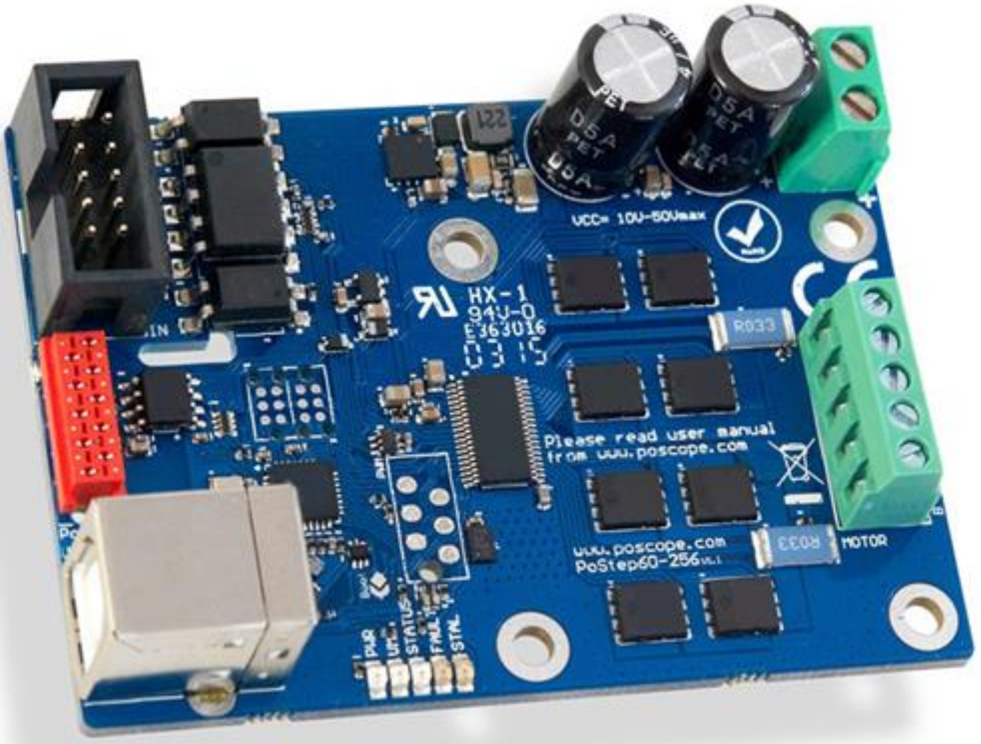
PoStep60

User manual

Version: 25/6/2016


VCC= 10V-50Vmax
HX-1
94V-0
E363016
R033
Please read user manual
from www.poscope.com
www.poscope.com
PoStep60-256
MOTOR
PWR
STATUS
STALL

Please read the following notes

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice.
2. PoLabs does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of PoLabs products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of PoLabs or others. PoLabs claims the copyright of, and retains the rights to, all material (software, documents, etc.) contained in this release. You may copy and distribute the entire release in its original state, but must not copy individual items within the release other than for backup purposes.
3. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of the products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. PoLabs assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
4. PoLabs has used reasonable care in preparing the information included in this document, but PoLabs does not warrant that such information is error free. PoLabs assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
5. PoLabs devices may be used in equipment that does not impose a threat to human life in case of the malfunctioning, such as: computer interfaces, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment, and industrial robots.
6. Measures such as fail-safe function and redundant design should be taken to ensure reliability and safety when PoLabs devices are used for or in connection with equipment that requires higher reliability, for example: traffic control systems, anti-disaster systems, anticrime systems, safety equipment, medical equipment not specifically designed for life support, and other similar applications.
7. PoLabs devices shall not be used for or in connection with equipment that requires an extremely high level of reliability and safety, as for example: aircraft systems, aerospace equipment, nuclear reactor control systems, medical equipment or systems for life support (e.g. artificial life support devices or systems), and any other applications or purposes that pose a direct threat to human life.
8. You should use the PoLabs products described in this document within the range specified by PoLabs, especially with respect to the maximum rating, operating supply voltage range and other product characteristics. PoLabs shall have no liability for malfunctions or damages arising out of the use of PoLabs products beyond such specified ranges.
9. Although PoLabs endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, PoLabs products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a PoLabs product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures.
10. Usage: the software in this release is for use only with PoLabs products or with data collected using PoLabs products.
11. Fitness for purpose: no two applications are the same, so PoLabs cannot guarantee that its equipment or software is suitable for a given application. It is therefore the user's responsibility to ensure that the product is suitable for the user's application.
12. Viruses: this software was continuously monitored for viruses during production, however the user is responsible for virus checking the software once it is installed.
13. Upgrades: we provide upgrades, free of charge, from our web site at www.poscope.com. We reserve the right to charge for updates or replacements sent out on physical media.
14. Please contact a PoLabs support for details as to environmental matters such as the environmental compatibility of each PoLabs product. Please use PoLabs products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. PoLabs assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
15. Please contact a PoLabs support at support@poscope.com if you have any questions regarding the information contained in this document or PoLabs products, or if you have any other inquiries.
16. The licensee agrees to allow access to this software only to persons who have been informed of and agree to abide by these conditions.
17. Trademarks: Windows is a registered trademark of Microsoft Corporation. PoKeys, PoKeys55, PoKeys56U, PoKeys56E, PoScope, PoLabs and others are internationally registered trademarks.

Contents

1 Introduction 6

  1.1 General symbols for this instructions 6

2 Description 6

3 PoStep60 features 6

4 Connection diagram 7

  4.1 Board use requirements 7

  4.2 Connection and setting 7

  4.3 PoStep pinout 8

  4.4 Stepper motor connection 8

    4.4.1 10 pin IDC connector pinout 8

    4.4.2 Stepper motor connections 9

  4.5 DC motor connection 11

    4.5.1 6 pin PWM connector pinout 11

    4.5.2 DC motor connection 11

  4.6 PoStep LEDs 12

    4.6.1 Status LED 12

  4.7 Requirements 12

5 Installation 13

6 Technical specifications 13

  6.1 Electrical specification – limiting values 13

  6.2 Electrical specification – static characteristic 13

7 User application 14

  7.1 PoStep60 main application GUI 14

    7.1.1 PoStep driver status 15

  7.2 Basic user control of the PoStep driver 16

  7.3 Advanced settings 18

  7.4 Step control 19

  7.5 Auto control 19

    7.5.1 Position control mode 19

    7.5.2 BINx Up/down mode 21

    7.5.3 Speed / Auto run mode 21

  7.6 DC motor control 22

  7.7 Log charts 23

  7.8 Firmware update 24

  7.9 Major changes in application version 0.61 24

    7.9.1 Major changes from 0.57 to 0.58: 24

    7.9.2 Major changes from 0.53 to 0.57: 24

8 PoStep60 I2C protocol specification 25

- 8.1 I2C address 25
- 8.2 PoStep60 I2C protocol 25
  - 8.2.1 Master write 25
  - 8.2.2 Master read 26
- 8.3 PoStep60 I2C commands 26
  - 8.3.1 0x01 - Loopback read 26
  - 8.3.2 0x03 – Set Run/Sleep mode 26
  - 8.3.3 0x0A – Read HW/FW info 27
  - 8.3.4 0x10 – Read voltage 27
  - 8.3.5 0x11 – Read temperature 27
  - 8.3.6 0x12 – Read pin statuses 27
  - 8.3.7 0x13 – Read driver status 27
  - 8.3.8 0x14 – Read driver mode 28
  - 8.3.9 0x20 – Read full scale current 28
  - 8.3.10 0x21 – Read idle current 28
  - 8.3.11 0x22 – Read overheat current 28
  - 8.3.12 0x23 – Read step mode 28
  - 8.3.13 0x24 – Read temperature limit 29
  - 8.3.14 0x25 – Read faults 29
  - 8.3.15 0x30 – Set full scale current 29
  - 8.3.16 0x31 – Set idle current 29
  - 8.3.17 0x32 – Set overheat current 30
  - 8.3.18 0x33 – Set step mode 30
  - 8.3.19 0x34 – Set temperature limit 30
  - 8.3.20 0x35 – Reset faults 30
  - 8.3.21 0x3F – Write settings to EEPROM 30
  - 8.3.22 0x40 – Read position 30
  - 8.3.23 0x41 – Read maximal speed 30
  - 8.3.24 0x42 – Read acceleration 30
  - 8.3.25 0x43 – Read deceleration 31
  - 8.3.26 0x50 – Set position 31
  - 8.3.27 0x51 – Set maximal speed 31
  - 8.3.28 0x52 – Set acceleration 31
  - 8.3.29 0x53 – Set deceleration 31
  - 8.3.30 0x5E – Set zero 32
  - 8.3.31 0x5F – Stop 32
- 9 Errata information 33
- 10 User manual changes 33
- 11 Grant of license 34
  - 11.1.1 Access 34
  - 11.1.2 Usage 34
  - 11.1.3 Copyright 34

11.1.4 Liability 34
11.1.5 Fitness for purpose 34
11.1.6 Mission Critical applications 34
11.1.7 Viruses 34
11.1.8 Support 34
11.1.9 Upgrades 34
11.1.10 Trademarks 34

1 Introduction

This manual contains information required to configure over an USB and run the PoStep60 driver. Please read the manual carefully to avoid damage to the driver. This section covers the instrument general description, instrument specifications and characteristics.

1.1 General symbols for this instructions

A few symbols are used throughout this manual that you should be aware of in order to complete certain tasks safely and completely. These symbols have different degrees of importance as described below:

NOTE!

Tells you other important information.

CAUTION!

Tells you something that could be considered to reduce a risk of failure or malfunction.

WARNING!

Tells you something that could cause damage to a property.

2 Description

The PoStep60 driver incorporate advanced stepper motor controller and external N-channel MOSFETs to drive a bipolar stepper motor or two brushed DC motors. A micro-stepping indexer is integrated, which is capable of step modes from full step to 1/256-step. An ultra-smooth motion profile can be achieved using adaptive blanking time and various current decay modes, including an auto-mixed decay mode. A simple step/direction or PWM interface allows easy interfacing to controller circuits. An I2C serial interface can further be used to control all the driver functions including position control on board. All running parameters (output current (torque), micro stepping, step mode, decay mode...) can be set over USB and stored on board for standalone operation. Internal shutdown functions are provided for over current protection, short circuit protection, under voltage lockout and over temperature. Fault conditions are indicated via a FAULT LED, and each fault condition can be read in configuration software.

3 PoStep60 features

- up to 6,0 Amps Phase Current
- advanced settings available through USB connection
- compatible with 4, 6, and 8 wire stepper motors of any voltage
- +10 VDC to +50 VDC Power Supply
- up to 256 Micro-steps per Step
- various modes of decay mode for smoother moving of motors
- 3,3V and 5V logic compatible inputs
- 250 kHz Max Step Rate
- Advanced “Auto-run” modes
- 0 °C To 85 °C Operating Temperature
- LED Power and Status Indicator
- Small Size: 54 mm X 75 mm

4 Connection diagram

4.1 Board use requirements

To properly operate the PoStep60 driver using external controller following connections need to be setup: step, direction and enable inputs; 10-50 VDC power supply, connected bipolar stepper motor before applying power.

