INSTRUCTION MANUAL

LRMO-E-CL Series Linear Piezoelectric Actuator Evaluation Kit
Python API Closed Loop System

Rev. 05212025

Table of Contents

OEM Evaluation Kit Instruction Manual

1. Introduction

Welcome to the realm of precision motion with the LRMO Series linear piezo actuator, crafted for unparalleled accuracy and efficiency. This manual will guide you through the operation and features of our innovative linear piezo actuator. The included evaluation kit provides all the essential components needed to fully explore the actuator's capabilities.

The LRMO Series sets new benchmarks for compact, high-performance linear piezoelectric actuators by merging a lightweight design with superior precision and functionality. These actuators are exceptionally energy-efficient, requiring no power in the hold position while still maintaining full torque. This efficiency makes them ideal for demanding OEM applications where performance and cost-effectiveness are crucial.

Each evaluation kit comes complete with an electronic driver PCB specifically designed for the LRMO piezo actuator, all necessary cables, and a dual-voltage 120/220 VAC to 5.0 VDC universal power supply.

The LRMO Series linear motors are offered with optional factory-fitted optical encoders and can also be customized to meet special application requirements.

2. Properties

The LRMO Series actuators boast several distinctive features:

- Construction from modern, reinforced engineered thermoplastics for reliability and affordability.
- Unmatched precision and resolution.
- Ultra-fast response times and exceptional start-stop capabilities.
- High linear force relative to size.
- Support for stepping and continuous modes of operation.
- A speed dynamic range spanning six orders of magnitude.
- Silent operation in continuous mode and low voltage design to minimize electrical arcing.

3. Unpacking and Preparation

After unpacking the LRMO series actuator evaluation kit, check the contents against the items listed in the table below. If any items are missing contact your supplier/distributor immediately for replacement parts.

ESCRIPTION
• LRMO series linear piezo actuator (with installed encoder)
• Pre-Programmed Close-Loop Electronic Driver (Green PCB)
• USB to Micro USB adapter Cable – connects driver PCB to Computer

Table1– Description

Figure 1. LRMO-E-CL Close Loop Electronic Driver PCB (top view)

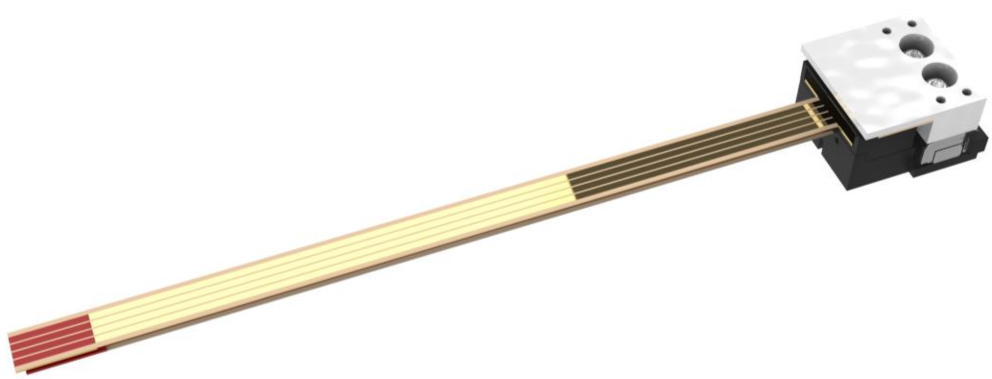


Figure 2. LRMO-E linear actuator (with installed encoder)

