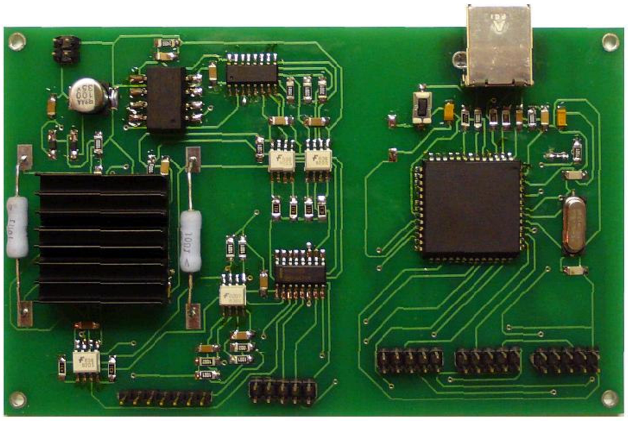
ASEQ MC35

USB stepper motor controller, driver



Products information:

- ST L6228 chip for bipolar stepper motors - maximum 2**.8A** drive current (1.4 A RMS)
- High speed USB controller can control up to 3 axis
- 1/2 microstep - higher accuracy, smoother operation ! with up to 256 microsteps !
- Overload & over-current & over-temperature protection, full protection of your computer and peripheral equipment
- Software full current settings control independently for each phase according to the user the actual current requirement
- Full closed-type optical driver isolation from USB controller to protect the user's computer and equipment and avoid errors on high-frequency USB connection
- End switches to perform emergency stop to avoid mechanical damage of your system and accurate calibration
- Development KID for Visual C and LabView is included

Attention:

- Please connect the power supply to the power supply connector of **12-36 V, 3A** (for one driver), 6 A – for two drivers and 9 A – for 3 drivers (for maximum load).
- The interface of driver shall be connected to one of the axis on USB-controller panel (cable is included in standard configuration).
- Limiting switches and motor shall be connected to the driver

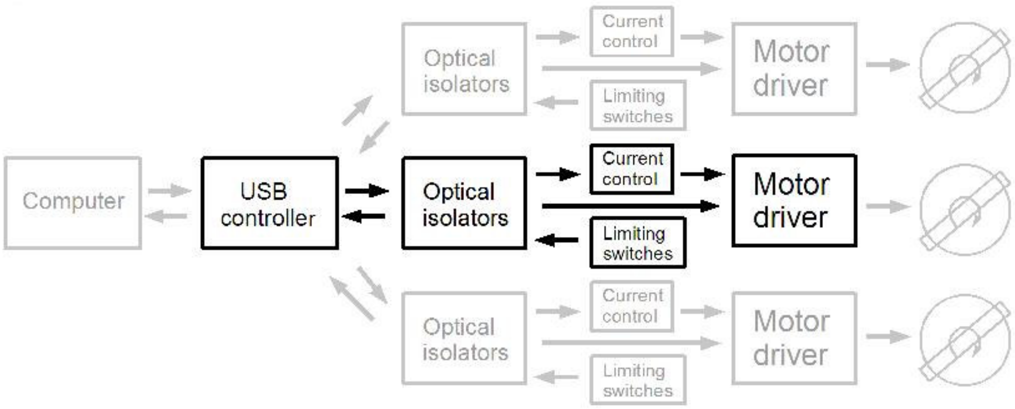
- This board can work with both **unipolar and bipolar** stepper motors

