



PiezoMotor Microstep Driver 101

Technical Manual

Imprint

Revision 02

11.04.2014

Document number 150086-02

© Copyright by PiezoMotor Uppsala AB
Stålgatan 14, SE-754 50 Uppsala, Sweden

All rights reserved, including those to the translation. No part of this description may be duplicated, reproduced, stored in an information system or processed or transferred in any other form without prior express written permission of PiezoMotor Uppsala AB.

Overview

This document is intended as a guide to the technical aspects of installing and using *PiezoMotor Microstep Driver 101* (hereafter referred to as PMD101). The unit is an advanced one axis driver for *Piezo LEGS* linear and rotary motors from PiezoMotor. The driver will give the motors resolution down in the nanometer/microradian range by dividing every full wfm-step into small and precise microsteps.

Detailed information about the different Piezo LEGS motors can be found on our homepage:

www.piezomotor.com

Contents

1 – Important Information

1.1 Symbols Used in this Document.....	1
1.2 Safety Instructions	2

2 – Delivery

2.1 Packaging and Delivery Contents.....	3
--	---

3 – Functional Principle

3.1 Piezo LEGS Linear and Rotary Motors	5
3.2 Motor Drive Principle	5
3.3 PiezoMotor Microstepping Driver.....	6
3.4 Waveforms and Voltages.....	6

4 – Quick Start

4.1 Getting Started	9
4.2 Installation.....	10
4.2.1 Cooling	
4.2.2 Power Supply	
4.2.3 Host Communication	
4.2.4 Motor Connection	
4.2.5 Encoder Connection	

5 – Detailed Technical Description

6 – Commands

6.1 Introduction.....	13
6.2 Commands	13
6.2.1 List of Commands	
6.2.2 Status Command	
6.2.3 Settings Commands	

7 – Installation and Assembly

7.1 Installation.....	21
7.2 Main Dimensions	21
7.3 External Connectors.....	22
7.3.1 Motor Port Pinouts	
7.3.2 Sensor Port Pinouts	
7.3.3 I/O Port Pinouts	
7.3.4 USB Connector	
7.3.5 Power Supply	

8 – Operation

8.1 Driving a Piezo LEGS Motor	25
8.2 Closed Loop	25
8.3 Open Loop	26
8.3.1 Internal Step Control	
8.3.2 External Step Control - Step/Direction Interface	
8.3.3 External Step Control - Analog Interface	
8.4 PiezoMotor Motion System Software	28

9 – Settings

9.1 Closed Loop Target Mode Parameters	29
9.1.1 StepsPerCount Parameter	
9.1.2 Target Mode Position Limits A and B	
9.1.3 Target Mode Minimum and Maximum Speed	
9.1.4 Target Mode Speed Ramp Up and Down	
9.1.5 Target Mode Stop Range	
9.1.6 Target Mode Model	
9.2 Other Settings	30
9.2.1 Parking and Initialization	
9.2.2 External Limits	
9.2.3 Encoder Direction	
9.2.4 Encoder Type	
9.2.5 Encoder Offset	
9.2.6 Save-and-Verify	
9.2.7 Prompt Character	
9.2.8 Analog Mode Stop Voltage	
9.2.9 Analog Mode Stop Range	

10 – Maintenance

10.1 General Maintenance Instructions	33
10.2 Troubleshooting	33
10.3 Firmware Updates	33

11 – Warranty

11.1 Warranty Conditions	35
--------------------------------	----

1 – Important Information

1.1 Symbols Used in this Document



Caution!

This pictogram with the wording “Caution!” indicates an imminent danger which can result in slight physical injuries or material damage.

► This arrow points out the appropriate precautions.



Regulation!

This pictogram with the wording “Regulation!” indicates a statutory regulation, guideline or directive which must be observed in respective context of the text.



Note!

This pictogram with the wording “Note” provides tips and recommendations for use and handling of the component.

1.2 Safety Instructions



Note!

The PMD101 driver is a high-end product intended for use with PiezoMotor's *Piezo LEGS* product line. In order to get best performance and reliability it is important the driver unit is handled according to the instructions given in this manual and other delivery documents.



Caution!

The piezoceramic elements in a Piezo LEGS motor act as capacitors and can sometimes hold substantial electrical charge.

- Make sure motors are discharged through suitable discharge resistors.



Caution!

Incorrect installation using improper mounting materials or methods can cause damage to the PMD101 driver unit.

- Observe the installation instructions.



Caution!

Depending on its use the PMD101 unit can get very hot.

- The unit should be installed in a clean and dry environment with access to proper ventilation. On installation, ensure that air can flow around the unit without obstruction. Additional heat sink must be applied if the unit is installed in secondary housing to ensure that the unit does not overheat. The PMD101 is intended for indoor operation.



Caution!

Electrostatic discharges at the driver unit connectors can cause irreparable damage to the electronics.

- Note and follow the ESD protective measures



Caution!

Incorrect connection of motor leads may cause irreparable damage to both motor and driver unit electronics.

- Connect in accordance with the specified pin assignment (see “Chapter 7 – Installation and Assembly” on page 21).

2 – Delivery

2.1 Packaging and Delivery Contents

The PMD101 driver unit has been cleaned prior to packaging, but does not carry any clean room classifications. The unit is shipped in a cardboard box with foam packaging material.

These items are included in the box at delivery:

- 1 x PMD101 driver unit
- 1 x USB cable
- 1 x AC-DC adaptor, 15V, 1.2 A
- 1 x CD with *PiezoMotor Motion System* software and pdf-version of this document
(latest version of software and manual is available on the PiezoMotor homepage)

3 – Functional Principle

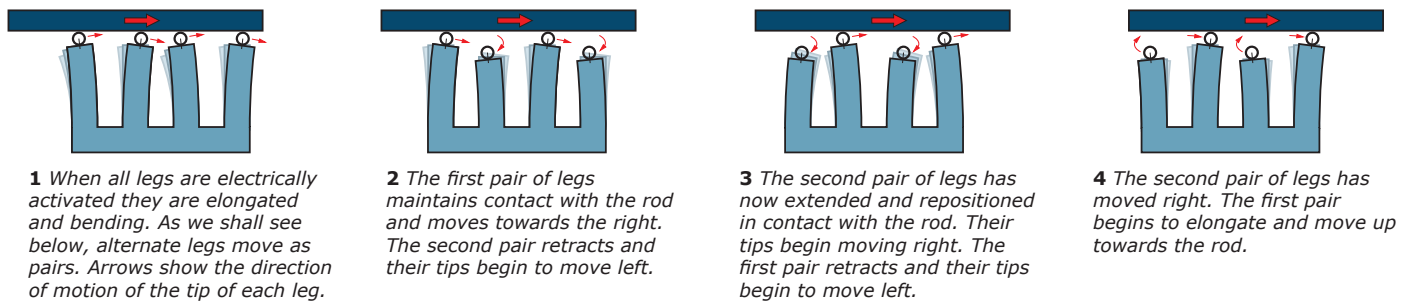
3.1 Piezo LEGS Linear and Rotary Motors

The PMD101 driver unit is made to control PiezoMotor's linear and rotary Piezo LEGS motors. The Piezo LEGS technology is characterized by its outstanding precision and high resolution. The motors are backlash free because of the friction based direct drive. There is no need for gearboxes or linear screws, and the motor is very strong compared to its size.

The performance of a Piezo LEGS motor is different from that of a DC or stepper motor in several aspects. A Piezo LEGS motor is friction based, meaning motion is transferred through contact friction between the drive leg and the drive rod (Piezo LEGS Linear), or between the drive leg and drive disc (Piezo LEGS Rotary). You cannot rely on each step being equal to the next. This is especially true if the motor is operated under varying loads/torques.

3.2 Motor Drive Principle

You need to feed 4 electrical drive signals to the motor in order to activate the legs and start moving. The actuation of the legs will lift and push the drive rod forward, or in case of a rotary motor, lift and spin the drive disc and axle around. For each waveform signal period the legs will complete one full step (see illustration below). To alter the speed of the motor the frequency of the electrical signals is changed. The motor speed will depend on the magnitude of external force/torque. The maximum speed is limited to a few thousand steps per second, which translates to movements in the range of centimeters per second (linear motion).



