeating of selected program part with command go to if repetition counter.

Parameter settings:

- max 65535 [repetition]

Command	Parameter	Description
counter(12345)	12345	Value of repetition counter

Drawing 42. Programming window – repetition counter setting

Jump to a line

Command of unconditional jump to selected program line. This command enables declare of unconditional loop executing the same task.

Command	Parameter	Description
jump_line(89)	89	Line number to which should be made a jump

Drawing 43. Programming window – go to a line

Jump if input

This command executes a conditional jump to line with set number, if selected input has defined state. Negative outcome of comparison causes execution of further part of the program.

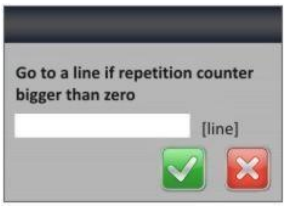
Command	Parameter	Description
if(in1 = off) jump_line(4)	in1 off 4	Number of checked input Input state(on – active ; off – inactive) Line number to which will be made a jump

Drawing 44. Programming window – go to program line in dependence on input

Jump if repetition counter

Conditional jump do program line with set number, executed if value of repetition counter (set by command „Set repetition counter”) is different from zero. Each jump decreases repetition counter value by 1. If counter state reaches zero, program will accomplish command in next line.

Command	Parameter	Description
if(counter>0) jump_line(12)	12	Line number to which will be made a jump



Drawing 45. Programming window – jump to program line in dependence on repetition counter state

Wait [ms]

Command cause delay in ms. After passing set timer, program goes to executing next command.

Parameter setting:

- max 86400000 [ms]

Command	Parameter	Description
wait_ms(123)	123	Waiting time [milliseconds]

Drawing 46. Programming window - delay

End of the program

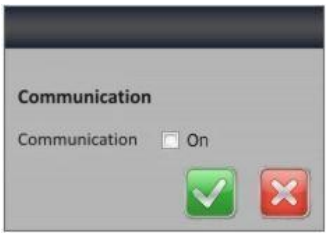
This command ends the program. In program may be several commands ending the program – separate for each program branch. It is necessary to place selected command in last line of the program for correct operation of the program.

Command	Parameter	Description
end()	-	-

Communication configuration

This command enables turn on/off sending message with command : USB message. It could be used for blocking of sending messages located in all program.

Command	Parameter	Description
set_USB(on)	on	Communication state (on – active; off – inactive)

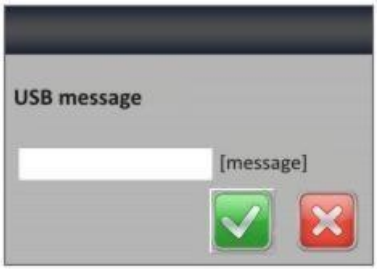


Drawing 47. Programming window – configuration of USB communication

USB message

This command enables sending of any, short, 7-character message through USB connector, if communication was activated by command: Communication configuration. This command can be useful at start of the program.

Command	Parameter	Description
print_USB(abcd,4)	abcd 4	Message Number of characters



Drawing 48. Programming window –USB message

Delete, edition and new command insertion

Program enables deleting of earlier inserted commands, edition and insert of new commands by context menu in main program window. To call the menu, firstly left-click on selected command, and when it will be highlighted right-click to call the menu.

Drawing 49. Context menu

As well „Delete" operation as „Insert" operation cause automatic command edition with jump to selected line. Additionally it is possible to delete commands by „Delete" button, when program line is highlighted. Change of command order in program is made by and buttons. Use of this commands don't introduce automatic command edition with jumps. Confirmation of all orders is made by „Enter" button, and resignation by „Esc" button.

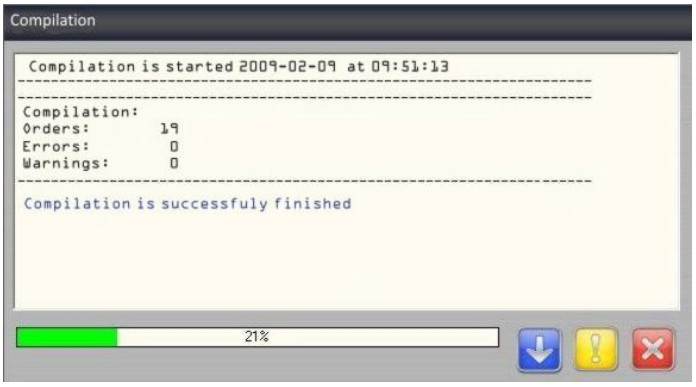
4.13 Description of program operation

All program commands are executed one by one, starting with first order. Each order apart of: Set position, HOME position, stop, and brake start operating when previous command

will finish. Program end its operation when is break by external signal STOP or END command.

