



Scan for more information

PA-12

Position Control
Communication Protocol

Table of Contents

| | |
|------------------------------------------------|-----------|
| 1. Before Use | 3 |
| Introduction | 3 |
| Safety | 3 |
| Storage | 4 |
| Important Note (Overload protection/Force-Off) | 5 |
| 2. Basic Info | 8 |
| Components | 8 |
| Dimensions | 8 |
| 3. Application | 9 |
| 4. Actuator Control | 10 |
| Circuit Connection | 10 |
| Communication | 12 |
| Specification | 12 |
| Packet Description | 14 |
| Data Map | 17 |
| Data Description | 18 |
| Command Example | 24 |
| 5. Optional Accessories | 30 |
| Metal Bracket | 30 |
| PC USB Interface | 31 |
| Arduino Actuator Tester Shield | 32 |
| 6. Warranty | 33 |

Have any queries? Our expert engineers are here to help!



sales@progressiveautomations.com



1-800-676-6123



progressiveautomations.com

1. Before Use

1.1 Introduction

Please carefully read through this manual before using our product for proper operation and to prevent any unexpected damage.

Progressive Automations' Micro Precision Actuators have been developed to provide reliable, high-quality linear solutions in a compact size. Our Micro Precision Actuators can be applied in various fields such as factory automation, medical devices, robotics, professional UAV, and radio control applications.

Features:

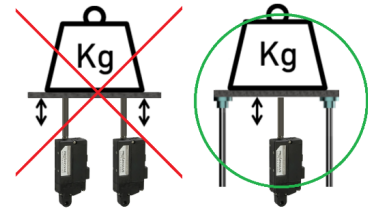
- Position Control (Positional Accuracy 50~90um – see spec chart of each model)
- Embedded Drive circuit
- 4096 Step High Resolution
- High-Performance Coreless Motor
- Minimal Mechanical Backlash (30um)
- Excellent Substitute for pneumatic cylinders which do not support position control

1.2. Safety

Please read our safety instructions below for operating our Micro Precision Actuators. Warranty will be void if products have been abused, altered, or tampered with in any way.

1. **DO NOT press the rod when the actuator is being operated.** Constantly applying a load that is higher than the rated force capacity may damage and burn out the motor.
2. **Apply proper input voltage.** Use a power supply or battery that has an output voltage within the proper range. For instance, apply 4.0~7.4V for 7.4V input products and 7.0~12V for 12V input products. The motor may burn out when 12V is applied to our 7.4V input products.
3. **Product lifespan may vary based on the load applied, duty cycle, etc.**
 - 1) **Usage below rated force capacity.** For instance, the rated force of our PA-12-10645012R is 4.50lbs. Using the PA-12-10645012R in applications that experience force conditions below 4.50lbs will maximize the product lifespan. The lower the applied load, the longer the motor lifespan.
 - 2) **Usage below 50% Duty Cycle rating:** If the motor operates continuously without any interval (rest), the motor will be overloaded and the overload protection feature will cut off power to the actuator motor. Therefore, it is important to consider the 50% duty cycle when using in your application. For example, a 20-minute period with a 50% duty cycle means that the motor must be off for, at least, 10 minutes and on for, no more than, 10 minutes to ensure it does not overheat. Usage below a 50% duty cycle will optimize lifespan. The lower the duty cycle, the longer the lifespan.
4. **Make sure not to damage the actuator motor case.** Installing your own brackets to fix the actuator on your system may risk dissipation and cause overheating.
5. **Set position commands within the mechanical limit:** The application should allow the actuator rod to have a full mechanical range of motion to reach the stroke limits without running into physical obstructions. Make sure the positional commands are made within the application's mechanical limit. If the positional command is out of the application's mechanical limit, the actuator will be overloaded at certain points in time and power will be cut off to protect the actuator due to the overload protection feature. If overload protection is inactivated by the user, the motor will not be protected. When considering precise position control, make sure to double-check this step before fully integrating the actuator.

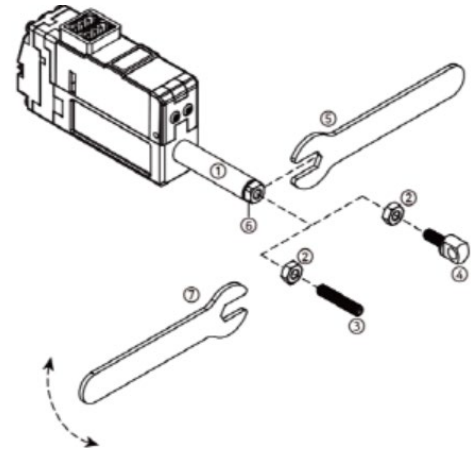
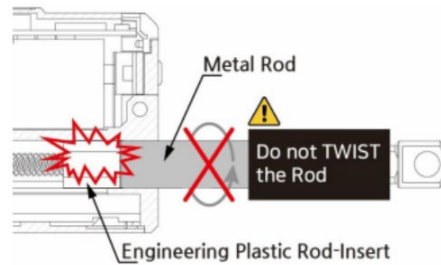
6. It is strictly banned to use multiple actuators for moving a single objective. Due to DC motor characteristics, each actuator's speed can vary slightly even if they are the same model and have the same goal position (this may overload one of the actuators).



7. Do not TWIST the rod with excessive force when tightening the rod end tip.

If you apply an excessive twisting force to the rod when tightening the rod end tip (while the body is fixed), it may cause damage to the internal part (plastic rod-insert). Please follow the instructions below to avoid damage.

- 1) Fix the rod-end with the included M3 spanner (⑤). This is to prevent the rod (①) from turning by mistake while tightening the M3 nut (②).



- 2) According to your preference, install the socket set screw (③) or rod end tip (④) to the proper position before tightening.

- 3) Adjust the angle of the rod end tip (④) to the desired angle. By using a long nose plier (or extra M3 spanner), fix the position by tightening the M3 nut (②) while the rod end is fixed with the M3 spanner (see above image). This is called the "double nut" concept that fixes a mechanical position by friction.

8. Use the "Overload Protection" feature to protect the actuator and your system from damage. The overload protection feature is activated from the factory. For other protection settings, set the "Alarm Shutdown" feature according to your system's condition when necessary.