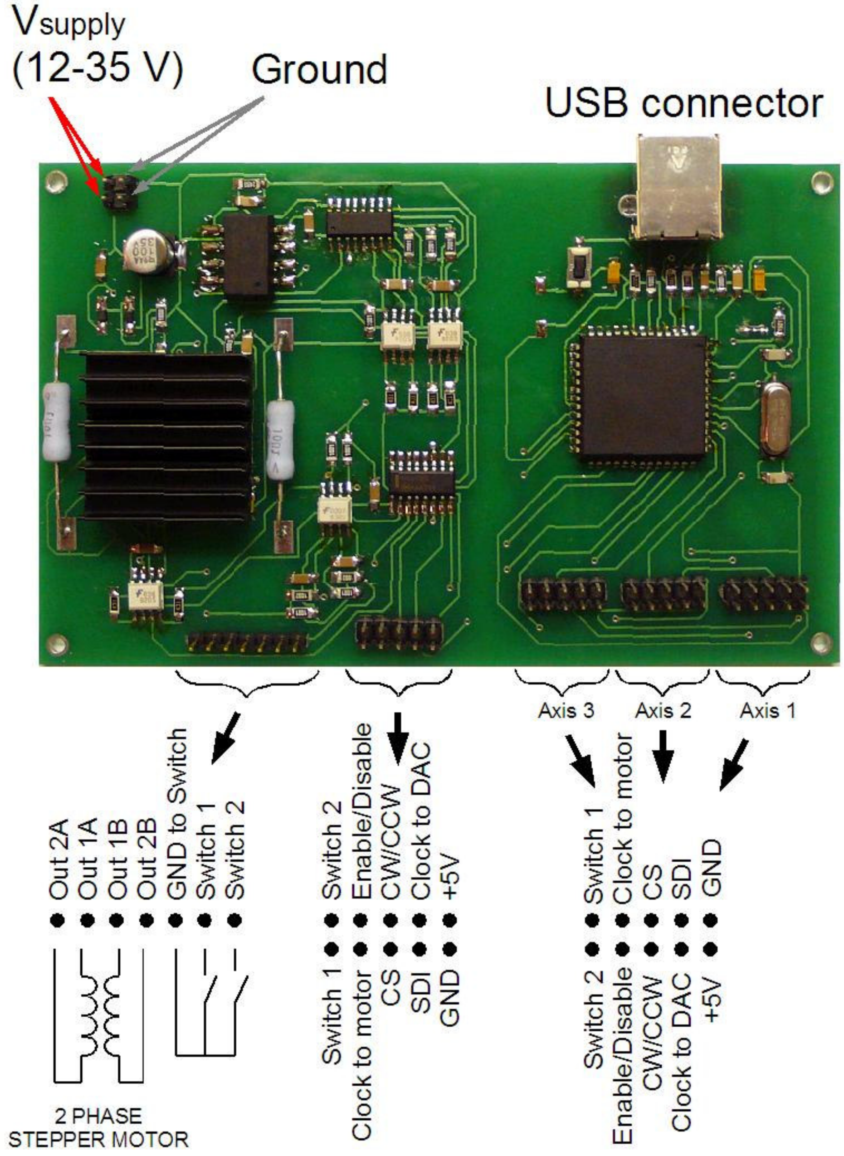
Standard configuration includes:

- USB controller (supports up to 3 axis)
- Driver for one stepper motor
- USB cable to connect USB controller to the computer
- CD with software

Separately can be ordered:

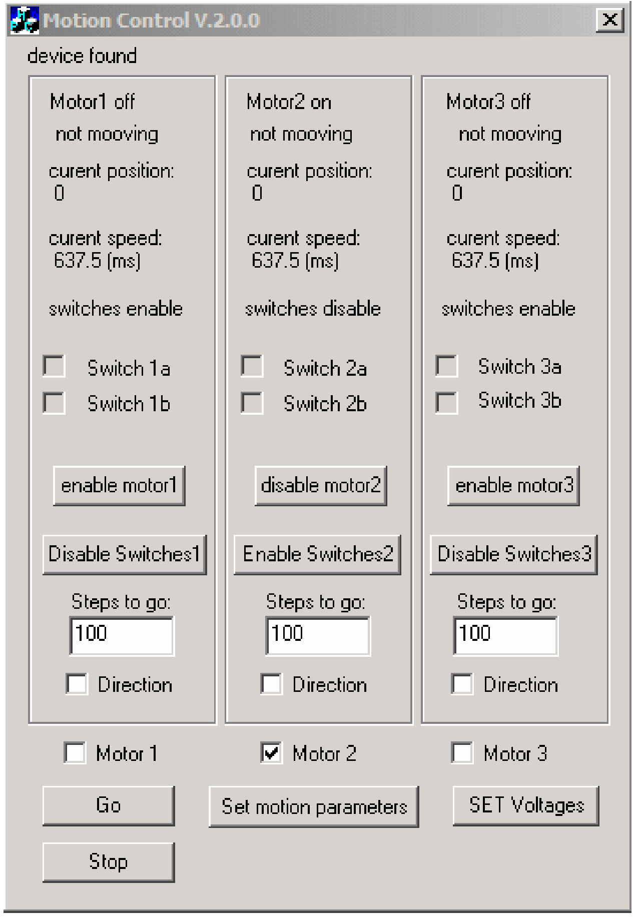
- Additional drivers
- Power supply





Size: 76 mm x 122 mm.

Current on the stepper motor is regulated by adjusting voltage on the Digital to Analog Converters outputs. When the voltage drop across the sense resistor becomes greater than the voltage at the DAC outputs the sense comparator triggers the monostable switching the bridge off.
In order not to damage stepper motor and driver check the absolute maximum parameters of your stepper motor as well as maximum allowed current trough sense resistors before increasing current on stepper motor by increasing output voltage from DAC



Motion Control V.2.0.0

device found

Motor1 off
not mooving
curent position:
0
curent speed:
637.5 (ms)
switches enable
□ Switch 1a
□ Switch 1b
Steps to go:
100
□ Direction

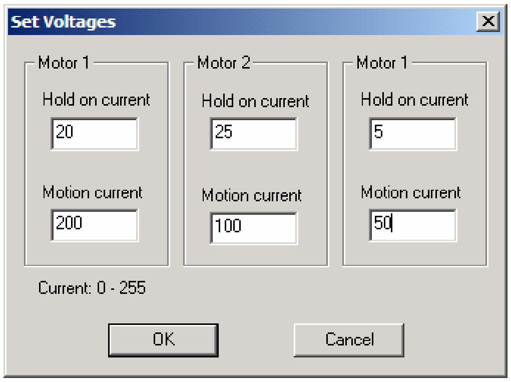
Motor2 on
not mooving
curent position:
0
curent speed:
637.5 (ms)
switches disable
□ Switch 2a
□ Switch 2b
Steps to go:
100
□ Direction

Motor3 off
not mooving
curent position:
0
curent speed:
637.5 (ms)
switches enable
□ Switch 3a
□ Switch 3b
Steps to go:
100
□ Direction

□ Motor 1 ☑ Motor 2 □ Motor 3

Standard view of executable file.

Initially by default when controller is connected to USB it disables drivers and sets output voltage of both Digital to Analog Converters to zero which corresponds to the zero voltage on load resistors. Enable driver and set non-zero voltage on the output of DAC to start stepper motor operation. 255 – maximum voltage (5V) can be set on the outputs of the DAC.



Set Voltages

Motor 1
Hold on current
20
Motion current
200

Motor 2
Hold on current
25
Motion current
100

Motor 1
Hold on current
5
Motion current
50

Current: 0 - 255

It is possible to set maximum voltage independently for both coils. This function allows to set motor in microstepping position.
Direction of the stepper motor rotation is set by three knobs (Direction Axis *i*) for each axis *i*.

To start motors rotation chose motors to rotate by clicking on corresponding axis. Then push button “Go”.

To stop rotation before motors reach desired position click “Stop”. It will stop motors with enabled “Axis *i*”.

Stepper motors rotates if both switches are “off” or “Switchers” function is off. It allows avoiding rotation out of the possible range and simultaneously finding reference point. The motor stops as soon as the motor reaches position when one of the Switches of the corresponding motor goes to “ON” position. In this case disable switchers and start rotation in the opposite direction. It is strongly recommended to enable Switchers again later.

Set rotation parameters for each motor before start rotation. Those parameters include:

a. Direction.