Definition

One *waveform-step* (*wfm-step*) is the step taken by the drive legs for each waveform signal period.

Note!

The wfm-step length/angle is not constant. Step length/angle will depend on external force/torque, and therefore the drive rod/disc will not always travel the same distance for one wfm-step.

Example

Piezo LEGS Linear: When taking a wfm-step the drive rod will move several micrometers.

Piezo LEGS Rotary: When taking a wfm-step the drive disc will rotate. The step angle depends on the diameter of the drive disc.

If there is an external force/torque present and the force/torque is opposing the direction of motion, the wfm-step length/angle will be decreased (shorter step). If the force/torque is working in the same direction as the direction of motion, the wfm-step length/angle will be increased (longer step). The speed of a Piezo LEGS motor is a product of the actual wfm-step length/angle (long or short) and the frequency of the drive signals. For varying loads/torques, using constant waveform frequency will not give a constant speed, simply because the wfm-step length is not constant.

To exemplify, the Piezo LEGS Linear 20N motor has a typical step length of $\sim 5 \mu\text{m}$ when there is no external load present. Running 2000 wfm-steps per second will generate a speed of $\sim 5 \mu\text{m} \cdot 2000 \text{ Hz} = \sim 10 \text{ mm/s}$. When the motor is loaded with 10 N of force opposing the direction of motion, the wfm-step length is decreased to $\sim 4 \mu\text{m}$, and resulting speed when driving at 2 kHz is of course $\sim 4 \mu\text{m} \cdot 2000 \text{ Hz} = \sim 8 \text{ mm/s}$. When the motor is loaded with 10 N of force in the same direction as the direction of motion, the wfm-step length is increased to $\sim 6 \mu\text{m}$. Resulting speed for 2 kHz drive frequency is $\sim 6 \mu\text{m} \cdot 2000 \text{ Hz} = \sim 12 \text{ mm/s}$. To conclude, with an unbalanced load the resulting speed will be different for forward and reverse motion.

3.3 PiezoMotor Microstepping Driver

The PMD101 is an high resolution microstepping driver made for Piezo LEGS linear and rotary motors. The PMD101 driver generates suitable waveforms to control one motor. Each waveform signal is constituted of numerous predefined discrete points. These points are voltage targets occurring at specific times along the waveform. When going from one point to the next, the legs of the motor are bent just a little bit, and the drive rod or drive disc is moved by so called *microsteps*. Resolution settings will decide how many microsteps there are within each complete wfm-step. The PMD101 driver unit has a maximum resolution of 2048 microsteps per 1 wfm-step. With typical wfm-step length of $\sim 5 \mu\text{m}$ (at no load), the microstep length at maximum resolution is $\sim 5 \mu\text{m} / 2048 = \sim 2.4 \text{ nm}$.

Definition

One *microstep* is a small fraction of a full wfm-step. Maximum resolution settings with the PMD101 gives 2048 microsteps within 1 wfm-step.

Note!

The microstep length is not constant. Microstep length will depend on resolution settings and on external forces/torques.

Example

Piezo LEGS Linear: The drive rod will not always travel the same distance for one microstep. Microsteps are close to 2.5 nm (using a resolution of 2048 microsteps per wfm-step).

Piezo LEGS Rotary: The drive disc will not always rotate the same distance for one microstep. The angular rotation of the drive axle for one microstep will depend on diameter of the drive disc.

3.4 Waveforms and Voltages

The driver unit delivers approximately $42 V_{pp}$ drive signals with optimal waveforms to operate the Piezo LEGS motors. Each repeating waveform signal period is constituted of numerous predefined discrete voltages. The maximum resolution of the PMD101 is 2048 target points per signal period, giving the Piezo LEGS motor resolution in the nanometer range (linear motion). There are several waveforms available in the PMD101 unit. The waveforms have been thoroughly tested for best performance with the Piezo LEGS motors. The waveforms are named *Delta*, *Rhomb*, and *RhombF*.

The Delta waveform gives a strong and smooth movement, and is recommended to use for best performance. It is the only waveform which can be set to highest resolution (2048 microsteps per wfm-step). The Delta waveform is best suited for microstepping and precision positioning, but is slower compared to driving motors with the waveform called Rhomb.

The Rhomb waveform is less suited for precision positioning but may give faster speeds. The RhombF waveform is rather similar to Delta, but is only available at maximum resolution of 256 microsteps per wfm-step. The waveform and resolution can be selected with the M and R commands. Waveform and resolution can also be selected using the C command which sets both parameters M and R at the same time. The waveforms are:

M3 = Delta	with resolution 2048 / 1024 / 512 / 256 microsteps per wfm-step
M2 = Delta	with resolution 256 / 128 / 64 / 32 microsteps per wfm-step
M1 = Rhomb	with resolution 256 / 128 / 64 / 32 microsteps per wfm-step
M0 = RhombF	with resolution 256 / 128 / 64 / 32 microsteps per wfm-step

Higher resolution limits the maximum number of wfm-steps per second (i.e. the waveform signal frequency). With highest resolution setting (2048 microsteps per wfm-step) the maximum waveform frequency is 61 wfm-steps per second. The speed of the motor depends on the actual wfm-step length/angle (which is load dependant) and the waveform frequency. For example, if the actual wfm-step length is $\sim 3 \mu\text{m}$ and the waveform frequency is 61 wfm-steps per second, the resulting speed is of course $\sim 3 \mu\text{m} \cdot 61 \text{ Hz} = \sim 183 \mu\text{m/s}$.

It is not necessary for the user to deal with resolution settings. With the H command, the user can set the waveform frequency (wfm-steps per second), and the driver will autoselect appropriate microstep resolution within the selected waveform. If either Delta waveform (M2 or M3) is selected, the H command to select waveform frequency will autoselect microstep resolution from 32 to 2048.

Giving command M4 will power down motor (park). Parking is done in a special sequence to minimize the positional disruption. Still the position might change by fractions of a wfm-step (depending on external loads/torques) when going from an unparked to a parked state.

The Piezo LEGS motors will consume power based on their size. When using the PMD101 driver you must consider this. If you try to drive a motor too fast, the internal voltage step-up will not be sufficient and the waveform output will be incorrect. Please consult your PiezoMotor contact for notes on maximum speed with the motor you have in mind to use with the PMD101.

4 – Quick Start

4.1 Getting Started

1. Unpack the PMD101 unit and verify that no damage has occurred during transportation. Place unit so that it gets sufficient cooling.
2. To install *PiezoMotor Motion System* software, run Windows Installer (msi file) from the CD, or find the latest release on our homepage: www.piezomotor.com.

Additionally, communication drivers from FTDI needs to be downloaded and installed separately. Find the virtual com port drivers on the FTDI homepage: www.ftdichip.com.

3. Connect your Piezo LEGS motor to the PMD101 unit. If you are using a Piezo LEGS Linear Twin motor, it needs to be connected with two cables in parallel (for example using the *Twin Connect Card*).
4. If you want to be able to run motor in closed loop, you will need to connect an encoder. The PMD101 supports quadrature and serial SSI encoders, or an analog signal. Quadrature encoders are either connected to the dedicated *Sensor* port (using a 6 pole JST SHR-06V-S connector) or to the I/O port (using a standard 16 pole IDC socket connector). Analog and serial encoders can only be connected directly to the I/O port.
5. Connect the USB cable between the host (PC) and the PMD101 unit.
6. Connect the power supply to the PMD101 unit.
7. Start *PiezoMotor Motion System* software and scan for driver units. Software will detect and list all connected driver units.
8. Double click on the driver unit in the list to start wizard. The Wizard will help you configure the basic settings.
9. When the wizard is completed, the motor can be tested in open loop and/or in closed loop operation (closed loop only if an encoder is connected). The software can also be used to send commands directly to the PMD101 in order to verify function and configure parameter settings. The software can also run command sequences with one or several driver to perform simple automation tasks.

4.2 Installation

4.2.1 Cooling

The driver unit is cooled by convection and therefore needs free space for this purpose. If the driver gets overheated it will automatically stop running the motor.