CAUTION!

Please make sure bipolar stepper motor is connected before applying power.

4.2 Connection and setting

1. Mount PoStep60 driver firmly to a stable surface. If PoStep60 is mounted to a conductive (metal) surface, please make sure the driver is properly isolated.
2. Connect USB cable (1) to your PC and run PoStep user application
3. Please setup running parameters (output current (torque), micro stepping, step mode, decay mode...) using PoStep user application.
4. Connect step, direction, enable and GND from PoStep60 (2) to controller or BOB
5. Connect steeper motor (5) using one of wiring options bellow(
6. Connect power supply (4)

WARNING!

Please make sure power input polarity is correct. Double check before powering up the driver. Reverse power input polarity results in permanent device failure.

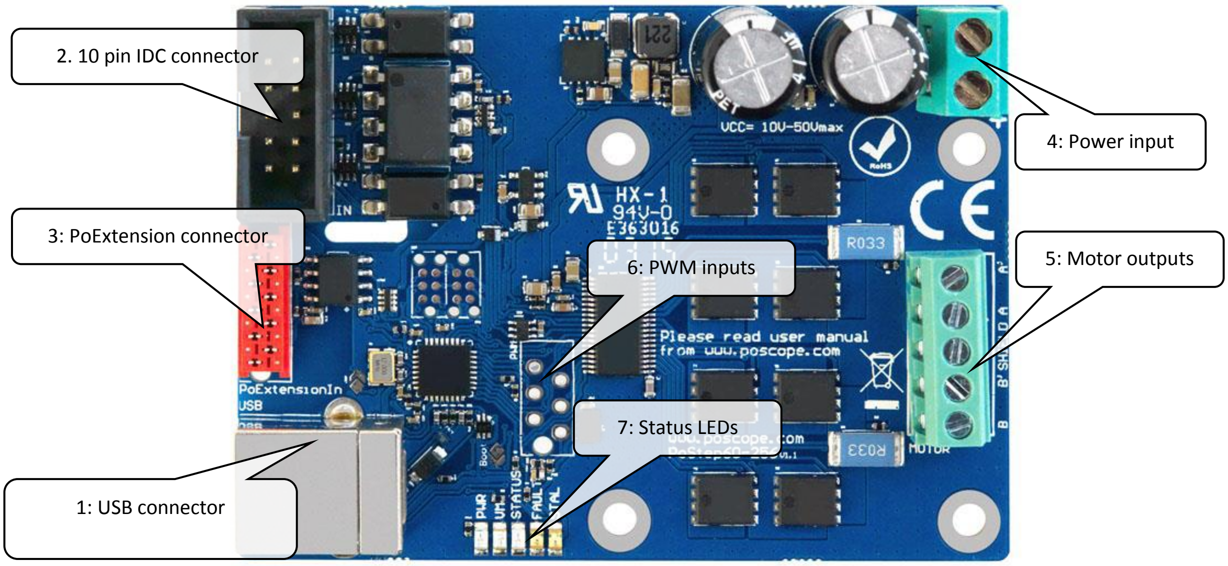
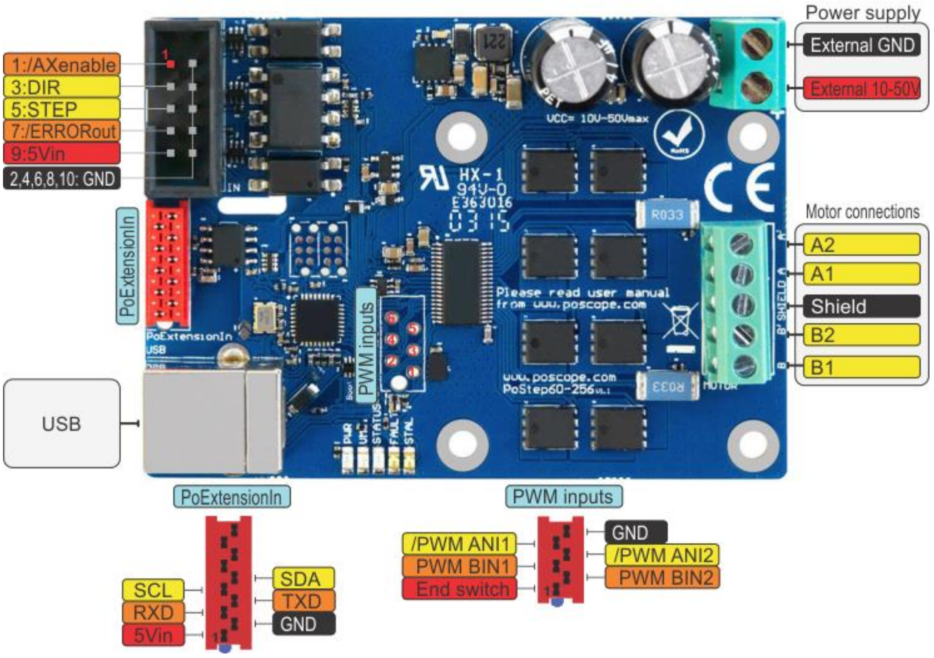

2. 10 pin IDC connector
4: Power input
3: PoExtension connector
6: PWM inputs
5: Motor outputs
7: Status LEDs
1: USB connector
VCC= 10V-50Vmax
Please read user manual from www.poscope.com
PoExtensionIn
USB
MOTOR
PWR
STATUS
FAULT

Figure 1: PoStep connections

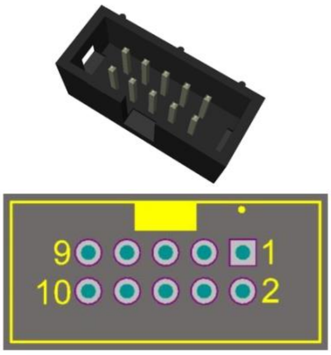
4.3 PoStep pinout


Power supply
External GND
External 10-50V
1:/AXenable
3:DIR
5:STEP
7:/ERROROut
9:5Vin
2,4,6,8,10: GND
PoExtensionIn
Motor connections
A2
A1
Shield
B2
B1
USB
PoExtensionIn
PWM inputs
SCL
RXD
5Vin
SDA
TXD
GND
/PWM ANI1
PWM BIN1
End switch
GND
/PWM ANI2
PWM BIN2

4.4 Stepper motor connection

4.4.1 10 pin IDC connector pinout

Pin	Function
1	Enable (inverted, 0=enable, 1=disable)
3	Direction
5	Step
7	Fault feedback (OC, 1=driver OK, 0=fault detected)*
9	+5V external supply voltage
2, 4, 6, 8, 10	GND



*Fault feedback signal is an open-colector signal. External pull-up resistor is needed to function properly (please see Figure 2). For input and open-collector output limiting values please refer to Section 6 - Technical specifications.

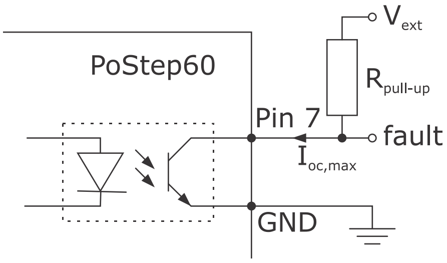

V_{ext}
PoStep60
R pull-up
Pin 7
fault
I oc,max
GND

Figure 2: Fault feedback pin connection diagram

NOTE!

Please note some of the drivers are showing failed Under Voltage Lockout (UVLO) fault status when disabled and supply voltage within required limits. The fault is removed after driver is enabled and supply voltage within limits.

4.4.2 Stepper motor connections

CAUTION!

To avoid malfunctions please make sure the phase winds are connected correctly. Resistance between leads of different phases is usually > 100kΩ. Resistance between leads of the same phase is usually < 100Ω.

NOTE!

Simple method of finding correct pairs of each winding wires is as follows (before proceeding none of the motor wires should be connected): take two random wires and short circuit them. Try rotating the stepper axis. If the pair is matching the resistance while trying to rotate the motor is much higher comparing to when none of the wires is short circuited.

NOTE!

On the motor connector on the PoStep board there is a shield pin available (in the middle). Please use proper shielding techniques for best performance.

The drive will work with 4-wire, 6-wire or 8-wire stepper motors.

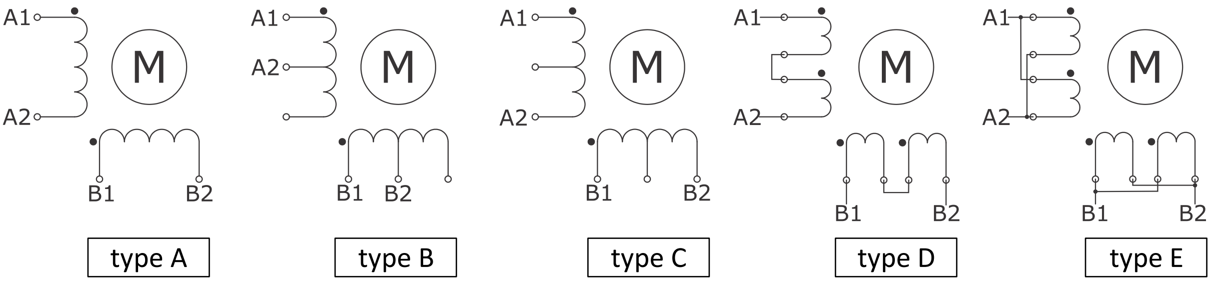


Figure 3: Stepper motor wiring options

4.4.2.1 4-wire motors (type A)

are truly bipolar, and can only be run as such.

4.4.2.2 6-wire motors

can be wired two ways to work with the bipolar drive.

The first is half-winding (type B): In this method, one end wire, and the center-tap wire of the phase is used. The other end is insulated and left unused. This method uses unipolar nameplate current specifications, and will produce nameplate torque.

The second is series winding (type C): In this configuration, the center-tap is insulated, and unused. This method uses all of the wiring per phase, but has double the number of wire turns as halfwinding or unipolar mode. Because of this, the amperage requirement becomes half the nameplate rating. Because the wire in the coil can handle more current than 'half', motor manufacturers will often "boost" the torque rating by specifying currents up to 71% of unipolar rated current while running in series mode. This is fine for FULL step motor drives, but not necessarily so good for microstepping drives. Using this much can smear microstepping smoothness and accuracy. Any extra torque achieved by this method will generally be lost to machine vibrations due to loss of microstepping smoothness. The best performance will be somewhere between the 50% and 71% current rating.

The advantage of using series winding is that lower power drives may be used. For example a unipolar motor rated for 4.0A/phase running in series requires only 2.0A/phase to achieve the same torque. The disadvantage of this method is that it raises motor inductance, which in turn, slows motor coil charging time. Since proper torque is reached only when the coil has charged to the required level, the longer it takes to charge, the longer until full torque is achieved. This leads to slower full torque stepping rates. Conversely, a half-winding configuration requires full nameplate rated current, but if the drive is capable of this, the advantage is that rated torque can be achieved twice as fast as series winding (using the same voltage, when comparing half-winding and series).