4.14 Program compilation and programming

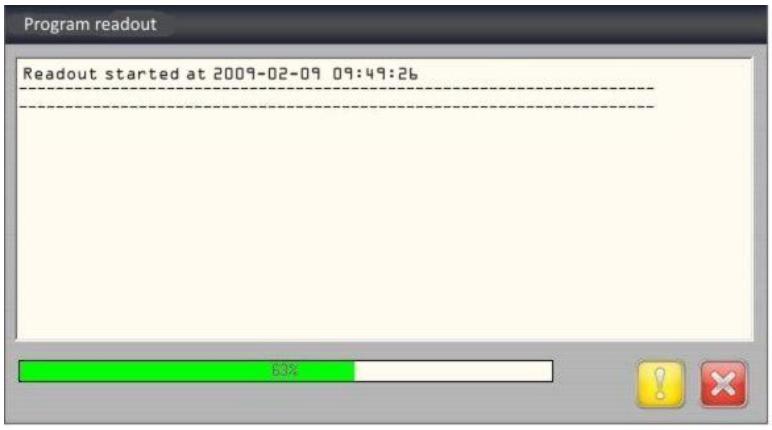
Program compilation means detection of basic errors created during writing of the program. We can pick out errors like: no defined motor parameters or program ending. Pressing of button cause close of the window. Button starts program recording to internal memory of the device, which can be stopped in any time by button. Progress bar informs about memory recording state.



Drawing 50. Compilation and programming

4.15 Program readout from internal memory

This application enables readout of program recorded in internal memory of trajectory generator MI389. Memory readout can be break at any time by button. Pressing of button cause close of window, not stop of readout. Readout of the program automatically appears in main widow of the program. Progress bar informs about memory readout state.

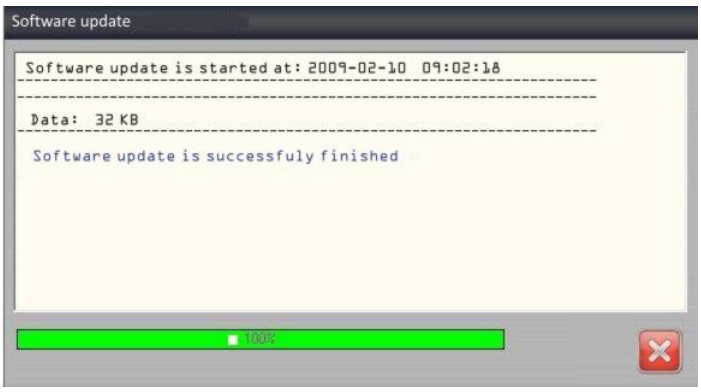


Drawing 51. Program readout from internal memory

4.16 Software update

Software update requires entering a password „**12345abcde**" to avoid accidental reprogramming of the device. After entering the password a diode UPLOAD will light on, program ask to show access path *.bin with current software. After confirmation, a

TRANSMISSION diode should light on, which informs about start of software update. Progress bar informs about update state. If everything is ok., all diodes should light on, then light off and in main window should appear and information with successful end of software update. When all diodes are light on it means that occurs an update error. In that case you should restart the device and make all steps again.



Drawing 52. Software update

Caution!
Software update is a process, which shouldn't be interrupted under no circumstances. Incorrect programming of the device may cause undetermined operations during its work.

5. Program example

Program show exemplary movement of three motors in dependence on input signal activity. Program works cyclically, in infinite loop.