9. DO NOT touch the actuator case right after actuator operation. It might be hot.

10. Keep away from water, humidity, dust, and oil.

11. Do not use outdoors. PA-12 is only designed for indoor use.

12. Keep out of children's reach. Keep hands off the actuator motor during operation.

1.3. Storage

Extreme conditions may cause malfunction or damage to the product. We do not recommend use/storage in:

- Direct sunlight and high temperatures over 70°C or low temperatures below -20°C
- Environment with high humidity
- Environment with vibration
- Dusty environment
- Electromagnetic environment

Important Note : Constant load / Overload Protection / Force Off Function

The overload protection (Overload shutdown) feature is to prevent motor burnouts and overload conditions which greatly affect the service life. Please continue reading for the overload protection and use it accordingly.

About the terminology and mechanism of overload protection (Overload Shutdown)

- The overload protection feature will be activated when the motor exceeds a 50% duty cycle. When the cumulative operating time is over 30 seconds without rest, the system will shut down.
- The duty cycle is the ratio of time ON versus time OFF.
- A 50% duty cycle means that the actuator is only ON for 50% of the time.
- Motor operating time includes the time in which the motor rotates/moves normally as well as the time in which the motor draws current during stuck conditions without motor rotation.

Overload condition and shutdown case

For a better understanding, here are some examples of overload conditions where the shutdown will happen after 30 seconds of continuous operation.

- When the actuator is continuously operated for more than 30 seconds.
- When the goal position command is made outside the application's mechanical limit range.
- When a stuck condition is maintained for more than 30 seconds.
- When the duty cycle is more than 50%. The greater the duty cycle conditions, the faster the shutdown occurs, and the lower the duty cycle, the later the shutdown occurs.

Overload shutdown disable

The overload shutdown function protects the actuator under overload conditions. In certain applications, there is a need to operate under severe conditions, even if the overall mechanism protection is more important than the actuator, or even if the life of the actuator is shortened. For this case, the shutdown function can be disabled through Progressive Automations' manager software so that the shutdown does not work during overload conditions.

Recovery after overload shutdown

Since the communication line is still functioning after shutdown, it can be restored to the initial state by the "Restart" command or by reconnecting power. Be sure to remove the cause of overloading before restarting.

Exception and Cautions

Exception: - Overload shutdown does not work when the actual load is heavier than the rated power even though it is an overloaded environment. To manage the actuator lifespan, please do not apply loads heavier than the rated force capacity.

Caution 1) When the spring is installed between the application and the actuator, or it is installed in the Z-axis (the direction of gravity), it applies an external force to the actuator. Under these conditions, the actuator operates slightly but continues to hold its position. If this condition persists, the actuator may invoke an Overload Shutdown in some cases. To prevent this, use the Force-Off command while external power is applied.

Caution 2) Each time the overload shutdown is executed, the cause of overload may eventually lead to motor damage. Therefore, the cause of overload must be removed after the first overload shutdown so future shutdowns do not occur.

Overload Protection Function

Using the overload protection function can protect against motor damage from an overload condition. The overload protection function is activated from the factory. When it is activated, motor power will be cut off in the case of overload condition to protect the actuator.

Another method to set overload protection is to use a command packet. You can send the “Store Data” command to the address (0X12) which is the address for Alarm Shutdown. Set bit 5 (see below) for overload error to “1” (overload activation) and send the “Store Data” command to the actuator motor. Actuator force will be cut off under overload conditions if the bit is set at “1”. (1=overload protection ON / 0=OFF)

| Error | bit |
|---------------------|-----|
| NONE | 7 |
| NONE | 6 |
| Overload Error | 5 |
| NONE | 4 |
| NONE | 3 |
| NONE | 2 |
| NONE | 1 |
| Input Voltage Error | 0 |

Refer to the example below for the “Store Data” command.

| HEADER | ID | Size | Command | Factor #1 Address | Factor #2 Data | Checksum |
|------------|------|------|---------|----------------------|-------------------|----------|
| 0xFFFFFFFF | 0x00 | 0x04 | 0xF3 | 0x12 | 0x20 | 0xD6 |



Caution

USE WITHIN RATED LOAD

For proper performance and better lifespan, we recommend using the actuator within its rated load range.

Force-OFF Function

- After the actuator moves to the designated position, the operation stops unless there is an external force that causes the position value to change. If the position value of the actuator is continuously changing due to vibration or external force, the actuator will be continuously operating without rest to hold the designated position value, this, in turn, affects the lifespan of the actuator.
- If the Force-OFF function is properly utilized, the motor power can be cut off to allow the motor to rest while maintaining its position with mechanical friction (mechanical self-lock function). That can be useful even when it is necessary to hold the position for a relatively long time after reaching a certain position. Shutting off the power of the motor using the Force-OFF parameter as an added safety function helps to manage the lifespan of the actuator.
- Under Force-OFF conditions, communication is still alive while motor power is off, so the actuator will move again when it gets a new position command without giving the “Force-ON” command.
- The Force-OFF feature can be available for the actuators having the “mechanical self-lock” feature. Please see the chart below to see availability. (The chart shows self-lock force of 1.06” stroke PA-12-R only. Refer to the separate actuator specifications for the self-lock force of 2.20” stroke models.)

| Rated Load | Mechanical Self-Lock |
|------------|----------------------|
| 4.5lbs | Not Available |
| 22.48lbs | Available |

For Force-OFF, send 0x00 as a “Store Data” command to the address (0X80) which is the address for Force ON/OFF. (For Force-ON, send 0x01)

Refer to the example below for the “Store Data” command.

| HEADER | ID | Size | Command | Factor #1 | Factor #2 | Checksum |
|------------|------|------|---------|-----------|-----------|----------|
| | | | | Address | Data | |
| 0xFFFFFFFF | 0x00 | 0x04 | 0xF3 | 0x80 | 0x00 | 0x88 |

TIP

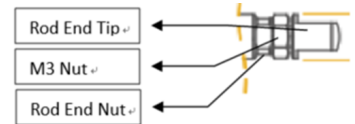
Under Force-OFF status, if user sends the "Goal Position" command, it is not necessary to send Force-ON packet additionally because the "Goal Position" command already includes the "Force-ON" packet in it.