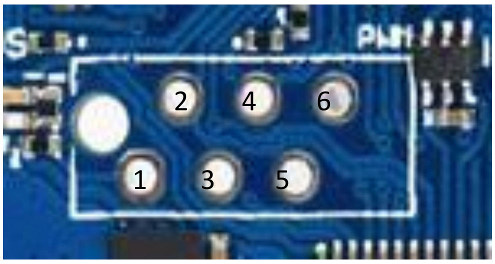
4.4.2.3 8-wire motors

can be run in paralell (type E) or serial (type D) mode. Parallel mode needs higher current, has lower inductance and better torque, Serial mode needs lower current and has lower torque. Please read also 6-wire motors.

4.5 DC motor connection

4.5.1 6 pin PWM connector pinout

Pin	Function
1	End Switch
2	PWM BIN2
3	PWM BIN1
4	PWM AIN2 (inverted)
5	PWM AIN1 (inverted)
6	GND



All PWM connector input pins has on board pull-up resistors. The pins can be either open-collector toward GND or push-pull driven. For input limiting values please refer to Section 6 - Technical specifications.

NOTE!

PWM AIN1 and PWM AIN2 inputs are inverted.

4.5.2 DC motor connection

The PoStep driver can be configured to enable direct control the state of the output drivers. This allows for driving up to two brushed DC motors. First DC motor is connected to terminals A1 and A2 and second DC motor to terminals B1 and B2. Speed and direction of DC motor rotation is controlled by PWM AINx and PWM BINx for first and second motor respectively. Driver allows connection of only one DC motor on A or B terminals as well as two DC motors.

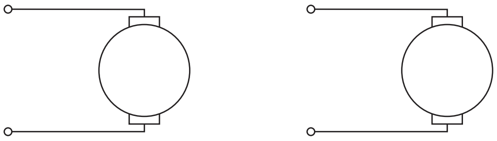


Figure 4: Connection of up to 2 DC motors

CAUTION!

Please make sure driver is configured to support DC motor drive. If not the DC motor can randomly rotate in speed and direction due to active micro stepping indexer.

NOTE!

Full-scale current value still applys when in DC drive mode. Idle current is not applicable and is not being used.

4.6 PoStep LEDs

There are five LEDs on the PoStep board showing status.

LED	Status
PWR	LED is on when PoStep is connected to a PC over USB or power supply is connected
VM	LED is on when power supply is connected
STATUS	Indicates various driver states
FAULT	LED is on when driver fault occurred. To see the fault cause the PoStep has to be connected to a PC and application running
STALL	LED is on when motor stall is detected in hardware.

4.6.1 Status LED

Status LED is used to indicate various PoStep driver states.

Pattern	Status
……….	LED is fast blinking when driver is in firmware update mode.
-.. -..	One long two short blinks indicates default settings are being loaded. Please use PoStep user application for proper configuration setup and store.
- -	Slow blinks indicate user configuration has been loaded

4.7 Requirements

- One available USB 1.1, USB 2.0, or USB 3.0 port
- USB HID device driver enabled operating system (Windows 98 SE/ME/2000/XP/Vista/7,8,10)
- Included software requires Windows 2000/XP/Vista/7,8,…
- Microsoft Visual C++ 2010 Redistributable Package (x86 or x64) or Microsoft Visual Studio 2010 needs to be installed on the system prior running PoStep60 application.

5 Installation

PoStep60 is USB HID compliant device and as such require no additional drivers for operation.

To operate the device user software installation is necessary on a target system.

NOTE! - The program can't start because MSVCR100.dll is missing from your computer.

If application fails to start showing above message most likely "Microsoft Visual C++ 2010 Redistributable Package" has to be downloaded from MS pages and installed prior running PoStep60 application.

NOTE! - The application was unable to start correctly (0xc000007b). Click OK to close the application.

If application fails to start showing 0xc000007b error please install **"Microsoft Visual C++ 2010 Redistributable Package (x86)"** no matter you have 64-bit MS Windows operating system running.

6 Technical specifications

6.1 Electrical specification – limiting values

Symbol	Parameter	Min	Max	Unit
V_M	power supply	-0.3	50	V
V_I	input voltage on 10 pin IDC input pins	-0.3	5.8	V
V_{ext}	supply voltage on pull-up resistor for fault signal	-0.3	50	V
$I_{oc,max}$	Fault signal maximum collector current		50	mA
$V_{DI,MAX}$	maximum voltage applied to PWM input pins		5.5	V
V_{iso}	isolation voltage (AC for 1min, R.H. 40-60%)		3000	V_{rms}
V_{esd}	electrostatic discharge	-4000	4000	V
$I_{drv,max}$	maximum driver output full scale current		6.0	A

6.2 Electrical specification – static characteristic

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{I,\ HIGH}$	applied voltage for HIGH state on 10 pin IDC input pins		3.0		5.8	V
$V_{I,\ LOW}$	applied voltage for LOW state on 10 pin IDC input pins		-0.3		0.4	V
$V_{DI,HIGH}$	applied voltage for HIGH state on PWM pins		1.6			V
$V_{DI,LOW}$	applied voltage for LOW state on PWM pins		-0.3		0.2	V

7 User application

User application enables various interactions with PoStep60 driver. The application enables setting of all the vital driver parameters including driver current (active, idle, and overheated values), micro-stepping value, driver name for later recognition, and advanced control setting values. Moreover, the application includes basic driving capabilities for stepper motors as well as two DC motors. For the stepper motor there is simple step/direction mode and more advanced onboard speed profile feature for out-of-the box drive of stepper motor. For DC motors drive there are two PWM channels with adjustable PWM values and directions. The application also monitors input statuses and driver status itself (temperature, mode, fault,…)

7.1 PoStep60 main application GUI

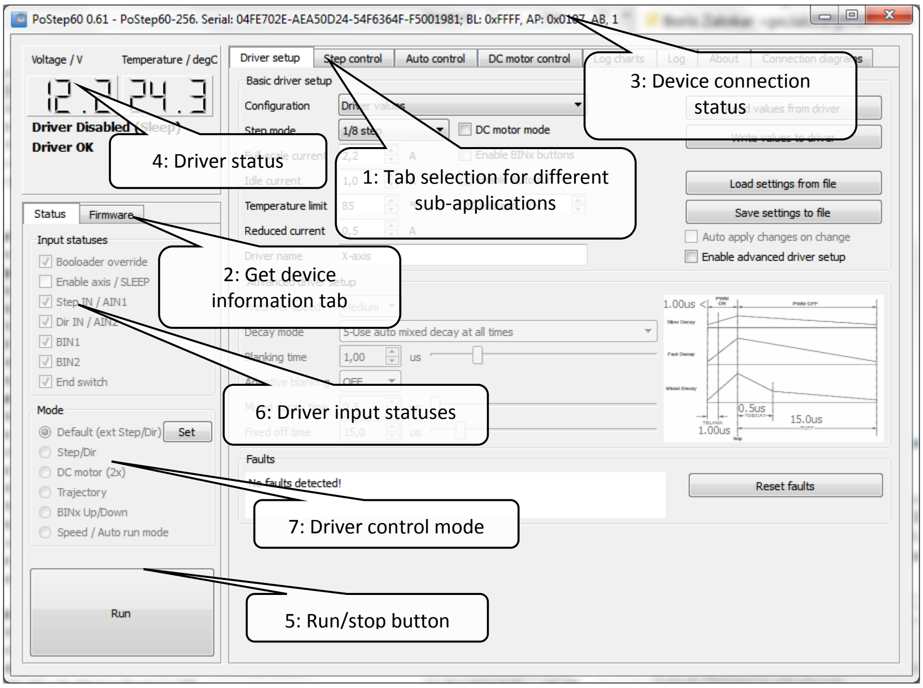

PoStep60 0.61 - PoStep60-256. Serial: 04FE702E-AEA50D24-54F6364F-F5001981; BL: 0xFFFF, AP: 0x0107, AB, 1
Voltage / V
Temperature / degC
12.2
24.3
Driver Disabled (sleep)
Driver OK
Status
Firmware
Input statuses
Booloader override
Enable axis / SLEEP
Step IN / AIN1
Dir IN / AIN2
BIN1
BIN2
End switch
Mode
Default (ext Step/Dir)
Set
Step/Dir
DC motor (2x)
Trajectory
BINx Up/Down
Speed / Auto run mode
Run
Driver setup
Step control
Auto control
DC motor control
Log charts
Log
About
Connection diagrams
Basic driver setup
Configuration
Driver values
Step mode
1/8 step
DC motor mode
Temperature limit
Reduced current
Decay mode
5-Use auto mixed decay at all times
Blanking time
1,00
us
Load settings from file
Save settings to file
Auto apply changes on change
Enable advanced driver setup
Faults
No faults detected!
Reset faults
1: Tab selection for different sub-applications
2: Get device information tab
3: Device connection status
4: Driver status
5: Run/stop button
6: Driver input statuses
7: Driver control mode

Figure 5: PoStep60 main application GUI

Figure 5 notes:

1. Tab selection for different sub-applications – please use tabs to navigate between sub-applications
2. Click to get device information such as serial number, mode status, …
3. The application title bar displays connection status of PoStep device. If a device is connected a serial number is displayed and if not “not connected” message is displayed. Please note only one device at same time can be connected.
4. Driver status – The section displays real-time data of a driver connected if the driver is in application mode.
5. Stop/Run button disables and enables PoStep driver. Useful feature when in various control modes.

NOTE!

Please note enable signal is shared between external “Enable” input. Disconnected external signal when using application driver control features.

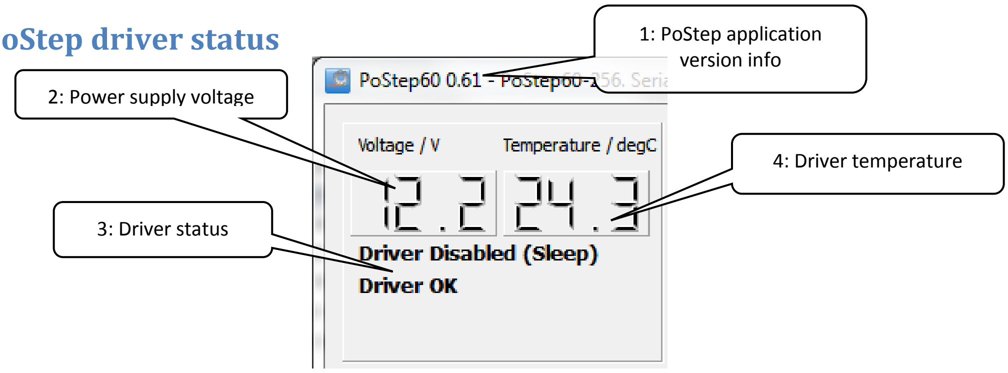
6. Driver input statuses displays current driver input statuses

7. Driver control mode displays current driver control mode

NOTE!

Driver real-time values are only available when driver is in application mode. In bootloader mode all driver features are disabled.