4.2.2 Power Supply

The driver unit is supplied with a AC-DC adapter, input AC 100-240 V, 50/60 Hz and output 12 VDC, 1.5 A. The driver can be used with any 12-24 V DC supply (be sure to check the polarity of the connector; outer shell is GND).

4.2.3 Host Communication

The driver unit connects to a PC with a USB cable and uses a serial interface with ASCII commands. The driver unit is equipped with a FT232R USB-to-RS232 chip from FTDI. The FTDI driver will by default add a virtual RS232 port so that commands can be tested manually via any terminal program, such as Hyperterminal (57600n81, local echo ON, CR+LF). If you type a wrong command in the terminal program, just press Esc to cancel the command line. Programmers may also use the D2XX Dynamic-link library (dll) to communicate with the driver unit (see FTDI manuals).

Communication Settings					
Baud Rate	Start Bit	Data Bits	Parity	Stop Bits	Handshaking
57600	1	8	none	1	none

4.2.4 Motor Connection

The connection to the motor is via a straight five pole cable with JST connectors. All Piezo LEGS twin motors need a *Twin Connect Card* in order to split the signal to the two motor connectors.



Caution!

Failure to connect all motor phases, or switching phases, can permanently damage the motor. Functional wiring can be verified by measuring motor phase capacitances through the motor cable from the driver side before attempting to run motor.

4.2.5 Encoder Connection

Encoder for position feedback may be connected to the dedicated sensor port or to the I/O port. Read more about supported encoder types in “Chapter 8 – Operation” on page 25.

5 – Detailed Technical Description

Editorial note - this chapter will be included in a later revision of this document.

6 – Commands

6.1 Introduction

The PMD101 is a versatile driver, made to work on its own (receiving commands from the host) or in close interaction with for example a servo motor controller. In this chapter, the command structure and commands are described so that a programmer can build software to control one or several drivers. When connected to the host (PC) the PMD101 unit is identified with the serial code of the internal FTDI chip. If multiple units are connected to the same host, each unit will be recognizable by the unique serial code.

6.2 Commands

All commands are in plain ASCII format and consist of a single letter. Read commands are executed immediately. All responses ends with CR and optionally a prompt character. Set commands are followed by an ASCII number and must be terminated with CR or any of the other valid command delimiters. There is no response to a set command but it may be combined with a read command, e.g. **M3m** to set waveform (**M3**) and read back setting (**m**).

6.2.1 List of Commands

The main commands to use with the PMD101 unit are listed in the table below. To better understand the Piezo LEGS technology, and definitions of *wfm-step* and *microstep*, read “3 – Functional Principle” on page 5.

PMD101 Commands			
Command	Type	Description	Example and notes
CR		Valid command delimiter	
LF		Valid command delimiter	
;		Valid command delimiter	
0..9		Numerical data. Some commands will also handle ":" for fractional data	
?	Read	Reads PMD101 firmware version	Response can be: BB-090 V3.0
*	Read	Reads run status	Response when running: 1 Response when stopped: 0
u	Read	Reads status	Response is: u <d₁d₂d₃d₄> Detailed information in table "Status Command (u-Command)" on page 17.
O<value>	Set	Sets encoder position value	Send O25 to set encoder position to 25
e (or E)	Read	Reads encoder position value	Response is for example e 25
T<value>	Set	Runs motor in <i>target mode</i> to specified target position (closed loop operation)	Send T4321 to guide motor to target count 4321 in closed loop operation
t	Read	Reads last given target position	Response can be: t 4321
S	Set	Stop running (and exit closed loop operation)	
Y<x>=<value>	Set	Sets parameter x (closed loop target mode settings and other special functions) Detailed information in table "Setting Commands (Y-Commands)" on page 18.	Send Y33=1 to set Y parameter 33 to state 1
Y<x>?	Read	Reads the setting for parameter x Detailed information in table "Setting Commands (Y-Commands)" on page 18.	Send: Y33? response: Y33=0 or Y33=1 You can set and read back in the same command: Y33=1? will set Y parameter 33 to state 1 and ask for response

PMD101 Commands

Command	Type	Description	Example and notes
M<digit>	Set	Open and closed loop waveform select and parking state. When selecting a waveform the motor is automatically unparked.	M4 to park motor (power down) M3 to set wfm <i>Delta</i> (res. 2048/1024/512/256) M2 to set wfm <i>Delta</i> (res. 256/128/64/32) M1 to set wfm <i>Rhomb</i> (res. 256/128/64/32) M0 to set wfm <i>RhombF</i> (res. 256/128/64/32)
m	Read	Read waveform setting	Response can be m 3 (if <i>Delta</i> high resolution is selected)
J<value>	Set	Run <i>generic microsteps</i> , where 2048 generic microsteps always equals one wfm-step regardless of actual resolution setting. Enter negative value to run reverse direction.	Send J512 to run 512 generic microsteps (equals 1/4 of 1 wfm-step).
J<val1>:<val2>	Set	Run <i>generic microsteps</i> with multiplier	Send J5:1024 to run 5.5 wfm-steps, which equals 11264 generic microsteps ($5 \cdot 2048 + 1024 = 11264$) Send J1:256 to run 1.125 wfm-steps, which equals 2304 generic microsteps ($1 \cdot 2048 + 256 = 2304$)
j	Read	Report <i>generic microsteps</i> left to run	Response is for example j 2:1963 if there is 6059 generic microsteps left to run ($2 \cdot 2048 + 1963 = 6059$)
H<value>	Set	Sets speed in number of wfm-steps per second (1...2500 wfm-steps/sec) with autoselect microstep resolution. This command changes settings but will not start running the motor.	Send H1200 to set 1200 wfm-steps per second Use H and J commands in sequence: H1200J5:1024 to set speed to 1200 wfm-steps per second and run 5.5 wfm-steps (11264 generic microsteps). Use H<60 in order to get best resolution within a waveform.
h	Read	Reports currently set speed settings (wfm-steps per second)	Response is a calculation formula.
N<digit>	Set	Enter/exit index mode	Send N4 to enter index mode for ABN=111 When index mode is active, you have to run open loop to find the index position. Whenever the encoder index mark is reached, the encoder position value will be reset at once (to whatever value is stored in the Y14 parameter) and stop the motor. Depending on speed and external load, the motor might not stop on the mark. If not, just issue the T command to return to the newly reset index position. Send N0 to exit index mode
n	Read	Report the index mode value	The status command also reports index mode bit
R<digit>	Set	Sets open loop microstep resolution. The new setting is not activated until next run command.	R3 for highest resolution within selected wfm R2 for second highest resolution within selected wfm R1 for second lowest resolution within selected wfm R0 for lowest resolution within the selected wfm For waveform M3 (<i>Delta</i>) the R2 setting will give resolution of 1024 microsteps per wfm-step. For waveform M1 (<i>Rhomb</i>) the R2 setting will give resolution of 128 microsteps per wfm-step.
r	Read	Reports open loop microstep resolution	Response can be r 2
C<dig1><dig2>	Set	Sets waveform and resolution in one command. Digit 1 is waveform (0...3) and digit 2 is resolution (0...3).	Send C32 to set waveform 3 (<i>Delta</i>) with resolution 2 (1024 microsteps per wfm-step)
c	Read	Reads currently used wfm and resolution	Response can be c 32 Note that currently used settings may differ from the settings that will take effect at next run