2. Basic Information

2.1 Components



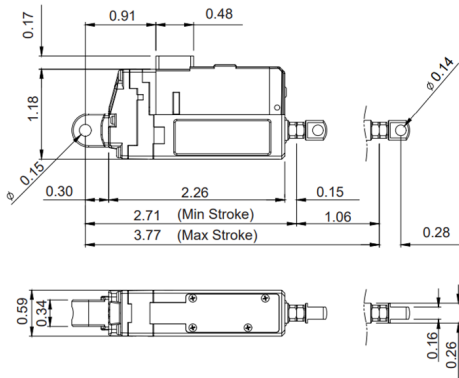
#3 M3 nut can be used to fix the hinge and hinge base. Also, the M3 nut can be used between the rod-end nut and the rod-end tip as a stopper.



2.2 Dimensions

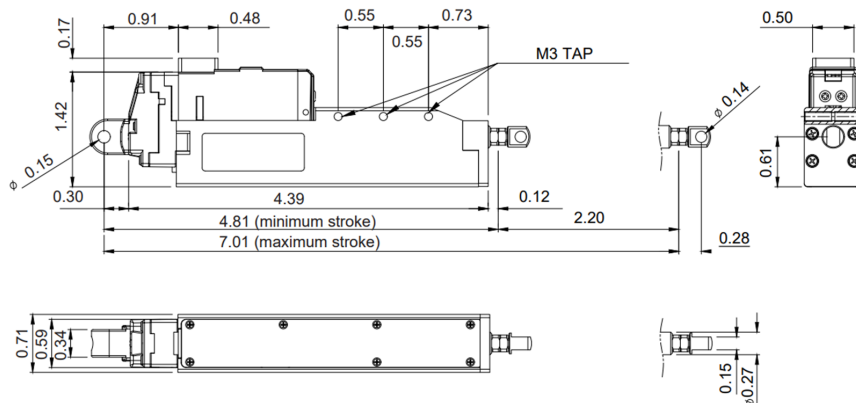
Please refer to the detailed dimensions under the 3D drawing on our website. (progressiveautomations.com → Resources → 3D Models)

[1.06" Model]



**** For 1.18" stroke actuator, the factory default stroke range is set at 1.06". Users may extend it to 1.18" using our Actuator manager software if necessary. (For better mechanical stability, 1.06" is recommended.)**

[2.20 " Model]



3. Application



Factory Automation

- Replacement for Pneumatic Cylinders
- Real-Time Automatic Width Adjustment Conveyer
- Real-Time Automatic Product Alignment (Up/Down or Left/Right)
- Automatic Valve Control (Oil or Water)
- Automatic Dispensing with Syringe
- Automatic Clamping System
- Fitting or Adjusting Distance
- Pick & Place
- In & out / Extension & Retraction
- On-Off Controls
- Changing Directions of Hexapod/Tripod Movement



Production & Test JIGs

- Hole-Punching Jig
- Hole Inspection Jig
- Switch Inspection Jig
- Touch Panel Inspection Jig
- PC Board Testing Jig



Robotics

- Robot Joints
- Robotic Grippers
- Linear Controls of Surgical Robot



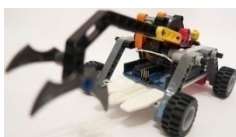
UAV / Professional Drone

- Fixed-Wing (Aileron/Elevator/Throttle/Flap/Air Brake/ Rudder/ Throttle)
- Helicopter (Swash Plate Control/Rudder)
- Multi-copter (Retract, Dropping Device)
- Linear Control Parts for Military products
- Pan/Tilt Camera Control



Medical / Lab Equipment

- Linear Position Control for Medical Devices (HIFU, etc.)
- Camera or Laser Focusing Control
- Laboratory Test Equipment



Education / Hobby

- 3D Printer
- Arduino or Raspberry Pi Control
- Maker's DIY Project

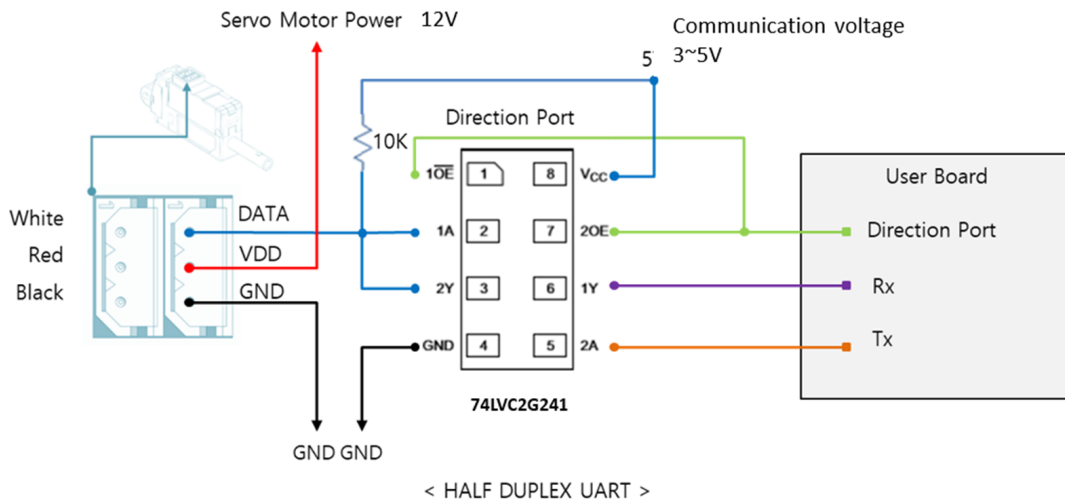
4. Actuator Control

4.1. Circuit Connection

The data structure for communication with the PA-12 on-board microcontroller is half-duplex UART. A full-duplex communication system allows both devices to transmit and receive data simultaneously. In the case of PA-12, the system is half-duplex, also known as semi-duplex. This means that the devices can communicate with each other, but not simultaneously. At any time during communication, one device must transmit while the other one receives, and vice-versa.

For this reason, if you are trying to communicate with a PA-12 through a full-duplex serial COM device, you will need to use a buffer in between (we recommend implementing a 74LVC2G241 chip).

TTL/PWM - 3Pin Connector for PA-12-T



The direction of the data signal for Tx and Rx of TTL level will be determined according to the level of direction_port as below.

- If the level of "direction_port" is LOW: Rx will receive Data.
- If the level of "direction_port" is HIGH: Tx will transmit Data.