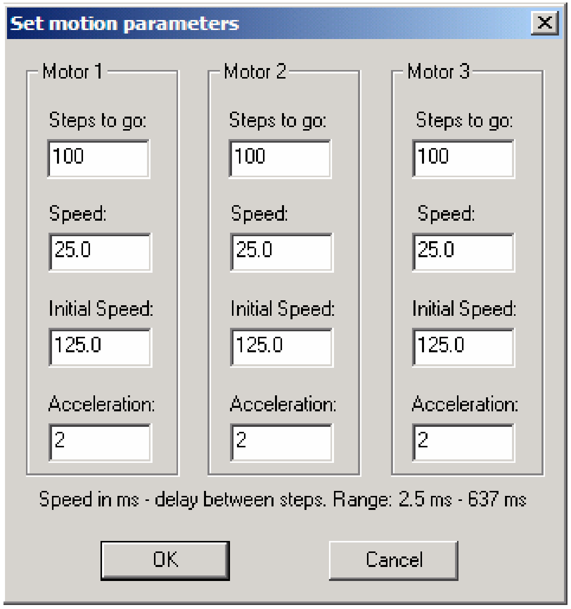
b. Enable or disable switchers.

c. Number of steps to rotate (65535 steps-maximum).

d. Maximum rotation speed (2.5 ms- minimum delay time between two consecutive steps; 637 ms - minimum).

e. Initial rotation speed (the same limits as parameter *d*). This parameter should be bigger that parameter *d*.

f. Acceleration –speed change at each step multiplied by 2.5 ms (1 – minimum, 255 – maximum).



	Motor 1	Motor 2	Motor 3
Steps to go:	100	100	100
Speed:	25.0	25.0	25.0
Initial Speed:	125.0	125.0	125.0
Acceleration:	2	2	2

Speed in ms - delay between steps. Range: 2.5 ms - 637 ms

Functions:

1. Enable/Disable drivers

void Enmotors(bool motor1, bool motor2, bool motor3, bool DisEn)

Output parameters: void
Input parameters:

bool motor1 – first driver is selected if TRUE; first driver is not selected if FALSE.

bool motor2 – second driver is selected if TRUE; second driver is not selected if FALSE.

bool motor3 – third driver is selected if TRUE; third driver is not selected if FALSE.

bool DisEn – selected drivers will be enabled if TRUE; selected drivers will be disabled if FALSE.

2. Set parameters for the first driver

void SetParametersForMotor1(unsigned short StepsToGo, char Speed, char InitSpeed, char Acceleration)

Output parameters: void
Input parameters:

unsigned short StepsToGo – number of steps to rotate (1-65535).

char Speed - Rotation speed. Range: 1-255. Delay time between two consecutive steps = Speed*2,5 ms.

char InitSpeed – Initial rotation speed. Range: 1-255. Delay time between two consecutive steps = InitSpeed *2,5 ms. Should be bigger than Speed.

char Acceleration – acceleration. Range: 1-255.

3. Set parameters for the second driver

void SetParametersForMotor2(unsigned short StepsToGo, char Speed, char InitSpeed, char Acceleration)

Output parameters: void
Input parameters:

unsigned short StepsToGo – number of steps to rotate (1-65535).

char Speed - Rotation speed. Range: 1-255. Delay time between two consecutive steps = Speed*2,5 ms.

char InitSpeed – Initial rotation speed. Range: 1-255. Delay time between two consecutive steps = InitSpeed *2,5 ms. Should be bigger than Speed.

char Acceleration – acceleration. Range: 1-255.

4. Set parameters for the third driver

void SetParametersForMotor3(unsigned short StepsToGo, char Speed, char InitSpeed, char Acceleration)

Output parameters: void
Input parameters:

unsigned short StepsToGo – number of steps to rotate (1-65535).

char Speed - Rotation speed. Range: 1-255. Delay time between two consecutive steps = Speed*2,5 ms.

char InitSpeed – Initial rotation speed. Range: 1-255. Delay time between two consecutive steps = InitSpeed *2,5 ms. Should be bigger than Speed.

char Acceleration – acceleration. Range: 1-255.