PMD101 Commands

Command	Type	Description	Example and notes																																																		
G<value>	Set	<p>Set microstep delay in multiples of 0.0625 μs. Minimum value is 128 and maximum value is 4,194,240.</p> <p>Takes immediate effect unless combined with the J command.</p> <p>Historic F command is still featured, but differs from how it was on PMD90. It can be used to set delay in multiples of 4 μs.</p>	<p>Send G128 to set delay 128 · 0.0625 μs = 8 μs.</p> <p>With microstep delay of 8 μs, you can calculate the number of wfm-steps/second: 1/(8 μs · [resolution])</p> <p>With resolution 2048 microsteps per wfm-step, you get 1/(8 μs · 2048) = 61 wfm-steps per second</p> <p>Suggested run command sequence M-R-G-J, e.g. M3R3G128J100:0</p>																																																		
g	Read	Reads currently used step delay multiplier (128...4,194,240)	Response can be g 128																																																		
D<value>	Set	<p>Run <i>actual microsteps</i> with currently used resolution settings. Enter negative value to run reverse direction.</p> <p>This run command is different from the J command because actual microsteps are used, and resolution settings will determine the traveling distance</p>	<p>Send D2048</p> <table><tr><td>If resolution is set to</td><td>Motor will go x wfm-steps</td></tr><tr><td>2048</td><td>1</td></tr><tr><td>1024</td><td>2</td></tr><tr><td>512</td><td>4</td></tr><tr><td>256</td><td>8</td></tr><tr><td>128</td><td>16</td></tr><tr><td>64</td><td>32</td></tr><tr><td>32</td><td>64</td></tr></table>	If resolution is set to	Motor will go x wfm-steps	2048	1	1024	2	512	4	256	8	128	16	64	32	32	64																																		
If resolution is set to	Motor will go x wfm-steps																																																				
2048	1																																																				
1024	2																																																				
512	4																																																				
256	8																																																				
128	16																																																				
64	32																																																				
32	64																																																				
d	Read	Read the <i>actual microsteps</i> left to run.	When D command is issued a counter is set. This counter immediately starts counting down towards zero when the motor runs. The d command will return the microsteps left to run.																																																		
+<value>	Set	Run forward <i>actual microsteps</i> (like the D command but <u>not</u> a command delimiter)	<p>Send +2048</p> <p>Motor will go x wfm-steps according to example for D command.</p>																																																		
-<value>	Set	Run reverse <i>actual microsteps</i> (like the D command but <u>not</u> a command delimiter)	<p>Send -2048</p> <p>Motor will go x wfm-steps according to example for D command.</p>																																																		
W	Set	Write current settings to non volatile memory (Flash/EEPROM)	No response, consider combining Ww to write and read back status.																																																		
w	Read	Read the settings stored in non volatile memory (Flash/EEPROM)	Response is w 0 if stored settings are equal to current settings, else the response is w 1																																																		
i (or I)	Read	Reads input pins 4, 7, 6 on I/O port.	<p>Response is a decimal number 1..7, i.e. if the response is i 7 it means all three pins are high. If the response is i 0 it means all three pins are low.</p> <table><tr><th>Input Pins</th><th>4</th><th>7</th><th>6</th><th></th></tr><tr><th>Bit</th><th>2</th><th>1</th><th>0</th><th>Dec</th></tr><tr><td></td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td></td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td></td><td>0</td><td>1</td><td>0</td><td>2</td></tr><tr><td></td><td>0</td><td>1</td><td>1</td><td>3</td></tr><tr><td></td><td>1</td><td>0</td><td>0</td><td>4</td></tr><tr><td></td><td>1</td><td>0</td><td>1</td><td>5</td></tr><tr><td></td><td>1</td><td>1</td><td>0</td><td>6</td></tr><tr><td></td><td>1</td><td>1</td><td>1</td><td>7</td></tr></table>	Input Pins	4	7	6		Bit	2	1	0	Dec		0	0	0	0		0	0	1	1		0	1	0	2		0	1	1	3		1	0	0	4		1	0	1	5		1	1	0	6		1	1	1	7
Input Pins	4	7	6																																																		
Bit	2	1	0	Dec																																																	
	0	0	0	0																																																	
	0	0	1	1																																																	
	0	1	0	2																																																	
	0	1	1	3																																																	
	1	0	0	4																																																	
	1	0	1	5																																																	
	1	1	0	6																																																	
	1	1	1	7																																																	

PMD101 Commands

Command	Type	Description	Example and notes																														
X<digit>	Set	Sets output pins 8 and 5 on I/O port.	Send X7 to set both pins 8 and 5 high.																														
			<table><tr><th>Output Pins</th><th></th><th>8</th><th>5</th><th></th></tr><tr><th>Bit</th><th>2</th><th>1</th><th>0</th><th>Dec</th></tr><tr><td></td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td></td><td>1</td><td>0</td><td>1</td><td>5</td></tr><tr><td></td><td>0</td><td>1</td><td>0</td><td>2</td></tr><tr><td></td><td>1</td><td>1</td><td>1</td><td>7</td></tr></table>	Output Pins		8	5		Bit	2	1	0	Dec		0	0	0	0		1	0	1	5		0	1	0	2		1	1	1	7
			Output Pins		8	5																											
			Bit	2	1	0	Dec																										
				0	0	0	0																										
				1	0	1	5																										
	0	1	0	2																													
	1	1	1	7																													
x	Read	Reads all pins on I/O port (outputs and inputs).	First digit is output pins (0...7), second digit is input pins (0...7). Response would be x 77 for all pins high (see the two tables above for a complete understanding).																														
p (or P)	Read	Read the value of the I/O analog input pin 3 (-10...10 V) or pin 4 (0...5 V)	Response is p <value> where <value> is an integer 0...1023.																														
v (or V)	Read	Reads the internal 48 V motor supply voltage	Response v <value> where <value> is an integer 0...1023 = 0...55 V.																														
a	Read	Reports position every 2 ms for 200 ms or until next incoming character from host.	<p>Note that this command should not be followed by CR or any other character. Response is all buffered encoder values in a text string.</p> <p>T5000a will set target position=5000 and start buffering position every 2 ms. This can be useful to monitor the performance. Note that only the first 200 ms will be logged. If you want to see performance when closing in on target you need to start at a point not so far away.</p> <p>To test for mechanical system resonances, try H1000J10a and look at the positional response at start/stop.</p>																														
...<Esc>		ASCII 27 will cancel any incomplete command.	Response is <esc>																														
...<Backspace>		ASCII 8 will silently cancel any incomplete command.	No response																														

6.2.2 Status Command

When sending the **u** command, the response is **u <d₁d₂d₃d₄>**. Each digit in the response holds four status flags with values 8, 4, 2, 1 respectively. In the table below, the meaning of each flag is explained. Each digit is between 0...9.

Status Command (u-Command)			
Status Digit	Flag	Name	Description
d₁	8	comErr	A host frame error was detected (or send buffer full)
	4	sensorErr	A sensor error was detected
	2	v48low	The driver unit 48 V voltage level was below limit (running stops)
	1	cmdWarning	An unknown command was detected
d₂	8	reset	The driver unit was reset (for instance when powering on)
	4	[not in use]	
	2	xlim	I/O port limit switch was activated (running stops)
	1	xrun	Analog run mode (on I/O port) is currently active
d₃	8	overheat	The driver unit is currently overheated (running stops)
	4	targetLimit	Target mode ended because encoder limit was reached
	2	targetMode	Currently in target mode
	1	indexMode	Currently in index mode
d₄	8	parked	Motor is currently parked
	4	tStop	Motion has been stopped at target position
	2	forward/clockwise	Running direction currently in use
	1	running	Currently running

Example

You can get a response **u 1827**:

d1=1 - cmdWarning

d2=8 - reset

d3=2 - targetMode

d4=7 - tStop, forward, running

indicates that an unknown command was detected

the driver unit was reset when powering on

currently in target mode

motor was stopped at target, but is now running forward

6.2.3 Settings Commands

The PMD101 must be configured correctly in order to get best performance out of the Piezo LEGS motor in your application. For instance, to run in closed loop you need to have ramping parameters set according to the actual external loads in your application. Most important for closed loop operation is to have the Y11 setting (*Target Mode StepsPerCount*) properly set up. The Piezo LEGS motor will perform very differently depending on external load/torque because the step length/angle is not constant. With target mode parameters properly configured, you will get the desired function in closed loop operation. See table below for explanation of each setting. You can learn more about the setting parameters in “9 – Settings” on page 29.