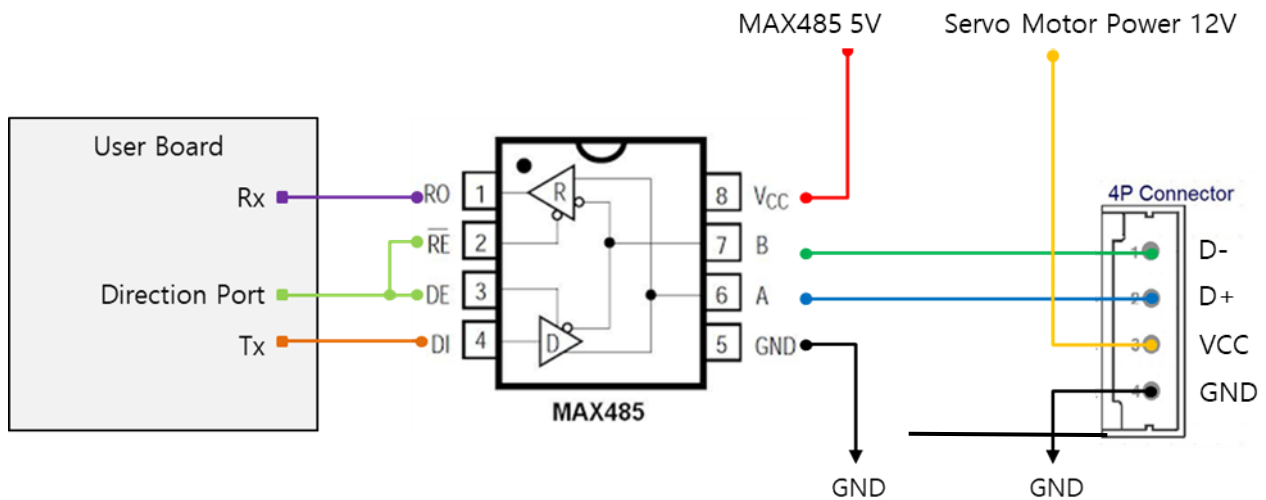
Both GNDs between actuator and controller should be connected as above diagram.

- For PWM control, connect PWM signal (3~5V) to the Data pin above.
- When using PWM communication, position control is possible, but feedback data, such as current position value, cannot be received, and serial daisy chain connection is not supported since data communication is not supported.
- Feedback data reception and daisy chain connections are possible with TTL or RS-485 communication.

RS-485 - 4Pin Connector for PA-12-R

Our PA-12-R uses RS-485 communication. For RS-485 type actuators, we recommend using the MAX485 chip as a buffer between a full-duplex controller and the half-duplex PA-12 device. Of course, communication devices that are themselves half-duplex would have no problem communicating with PA-12 directly. For example, the Allen-Bradley 1769-ASCII PLC module can communicate with the PA-12 directly.

| PIN NUMBER(COLOR) | PIN NAME | FUNCTION(RS485) |
|-------------------|----------|-----------------|
| 1(Yellow) | D- | RS485 - |
| 2(White) | D+ | RS485 + |
| 3(Red) | VCC | Power + |
| 4(Black) | GND | Power - |



If power is supplied from the outside, you can connect to 485 D+, 485 D- only.

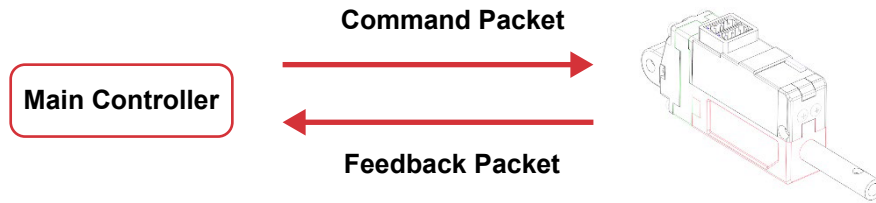
You can convert Tx and Rx mode by controlling "Direction_Port" pin in the circuit above.

- When the level of "direction_port" is LOW: Rx will receive Data.
- When the level of "direction_port" is HIGH: Tx will transmit Data.

Both GNDs between the actuator and controller should be connected as seen in the diagram.

4.2. Communication

The actuator and your main controller will communicate by exchanging data packets. These will be command packets (main controller to actuator) and Feedback packets (actuator to your main controller)



1. Specification

1) Communication specification

- **2 in One Mode** (Pulse / Data Mode Auto-Switching)

The PA-12-T will automatically recognize the input signal between data mode (TTL) and pulse mode (PWM).

- **Data Mode (TTL or RS-485)**

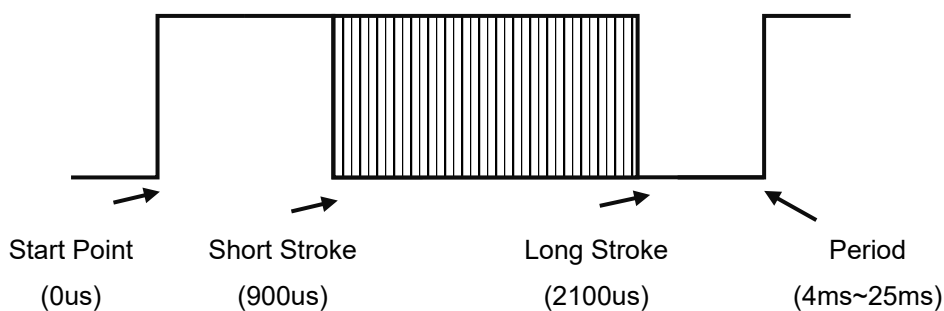
Asynchronous Serial communication (8 bit, 1 Stop bit, Non-Parity)

| Item | Spec |
|-----------|-------------------|
| Structure | Half-duplex UART |
| Baud Rate | 57600bps(default) |
| Data Size | 8bit |
| Parity | non-parity |
| Stop Bit | One bit |



Caution

- The actuator uses half-duplex communication and needs proper delay time to prevent communication errors.
- **The recommendable delay time is 5msec for data writing and 10msec for data reading.**
- Otherwise, there can be communication collision and motor failure.
- The delay time above is not the minimum; however, it is recommended for safety.