7.1.1 PoStep driver status



PoStep60 0.61 - PoStep60-256. Seri

Voltage / V: 12.2

Temperature / degC: 24.3

Driver Disabled (Sleep)

Driver OK

Figure 6: PoStep driver status section

Figure 6 notes:

1. PoStep application version info – please check for latest update of PoStep application.
2. Power supply voltage – displays main power supply voltage.
3. Driver status – displays driver statuses
4. Driver temperature – displays main power supply voltage.

CAUTION!

The displayed temperature value may differ from the actual driver temperature due to indirect contact. Parts of the diver can have higher temperature than one displayed and as such might cause you burns if touched.

7.1.1.1 *Driver statuses*

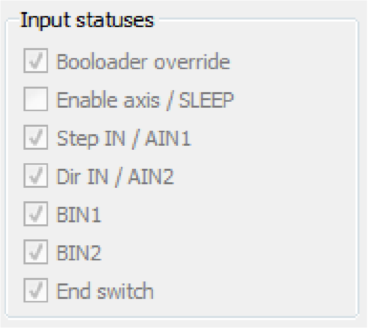
- Driver active – there was a signal change in step/direction inputs in the last 10 seconds. Full-scale active current value is set
- Idle – there was no change on step/direction inputs for more than 10 seconds. Idle current value is set
- Overheated – when driver exceeds limit temperature values drivers goes into overheated mode. Reduced current value is set
- Fault detected – displays that at least one of possible driver faults were detected. Please check fault type in Advanced settings.

7.1.1.2 *Input statuses*

The application shows real-time data of all the driver IO pins - Figure 7 (tick represents high level):

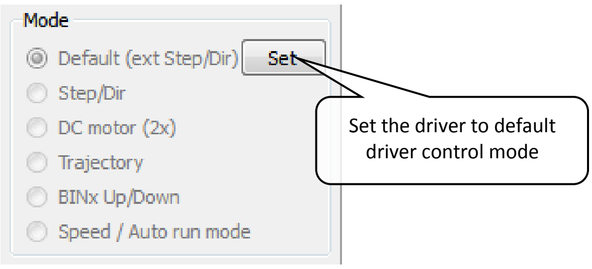
- Bootloader override – the pin should always be high
- Enable axis / SLEEP – high level enables diver operation. Low level puts the driver to sleep mode
- Step IN / AIN1 – this is Step input signal in stepper motor operation or Motor 1 PWM control AIN1 when in DC motor control mode
- Dir IN / AIN2 – this is Direction input signal in stepper motor operation or Motor 1 PWM control AIN2 when in DC motor control mode
- BIN1 – Motor 2 PWM control BIN1 when in DC motor control mode

- BIN2 – Motor 2 PWM control BIN2 when in DC motor control mode
- End switch – end switch input status


Input statuses

- ☑ Booloader override
- ☐ Enable axis / SLEEP
- ☑ Step IN / AIN1
- ☑ Dir IN / AIN2
- ☑ BIN1
- ☑ BIN2
- ☑ End switch

Figure 7: Input statuses


Mode

- ● Default (ext Step/Dir)
- ○ Step/Dir
- ○ DC motor (2x)
- ○ Trajectory
- ○ BINx Up/Down
- ○ Speed / Auto run mode

Set the driver to default driver control mode

Figure 8: Driver control modes

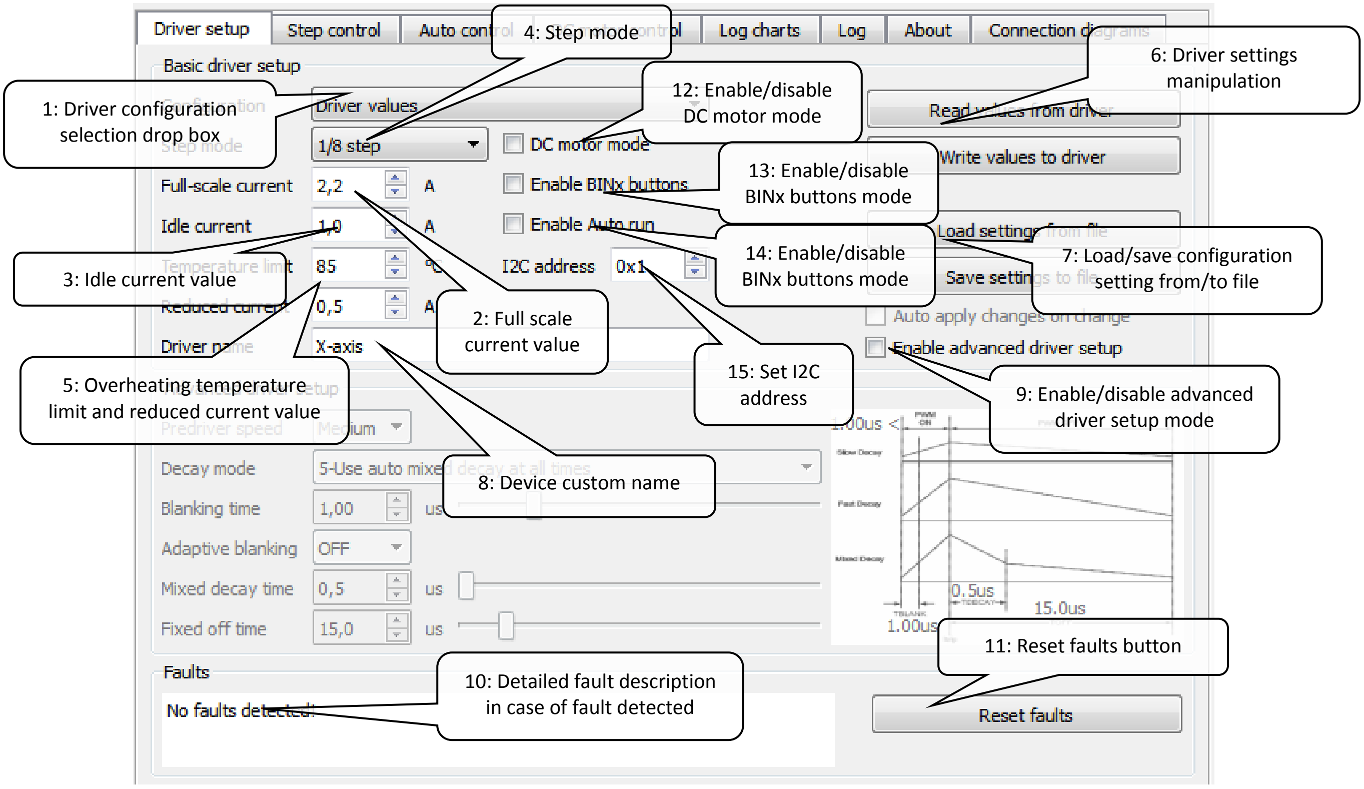
7.1.1.3 Driver control modes

Driver control mode section displays current mode of the driver - Figure 8.

- Default (ext Step/Dir) – this is the driver default state which enables Step/direction control using external controller. Enable axis, Step IN, and Dir IN inputs shall all be connected for proper operation.
- Step/Dir – in this mode driver generates the driving signals. The mode is mainly used for driver settings adjustments – fine tuning. Please see subsection 7.4 Step control.
- DC motor – two DC motors can be controlled using direct PWM signals. Please see subsection 7.6 DC motor control.
- Trajectory – in this mode the driver take over controller parts of motion planning. Required position and speed profile parameters can be set. Please see subsection 7.5 Auto control.
- BINx Up/Down – upgrade to “Trajectory” mode where movement is triggered by external buttons. Please see subsection7.5.2 BINx Up/down mode.
- Speed / Auto run mode – the mode enables automatic starting of driver at power up or start/stop using external switch. Please see subsection 7.5.3 Speed / Auto run mode.

7.2 Basic user control of the PoStep driver

Basic user control section enables user selection of predefined driver configuration sets, setting currents: full scale, idle, overheated reduction, and temperature limit. Moreover, the basic manipulation with driver data is enabled: readout, change, and store. Furthermore, each of the drivers can be set an unique name for easier identification.


Driver setup
Step control
Auto control
Log charts
Log
About
Connection diagrams
Basic driver setup
Read values from driver
Write values to driver
Load settings from file
Save settings to file
Advanced driver setup
Reset faults
1.00us
0.5us
15.0us
Slow Decay
Fast Decay
Mixed Decay
PWM ON
TBLANK
TDECAY
1: Driver configuration selection drop box
2: Full scale current value
3: Idle current value
4: Step mode
5: Overheating temperature limit and reduced current value
6: Driver settings manipulation
7: Load/save configuration setting from/to file
8: Device custom name
9: Enable/disable advanced driver setup mode
10: Detailed fault description in case of fault detected
11: Reset faults button
12: Enable/disable DC motor mode
13: Enable/disable BINx buttons mode
14: Enable/disable BINx buttons mode
15: Set I2C address

Configuration: Driver values
Step mode: 1/8 step
Full-scale current: 2,2 A
Idle current: 1,0 A
Temperature limit: 85 °C
Reduced current: 0,5 A
Driver name: X-axis
☐ DC motor mode
☐ Enable BINx buttons
☐ Enable Auto run
I2C address: 0x1
☐ Auto apply changes on change
☐ Enable advanced driver setup
Predriver speed: Medium
Decay mode: 5-Use auto mixed decay at all times
Blanking time: 1,00 us
Adaptive blanking: OFF
Mixed decay time: 0,5 us
Fixed off time: 15,0 us
Faults: No faults detected!

Figure 9: Basic user controls

Figure 9 notes:

1. Driver configuration selection drop box – select one of predefined configurations. Use default configuration if you are not familiar with motor performance parameters.
2. Full scale current value – setup required motor current value.

NOTE!

Please follow stepper motor datasheet when setting driver values. Setting higher current value than nominal motor current may result in changed characteristics of PoStep motor driver (overheating, motor ringing,…) due to motor running out of specs.

3. Idle current value – setup idle current – idle current is a driver current when there is no activity on “Step/Dir” inputs for more than 10s.
4. Step mode – select a step mode between full step and 1/256-step (256 micro steps) mode.
5. Overheating temperature limit and reduced current value – setup a temperature limit at which driver reduces driver current to protect driver from possible overheating.
6. Driver settings manipulation – enables reading/writing configuration to/from PoStep60 driver
    a. Read values from driver – reads driver configuration settings currently set on the driver.
    b. Write values to driver – writes driver configuration settings and store them to a non-volatile memory on the driver. The settings are read and loaded each time driver is powered or reset.

NOTE!

Any change in configuration settings made in application is not valid to driver until confirmed by “Write values to driver”.