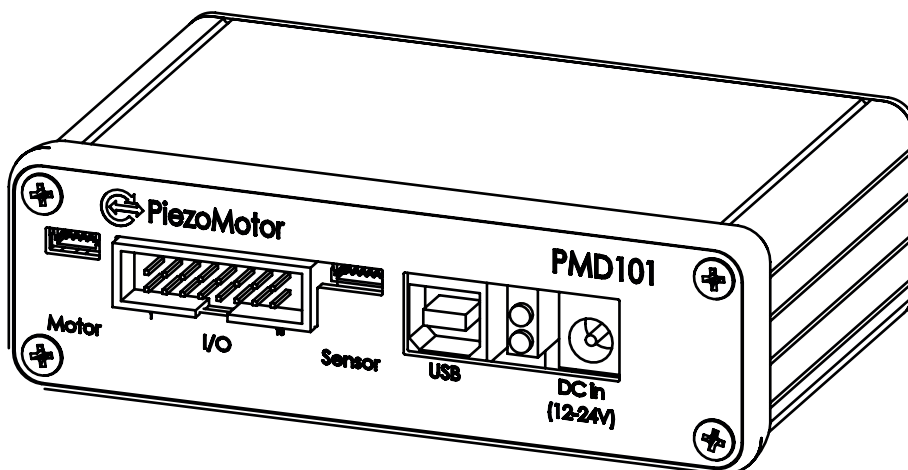
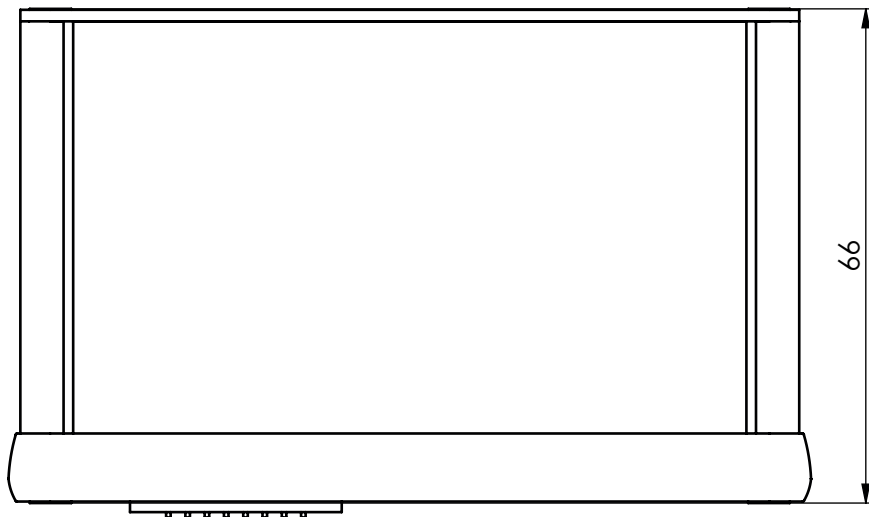
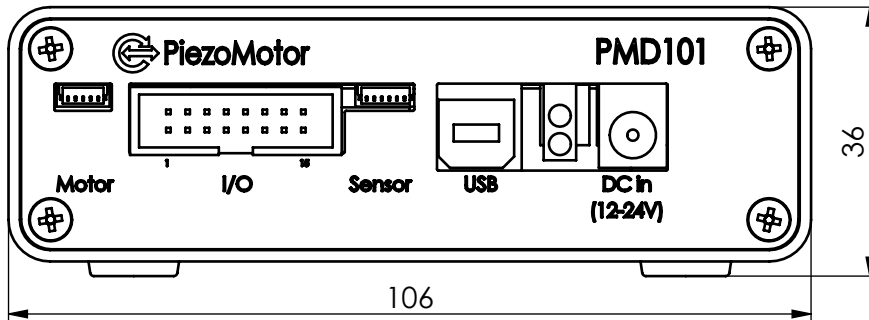
Setting Commands (Y-Commands)		
Y	Description	Notes
1	<p><i>Parking and initialization</i></p> <p>Y1=0 to unpark Read status with Y1? Y1=1 to park Response: Y1=0 - unparked (ready to run) Y1=2 to initiate from flash* Response: Y1=1 - parked (powered down) Y1=3 to initiate default values**</p> <p>* Parameters Y2...14, and Y38...40, waveform and resolution, parking state. ** Parameters Y2...14, and Y38...39, C33 (and motor is unparked).</p>	<p>Any run command will also unpark but stop command will not park. When issuing the park command (Y1=1) it will take around 300 ms to complete.</p> <p>Save to flash with Y32</p>
2	<p><i>External limits</i></p> <p>Y2=0 disabled Y2=2 enabled, active low Y2=1 enabled, active low Y2=3 enabled, active high</p>	Default: Y2=1 (U8)
3	<i>Target mode position limit A</i> - target mode will stop when encoder count <A	Default: Y3=-1000 (I32)
4	<i>Target mode position limit B</i> - target mode will stop when encoder count >B	Default: Y4=1000 (I32)
5	<i>Target mode stop range</i> - how many encoder counts from target is acceptable to stop	Default: Y5=0 (U16)
6	<p><i>Encoder direction</i> - to set the encoder counting direction</p> <p>Y6=0 for positive counting Y6=1 for negative counting</p>	Default: Y6=0 (U1)
7	<i>Target mode minimum speed</i> - wfm-steps per second	Default: Y7=1 (U16)
8	<i>Target mode maximum speed</i> - wfm-steps per second	Default: Y8=2000 (U16)
9	<i>Target mode speed ramp up</i> - wfm-steps per second per millisecond	Default: Y9=200 (U16)
10	<i>Target mode speed ramp down</i> - wfm-steps per second at 1 wfm distance from target	Default: Y10=200 (U16)
11	<p><i>Target mode StepsPerCount</i> - this parameter tells the controller how many wfm-steps to run to reach target.</p> <p>Y11=<SPC> where SPC is a integer which needs to be calculated by the user.</p> $SPC = \frac{2^{18} \cdot \text{encoder resolution [nm/count]}}{\text{distance per wfm-step [nm/wfm-step]}} = \frac{2^{18}}{\text{encoder counts per wfm-step}}$ <p>Example: You have a linear encoder with 20 nm resolution mounted on the motion axis, and you know that for your application the distance travelled for one wfm-step is 3 µm (3000 nm). You calculate the $SPC = 2^{18} \cdot 20 / 3000 = 1748$ and enter Y11=1748.</p> <p>Example: You have a linear encoder with 20 nm resolution mounted on the motion axis, but you are uncertain of the travel distance for one wfm-step. Read encoder value (e) then run at least ten wfm-steps (J10:) and read encoder again (e). You get a difference of 1500 counts, i.e. averaging 150 counts per each of the ten wfm-steps. You calculate the $SPC = 2^{18} / 150 = 1748$ and enter Y11=1748.</p> <p>Note! The average wfm-step may not be the same in both directions because the wfm-step length/angle is load dependant. You must either calculate the SPC for both directions and use the mean value, or reset SPC on each new run.</p>	<p>Default: Y11=3172 (U32)</p> <p>Note: $2^{18} = 4 \cdot 65536$</p>

7 – Installation and Assembly

7.1 Installation

The PMD101 driver unit should be used in a clean and dry environment with proper ventilation. On installation, ensure that air can flow around the unit without obstruction. Additional heat sink or fan may be applied if the unit is installed in secondary housing to ensure that the unit does not overheat. The PMD101 is intended for indoor operation.

7.2 Main Dimensions



7.3 External Connectors

The PMD101 has several ports for connecting motors and encoders, as well as I/O signals. The port pin-outs are listed in the tables below.



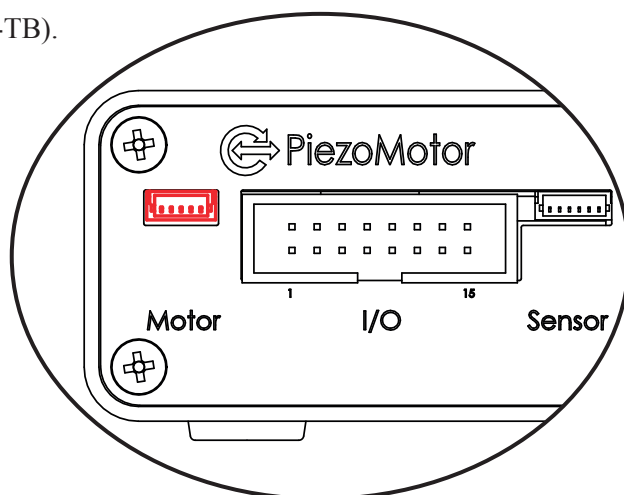
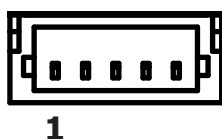
Note!