2) Data specification

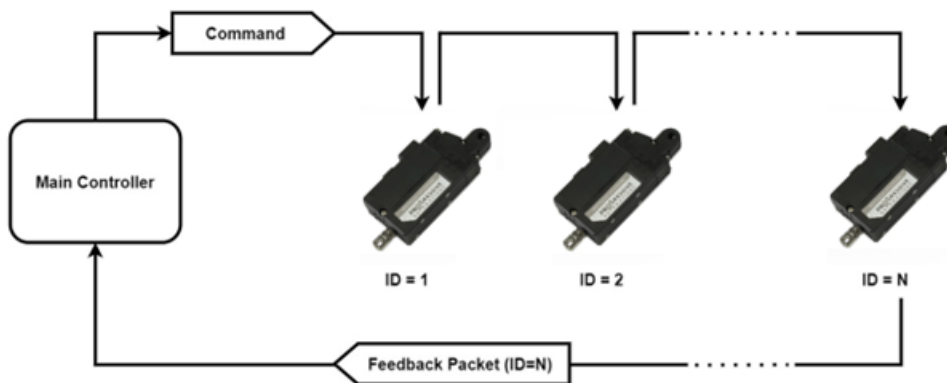
The data range is basically determined as seen below in both Data and Pulse modes.

| Rod Stroke | Data Mode (TTL/RS485) | Pulse Mode (PWM) |
|--------------------------------------|--------------------------|---------------------|
| Short Stroke (Max Retracting stroke) | 0 | 900us |
| Half Stroke | 2047 | 1500us |
| Long Stroke (Max Extending stroke) | 4095 | 2100us |

**The long-stroke limit for 1.18" stroke products is set to 1.06" from the factory and users are able to extend it to 1.18" if necessary. The data value of 1.06" is 3686. (It is recommended to use 1.06" for better mechanical stability related to lateral load)

3) Daisy-Chain Connection

When a Command Packet is sent to multiple actuators, only the actuator with ID=N will respond/operate. (Only ID=N actuator will send Feedback Packet and execute the command.)



- Factory default ID is 0 and the users need to assign ID on the Manager software or by API.
- The PWM mode does not support a daisy chain connection nor feedback data option.



Caution

Unique ID

- Each actuator must have an individual ID to prevent interference between the same IDs. Therefore, you need to set individual IDs for each actuator in the network node.
- Users may assign 253 different IDs and connect 253pcs actuators in serial via TTL protocol. For RS-485 protocol, 253 IDs can be assigned, but the available serial connection is up to 32 pcs of actuator motors due to RS-485 node regulation.
- As the factory default ID is 0, please assign different individual IDs for each actuator from 1~253 for daisy chain connection.

2. Packet Description

1) COMMAND PACKET

The structure and elements for Command Packets to operate your actuator are found below.

▪ Structure



▪ Element

| Index | Data | Description |
|-------|--------------|-----------------------------------------------------------------------------------|
| 0 | Start Byte 1 | Start Byte 1 (0xFF) |
| 1 | Start Byte 2 | Start Byte 2 (0xFF) |
| 2 | Start Byte 3 | Start Byte 3 (0xFF) |
| 3 | ID | Actuator ID (Range: 0 ~ 253, Broadcast ID: 254) |
| 4 | SIZE | Packet Size (COMMAND+FACTOR+CHECKSUM) |
| 5 | COMMAND | Instruction |
| 5+1 | FACTOR #1 | First Parameter |
| 5+m | FACTOR #m | m th Parameter |
| 5+N | FACTOR #N | Last Parameter |
| 5+N+1 | Check Sum | Check Sum = 0xFF - (LOWER_BYTE (ID + SIZE + COMMAND + FACTOR#1 + ... + FACTOR#N)) |

▪ Element Description

1. HEADER (3Byte)

- Data indicating the start of transmission: 0xFFFFFFFF

2. ID (1Byte)

- The ID is a unique number used to address each actuator in a daisy chain connection.
- Factory default ID value is 0.
- In case of ID = 0 ~253, only the actuator with the reference ID will be operated.
- In case of ID = 254 (0xFE), "Broadcasting Mode" will be activated which operates all actuators at once and Feedback Packet will not work.

3. SIZE (1Byte)

- Packet length in Byte unit.
- Data counting value after "Size" data (COMMAND+FACTOR+CHECKSUM).
- That is, Size value = Number of byte of "Factor" + 2

4. COMMAND (1Byte)

- Command codes define the purpose of a packet.

| Function | CODE | Description |
|-----------------|------|-----------------------------------------------------------|
| Echo | 0xF1 | Feedback Packet Reception |
| Load Data | 0xF2 | Send "Address" and get feedback of Data |
| Store Data | 0xF3 | Send "Address" and "Data". Then Save. |
| Send Data | 0xF4 | Send "Address" and "Data" for temporary storage |
| Execution | 0xF5 | Execute temporarily stored data that is made by SendData. |
| Factory Reset | 0xF6 | Reset to factory default parameter value |
| Restart | 0xF8 | Restart actuator system |
| Symmetric Store | 0x73 | Store data in the same address of multiple qty actuators. |

5. FACTOR

- Additional packet factor according to Command.

6. CHECKSUM

Verification data to check omission and any changes of packet data. The interaction formula will be:

- Checksum = BinaryInvert (LOWER_BYTE (ID + SIZE + COMMAND + FACTOR#1 + ... + FACTOR#N))
- LOWER_BYTE = Only the lower 1 byte is taken among the summed data values.
= Divide the summed data value by 0x100 and take only the remainder.
- Here is formula for above:

$$\text{LOWER_BYTE} (\text{ID} + \text{SIZE} + \text{COMMAND} + \text{FACTOR\#1} + \dots + \text{FACTOR\#N}) == (\text{ID} + \text{SIZE} + \text{COMMAND} + \text{FACTOR\#1} + \dots + \text{FACTOR\#N}) \% 0x100$$

2) FEEDBACK PACKET

After reception of Command Packet, the actuator sends Feedback Packets including the requested information. Its structure is as shown below.

▪ Structure



▪ Element

| Index | Data | Description |
|-------|--------------|---------------------------------------------------------------------------------|
| 0 | Start Byte 1 | Start Byte 1 (0xFF) |
| 1 | Start Byte 2 | Start Byte 2 (0xFF) |
| 2 | Start Byte 3 | Start Byte 3 (0xFF) |
| 3 | ID | Actuator ID (Range: 0 ~ 253, Broadcast ID: 254) |
| 4 | SIZE | Packet Size (COMMAND+FACTOR+CHECKSUM) |
| 5 | ERROR | Error Code |
| 5+1 | FACTOR #1 | First Parameter |
| 5+m | FACTOR #m | m th Parameter |
| 5+N | FACTOR #N | Last Parameter |
| 5+N+1 | Check Sum | Check Sum = 0xFF - (LOWER_BYTE (ID + SIZE + ERROR + FACTOR#1 + ... + FACTOR#N)) |

- **Element Description**

- 1. HEADER (3Byte)**

- Data indicating the start of transmission: 0xFFFFFFFF

- 2. ID (1Byte)**

- Individual ID number for each actuator (1 ~253).