7. Setting files – enables loading/saving driver configuration settings from/to file
    a. Load settings from file – load driver configuration settings previously saved.
    b. Save settings to file – save currently set configuration to specified file.
8. Device custom name – displays or change driver custom name
    a. Read custom name – reads driver name
    b. Write custom name – writes and stores driver name
9. Enable/disable advanced driver setup mode – by checking advanced setup mode is enabled allowing setting advanced parameters
10. Detailed fault description in case of fault detected – further describes detected fault and possible cause of the fault
11. Reset faults – press the button to reset driver faults
12. DC motor mode – Enable or disable DC motor mode
13. BINx Up/Down – Enable or disable BINx buttons mode
14. Auto run – Enable or disable Auto run mode
15. I2C address – Sets an address of the PoStep60 driver when using I2C communication

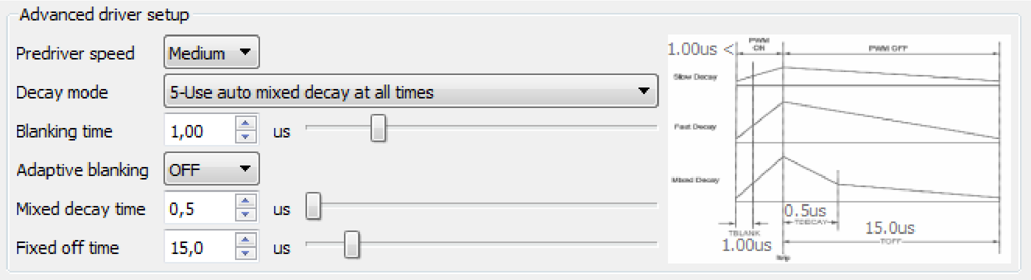
7.3 Advanced settings

WARNING!

The advanced settings shall only be used by a person with a strong knowledge in motor control. There is a chance of setting combination that may results in permanent device failure.

In stepping motors, current regulation is used to vary the current in the two windings in a sinusoidal fashion to provide smooth motion. An ultra-smooth motion profile can be achieved using advanced settings such as blanking and decay time, and various current decay modes, including an auto-mixed decay mode.

The current through the motor windings is regulated by an adjustable fixed-off-time PWM current regulation circuit. When an H-bridge is enabled, current rises through the winding at a rate dependent on the DC voltage and inductance of the winding and the magnitude of the back EMF present. Once the current hits the current chopping threshold, the bridge disables the current for a fixed period of time, which is programmable between 500 nS and 128 us. After the off time expires, the bridge is re-enabled, starting another PWM cycle.


Advanced driver setup

Predriver speed: Medium
Decay mode: 5-Use auto mixed decay at all times
Blanking time: 1,00 us
Adaptive blanking: OFF
Mixed decay time: 0,5 us
Fixed off time: 15,0 us

Figure 10: Advanced driver setup settings

NOTE!

When setting up the driver various compromises need to be considered such as:

- Higher the input voltage longer the fixed off time has to be to control the required current,
- Higher the input voltage lower the step mode,
- …

7.4 Step control

The application enables simple internal control of step/dir pins. It allows a test of basic driving capabilities of the PoStep60 driver.

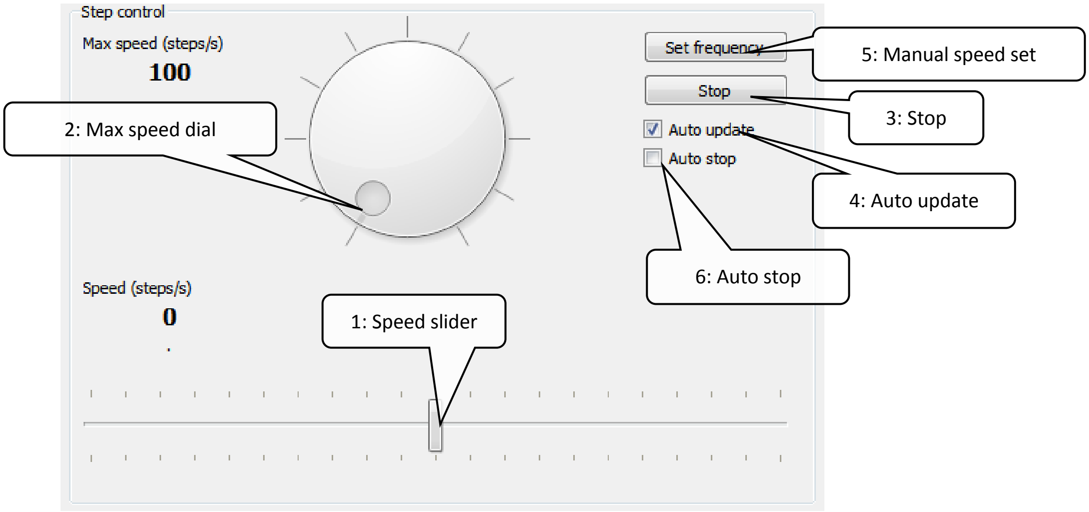


Figure 11: Step control controls

1. Speed slider – use this slider to set step speed. Please note the maximum value is limited by “Max speed” value.
2. Max speed dial – set maximum step speed to limit the step speed value set by “Speed” value.
3. Stop – stops internal step/dir generator and switch the pins control to external controller.
4. Auto update – if checked, any change in step speed value is sent to driver immediately.
5. Manual speed set – send new step speed value to driver if “Auto update” is disabled.
6. Auto stop – when checked step speed resets to zero when slider is released.

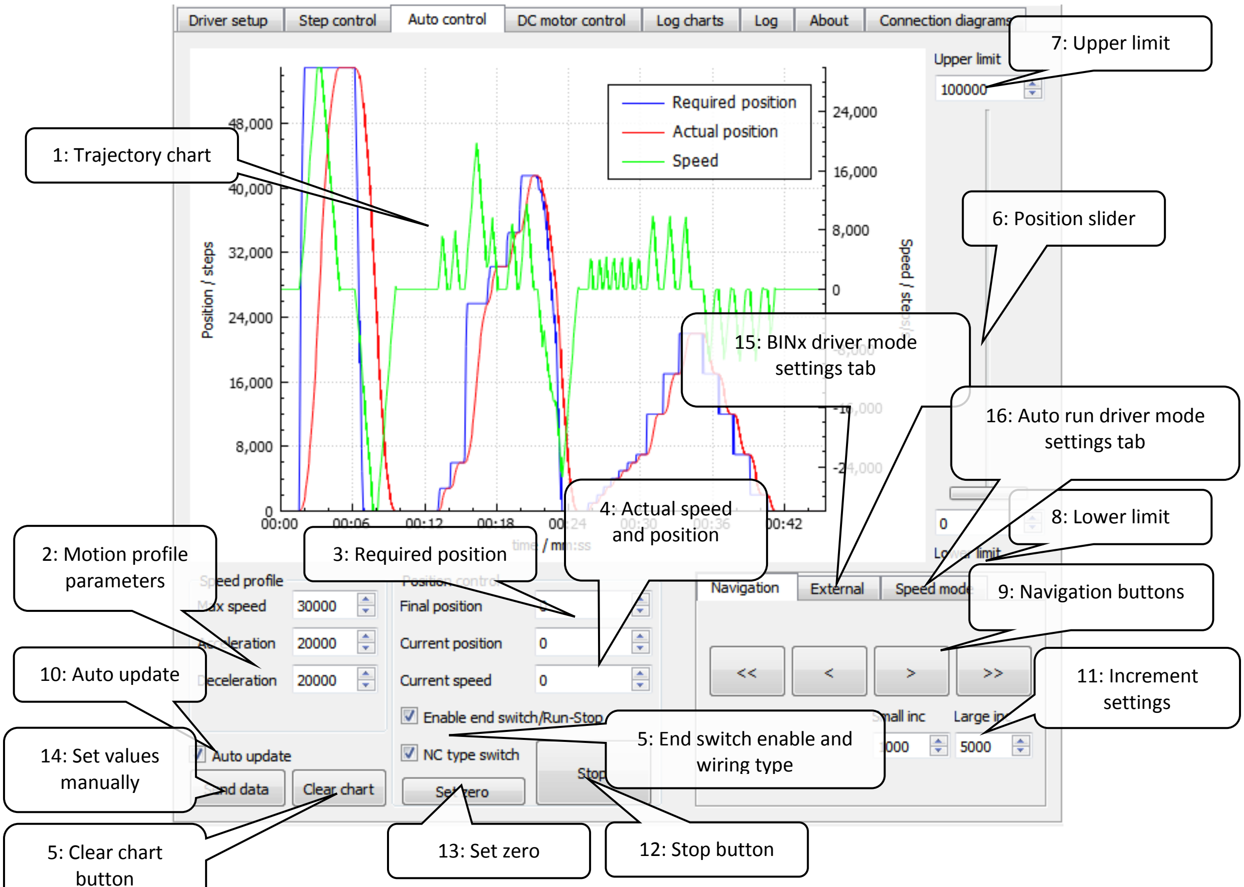
NOTE!

If driver is used in combination with external controller or BOB please make sure to stop the application step/dir control by pressing button “Stop”. This way external controller takes over control of the step/dir pins again.

7.5 Auto control

7.5.1 Position control mode

Simple position control algorithm is implemented in the driver. The driver moves to required position using standard “Trapezoidal motion profile” where acceleration, deceleration, and maximal speed is defining the profile. Required position can be set using input box or slider. End switches (NC or NO; please note – end switch reacts only on change of end switch state.) and position limits can be used to define/limit a range of movement. The provided position control allows a test of basic driving capabilities of the PoStep60 driver and is not intended to replace external controller.


Driver setup
Step control
Auto control
DC motor control
Log charts
Log
About
Connection diagrams
Required position
Actual position
Speed
48,000
40,000
32,000
24,000
16,000
8,000
0
Position / steps
24,000
16,000
8,000
0
-16,000
-24,000
Speed / steps/
00:00
00:06
00:12
00:18
00:24
00:30
00:36
00:42
time / mm:ss
Speed profile
Position control
Navigation
External
Speed mode
<<
<
>
>>
Send data
Clear chart
Set zero
Stop
1: Trajectory chart
2: Motion profile parameters
3: Required position
4: Actual speed and position
5: End switch enable and wiring type
5: Clear chart button
6: Position slider
7: Upper limit
8: Lower limit
9: Navigation buttons
10: Auto update
11: Increment settings
12: Stop button
13: Set zero
14: Set values manually
15: BINx driver mode settings tab
16: Auto run driver mode settings tab

Upper limit 100000

Lower limit 0

Max speed 30000

Acceleration 20000

Deceleration 20000

Final position

Current position 0

Current speed 0

☑ Enable end switch/Run-Stop

☑ NC type switch

☑ Auto update

Small inc ...000

Large inc 5000

Figure 12: Position control controls

1. Trajectory chart – displays required and actual position and speed charts
2. Motion profile parameters – Trapezoidal motion profile parameters: acceleration, deceleration, and maximum speed
3. Required position – set and displays (slider) required position
4. Actual speed and position – displays actual speed and position values
5. End switch – enable/disable end switch control and end switch wiring type settings

NOTE!

End switch reacts only on change of end switch state. Position change request after end switch touched can be set in either direction.

6. Position slider – required position can be set using slider
7. Upper limit – limits final position.

NOTE!

If current position is out of new position limit values set then required position will adjust accordingly and motor shall move if Auto update is enabled.