Program example

001: drv(M1; 1/1; 200; 1; 1)	Set step division on 1 for first motor with 1,8° step
002: drv(M2; 1/8; 200; 1; 1)	Set step division on 8 for second motor with 1,8° step
003: drv(M3; 1/16; 2001; 1)	Set step division on 16 for third motor with 1,8° step
004: v_min(M1; 0,01)	Set min velocity: 0,01 [rev/sec] – for first motor
005: v_min(M2; 0,02)	Set min velocity: 0,02 [rev/sec] – for second motor
006: v_min(M3; 0,01)	Set min velocity: 0,03 [rev/sec] – for third motor
007: v_max(M1; 4,00)	Set max velocity: 4 [rev/sec] – for first motor
008: v_max(M2; 5,00)	Set max velocity: 5 [rev/sec] – for second motor
009: v_max(M3; 6,00)	Set max velocity: 6 [rev/sec] – for third motor
010: a_max(M1; 50,00)	Set max acceleration: 50 [rev/sec^2] – for first motor
011: a_max(M2; 60,00)	Set max acceleration: 60 [rev/sec^2] – for second motor
012: a_max(M3; 70,00)	Set max acceleration: 70 [rev/sec^2] – for third motor
013: set_out(out1 = off)	Set 1 output in inactive state
014: set_out(out2 = off)	Set 2output in inactive state
015: set_out(out3 = off)	Set 3 output in inactive state
016: set_out(out4 = off)	Set 4 output in inactive state
017: set_out(out5 = off)	Set 5 output in inactive state
018: set_out(out6 = off)	Set 6 output in inactive state
019: set_out(out7 = off)	Set 7 output in inactive state
020: set_out(pk1 = off)	Set 1 relay in inactive state
021: set_out(pk2 = off)	Set 2 relay in inactive state
022: counter(2)	Set repetition counter: 2
023: set_out(out1 = on)	Set 1 output in active state
024: wait_ms(200)	Wait 200 [ms]
025: set_out(out1 = off)	Set 1 output in inactive state
026: wait_ms(200)	Wait 200 [ms]
027: if(counter>0) jump_line(23)	If counter bigger than zero jump to 23 line and decrease the counter
028: if(in1 = off) jump_line(28)	If 1 input is inactive jump to 28 line
029: enable(on)	Set ENABLE output in active state
030: home(M1; 3,00; 0,01)	Home first motor with velocity 3 [rev/sec]
031: home(M2; 4,00; 0,01)	Home second motor with velocity 4 [rev/sec]
032: home(M3; 5,00; 0,01)	Home third motor with velocity 5 [rev/sec]
033: wait_motor(M1) wait_motor(M2) wait_motor(M3)	Wait for end of movement of all three motors
034: counter(1)	Set repetition counter: 1
035: set_out(out1 = on)	Set 1 output in active state
036: wait_ms(200)	Wait 200 [ms]
037: set_out(out1 = off)	Set 1 output in inactive state
038: wait_ms(200)	Wait 200 [ms]
039: if(counter>0) jump_line(35)	If counter bigger than zero jump to 35 line and decrease the counter
040: go_to(M1; 1111)	Go to set position for first motor: 1111
041: go_to(M2; 1111)	Go to set position for second motor: 2222
042: go_to(M3; 1111)	Go to set position for third motor: 3333
043: wait_motor(M1) wait_motor(M2) wait_motor(M3)	Wait for end of movement of all three motors
044: if(in2 = on) jump_line(49)	If input 2 is inactive jump to 49 line
045: if(in3 = on) jump_line(56)	If input 3 is inactive jump to 56 line
046: if(in4 = on) jump_line(67)	If input 4 is active jump to 67 line
047: if(in5 = on) jump_line(74)	If input 4 is active jump to 74 line
048: jump_line(44)	Jump to 44 line
049: set_out(out2 = on)	Set 2 output in active state
050: go_to(M1; 2222)	Go to set position for first motor: 2222
051: go_to(M2; 2222)	Go to set position for second motor: 2222
052: go_to(M3; 2222)	Go to set position for third motor: 2222
053: wait_motor(M1) wait_motor(M2) wait_motor(M3)	Wait for end of movement of all three motors
054: set_out(out2 = off)	Set 2 output in inactive state
055: jump_line(40)	Jump to 40 line
056: set_out(out3 = on)	Set 3 output in active state
057: v_dest(M1; -2,00)	Set selected velocity for first motor: -2 [rev/sec]
058: v_dest(M2; -2,00)	Set selected velocity for second motor: -2 [rev/sec]
059: v_dest(M3; -2,00)	Set selected velocity for third motor: -2 [rev/sec]
060: wait_ms(333)	Wait 333 [ms]
061: brake(M1)	Brake first motor
062: brake(M2)	Brake second motor
063: brake(M3)	Brake third motor
064: wait_motor(M1) wait_motor(M2) wait_motor(M3)	Wait for end of movement of all three motors
065: set_out(out3 = off)	Set 3 output in inactive state
066: jump_line(40)	Jump to 40 line
067: set_out(out4 = on)	Set 4 output in active state
068: go_to(M1; 3333)	Go to set position for first motor: 3333
069: go_to(M2; 3333)	Go to set position for second motor: 3333
070: go_to(M3; 3333)	Go to set position for third motor: 3333
071: wait_motor(M1) wait_motor(M2) wait_motor(M3)	Wait for end of movement of all three motors
072: set_out(out4 = off)	Set 4 output in inactive state
073: jump_line(40)	Jump to 40 line
074: enable(off)	Set ENABLE output in inactive state
075: set_out(out1 = off)	Set 1 output in inactive state
076: set_out(out2 = off)	Set 2output in inactive state
077: set_out(out3 = off)	Set 3 output in inactive state
078: set_out(out4 = off)	Set 4 output in inactive state
079: set_out(out5 = off)	Set 5 output in inactive state
080: set_out(out6 = off)	Set 6 output in inactive state