- 3. SIZE (1Byte)**

- Packet length in Byte unit.
- Data counting value after "Size" data (COMMAND+FACTOR+CHECKSUM).
- That is, Size value = Number of byte of "Factor" + 2

- 4. HARDWARE ERROR (1Byte)**

- Error status during operation for each bit

| Error | bit | Description | LED |
|---------------------|-----|-------------------------------------------------------------------------------------------------------------------------|-----------|
| None | 7 | TBD | None |
| None | 6 | TBD | None |
| Overload Error | 5 | In case the current load cannot be controlled with the designated maximum force, it will be set as "1". | Red Blink |
| None | 4 | TBD | None |
| None | 3 | TBD | None |
| None | 1 | TBD | None |
| Input Voltage Error | 0 | In case the input voltage is out of the operating voltage range designated in the control table, it will be set as "1". | Red On |

- 5. FACTOR**

- Additional Packet factor according to Feedback Data.

- 6. CHECKSUM**

Verification data to check omission and any changes of packet data. The interaction formula will be:

- Checksum = 0xFF - (LOWER_BYTE (ID + SIZE + ERROR + FACTOR#1 + ... + FACTOR#N))
- LOWER_BYTE = Only the lower 1 byte is taken among the summed data values.
= Divide the summed data value by 0x100 and take only the remainder.
- Here is the formula for above.

$$\text{LOWER_BYTE (ID + SIZE + ERROR+ FACTOR\#1 + ... + FACTOR\#N)} == (\text{ID + SIZE + ERROR+ FACTOR\#1 + ... + FACTOR\#N}) \% 0x100$$

3) DATA MAP

1. Data Memory Map

▪ Memory using data (Non-volatile)

- Data will be saved in a non-volatile memory which maintains data even after power loss.
- All data will be reset to the default value when factory Reset command is executed.

| Address | Name | Description | Access | Default |
|-----------|------------------------------|----------------------------------------|--------|------------|
| 0 (0x00) | Model Number (L) | Lower byte of Model Number | R | |
| 1 (0x01) | Model Number (H) | High byte of Model Number | R | |
| 2 (0x02) | Version of Firmware | Firmware version info | R | - |
| 3 (0x03) | ID | Actuator ID | RW | 0 (0x00) |
| 4 (0x04) | Baud Rate | Communication speed | RW | 32 (0x20) |
| 6 (0x06) | Short Stroke Limit (L) | Lower byte of Short Stroke Limit | RW | 0 (0x00) |
| 7 (0x07) | Short Stroke Limit (H) | High byte of Short Stroke Limit | RW | 0 (0x00) |
| 8 (0x08) | Long Stroke Limit (L) | Lower byte of Long Stroke Limit | RW | 102 (0x66) |
| 9 (0x09) | Long Stroke Limit (H) | High byte of Long Stroke Limit | RW | 14 (0x0E) |
| 12 (0x0C) | the Lowest Limit Voltage | Lowest Voltage Limit | R | Each SPEC |
| 13 (0x0D) | the Highest Limit Voltage | Highest Voltage Limit | RW | Each SPEC |
| 14 (0x0E) | Motor Operating Rate (L) | Lower byte of Motor Operating Rate | RW | 255 (0xFF) |
| 15 (0x0F) | Motor Operating Rate (H) | High byte of Motor Operating Rate | RW | 3 (0x03) |
| 16 (0x10) | Feedback Return Mode | Feedback Return Mode | RW | 1 (0x01) |
| 17 (0x11) | Alarm LED | Alarm LED Function | RW | 33 (0x21) |
| 18 (0x12) | Alarm Shutdown | Alarm Shutdown Function | RW | 33 (0x21) |
| 19 (0x13) | Start Compliance Margin | Start Compliance Margin | RW | Each SPEC |
| 20 (0x14) | End Compliance Margin | End Compliance Margin | RW | Each SPEC |
| 24 (0x18) | Calibration Short Stroke (L) | Lower byte of Calibration Short Stroke | R | 0 (0x00) |
| 25 (0x19) | Calibration Short Stroke (H) | High byte of Calibration Short Stroke | R | 0 (0x00) |
| 26 (0x1A) | Calibration Long Stroke (L) | Lower byte of Calibration Long Stroke | R | 255 (0xFF) |
| 27 (0x1B) | Calibration Long Stroke (H) | High byte of Calibration Long Stroke | R | 15 (0x0F) |
| 33 (0x21) | Acceleration Ratio | Moving Acceleration Ratio | RW | Each SPEC |
| 34 (0x22) | Deceleration Ratio | Moving Deceleration Ratio | RW | Each SPEC |
| 37 (0x25) | Position D Gain | Derivative Gain | RW | Each SPEC |
| 38 (0x26) | Position I Gain | Integral Gain | RW | Each SPEC |
| 39 (0x27) | Position P Gain | Proportional Gain | RW | Each SPEC |
| 46 (0x2E) | Min Position Calibration | Min Position Value Trim | RW | Each SPEC |
| 47 (0x2F) | Max Position Calibration | Max Position Value Trim | RW | Each SPEC |

2. Parameter Map

Parameter Using Data (Volatile)

- All data will be reset to default after power loss.

| Address | Name | Description | Access | Default |
|------------|----------------------------------|--------------------------------------------|--------|----------|
| 128 (0x80) | Force ON/OFF | Force On/Off | RW | 0 (0x00) |
| 129 (0x81) | LED | LED On/Off | RW | 0 (0x00) |
| 134(0x86) | Goal Position (L) | Lower byte of Goal Position Value | RW | - |
| 135 (0x87) | Goal Position (H) | High byte of Goal Position Value | RW | - |
| 140 (0x8C) | Present Position (L) | Lower byte of Present Position Value | R | - |
| 141(0x8D) | Present Position (H) | High byte of Present Position Value | R | - |
| 144 (0x90) | Present Motor Operating Rate (L) | Lower byte of Present Motor Operating Rate | R | - |
| 145 (0x91) | Present Motor Operating Rate (H) | High byte of Present Motor Operating Rate | R | - |
| 146 (0x92) | Present Voltage | Present Voltage | R | - |
| 150 (0x96) | Moving | Moving Status | R | 0 (0x00) |

4) DATA DESCRIPTION

Non-Volatile Memory

1. Model Number

Model number refers to PA's actuator model. For example, PA-12-10645012R.