8. Lower limit – limits starting position
9. Navigation buttons – set required position by a step defined by increment settings (11)
10. Auto update – send new values to driver automatically if checked
11. Increment settings – defines Small and Large increments for navigation steps

12. Stop button – stops position movement immediately (no deceleration profile applied)

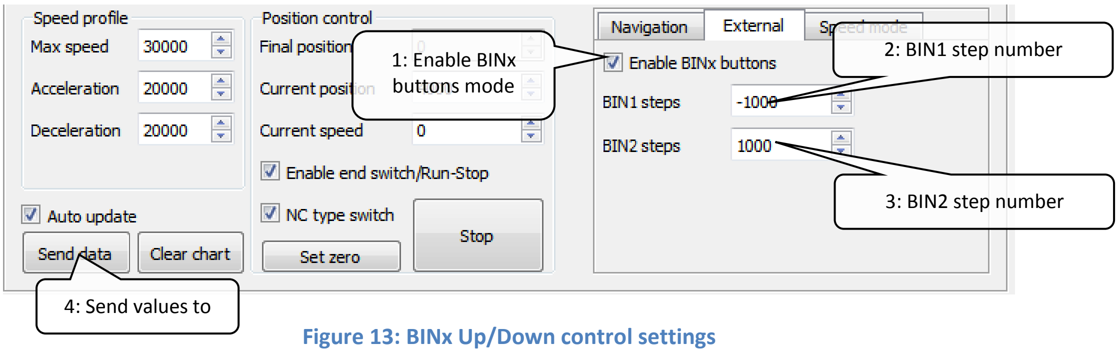
NOTE!

If driver is used in combination with external controller or BOB please make sure to stop the application Position control control by pressing button “Stop”. This way external controller takes over control of the step/dir pins again.

13. Set zero – set both the required and actual position values to zero
14. Set values manually – send new values to driver manually
15. External – External navigation tab used to configure BINx driver mode settings
16. Speed mode – Speed mode navigation tab used to configure Auto run driver mode settings

7.5.2 BINx Up/down mode

In a BINx driver control mode the driver is set to be used alone (no external controller or BOB is needed) just with two external buttons on pins BIN1 and BIN2 (please see subsection 4.5.1 - 6 pin PWM connector pinout for proper wiring). Connect NO button between pin BIN1 and GND. Connect NO button between pin BIN2 and GND. When BIN1 or BIN2 button is pressed motor moves number of steps defined in BIN1 and BIN2 steps respectively. Same motion profile is valid as in position control mode settings.



Speed profile
Max speed 30000
Acceleration 20000
Deceleration 20000
☑ Auto update

Position control
Current speed 0
☑ Enable end switch/Run-Stop
☑ NC type switch

☑ Enable BINx buttons
BIN1 steps -1000
BIN2 steps 1000

Figure 13: BINx Up/Down control settings

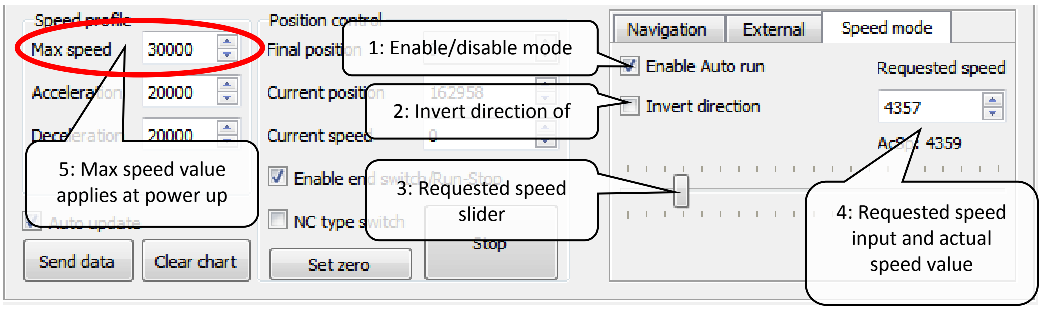
1. Enable BINx buttons – when checked driver is in BINx buttons control mode
2. BIN1 steps – Defines number of steps to move when BIN1 button pressed
3. BIN2 steps – Defines number of steps to move when BIN2 button pressed
4. Send data – send new values to driver (automatically if “Auto update” checked)

NOTE!

Make sure to confirm the change of driver mode and save mode settings by clicking “Write values to driver”.

7.5.3 Speed / Auto run mode

In Auto run mode PoStep60 driver automatically starts motion of stepper motor after power up. No external controller or BOB is needed. Optionally by enabling “Run/stop” a “Normally open - NO” or “Normally closed – NC” switch can be used to control the motion on or off. For wiring please see subsection 4.5.1 6 pin PWM connector pinout. Same motion profile settings apply as in Position control mode above. At power up “Max speed” value applies and not “Requested speed” value. Requested speed value applies only during active USB connection of PoStep60 driver.



Max speed 30000

Acceleration 20000

Deceleration 20000

Current position 162958

Current speed 0

☑ Enable Auto run

☐ Invert direction

Requested speed 4357

Act: 4359

Speed profile
Position control
Final position
Navigation
External
Speed mode
☑ Enable end switch/Run-Stop
☐ NC type switch
☑ Auto update
Send data
Clear chart
Set zero
Stop
1: Enable/disable mode
2: Invert direction of
3: Requested speed slider
4: Requested speed input and actual speed value
5: Max speed value applies at power up

Figure 14: Speed / Auto run control settings

1. Enable Auto run – when checked driver is in Auto run control mode
2. Invert direction – Inverts direction of stepper movement
3. Requested speed slider – Use the slider to set requested speed. Maximum speed value is limited by "Max speed"
4. Requested speed – Requested and actual speed value. Due to hardware limitation actual speed can differ from requested speed.
5. Max speed – Max speed value – this is the value of the speed at power up

NOTE!

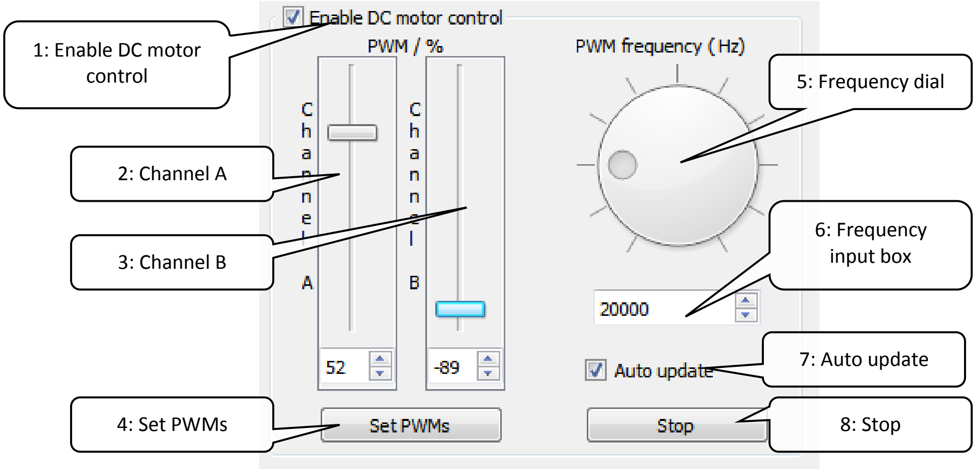
At power up "Max speed" value applies and not "Requested speed" value. Requested speed value applies only during active USB connection of PoStep60 driver.

NOTE!

Make sure to confirm the change of driver mode and save mode settings by clicking "Write values to driver".

7.6 DC motor control

PoStep60 driver can be configured to drive up to two dc motors bi-directionally. Enabling DC motor control bypass driver internal indexer control used for stepper motor micro-stepping control. All the driver parameters except idle current apply for DC motor control as well. This enable current limited DC motor drive using PWM control. The PoStep60 application enables simple PWM control where PWM and PWM frequency (period) can be set.



☑ Enable DC motor control

Channel A: 52

Channel B: -89

PWM frequency (Hz): 20000

☑ Auto update

PWM / %
Channel A
Channel B
PWM frequency (Hz)
Set PWMs
Stop
1: Enable DC motor control
2: Channel A
3: Channel B
4: Set PWMs
5: Frequency dial
6: Frequency input box
7: Auto update
8: Stop

Figure 15: DC motor control controls

1. Enable DC motor control – enables and disables DC motor control.

NOTE!

Enabling or disabling DC motor control puts driver into stand-by to prevent sudden movement of motor or accidental overheating of stepped motor winding.

2. Channel A – Slider and input box for PWM value for DC motor connected to output terminals A
3. Channel B – Slider and input box for PWM value for DC motor connected to output terminals B
4. Set PWMs – send new values to driver manually
5. Frequency dial – set PWM frequency (period)
6. Frequency input box – set PWM frequency (period)
7. Auto update – send new values to driver automatically if checked
8. Stop – sets PWMs to zero and put open collector inputs to high state enabling external PWM controller PWM control

NOTE!

If driver is used in combination with external controller or BOB please make sure to zero PWM values by pressing button “Stop”. This way external controller takes over control of the PWM inputs again.

7.7 Log charts

PoStep60 application enables simple log of major driver parameter through time. By the chart the driver temperature can be monitored for possible overheating. This way driver setting can be adjusted or heat dissipation elements checked for its performance.

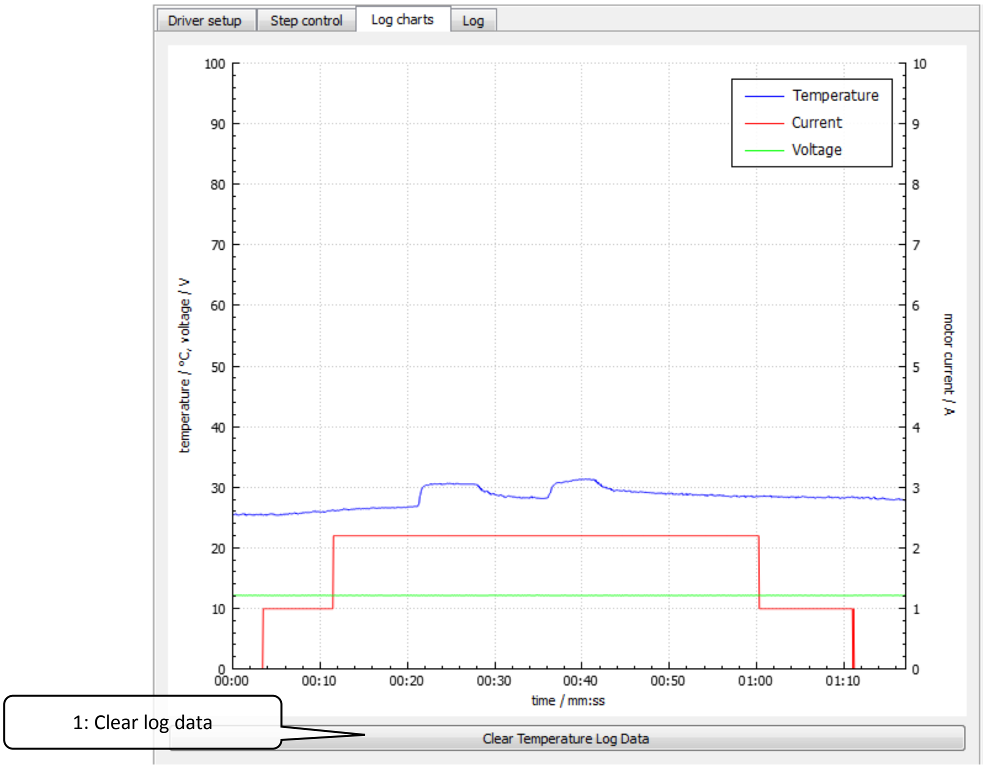

Driver setup
Step control
Log charts
Log
100
90
80
70
60
50
40
30
20
10
0
temperature / °C, voltage / V
10
9
8
7
6
5
4
3
2
1
0
motor current / A
Temperature
Current
Voltage
00:00
00:10
00:20
00:30
00:40
00:50
01:00
01:10
time / mm:ss
1: Clear log data
Clear Temperature Log Data

Figure 16: PoStep60 driver parameters log chart

1. Clear log data – Clears log charts data.

7.8 Firmware update

Application automatically checks for PoStep60 driver firmware version and advises when update needed. If by any chance automatic firmware update fails a manual update is possible. Please select “Firmware” tab and manually reset PoStep60driver to enter boot mode ie. when “Full update” button is enabled. Trigger update by pressing “Full update” and wait for finished update. After update the driver automatically resets and enters normal mode.