4. Technical Specifications of LRMO Series Linear Piezo actuator

Specification	Model
Power Supply Voltage	5.0 V DC
Push/Pull Force	≥0.2 N
Self-Braking Force	≥0.25 N
Motor Response Time	≈30 µs
Travel Range	9.0 mm
Max Speed (continuous mode)	≥150 mm/s
Minimum Linear Step	<0.04 µm
Encoder Resolution (after quadrature) *	2.66 µm
Minimum Controlled Linear Step*	2.66 µm
Uni-directional Repeatability	2.66 µm
Linear Backlash at Change of Direction	<0.1 µm
Elastic Stiffness	≈200 mN/µm
Linear Hysteresis	<2.0 µm
Pitch	<1 mrad
Maximum Moment Mx	0.07 Nm
Roll	<0.5 mrad
Maximum Moment My	0.12 Nm
Yaw	<1 mrad
Maximum Moment Mz	0.9 Nm
Vertical Runout	3.0 µm
Horizontal Runout	6.0 µm
Frequency Response	4 kHz
Operating Temperature	-20 °C to 80 °C
Maximum Load (at listed specification)	20 g
Maximum Tolerable Load	4.2 kgf
Max Current (continuous mode)	300 mA
Max. Current at the velocity 10mm/s (PWM mode)	30-40 mA
Motor Weight	6.8 g
Motor Dimensions	16.6x16.5x5.8mm
Driver PCB Dimensions	31x28x9.6 mm
Drive PCB Weight	20 g

*Model with factory installed encoder

Table 2–LRMO Series Linear Piezo Actuator

5. Mechanical Drawings of LRMO Series Linear Piezo actuator

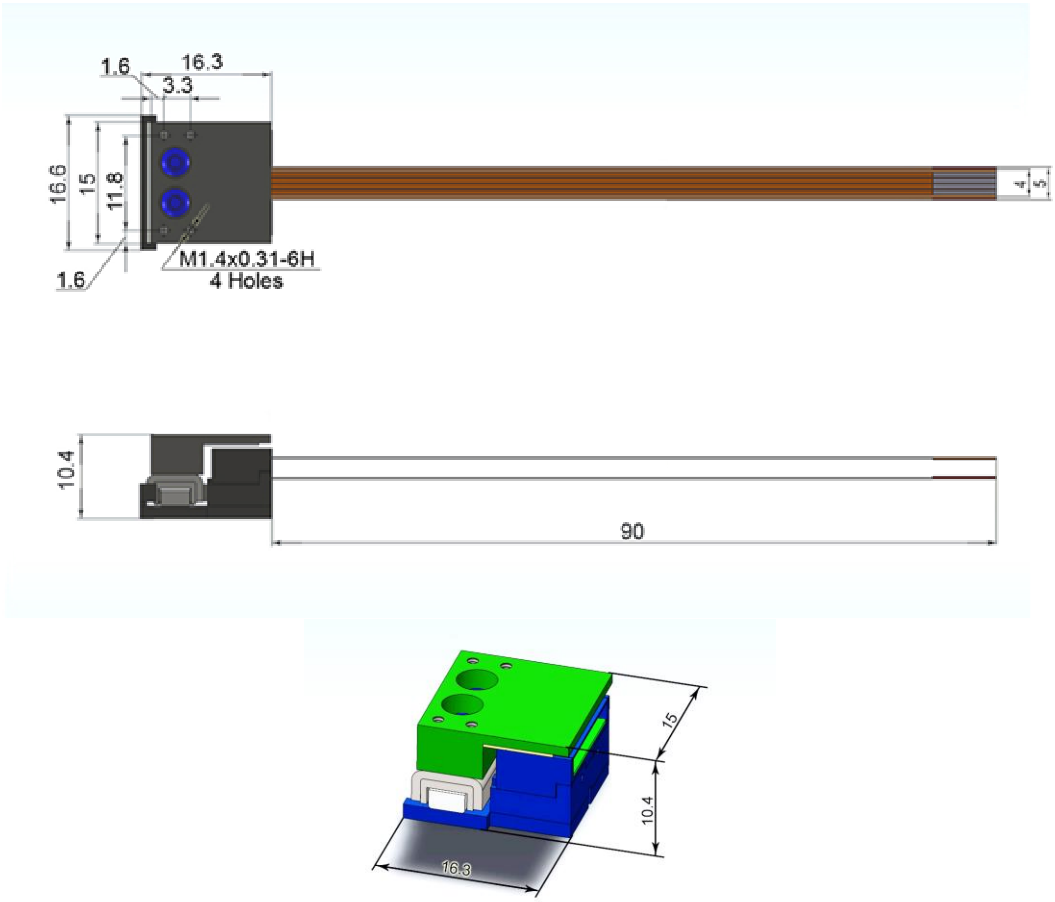


Figure 3. LRMO piezo actuator with factory-fitted encoder. Dimensions (mm)