5. Set Voltage for the first driver

void SetVoltage1(char Voltage)

Output parameters: void
Input parameters:

char Voltage – voltage. Range: 0-255. 0 corresponds to 0V, 255 corresponds to 5V

6. Set Voltage for the second driver

void SetVoltage2(char Voltage)

Output parameters: void
Input parameters:

char Voltage – voltage. Range: 0-255. 0 corresponds to 0V, 255 corresponds to 5V

7. Set Voltage for the third driver

void SetVoltage3(char Voltage)

Output parameters: void
Input parameters:

char Voltage – voltage. Range: 0-255. 0 corresponds to 0V, 255 corresponds to 5V

8. Start rotation

void Go(bool motor1, bool Direction1, bool DisSwitch_1, bool motor2, bool Direction2, bool DisSwitch_2, bool motor3, bool Direction3, bool DisSwitch_3)

Output parameters: void
Input parameters:

bool motor1 – first driver is selected if TRUE; first driver is not selected if FALSE.

bool Direction1 – direction for the first driver

bool DisSwitch_1 – switchers function for the first driver is enabled if FALSE, switchers function is disabled if TRUE

bool motor2 – second driver is selected if TRUE; second driver is not selected if FALSE.

bool Direction2 – direction for the second driver

bool DisSwitch_2 – switchers function for the second driver is enabled if FALSE, switchers function is disabled if TRUE

bool motor3 – third driver is selected if TRUE; third driver is not selected if FALSE.

bool Direction3 – direction for the third driver

bool DisSwitch_3 – switchers function for the third driver is enabled if FALSE, switchers function is disabled if TRUE

9. Stop rotation

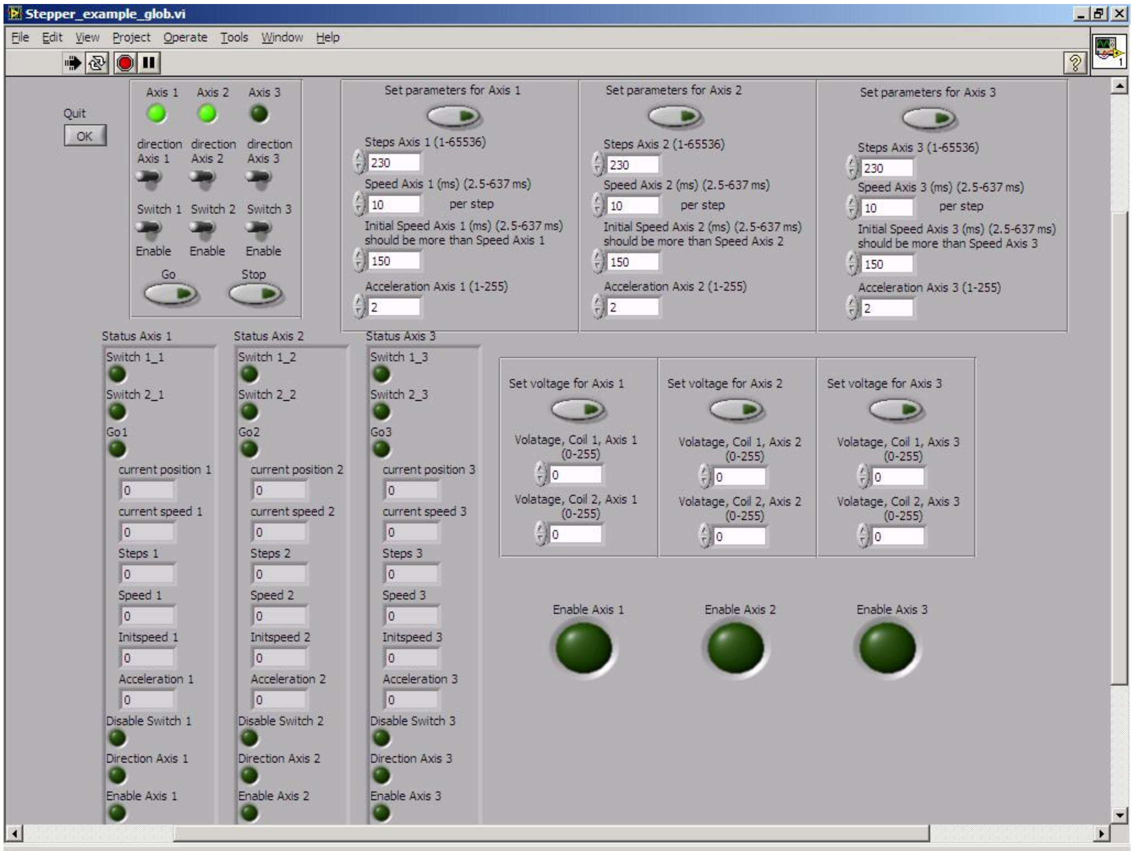
void Stop(bool motor1, bool motor2, bool motor3)