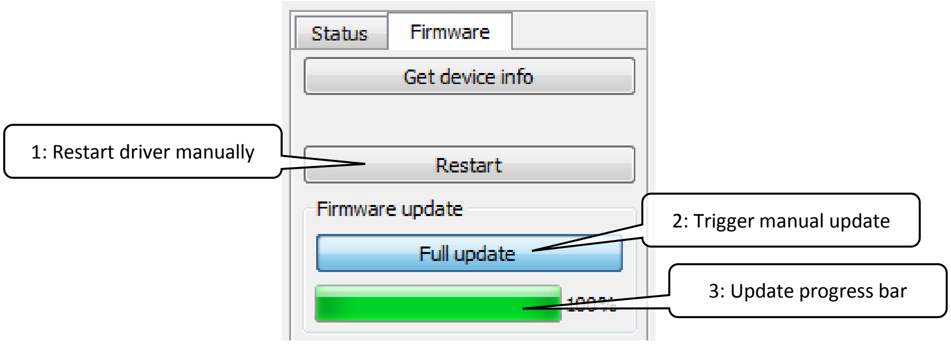

Status
Firmware
Get device info
1: Restart driver manually
Restart
Firmware update
2: Trigger manual update
Full update
3: Update progress bar
100%

Figure 17: Manual firmware update

1. Restart driver manually – trigger driver restart and enables the driver to enter boot mode.
2. Full update – trigger start of full firmware update
3. Update progress bar – shows status of update progress.

7.9 Major changes in application version 0.61

- Added Auto run mode
- Position control navigation tab changed to Auto control
- Auto control tab restructuring

7.9.1 Major changes from 0.57 to 0.58:

- Voltage Too Low when disabled bug fix
- BINx mode - Button Up/Down control added
- I2C communication added

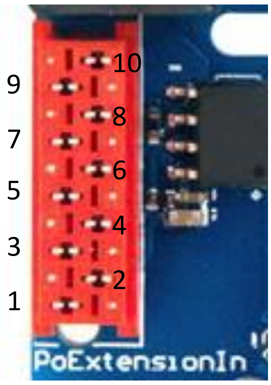
7.9.2 Major changes from 0.53 to 0.57:

- Predriver Fault bug fix
- Position control added
- DC motor mode setting added
- DC motor control added
- Improved Voltage and Temperature filtering
- Added installation notes (MS VS 2010 redistributable package)

8 PoStep60 I2C protocol specification

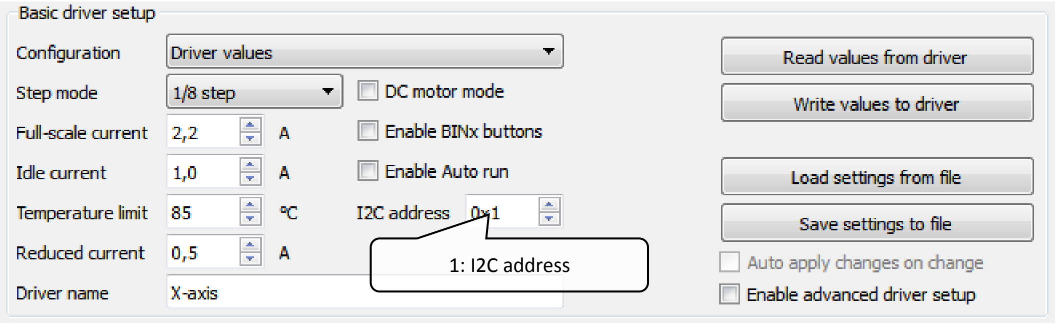
PoStep60 has implemented I2C communication connection which enables setup and read most of the driver configuration parameters and statuses. I2C pins are located on PoExtension connector (please see Figure 1).

Pin	Function
1	5V
2	GND
5	SCL – I2C clock pin
6	SDA – I2C data pin
3, 4, 7, 8, 9, 10	NC – not connected



8.1 I2C address

PoStep60 default I2C address is set to 1 (0x01), but can be configured within PoStep60 application to any value between 1 and 127 (please see Figure 18). This way multiple PoStep60 drivers can be connected to same I2C bus and be monitored and configured externally in real-time. To confirm the I2C address “Write values to driver” has to be clicked.



Basic driver setup

Configuration: Driver values
Step mode: 1/8 step
Full-scale current: 2,2 A
Idle current: 1,0 A
Temperature limit: 85 °C
Reduced current: 0,5 A
Driver name: X-axis
☐ DC motor mode
☐ Enable BINx buttons
☐ Enable Auto run
I2C address: 0x1
☐ Auto apply changes on change
☐ Enable advanced driver setup

Figure 18: PoStep60 I2C address setup

8.2 PoStep60 I2C protocol

I2C communication is driven by a single I2C master which communicates to multiple slaves (PoStep60 drivers). Every packet consist of I2C address, Read/Write command bit, command byte (in case of write) and multiple data bytes. In write operation master specifies command for either write or read operation. In a Master read operation, based on previously written command, data bytes are sent back from PoStep60 driver (slave). Always applies the last command send by the master.

NOTE!

In a Master read operation always applies the last command send by the master in a Master write operation. This way same parameter can be read continuously.

8.2.1 Master write

In Master write operation master always provide a command as a first data byte. Following bytes represent data to be interpreted by PoStep driver.

STOP	I2C ADDRESS	R/W=0	A	COMMAND	A	BYTE0	A	...	BYTEn	A	END

☐ From Master to PoStep driver
☐ From PoStep driver to Master

8.2.2 Master read

Before read operation we shall always perform a Master write operation providing command for read operation. After command is provided the Master read can be performed. In the Master read operation PoStep response with multiple bytes of data. The data has to be properly interpreted by the master.

STOP	I2C ADDRESS	R/W=1	A	BYTE0	A	BYTE1	A	…	BYTEn	A	END

▨ From Master to PoStep driver
☐ From PoStep driver to Master

8.3 PoStep60 I2C commands

8.3.1 0x01 - Loopback read

In a Master write operation master sends 0x01 command followed by 4 bytes. In a Master read operation the 4 bytes from previous write operation are send back from the PoStep driver. The main purpose of the command is to test a response of PoStep60 driver.

8.3.2 0x03 – Set Run/Sleep mode

To enable or disable (Run/Sleep) the driver Master shall write 0x03 command followed by one byte of data – BYTE0. BYTE0 relates to Run/Sleep mode as follows:

Mode	BYTE0
Run	0xDA
Sleep	0x0F

NOTE!

Setting run over I2C overrides external enable signal. Similarly, external enable overrides internal sleep signal.

8.3.3 0x04 – Set I2C address

Master writes 0x04 command followed by two bytes of data. BYTE0 shall be equal to current I2C address (basic check before applying new address) and BYTE1 shall represent new I2C address.

8.3.4 0x05 – Set driver mode

Master writes 0x05 command followed by one BYTE of data.

BYTE	Mode
1	Default mode – external control
6	Auto run mode

CAUTION!

Please make sure to request Stop (8.3.38 0x5F – Stop) and Zero (8.3.37 0x5E – Set zero) before changing driver mode to Auto run mode to prevent sudden motor movement.

8.3.5 0x0A – Read HW/FW info

First Master writes 0x0A command (no data bytes are needed). In a next step Master reads 5 bytes of data where meaning of byte is described in a table below.

BYTE	Info
0	0x41 – driver ID
1	HW version major release
2	HW version minor release
3	FW version major release
4	FW version minor release

8.3.6 0x10 – Read voltage

To read PoStep supply voltage first Master writes 0x10 command (no data bytes are needed). In a next step Master reads two bytes of data. Calculated voltage is:

Voltage/V = (BYTE1 * 256 + BYTE0) * 0.072 = (BYTE1 << 8 | BYTE0) * 0.072

8.3.7 0x11 – Read temperature

To read PoStep temperature first Master writes 0x11 command (no data bytes are needed). In a next step Master reads two bytes of data. Calculated temperature is:

Temperature/°C = (BYTE1 * 256 + BYTE0) * 0.125 = (BYTE1 << 8 | BYTE0) * 0.125

8.3.8 0x12 – Read pin statuses

First Master writes 0x12 command (no data bytes are needed). In a next step Master reads one byte of data. Each bit in the read byte represents one pin. 0 represents logic low level and 1 represents logic high.

Bit	Pin
0 (LSB)	Bootloader override
1	Sleep
2	Step / AIN1
3	Direction / AIN2
4	BIN1
5	BIN2
6	End switch
7 (MSB)	/

8.3.9 0x13 – Read driver status

First Master writes 0x13 command (no data bytes are needed). In a next step Master reads one byte of data. Read value represents driver status:

Status value	Status
1	Driver in sleep mode
2	Driver active
3	Driver in idle mode
4	Driver overheated
5	Driver in DC motor control mode

8.3.10 0x14 – Read driver mode

First Master writes 0x14 command (no data bytes are needed). In a next step Master reads one byte of data. Read value represents driver mode:

Mode value	Mode
1	Default mode – external control
2	Step control
3	DC motor control
4	Position control
5	BINx buttons
6	Auto run mode

NOTE!

Driver mode is read only value for I2C communication and can only be set by using USB connection and PoStep60 application!

8.3.11 0x20 – Read full scale current

To read PoStep full scale current first Master writes 0x20 command (no data bytes are needed). In a next step Master reads two bytes of data. Calculated current is:

current/A = 0.065*BYTE0 / (2^{BYTE1})

8.3.12 0x21 – Read idle current

First Master writes 0x21 command (no data bytes are needed). In a next step Master reads two bytes of data. Same formula applies as in subsection 8.3.11 0x20 – Read full scale current to calculate idle current value.