2. Version of Firmware

Shows the current firmware version of the selected actuator.

3. ID [0~254 / Default: 0]

ID to discriminate each actuator. Different IDs should be assigned in Daisy-Chain system.

- In case of ID = 0 ~253, only the actuator with the reference ID will be operated.
- In case of ID = 254 (0xFE), "Broadcasting Mode" will be activated which operates all actuators at once and Feedback Packet will not work.

ID is a non-volatile memory area. If you change the data, communication may stop for a short time during saving process. Therefore, please be careful of frequent value changes during operation.

4. Baud Rate

- Refers to the communication speed. The default value is 57600bps.
- Actuator should always be rebooted if the baud rate is updated.

[Setting Value]

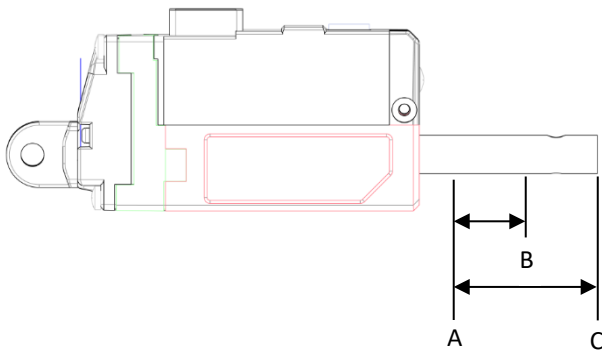
| Value | Baud Rate (bps) |
|------------------|-----------------|
| 16 (0x10) | 115200 |
| 32 (0x20) | 57600 |
| 48 (0x30) | 38600 |
| 64 (0x40) | 19200 |
| 128 (0x80) | 9600 |

Baud Rate is a non-volatile memory area. If you change the data, communication may stop for a short time during the saving process. Therefore, please be careful of frequent value changes during operation.

38600bps is not available in older firmware, such as version 1.5 or lower.

5. Stroke Limit [0~4095]

Stroke limit refers to the maximum and minimum value of the Goal Position between Short Stroke (A) and Long Stroke (C). If the Goal Position value is smaller than the Short Stroke Limit value or greater than the Long Stroke Limit value, the Goal Position value is replaced with the Stroke Limit value.



Non-volatile memory area. If you change the data, communication may stop for a short time during the saving process. Therefore, please be careful of frequent value changes during operation.

6. The Highest / Lowest Limit Voltage

Refers to the maximum and minimum values for the actuator input voltage (unit: 0.1V). Depending on the input voltage, the speed and force of an actuator may vary. For more details, please refer to the datasheet of each model.

| Input Volt | Lowest Voltage | Highest Voltage |
|-------------|----------------|-----------------|
| 7.4V Models | 6 [V] | 8.9[V] |
| 12V Models | 7[V] | 13[V] |

7. Motor Operating Rate [0~1023 / Default: 1023]

Motor Operating Rate refers to the maximum operating rate of the motor and the maximum PWM value supplied to the motor. If it is set to less than 400, the motor may not operate. Please note that changing the Motor Operating Rate also changes the speed and stall force.

Motor Operating Rate is a non-volatile memory area. If you change the data, communication may stop for a short time during the saving process. Therefore, please be careful of frequent value changes during operation.

8. Feedback Return Mode

After receiving a Command Packet, the actuator will be in the Feedback Packet Return mode.

| Mode | Feedback Packet Return or NOT |
|------|-----------------------------------------------------------------------|
| 0 | DO NOT send FeedbackPacket for all Commands (except for Echo command) |
| 1 | Sending FeedbackPacket only for Load Data Command. |
| 2 | Sending FeedbackPacket for all Commands. |

TIP

Under Broadcast ID (0xFE) mode, Feedback Packets will NOT be sent regardless of the Feedback Return Mode values.

9. Alarm LED

If the concerned bit is set as "1" when an error occurs, the error LED indication will be activated. (1: activate, 0: deactivate)

| Error | bit | LED Indicate |
|---------------------|-----|--------------|
| Overload Error | 5 | Red Blink |
| Input Voltage Error | 0 | Red |

In case of an Input Voltage Error, the alarm is immediately cleared when the error is resolved.

In case of an Overload Error, the alarm will not be cleared even after the overload condition is resolved. Rebooting the power or restarting the system can clear an Overload Error.

Alarm LED is a non-volatile memory area. If you change the data, communication may stop for a short time during saving process. Therefore, please be careful of frequent value changes during operation.

10. Alarm Shutdown

Force will be OFF if the concerned bit is set as "1" when an error occurs. (1: activate, 0: deactivate)

| Error | bit |
|---------------------|-----|
| Overload Error | 5 |
| Input Voltage Error | 0 |

TIP

Overload Error / Input Voltage Error shutdown feature is activated from the factory to protect the motor. These can be disabled by the user using our PA-12 manager software, but please consider carefully.

Alarm Shutdown is a non-volatile memory area. If you change the data, communication may stop for a short time during the saving process. Therefore, please be careful of frequent value changes during operation.

11. Stroke Compliance Margin

Start Compliance Margin (Recommended margin value: 7)

- Refers to the minimum margin value for the actuator to start moving position.
- For example, if the compliance margin is 7 and the current position value is 400, the actuator will start moving when a positional value between 407 (400+7) and 393 (400-7) is set.
- Likewise, when a positional change occurs by more than +/-7 (out of 393~407) from the present position value due to physical external pressure or electrical noise, the motor will start moving to compensate for the position.
- For this reason, a larger value for compliance margin means more stable operation without jittering even in an environment where the external pressure, electrical noise, or clearance increases, however, the sensitivity to drive to the desired position may be reduced. Basically, increasing this value increases durability while reducing this value increases precision.
- This value must be equal to or greater than the "End compliance margin value" as shown below (setting it to a lower value may cause an error).

- Refers to the minimum margin value for the actuator to complete its movement.
- For example, if the actuator is instructed to move to a position value of 400, and assuming that it cannot physically stop at a position value of 400 exactly due to software & mechanical clearance, acceleration, etc. of the actuator, the End compliance margin will judge if the positional command was performed properly. If this value is set to 4 and the position command value is set to 400, the actuator will judge that the positional movement has been made properly when it reaches within the 396~404 range and then stops positional movement.
- If this value is increased for stable operation, you should not increase it beyond the "Start Compliance Margin" value described above, and if this value is decreased too much to increase the accuracy, it may bring adverse effects such as jitter.
- The smaller the End Compliance Margin, the more sensitive and greater the positioning accuracy. However, if it is reduced below a certain value, the effect becomes insignificant.
- If the End Compliance Margin is increased, the operation becomes cleaner and more stable when the target position is reached. The faster the product, the better the effect. However, if it is too large, the precision may deteriorate.