081: set_out(out7 = off)	Set 7 output in inactive state
082: set_out(pk1 = off)	Set 1 relay in inactive state
083: set_out(pk2 = off)	Set 2 relay in inactive state
084: end()	End operation of the program

Description of program operation

Line: 001 – 012	Initialization of basic motor operations parameters (motor step division, driver step division, min velocity, max velocity, max acceleration). All settings are necessary for correct execution of movement tasks.
Line: 013 – 021	Deactivation of all outputs (digital and relay) by setting them in inactive state.
Line: 022 – 027	Triple execution of sequence: set output in active state, wait, set output in inactive state, wait.
Line: 028	Waiting for appearing high state on 1 input. After its detection, program goes to execution of further commands.
Line: 029	Activation ENABLE signal for controlling
Line: 030 – 032	Homing of all three motors
Line: 033	Waiting for reaching HOME position by all three motors
Line: 034 – 039	Twice execution of the sequence: set output in active state, wait, set output in inactive state, wait
Line: 040 – 042	Go to set position for all motors
Line: 043	Waiting for reaching set position by all three motors
Line: 044 – 048	Waiting for appearing high state on 1,2 or 3 input. In dependence on input is made a jump to proper line.
Line: 049	Setting of 2 output in active state
Line: 050 – 052	Setting of selected position for all three motors
Line: 053	Waiting for reaching set position by all three motors
Line: 054	Setting of 2 output in inactive state
Line: 055	Jump to set line
Line: 056	Setting of 3 input in active state
Line: 057 – 059	Go to set velocity for all three motors
Line: 060	Waiting 333 ms
Line: 061 – 063	Start of braking by all three motors
Line: 064	Waiting for reaching zero velocity by all three motors
Line: 065	Setting of 3 output in inactive state
Line: 066	Jump to set line
Line: 067	Setting of 4 output in active state
Line: 068 – 070	Go to set position for all three motors
Line: 071	Waiting for reaching set position by all three motors
Line: 072	Setting of 4 output in inactive state
Line: 073	Jump to set line
Line: 074	Set ENABLE signal in inactive state
Line: 075 – 083	Deactivation of all outputs (digital and relay) by setting them in inactive state.
Line: 084	End of the program

6. Safety and assembly rules

- Use of described devices in special importance systems (e.g.: medical systems, vehicles, etc.) requires use of additional protection, to prevent any possible interruptions of the device operation.
- Devices must be correctly mounted in the panel. Not following to safety measures can cause an electric shock.
- Do not power external devices when the device is on.
- Do not disassemble and make any adaptation of the device on your own. If you need any help please contact with WObit technical department. Unauthorized changes can cause injury or fire. It results also the device invalidation.
- This devices can't be exploited outside. It may cause an electric shock and shorten device life.
- External connections should be made by ZOAWG connectors.
- Exceeding of recommended operations parameters can cause damage of the device or fire.

- For clearing of the device it is not allowed to use substances containing water or oils.