8.3.13 0x22 – Read overheat current

First Master writes 0x22 command (no data bytes are needed). In a next step Master reads two bytes of data. Same formula applies as in subsection 8.3.11 0x20 – Read full scale current to calculate overheat current.

8.3.14 0x23 – Read step mode

First Master writes 0x23 command (no data bytes are needed). In a next step Master reads one byte of data. Read value represents step mode:

Mode value = BYTE0 & 0x0F

Mode value	Mode
0	Full step
1	Half step
2	1/4 step
3	1/8 step
4	1/16 step
5	1/32 step
6	1/64 step
7	1/128 step
8	1/256 step

8.3.15 0x24 – Read temperature limit

To read temperature limit first Master writes 0x24 command (no data bytes are needed). In a next step Master reads one byte of data. Calculated temperature is:

Temperature/°C = BYTE0

8.3.16 0x25 – Read faults

First Master writes 0x25 command (no data bytes are needed). In a next step Master reads one byte of data. Each bit in the read byte represents one fault. 0 represents no faults and 1 represents fault occurred.

Bit	Error	Description
0 (LSB)	OTS	Device has entered over temperature shutdown. OTS bit will clear once temperature has fallen to safe levels
1	AOCP	Channel A overcurrent shutdown. Please check wiring or possible short circuit.
2	BOCP	Channel B overcurrent shutdown. Please check wiring or possible short circuit.
3	APDF	Channel A predriver fault. Please check driver settings.
4	BPDF	Channel B predriver fault. Please check driver settings."
5	UVLO	Power supply voltage too low Bit clears after voltage rises above lower limit.
6	*STD*	*Stall detected.*
7 (MSB)	*STDLAT*	*Latched stall detect.*

Fault 0 (OTS) cannot be cleared due to the nature of the fault. Faults 1 – 5 can be cleared and operation resumed by writing 0 to the bit location. Please see subsection 8.3.22 0x35 – Reset faults. Faults 6 and 7 are faults related to stall detection and are not real faults – they are not stopping operation of the PoStep60 driver. The faults 6 and 7 can be ignored.

NOTE!

When there is no external power supply available the fault data shows 0xFF – all errors active! Please ignore the value while the power supply is not available.

8.3.17 0x30 – Set full scale current

Master writes 0x30 command followed by two bytes of data. BYTE0 and BYTE1 are calculated as follows:

```
int Tq = 123 * xxCurrent; //xxCurrent: current value in A
int Ai = 3;
while(Tq > 255){
        Ai--;
        Tq = Tq >> 1;
}
(uchar)BYTE0 = Tq;
(uchar)BYTE1 = Ai;
```

NOTE!

Limiting values from subsection 6.1 - Electrical specification – limiting values applies.

8.3.18 0x31 – Set idle current

Master writes 0x31 command followed by two bytes of data. BYTE0 and BYTE1 are calculated using same algorithm as in subsection 8.3.17 0x30 – Set full scale current.

8.3.19 0x32 – Set overheat current

Master writes 0x32 command followed by two bytes of data. BYTE0 and BYTE1 are calculated using same algorithm as in subsection 8.3.17 0x30 – Set full scale current.

8.3.20 0x33 – Set step mode

Master writes 0x33 command followed by one byte of data – BYTE0. BYTE0 relates to step mode as follows:

Mode	BYTE0
Full step	0
Half step	1
1/4 step	2
1/8 step	3
1/16 step	4
1/32 step	5
1/64 step	6
1/128 step	7
1/256 step	8

Other values are ignored.

8.3.21 0x34 – Set temperature limit

Master writes 0x34 command followed by one byte of data – BYTE0. BYTE0 is calculated as follows:

BYTE0 = Temperature/°C

8.3.22 0x35 – Reset faults

To reset all faults Master shall write 0x35 command (no data bytes are needed).

8.3.23 0x3F – Write settings to EEPROM

To store changes made with any of the “Set” commands a Master shall write 0x3F command (no data bytes are needed).

Following commands are related to internal position controller – please see subsection 7.5 Auto control.

8.3.24 0x40 – Read position

To read PoStep position first Master writes 0x40 command (no data bytes are needed). In a next step Master reads four bytes of data. Calculated position is:

position/steps = BYTE3 * 16777216 + BYTE2 * 65536 + BYTE1 * 256 + BYTE0 =
= BYTE3 << 24 | BYTE2 << 16 | BYTE1 << 8 | BYTE0

8.3.25 0x41 – Read maximal speed

To read PoStep maximal speed first Master writes 0x41 command (no data bytes are needed). In a next step Master reads two bytes of data. Calculated maximal speed is:

Maximal speed/steps/s = BYTE1 * 256 + BYTE0 = BYTE1 << 8 | BYTE0

8.3.26 0x42 – Read acceleration

To read PoStep acceleration first Master writes 0x42 command (no data bytes are needed). In a next step Master reads two bytes of data. Calculated acceleration is:

acceleration/steps/s^2 = BYTE1 * 256 + BYTE0 = BYTE1 << 8 | BYTE0

8.3.27 0x43 – Read deceleration

To read PoStep deceleration first Master writes 0x43 command (no data bytes are needed). In a next step Master reads two bytes of data. Calculated deceleration is:

deceleration/steps/s^2 = BYTE1 * 256 + BYTE0 = BYTE1 << 8 | BYTE0

8.3.28 0x44 – Read current speed

To read PoStep current speed first Master writes 0x44 command (no data bytes are needed). In a next step Master reads two bytes of data. Calculated deceleration is:

current speed/steps/s = BYTE1 * 256 + BYTE0 = BYTE1 << 8 | BYTE0

8.3.29 0x45 – Read requested speed

To read PoStep requested speed first Master writes 0x45 command (no data bytes are needed). In a next step Master reads two bytes of data. Calculated requested speed is:

requested speed/steps/s = BYTE1 * 256 + BYTE0 = BYTE1 << 8 | BYTE0

8.3.30 0x46 – Read Auto run invert direction status

To read PoStep Auto run invert direction status first Master writes 0x46 command (no data bytes are needed). In a next step Master read one byte of data. Calculated direction is:

Auto run invert direction = BYTE0 & 0x01

8.3.31 0x50 – Set position

NOTE!

Set position and internal position control profile settings are applicable only when driver is in Position control or BINx buttons mode. To enable position control over I2C after reset and (in standalone mode) please enable BINx buttons mode and write values to driver.

To set required position of internal position controller Master writes 0x50 command followed by four bytes of data where:

MSB	Required position (32 bit)		LSB
BYTE3	BYTE2	BYTE1	BYTE0

8.3.32 0x51 – Set maximal speed

To set required maximal speed of internal position controller Master writes 0x51 command followed by two bytes of data where:

MSB	Maximal speed (16 bit)	LSB
BYTE1		BYTE0

8.3.33 0x52 – Set acceleration

writes 0x52 command followed by two bytes of data where:

MSB	Acceleration (16 bit)	LSB
BYTE1		BYTE0

8.3.34 0x53 – Set deceleration

Master writes 0x53 command followed by two bytes of data where:

MSB	Deceleration (16 bit)	LSB
BYTE1		BYTE0

8.3.35 0x54 – Set requested speed

NOTE!

Set speed and invert direction settings are applicable only when driver is in Auto run/Speed mode. To enable Speed mode control over I2C set driver mode to Auto run mode (8.3.4 0x05 – Set driver mode0x05 – Set driver mode) or enable Speed mode within PoStep application and write values to driver (standalone mode).

Master writes 0x54 command followed by two bytes of data where:

<table>
<tr><th colspan="2">MSB Requested speed (16 bit) LSB</th></tr>
<tr><td>BYTE1</td><td>BYTE0</td></tr>
</table>

8.3.36 0x55 – Set invert direction

Master writes 0x55 command followed by one byte of data where only LSB bit defines direction:

<table>
<tr><th colspan="2">MSB Invert direction (8 bit) LSB</th></tr>
<tr><td>BYTE0</td><td>BIT0</td></tr>
</table>

8.3.37 0x5E – Set zero

Master shall write 0x5E command (no data bytes are needed).

8.3.38 0x5F – Stop

Master shall write 0x5F command (no data bytes are needed).

9 Errata information

- Auto run invert direction set cannot be applied over I2C. Driver update is needed to a software version 0x0109 or above.
    - **Affected:** All drivers with a software version above 0x0107 and bellow 0x0109.
- I2C communication fails from time to time. Driver update is needed to a software version 0x0105 or above.
    - **Affected:** All drivers with a software version bellow 0x0105.
- Predriver Fault bug causing faulty driver failure status. Driver update is needed to a software version 0x0103 or above.
    - **Affected:** All drivers with a software version bellow 0x0103.

10 User manual changes

Changes in 6/25/2016 version:

- Added Driver set mode over I2C
- Fixed bug for Auto run invert direction set over I2C

Changes in 6/4/2016 version:

- Added Auto run I2C commands
- Added I2C address change I2C command

Changes in 5/2/2016 version:

- Added Auto run control mode

Changes in 5/12/2015 version:

- Fixed I2C communication breakage bug
- Added new I2C commands

Changes in 10/9/2015 version:

- Added BINx and I2C functionality related to application
- Added I2C protocol specifications and commands
- Added PoExtension pinout

Changes in 2/6/2015 version:

- 10 pin IDC connection diagram updated
- PWM input pinout description and note updated
- Technical specifications section updated

11 Grant of license

The material contained in this release is licensed, not sold. PoLabs grants a license to the person who installs this software, subject to the conditions listed below.

11.1.1 Access

The licensee agrees to allow access to this software only to persons who have been informed of and agree to abide by these conditions.

11.1.2 Usage

The software in this release is for use only with PoLabs products or with data collected using PoLabs products.

11.1.3 Copyright

PoLabs claims the copyright of, and retains the rights to, all material (software, documents etc) contained in this release. You may copy and distribute the entire release in its original state, but must not copy individual items within the release other than for backup purposes.

11.1.4 Liability

PoLabs and its agents shall not be liable for any loss or damage, howsoever caused, related to the use of PoLabs equipment or software, unless excluded by statute.

11.1.5 Fitness for purpose

No two applications are the same, so PoLabs cannot guarantee that its equipment or software is suitable for a given application. It is therefore the user's responsibility to ensure that the product is suitable for the user's application.

11.1.6 Mission Critical applications

Because the software runs on a computer that may be running other software products, and may be subject to interference from these other products, this license specifically excludes usage in 'mission critical' applications, for example life support systems.

11.1.7 Viruses

This software was continuously monitored for viruses during production; however the user is responsible for virus checking the software once it is installed.

11.1.8 Support

No software is ever error-free, but if you are unsatisfied with the performance of this software, please contact our technical support staff, who will try to fix the problem within a reasonable time.

11.1.9 Upgrades

We provide upgrades, free of charge, from our web site at www.poscope.com. We reserve the right to charge for updates or replacements sent out on physical media.

11.1.10 Trademarks

Windows is a registered trademark of Microsoft Corporation. PoStep, PoDDS, PoRef, PoScope, PoLabs and others are internationally registered trademarks.

Support:

www.poscope.com

http://blog.poscope.com/stepper-motor-driver/