All I/O digital signals are 5 V logic and inputs have internal pull-up.

7.3.1 Motor Port Pinouts

The motor connector is a 5 pin JST connector (JST BM05B-SRSS-TB).

Motor Port	
Pin	Description
1	Phase 1
2	Phase 2
3	Phase 3
4	Phase 4
5	Ground (GND)



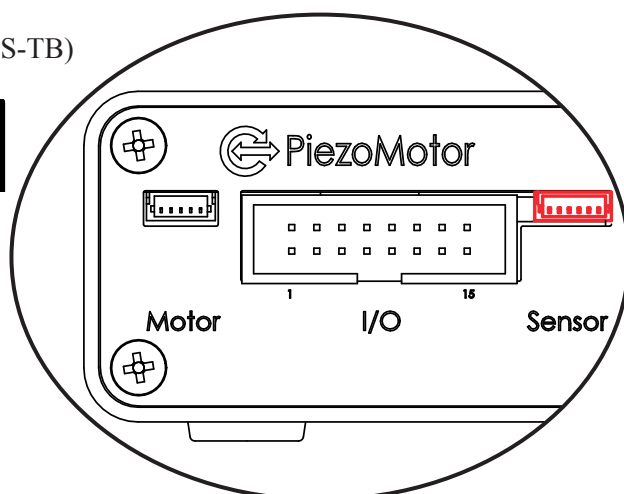
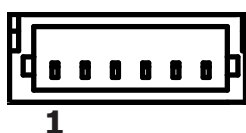
Caution!

Switching motor phases, or failure to connect all motor phases, can permanently damage the motor. Functional wiring can be verified by measuring motor phase capacitances through the motor cable from the driver side before attempting to run motor.

7.3.2 Sensor Port Pinouts

The encoder connector is a 6 pin JST connector (JST BM06B-SRSS-TB)

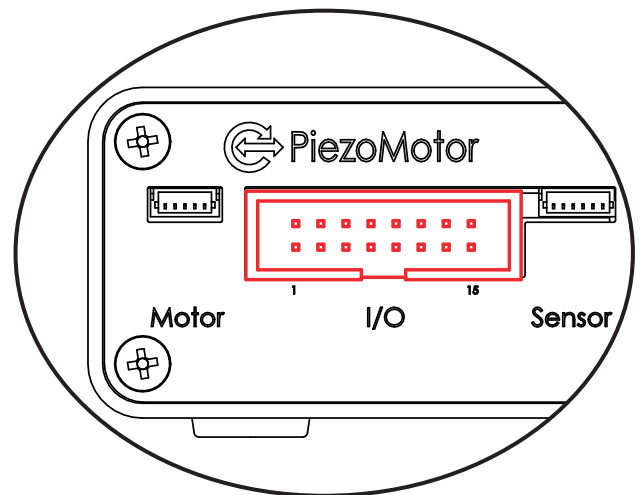
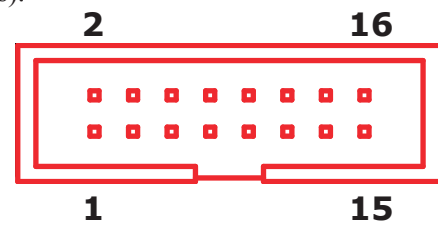
Sensor Port		
Pin	Type	Description
1	-	-
2	Input	Quadrature Signal A+
3	Input	Quadrature Signal B+
4	Output	+5 V (max 200 mA)
5	-	GND
6	Input	Quadrature Z Index



7.3.3 I/O Port Pinouts

The I/O port connector is a standard 2 x 8 pin header (for IDC sockets).

I/O Port			
Pin	Type	Description	Secondary Use
1	Output	+5 V (max 200 mA)	
2	-	Ground (GND)	
3	Input	Analogue -10...10 V	
4	Input	Analogue 0...5 V	In 2
5	Output	Out 0	SSi SDA
6	Input	In 0	External limit switch R
7	Input	In 1	External limit switch F
8	Output	Out 1	SSi SCK
9	Input	Analogue_run	
10	Input	Dir_x	
11	Input	Step_x	
12	Input	Quadrature Z Index	
13	-	Ground (GND)	
14	Output	+5 V (max 200 mA)	
15	Input	Quadrature Signal B+	
16	Input	Quadrature Signal A+	



7.3.4 USB Connector

The PMD101 uses USB communication between host (PC) and unit. A USB-to-RS232 chip inside the driver adds a virtual serial com port when connecting via USB. Appropriate FTDI drivers needs to be installed on your PC.

7.3.5 Power Supply

The PMD101 can be powered with a 12-24 V DC voltage. The unit is supplied with a 12 V (1500 mA) AC-DC adapter.

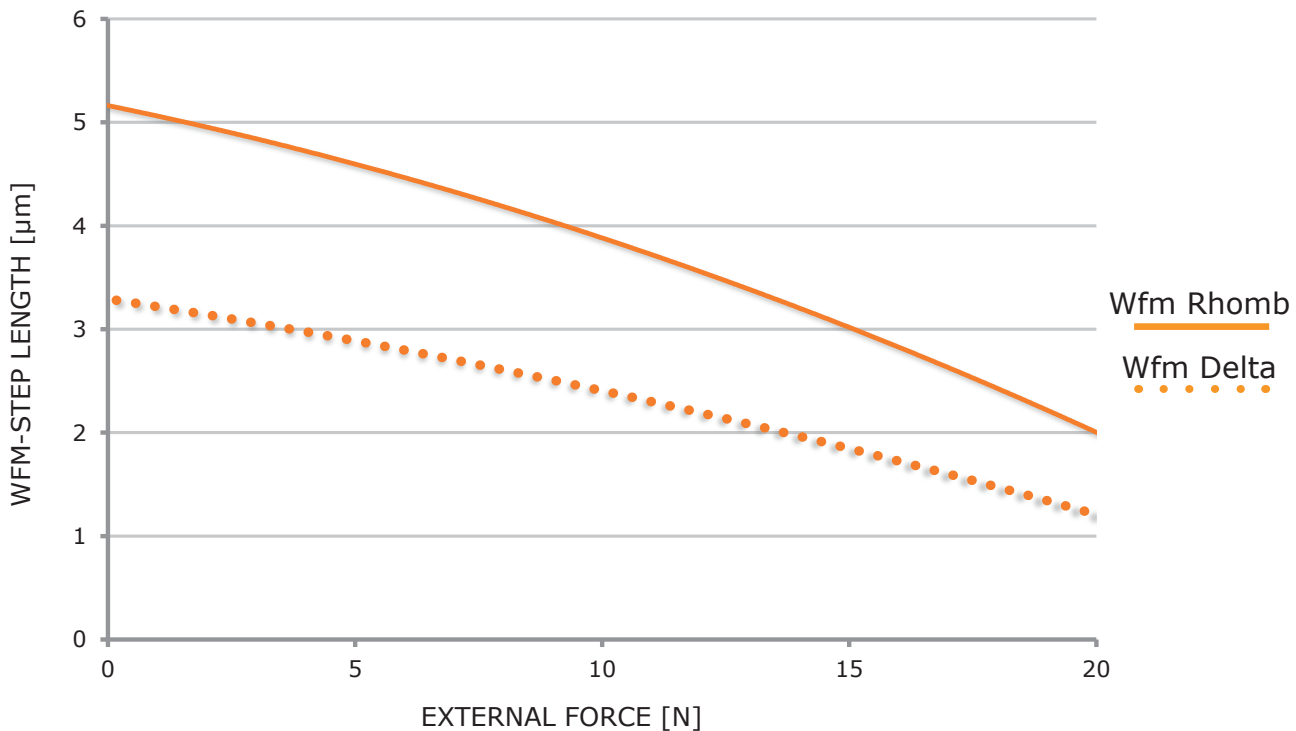
8 – Operation

8.1 Driving a Piezo LEGS Motor

When driving a Piezo LEGS motor, you cannot count wfm-steps or microsteps and know your position. Because of the friction drive you cannot rely on each step being equal to the next. For position control, you will have to use some form of feedback from positioning sensor or limit switches. The PMD101 allows for open or closed loop operation. There is no way to control the actual speed, only the number of wfm-steps per second.