6. Motion Control Closed-Loop (Feedback Control) using Piezo Motion Python API

In closed-loop control (feedback control) mode, output from the motor's optical encoder is fed to the encoder board and used to close the loop. The position and speed of the motor can then be controlled through an elaborate set of commands via the MicroUSB port. Two output signals from the encoder (channel A and channel B, with a phase difference of 90°) can be monitored on the pins of the Encoder Output connector.

7. Operation and Control of LRMO Series Rotary Piezo Actuator

Connecting the Power Supply

Power to the electronic driver board can be supplied using either of the following three methods:

i. Connect the Micro USB connector located on the electronic driver PCB to a suitable USB port on the computer, Figure 4.

ii. If you wish to power the unit from your own system instead of through the Micro USB connector, you can use the J1 connector, Figure 5. Apply +5V (Vdd) to pin 1 or 2, with pin 3 serving as the ground (GND).

Mounting and Connecting the Driver Board PCB

Connect the piezo actuator to the driver board using the two flexible PCB cables, as illustrated in Figure 1. This cable fits into the actuator connector on the board according to Figure 1. To connect the cables, lift the 'locking clip' (motor connector on Figure 6), insert the cable and then lower the 'locking clip' to lock the cable in place. To remove the cable, lift the 'locking clip' and remove the cable.

Driver Board Operation

The electronic driver PCB is responsible for generating the signals needed to drive the piezo actuator resulting in linear movement, Figure 4. S1 and S2 buttons are used for manual motor control; S1 causes movement in Left direction, and S2 enables movement in Right direction.

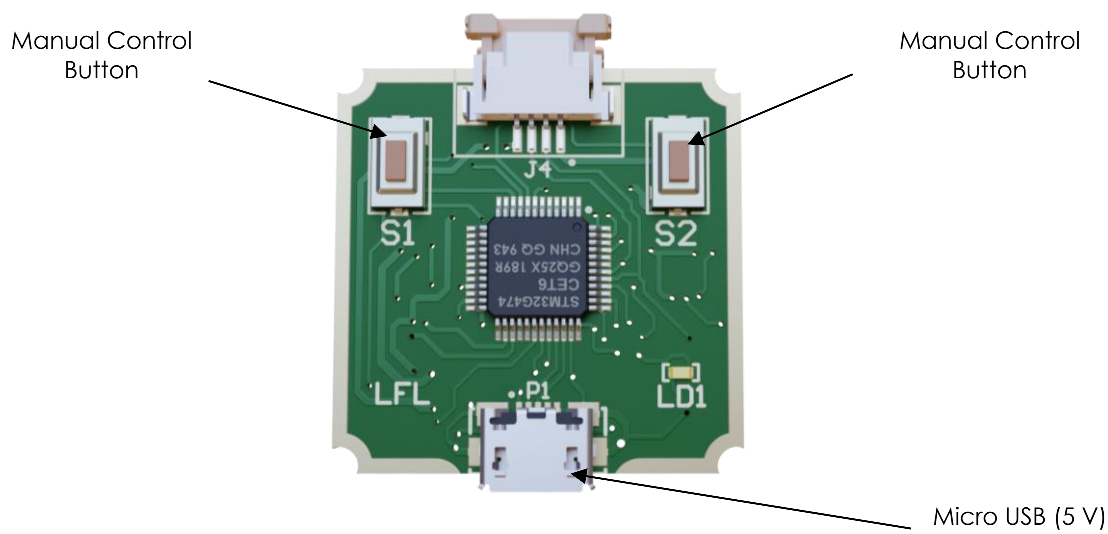


Figure 4. Driver board top view

8. Setting Up the Piezomotor Close Loop system

Connecting the Driver Board and Encoder

IMPORTANT – To Avoid Unnecessary Damage Please Connect the motor cable and encoder cable to the driver PCB BEFORE connecting the 5 VDC power.