Output parameters: void
Input parameters:

bool motor1 – first driver is selected if TRUE; first driver is not selected if FALSE.

bool motor2 – second driver is selected if TRUE; second driver is not selected if FALSE.

bool motor3 – third driver is selected if TRUE; third driver is not selected if FALSE.



Steps Axis 1 (1-65536): 230
Speed Axis 1 (ms) (2.5-637 ms): 10 per step
Initial Speed Axis 1 (ms) (2.5-637 ms) should be more than Speed Axis 1: 150
Acceleration Axis 1 (1-255): 2

Steps Axis 2 (1-65536): 230
Speed Axis 2 (ms) (2.5-637 ms): 10 per step
Initial Speed Axis 2 (ms) (2.5-637 ms) should be more than Speed Axis 2: 150
Acceleration Axis 2 (1-255): 2

Steps Axis 3 (1-65536): 230
Speed Axis 3 (ms) (2.5-637 ms): 10 per step
Initial Speed Axis 3 (ms) (2.5-637 ms) should be more than Speed Axis 3: 150
Acceleration Axis 3 (1-255): 2

Status Axis 1: Switch 1_1, Switch 2_1, Go1, current position 1: 0, current speed 1: 0, Steps 1: 0, Speed 1: 0, Initspeed 1: 0, Acceleration 1: 0, Disable Switch 1, Direction Axis 1, Enable Axis 1

Status Axis 2: Switch 1_2, Switch 2_2, Go2, current position 2: 0, current speed 2: 0, Steps 2: 0, Speed 2: 0, Initspeed 2: 0, Acceleration 2: 0, Disable Switch 2, Direction Axis 2, Enable Axis 2

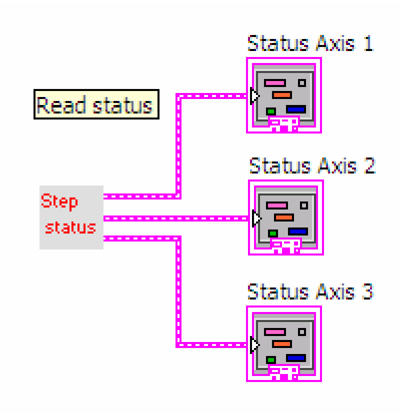
Status Axis 3: Switch 1_3, Switch 2_3, Go3, current position 3: 0, current speed 3: 0, Steps 3: 0, Speed 3: 0, Initspeed 3: 0, Acceleration 3: 0, Disable Switch 3, Direction Axis 3, Enable Axis 3