Example

The motor wfm-step length is affected by the external force it is acting against. A typical behavior for LT20 motor is exemplified in the graph below (for waveform *Rhomb* and waveform *Delta*). Graph shows only behavior when working against the external force. When working with the external force the step length will increase accordingly.



Note!

No opposing force will for this specific motor type (and *Rhomb* waveform) give typical no load wfm-step length ~5 μm. Note that step length is also temperature dependent, and that these typical values are given at 20°C. With waveform *Delta* or *RhombF* the typical wfm-step length will be shorter compared to *Rhomb*. The stall force for this motor is specified to 20 N. For optimal microstep performance, and best life time, you should design your system so that the motor works at maximum 50% of stall force. Graph values are typical, and standard deviation σ of 0.5 μm should be taken into account.

8.2 Closed Loop

Closed loop operation is the preferred way of running the Piezo LEGS motor. In closed loop mode, the motor will be guided to a certain position when the “go to target” command is issued (T command). Of course, the closed loop mode of operation will only work when the driver is receiving feedback from an encoder. The PMD101 will read quadrature encoders or analog sensors. All serial SSi encoders are also supported via the I/O port. If a special encoder is not supported, one may have the host (PC) report the current position so that the driver can make corrections based on each report.

When the target command is issued, the motor starts moving and the PMD101 runs a target loop. A new estimate of the distance to target is made every 2 ms. The target loop will take in to account all applicable settings including ramping parameters. It is possible to set encoder limits to stop motion if position comes outside of a certain range. External limit switches can also be connected to the I/O port. The waveform type used in target mode is the one which have been saved to Flash/EEPROM memory.



Note!

To get closed loop operation working properly it is important to set the target loop parameters for the actual system. Carefully read “Chapter 9.1 Closed Loop Target Mode Parameters” on page 29 to learn more about all important settings for closed loop operation. Most important is to set the *StepsPerCount* parameter which holds information about encoder resolution and wfm-step length.

8.3 Open Loop

8.3.1 Internal Step Control

To drive a motor in open loop means sending commands to the driver and not using encoder feedback in a control loop. You can command the driver to take a precise number of wfm-steps or microsteps, but you cannot say beforehand in what position the motor will stop.



Note!

The easiest way of driving in open loop is by using the J command for *generic microsteps* (where 2048 microsteps always equals one wfm-step) and the H command to select speed (wfm-steps per second). Appropriate microstep resolution will be auto-selected within the chosen waveform. Note however that if you use the H command to select speeds over 60 wfm-steps per second, it will not use the highest resolution settings for the Delta waveform. For highest resolution within a wfm-step it is necessary to use $H < 60$.

8.3.2 External Step Control - Step/Direction Interface

An open loop mode of operation can also be triggered using the step/direction interface via the I/O port. It is possible to connect a stepper motor controller to the external pins 10, 11 and 13 of the PMD101. The control signals are XORed together and bypass the internal control. Notably, the external direction signal (Dir_x) gets inverted if the last open loop run was in forward direction. Limit switches should be connected to the external motion controller and not to the I/O port of the PMD101. Note that a stepper motor controller without encoder feedback will not suffice for closed loop operation.



Note!

The external step pulse (Step_x) from the stepper motor controller will make the PMD101 run microstepping at the currently set resolution. External step pulse frequency should never exceed 120 kHz. The resulting speed of the motor depends on the resolution setting. At lowest resolutions (32 or 64 microsteps per wfm-step) the resulting speed may exceed motor specifications.

Example

If the resolution of the PMD101 is set to 512 microsteps per wfm-step, and the external microstep frequency is 50 kHz, the resulting speed is $50000 / 512 = \sim 98$ wfm-steps per second. The average wfm-step length could for example be $\sim 5 \mu\text{m}$ for a certain load condition. The resolution of linear motion with this microstep resolution setting is $\sim 5 \mu\text{m} / 512 = \sim 0.01 \mu\text{m}$. The resulting speed is $\sim 5 \mu\text{m} \cdot \sim 98 \text{ Hz} = \sim 0.5 \text{ mm/s}$.



Note!

When powering on the PMD101, the waveform and microstep resolution is initiated from Flash/EEPROM. All settings can be saved to the memory using the W command. If the motor is parked when saving, the motor will not automatically be unparked at power on. The step/direction mode will not work as long as the motor is parked.

8.3.3 External Step Control - Analog Interface

The analog interface is enabled by connecting pin 9 (Analog_run) to GND. An analog motion control signal may be connected either to I/O port pin 4 (0...5 V levels) or to pin 3 (-10...+10 V levels). Pin 10 can be tied to GND in order to reverse the direction. Pins 3 and 4 are interconnected internally via a resistor so that 0 V on pin 3 gives 2.5 V on pin 4 (but not the other way around).



Note!

The exact stop voltage and stop range can be set by command (see “Chapter 6.2.3 Settings Commands” on page 18). Default stop voltage is in the middle of the interval (i.e. 0 V for the -10...+10 V interface, and 2.5 V for the 0...5 V interface).



Note!

The dedicated limit switch pins on the I/O port are not functional when using the analog interface, but an external limit switch can be used to disable analog mode. Setting pin 9 high (or no connect) will stop motion by disabling the analogue interface.

The analogue interface utilizes I/O pins 6 and 7 for selecting waveform and resolution. The high dynamic range from highest speed down to microstepping at highest resolution makes it a challenge to control by a simple linear voltage-speed function. For this reason there is also a non-linear voltage-speed function available. The table below shows the different operating modes.

Analogue Interface Operating Modes		
Pin 6	Pin 7	Description
0	0	Same as (1,0).
0	1	High resolution Delta (2048 microsteps per wfm-step). Speed (wfm-steps per second) is linear to the input voltage.
1	0	Auto-select resolution according to a non-linear voltage-speed function. This gives good control for speeds 0.25...2300 wfm-steps per second. Best option to use with external servo motor controllers.
1	1	Use the latest open loop settings (waveform and resolution). Speed (wfm-steps per second) is linear to the input voltage.

The digital limits (target mode position limits A and B) are active in analog mode. They will limit the driving range of the motor when getting position feedback from a quadrature sensor. When disabling analog mode (pin 9 high), the controller will go straight in to target mode and hold position (closed loop with quadrature sensor). To disable this behavior, set limit A = limit B. When setting them equal you will also disable limit control and have no range restrictions in analog mode.

8.4 PiezoMotor Motion System Software

To make it easy for everyone to start using the Piezo LEGS motors, there is a Microsoft Windows compatible software called *PiezoMotor Motion System*. The software is found on the installation CD that comes with the PMD101, and can also be downloaded from the PiezoMotor homepage: **www.piezomotor.com**.

On the PiezoMotor homepage you can find video tutorials to get familiar with the software.

9 – Settings

9.1 Closed Loop Target Mode Parameters

To set up your system to run in closed loop target mode you need to use an encoder for position feedback. Quadrature or analogue encoders, as well as serial SSI encoders are supported. If you use the *PiezoMotor Motion System* software, most settings can be set in the options tab. If you are writing your own software or if you are running the driver from a terminal program, you have to use the settings command (Y command) to set parameters. Read more about command syntax and how to connect to the PMD101 in “Chapter 6 – Software Implementation” on page 13.

9.1.1 StepsPerCount Parameter

Most important is to set the *Target Mode StepsPerCount* (SPC) parameter. The parameter is used in the driver unit target loop to calculate the number of wfm-steps to run to reach target. The SPC value is a calculated relationship between encoder resolution and wfm-step length. When the SPC is correctly set, the driver unit has all information necessary to perform an optimized target mode run.

$$\text{Linear motor: } \text{SPC} = \frac{2^{18} \cdot \text{encoder resolution [nm/count]}}{\text{wfm-step length [nm/wfm-step]}} = \frac{2^{18}}{\text{encoder counts per wfm-step}}$$

$$\text{Rotary motor: } \text{SPC} = \frac{2^{18} \cdot \text{encoder resolution [mrad/count]}}{\text{wfm-step angle [mrad/wfm-step]}} = \frac{2^{18}}{\text{encoder counts per wfm-step}}$$

Example Linear

You have a linear encoder with 20 nm resolution mounted on the motion axis, and you know (or can estimate from the graph in the motor data sheet) that for your application the distance travelled for one wfm-step is about 3 μm (=3000 nm). You can calculate the SPC value: $2^{18} \cdot 20 / 3000 = 1748$.