Connect the two flexible PCB cables, attached to the encoder motor assembly, to the -PIN encoder connector and to the 3-PIN motor connector located on the driver board. To connect the cable, lift the 'locking clip', insert the cable and then lower the 'locking clip' to lock the cable in place. To remove the cable, lift the 'locking clip' and remove the cable.

9. Python Library Description

The LRMO closed-loop system provides a Python class interface for ease of control and integration. Below are the available methods and parameters:

Creating a Class Object:

PiezoMotor('serialPortName')

- Creates an instance of the PiezoMotor class.
    - Parameter ***serialPortName***: Specifies the name of the serial port through which the driver is connected to the control device.

Home(n)

Moves the motor to extreme left position and resets the position value to zero.

getPosition()

- Returns the motor position value in micrometers with an accuracy of one decimal point.

Velocity(value)

- Sets the motor velocity.
- Parameter:
    - *value*: Sets the motor velocity in mm/s
        - Range acceptable: 0.01 mm/s to 20 mm/s

If set to zero, the motor enters PWM mode without velocity stabilization. Configure PWM parameters using setPWMsettings.

Move(action)

Controls the motor's movement or stops it.

- Parameter:
    - action: Specifies the motor's action.

        - Valid values:
            - 'L', 'LEFT', 'Left': Move to the left
            - 'R', 'RIGHT', 'Right': Move to the right
            - Other values: Stop

Position(value)

Moves the motor to an absolute position in encoder pulses.

- Parameter:
    - value
        - 1 pulse = 2.6 microns

setPWMsettings(duty_cycle_percent, frequency_Hz)

Configures the PWM parameters for motor operation when the velocity is set to zero.

- Parameters:
    - *duty_cycle_percent*: Sets the duty cycle as a percentage of the PWM period.
        - Valid values: 0-100%
    - *frequency_Hz*: Sets the PWM frequency.
        - Valid values: 6 Hz to 2 kHz

10. Control of motor using Python environment

The closed loop LRMO-E motor can be controlled by using Python programs run on a PC or other computers, e.g. Raspberry Pi. To implement this the following steps need to be performed.

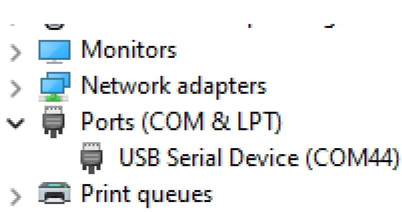
Install the software package provided with the motor, which contains the Python library. A view of the files contained in this package is shown below.

Name	Status	Date modified	Type	Size
Instruction Set LFL-E		12/23/2024 11:17 AM	Microsoft Word D...	19 KB
Library_description_LFL-E		12/23/2024 11:17 AM	Microsoft Word D...	15 KB
LibraryLFLE_fin		12/23/2024 11:17 AM	Python file	4 KB
UART_LFL-E_DEMO_Test		12/23/2024 11:17 AM	Python file	3 KB
UART_LFL-E_VELOCITY_ACCURACY_Test		12/23/2024 11:17 AM	Python file	1 KB
USB_LFL-E_DEMO_Test		12/23/2024 11:17 AM	Python file	3 KB
USB_LFL-E_VELOCITY_ACCURACY_Test		12/31/2024 2:18 PM	Python file	1 KB

Install a Python IDE on your computer – for example a Pyton IDE for beginners (Thonny) can be installed from this page - thonny.org.
Connect the two flexible cables of the motor to the blue driver board as described earlier.
Use the enclosed “USB to Micro USB adapter Cable” to connect one of the USB ports of your computer to the Micro USB port on the blue driver board, Figure 4. 5V power is automatically supplied to the driver board from the USB port.
Run Thonny and open one of the files in the folder, e.g. “USB_RFL-E_VELOCITY_ACCURACY_Test”. The following window will open:

The top window displays the program, which has been opened. The bottom window - “Shell” displays the results of running the program.