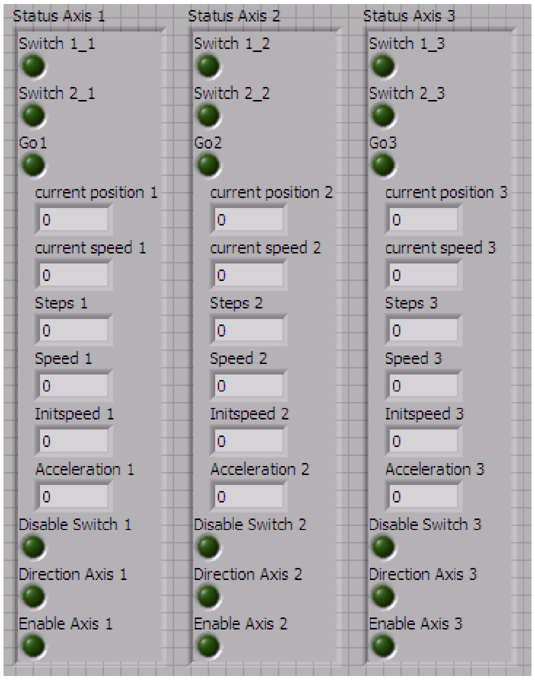
Volatage, Coil 1, Axis 1 (0-255): 0
Volatage, Coil 2, Axis 1 (0-255): 0
Volatage, Coil 1, Axis 2 (0-255): 0
Volatage, Coil 2, Axis 2 (0-255): 0
Volatage, Coil 1, Axis 3 (0-255): 0
Volatage, Coil 2, Axis 3 (0-255): 0

Software example written on LabView to operate with controller (included in standard configuration).

Functions (SubVIs):

1. Status VI:





Status Axis 1
Switch 1_1 ●
Switch 2_1 ●
Go1 ●
current position 1: 0
current speed 1: 0
Steps 1: 0
Speed 1: 0
Initspeed 1: 0
Acceleration 1: 0
Disable Switch 1 ●
Direction Axis 1 ●
Enable Axis 1 ●

Status Axis 2
Switch 1_2 ●
Switch 2_2 ●
Go2 ●
current position 2: 0
current speed 2: 0
Steps 2: 0
Speed 2: 0
Initspeed 2: 0
Acceleration 2: 0
Disable Switch 2 ●
Direction Axis 2 ●
Enable Axis 2 ●

Status Axis 3
Switch 1_3 ●
Switch 2_3 ●
Go3 ●
current position 3: 0
current speed 3: 0
Steps 3: 0
Speed 3: 0
Initspeed 3: 0
Acceleration 3: 0
Disable Switch 3 ●
Direction Axis 3 ●
Enable Axis 3 ●

Command returns status of three drivers based on information received from microcontroller.

“Switch 1_*i*”, “Switch 2_*i*” (where *i*=1, 2, 3 driver) shows status of both switchers for correspondent axis. If indicator is “ON” switcher is “ON”. If indicator is “OFF” switcher is “OFF”.

“Go*i*”, (where *i*=1, 2, 3 driver) shows motion status for correspondent axis. If “on” motor is moving.

“Current speed” – current speed (number has meaning only if motor is moving). Delay time between two consecutive steps = “Current speed”*2,5 ms.

“Steps” – targeted number of steps set by “Set parameters” function.

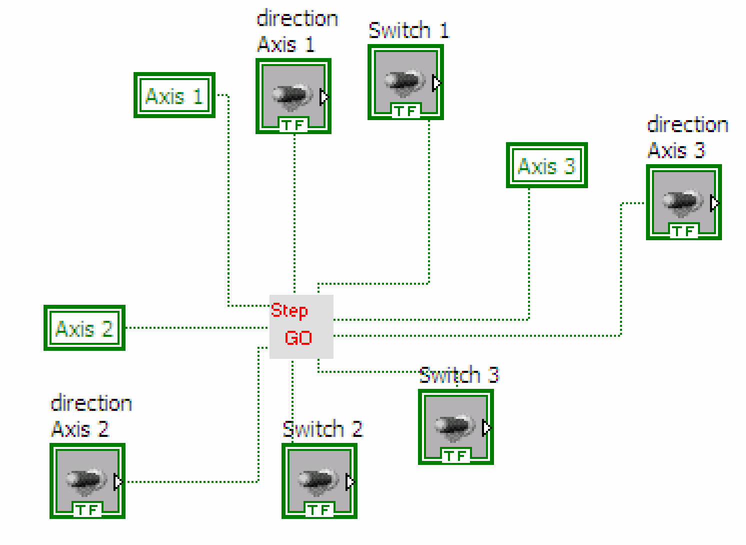
“Iitspeed” – initial speed set by “Set parameters” function. Delay time between two consecutive steps = “Iitspeed”*2,5 ms.