Example Linear

You have a linear encoder with 20 nm resolution mounted on the motion axis, but you are uncertain of the travel distance for one wfm-step. The wfm-step length can be estimated experimentally in the actual application. Run at least ten wfm-steps (open loop) to get a good average value. With an average of 150 counts per wfm-step you can calculate $\text{SPC} = 2^{18} / 150 = 1748$.

Example Rotary

You have a rotary encoder with 8192 CPR (counts per revolution). The encoder resolution is $2\pi/8192 \approx 0.77$ mrad. You know from your application (or can estimate from the graph in the motor data sheet) that the wfm-step angle is 0.9 mrad given the external load. You can calculate $\text{SPC} = 2^{18} \cdot 0.77 / 0.9 = 224279$

Example Rotary

You have a rotary encoder with 8192 CPR (counts per revolution). The encoder resolution is $2\pi/8192 \approx 0.77$ mrad. You are however uncertain of the angular displacement for one wfm-step and need to measure it experimentally. Run at least ten wfm-steps (open loop) to get a good average. With an average of 1.2 counts per wfm-steps you calculate the $\text{SPC} = 2^{18} / 1.2 = 218453$



Note!

The average wfm-step may not be the same in both directions because the wfm-step length/angle is load dependant. Recommended procedure is to run open loop in both directions to measure mean distance and calculate the average SPC.

9.1.2 Target Mode Position Limits A and B

In addition to external limit switches connected to the I/O port, you can set digital limits using the setting parameters *Target Mode Position Limit A* and *Target Mode Position Limit B*. When running in target mode the motor will stop when passing either of these limits. When motor is outside of the digital limits, it is no longer possible to run the motor in target mode. User must use open loop run commands to return back to within limits.

When running motor in analog mode (external step control), the digital limits will also control the drive range. To disable this, limit A must be set equal to limit B.

9.1.3 Target Mode Minimum and Maximum Speed

When running in target mode, the setting parameters called *Target Mode Minimum Speed* and *Target Mode Maximum Speed* defines the minimum and maximum speed of the run. The parameters are given in the unit wfm-steps per second. Default settings are minimum 1 wfm-step per second, maximum 2000 wfm-steps per second.

9.1.4 Target Mode Speed Ramp Up and Down

The setting parameters *Target Mode Speed Ramp Up* and *Target Mode Speed Ramp Down* will change the behavior of the target run. At the start of the target run the speed will be ramped up with the acceleration defined in the ramp up parameter. Default value is 200 wfm-steps per second per millisecond.

At the end of the target run motion will slow down according to what is set in the ramp down parameter. Ramp down parameter is defined as the cycle frequency (wfm-steps per second) at the estimated distance of one wfm-step from target. Default value is 200 wfm-steps per second.

9.1.5 Target Mode Stop Range

The setting parameter *Target Mode Stop Range* defines where it is acceptable for the PMD101 to stop when closing in on target. Default value is 1 encoder count, meaning the motion will stop in the range ± 1 encoder count from target.

9.1.6 Target Mode Model

The *Target Mode Model* setting relates to speed when closing in on target. If this setting is enabled, the PMD101 will run extra carefully towards target to minimize the risk for overshoot.

9.2 Other Settings

In this chapter, some of the more important settings are more thoroughly explained. All settings are listed in the table in “Chapter 6.2.3 Settings Commands” on page 18.

9.2.1 Parking and Initialization

The *Parking and Initialization* setting allows the user to park (power down) or unpark (power up) motor. The same command is also used for initiating various saved parameters from Flash/EEPROM or to initiate the factory set default values.

9.2.2 External Limits

The *External limits* setting will enable input pins 6 and 7 for use with external limit switches. External limit switches will work in closed loop target mode and in open loop, but not when running motor via the step-direction interface or via analogue interface. When the setting is enabled (active low), low levels on the I/O port pins will stop motion. Pin 6 low will stop reverse motion and pin 7 low will stop forward motion. Setting can also be enabled active high.

9.2.3 Encoder Direction

For proper closed loop operation, the encoder must count upwards when going in the forward direction. If the counting direction is wrong the *Encoder Direction* setting allows for digitally changing the target run direction.

9.2.4 Encoder Type

The *Encoder Type* setting defines what type of encoder is used. It is possible to use the PMD101 with quadrature encodes (with ABZ index), analogue encoders, and serial encoders (2-wire SSI, 8 to 32 bit, always extended to 32 bit by lap counting). The encoders are connected to the driver unit via the I/O port or via the dedicated sensor port (only quadrature encoder).

9.2.5 Encoder Offset

The *Encoder Offset* parameter setting defines the encoder count offset position from the index mark (for quadrature encoders). Default offset value is 0 so that encoder count will be set to 0 at the index mark.

Example

You enter index mode with command **N4** and run in open loop until index mark is detected (index mark is found when I/O pins 16, 15, and 12 are all high, i.e. ABZ=111). When detecting the index mark, the motor stops running and the encoder count is set to whatever value is stored in the *Encoder Offset* parameter. Driver also exits index mode after a successful detection. Depending on the speed and mass of the load, the motor might not come to a complete stop on the index mark. To return to the exact index mark position, run in closed loop target mode.

9.2.6 Save-and-Verify

This command is issued to save many different settings to the Flash/EEPROM memory. See details in “Chapter 6.2.3 Settings Commands” on page 18.

9.2.7 Prompt Character

If you want the driver to return a prompt character “>” after each command, you may enable the *Prompt Character* setting parameter. When enabled there will also be an indicator, “!>”, when unknown commands are detected.

9.2.8 Analog Mode Stop Voltage

The *Analog Mode Stop Voltage* setting parameter defines the voltage level that will be regarded as stop. The default stop voltage is 0 V on the I/O analogue ± 10 V input (pin 3), corresponding to 2.5 V on the I/O analog 0...5 V input (pin 4).

9.2.9 Analog Mode Stop Range

The *Analog Mode Stop Range* setting parameter defines the voltage range around the stop voltage that will still be regarded as stop. Default is ± 40 mV around the stop voltage setting on pin 3 (for the -10...10 V input), and ± 10 mV around the stop voltage setting on pin 4 (for the 0...5 V input).

10 – Maintenance

10.1 General Maintenance Instructions

The PMD101 does not require any regular maintenance. Follow given safety instructions!

10.2 Troubleshooting

If problem arise the status command (u command) error codes might provide useful information (see “Chapter 6.2.2 Status Command” on page 17). Check wirings to motor and encoder. Try driving in open loop to see if you can get movement. To get closed loop operation in order, make sure the SPC setting is correct. Check that you are calculating the SPC using the right encoder resolution. If you are using a quadrature encoder you need to check the interpolation factor. If you get readings from the encoder you can see if you have the correct resolution by taking a number of wfm-steps and checking how many encoder counts you have traveled. A wfm-step is typically $\sim 5\text{ }\mu\text{m} \pm \text{a few }\mu\text{m}$ for a standard Piezo LEGS linear motor at zero load.

10.3 Firmware Updates

When new features are added, the user can update the firmware. Please check our homepage for updates: [*www.piezomotor.com*](http://www.piezomotor.com).

When new firmware is installed, you should also look for the latest version of this manual to learn about the updates.

11 – Warranty

11.1 Warranty Conditions

General conditions for the supply of mechanical, electrical, and electronic products according to *Orgalime S2012*.

**Visit our website for application examples,
CAD files, videos and more...**

www.piezomotor.com



PiezoMotor Uppsala AB
Stålgatan 14
SE-754 50 Uppsala, Sweden

Telephone: +46 18 489 5000
Fax: +46 18 489 5001

info@piezomotor.com
www.piezomotor.com