Run the “Device manager” of your computer and note the COM port, which is assigned to the driver board, see below:



Change the COM port number displayed on line "5" of the program with the assigned port from the" Device Manager". Run the program by pressing the green "Run" button and examine the results in the "Shell" window. The maximum rotational speed and the positioning error are displayed.

You can modify this program or write your own using the available commands, see "Instruction Set for ROMO-E". Please note that all new programs need to be saved and started from the folder containing the Python library in order for the program to work properly.

11. Serial Communication Protocol via UART

The ROMO closed-loop system uses a serial communication protocol with Little Endian byte order and provides commands for controlling the motors' behavior using UART commands through the pins TX, RX and GND of J2 connector (see Figure 5 – driver PCB & Figure 6 driver PCB pin-outs).

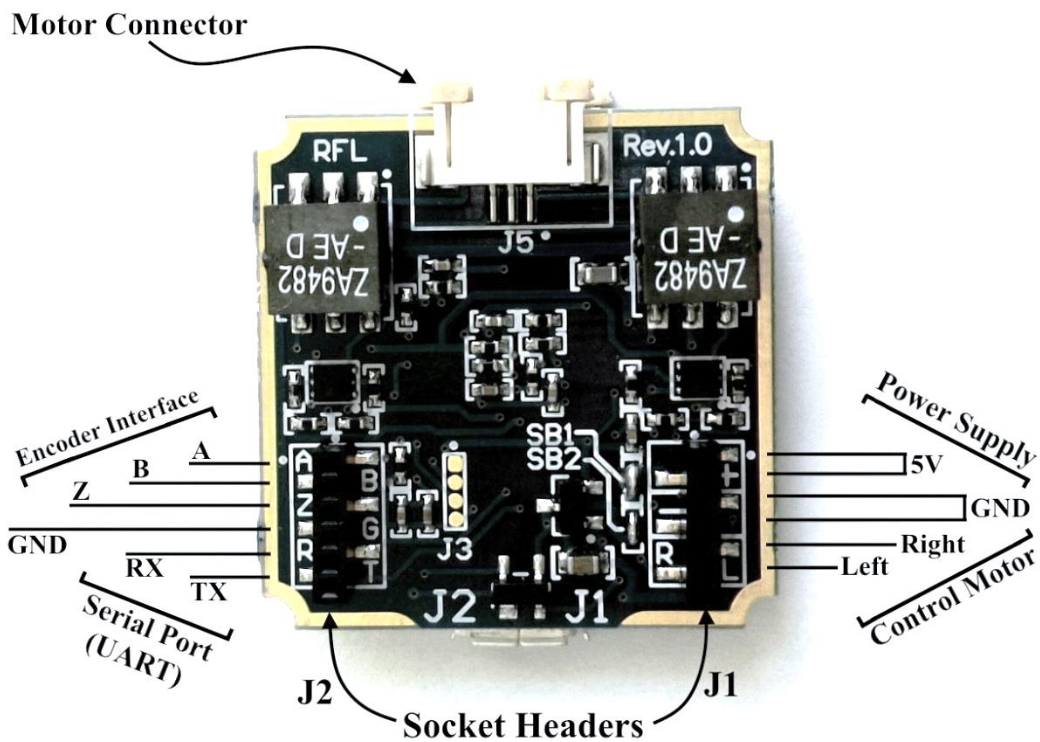


Figure 5 – Green Driver PCB

Pin No.	J1	J2
1	5V	A
2	5V	B
3	GND	Z (for RLF only)
4	GND	GND
5	Right	RX
6	Left	TX

Figure 6 Driver PCB pin-outs

Serial Port Configuration

To communicate with the motion control product, configure your serial port as follows:

- Baud rate: 115200
- Data bits: 8
- Parity: None
- Stop bits: 1
- Byte order: Little Endian

Instruction Set

The following commands are supported:

Home(direction)

Moves the motor to zero position.