“Acceleration” – acceleration set by “Set parameters” function.

“Disable switch” – status of the Switcher function. If “ON” – Switcher function is “ON”. If “OFF” – Switcher function is “OFF”.

“Enable Axis” – Enable/Disable driver. If “ON” – driver is “ON”. If “OFF” – driver is “OFF”.

2. GO VI:



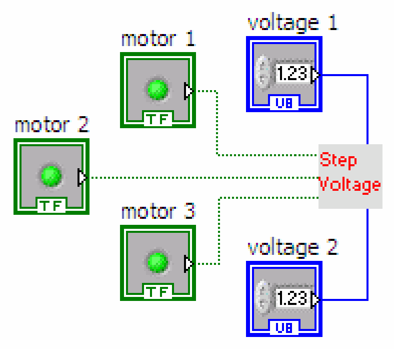
Command starts rotation of the selected motors.

“Axis 1”, “Axis 2”, “Axis 3” – if “ON” – corresponding motor is selected; if “OFF” corresponding motor is not selected.

“Direction Axis 1”, “Direction Axis 2”, “Direction Axis 3” – direction of the corresponding motor.

“Switch 1”, “Switch 2”, “Switch 3” - if “ON” – corresponding Switcher function is enabled; if “OFF” corresponding Switcher function is disabled.

3. Set Voltage VI:



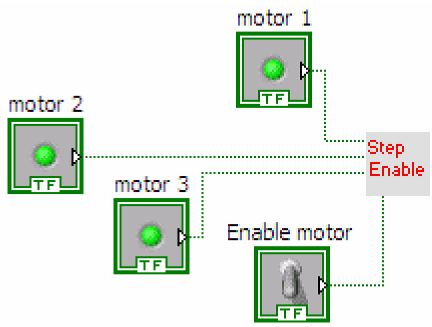
Command sets voltage on two sense resistors. Voltage will be set on all drivers if all motors are selected.

“motor 1”, “motor 2”, “motor 3” – if “ON” – corresponding motor is selected; if “OFF” corresponding motor is not selected.

“voltage 1” – voltage on the first sense resistor. Range: 0-255.

“voltage 2” – voltage on the second sense resistor. Range: 0-255.

3. Enable motor VI:

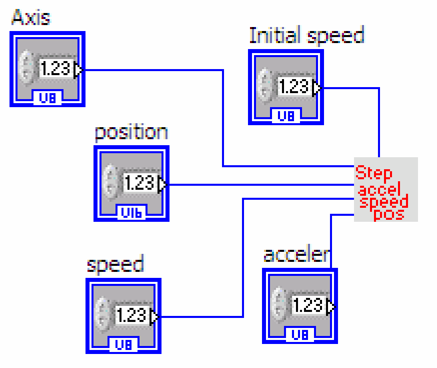


Command enables/disables drivers.

“motor 1”, “motor 2”, “motor 3” – if “ON” – corresponding motor is selected; if “OFF” corresponding motor is not selected.

“Enable motor” – If “ON” – selected motors will enabled, if “OFF” – selected motors will be disabled.

3. Set parameters VI:



Command sets motion parameters.

“Axis” if “1” – parameters will be set for driver 1,

if “2” – parameters will be set for driver 2,

if “3” – parameters will be set for driver 3,

“Speed” – Rotation speed. Range: 1-255. Delay time between two consecutive steps = “Speed”*2,5 ms.

“Steps” – Number of steps to move. Range: 1-65535

“Initial speed” – Initial rotation speed. Range: 1-255. Delay time between two consecutive steps = “Initial speed”*2,5 ms. Should be bigger than “Speed”.

“Acceleration” – acceleration. Range: 1-255.