Command code: 0
Size: 1 byte

- Parameter *direction*: Specifies the direction of movement.
  - Size: 1 byte
  - Valid values for *direction*:
    - 1: Counterclockwise
    - Other values: Clockwise

getPosition()

Returns the motor position value in encoder pulses. Each encoder pulse corresponds to 0.3515625 degrees (encoder has 1024 pulses per revolution).

- Command code: 1

Size: 1 byte

Returned size: 2 bytes (signed value)

Velocity(value)

Sets the motor velocity.

- Command code: 2

Size: 1 byte
Parameter *value*: Sets the motor velocity. Use the formula:

Value = 10 × Desired velocity (RPM)

- Size: 2 bytes
- Acceptable Range: 0.1 RPM to 600 RPM

If velocity is set to zero, the motor enters PWM mode without velocity stabilization. PWM parameters are configured using the *setPWMsettings* command.

Move(action)

Controls the motor's movement or stops it.

- Command code: 3
- Size: 1 byte
    - Parameter *action*: Specifies the motor's action.

Size: 1 byte
Valid values:

- 1: Counterclockwise movement
- 2: Clockwise movement
- Other values: Stop

Position(value)

Moves the motor to an absolute position in encoder pulses.

- Command code: 4
- Size: 1 byte
    - Parameter *value*: Specifies the absolute position in encoder pulses to which motor should move.
        - Size: 2 bytes (signed value)
        - Range: -32768 to 32767
        - 1 pulse = 0.3515625 degrees

When rotating clockwise, the position value increases; when counterclockwise, it decreases.

setPWMsettings(num_periods, run_periods)

Configures the PWM parameters for motor operation when the *velocity* is set to zero.

- Command code: 5
- Size: 1 byte

- Parameters *num_periods*: sets the number of periods of excitation of the motor within the PWM period.

    - Size: 2 bytes
    - Valid values: 0 to 65535
    - Time estimate (seconds) that corresponds to the specified number of periods of motor excitation use:

      Period Time = num_periods_in_PWM_period / Fgen

Where, Fgen - motor resonator excitation frequency (in Hz), changes during motor operation, the average value is approximately equal 334 kHz.

- Parameter *run_periods*: Sets the number of periods of excitation of motor run within the PWM period that the motor will run.

    - Size: 2 bytes
    - Valid values: 0 to 65535

To estimate run time (seconds) that corresponds to the specified number of periods of motor excitation, use:

Run Time = run_periods / Fgen

Response Behavior

Successful execution of a command returns a confirmation value of 1 (size: 1 byte), except for the getPosition command, which returns the current position of the motor.

12. Recommended settings to avoid overheating.

LRMO Series piezo actuators are designed for precise control applications using a duty cycle. They are not designed for prolonged operation in Continuous (non-stepping) Mode, which can lead to overheating of the actuator and possible internal damage not protected under warranty.

To avoid overheating of the actuator please follow the guidelines in the table below and ensure that motion control settings for Continuous Mode and/or Stepping (PWM) Mode are within the limits specified in the table below.

For applications requirements exceeding the recommended guideline, please contact our Technical Support.

Model #	Linear Speed (mm/s)	Recommended PWM Duty Cycle	Maximum Duration in Continuous Mode
LRMO-E-CL	>100 mm/s	<20%	10 s

Table 4–Recommended settings to avoid overheating.

13. Technical Support

For technical support please contact: info@piezomotors.com

Piezo Motor Company, LLC
Boca Raton
Florida 33496
USA

www.piezomotors.com