Stroke Compliance Margin is a non-volatile memory area. If you change the data, communication may stop for a short time during the saving process. Therefore, please be careful of frequent value changes during operation.

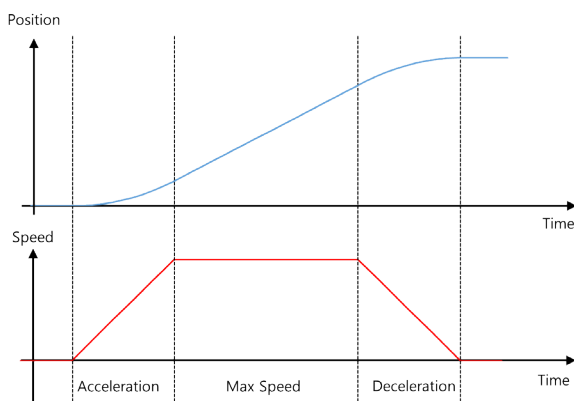
12. Calibration Stroke

- Calibration Short Stroke: Refers to the Short Stroke calibration value. The Short Stroke Calibration value that was set at the factory is saved by default.
- Calibration Long Stroke: Refers to the Long Stroke calibration value. The Long Stroke Calibration value that was set at the factory is saved by default.

13. Acceleration / Deceleration [0~255 / Default: Individual Spec]

Indicates the acceleration/deceleration rate of the motor.

- **Acceleration:** Refers to the acceleration value when the motor starts moving. If the value is high, the motor accelerates rapidly. Low values allow for smooth acceleration; however, the motor may not move at all if the value is too low.
- **Deceleration:** Refers to the deceleration value when the motor reaches the goal position. If deceleration is too high, the actuator may stop outside the goal position and result in oscillation between the deviating position and the goal position. In this case, normal braking may not be achieved.
- If the deceleration is too low, the actuator becomes too slow and the time to reach the goal position may be delayed.



It is a non-volatile memory area. If you change the data, communication may stop for a short time during saving process. Therefore, please be careful of frequent value changes during operation.

When correcting Acceleration/Deceleration, test after applying a small change.

14. Position PID

Refers to the PID value for motor position control. If a PID value is larger than the set value is applied, the motor may vibrate without stopping at the set position value due to an overshoot or over-response state and leading to a rough operation against the error with the goal position.

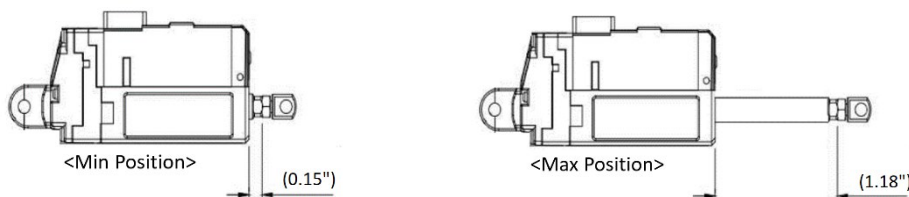
If a PID value is smaller than the set value is applied, it operates smoothly with the goal speed, however, the error with the goal speed value may appear large.

- Be careful about changing the value. Start by making small changes to the value first.

Position PID is a non-volatile memory area. If you change the data, communication may stop for a short time during the saving process. Therefore, please be careful of frequent value changes during operation.

15. Min/Max Position Calibration [0~255 / Default: Individual Spec]

- Min Position: Refers to the position of the minimum stroke when the Goal Position value is '0'.
- Max Position: Refers to the maximum stroke position when the Goal Position value is '4095'.
- Unlike the Stroke Limit command, the range of the Goal Position value [0~4095] is not limited and the length of the used stroke gets changed.
- Min/Max Position value of each actuator has a mechanical tolerance of $\pm 0.5\text{mm}$
So, Position Calibration Command can be used to synchronize the start and end positions of slightly different actuators for the same Goal Position value by correcting each error value.



| Parameter | Goal Position Range | Remark |
|----------------------|----------------------------------------|-------------------------------------------------------------------------------|
| Stroke Limit | Short Stroke Limit ~ Long Stroke Limit | No limit on the available stroke range, only limit G/P input range (0~4095) |
| Position Calibration | 0 ~ 4095 (Full range) | Input range (0~4095) is not limited, but the available stroke range may vary. |

- For example, if the Min Position Calibration value at Min Position 0.15" of PA-12-10645012R is 5, increasing the Min Position Calibration value increases the Min Position value and the entire stroke range will be reduced as the Min Position increases.

It is a non-volatile memory area. If you change the data, communication may stop for a short time during saving process. Therefore, please be careful of frequent value changes during operation.

Volatile Memory

16. Force ON/OFF

Refers to the setting for Force ON and OFF (0: OFF, 1: ON)

| Value | Description |
|-------|----------------------------------------------------|
| 0 | Cut off power to the motor and Force is OFF. |
| 1 | Power to be supplied to the motor and Force is ON. |

TIP

The mechanical design of our PA-12 allows it to keep its position even after motor power loss. For instance, the PA-12 experiencing 9lbs force against the rod will stick to its position firmly after motor power is off.

So, in case the actuator needs to keep a certain position (if mechanical frictional force is able to keep its position under power-off condition against your load), apply the FORCE OFF parameter. In this case, the communication line is still alive while the motor power is off to help the actuator have a longer lifespan. Upon receiving a new positional command, the actuator will have FORCE on and then proceed with its next movement.

17. LED

Refers to the control LED when there is no Error indication.

| bit | Description |
|-----|--------------------------------------------------|
| 0 | LED Disable (All LEDs will be Off when it is 1.) |
| 1 | RED LED Control |
| 2 | GREEN LED Control |
| 3 | BLUE LED Control |

18. Goal Position

- Goal position is a value that represents the desired positional value to move to. The goal position value will be affected by both the short and long stroke limits. (i.e., if the position command is out of the stroke limit range, it will only move to the stroke limit position)
- For the 1.06" stroke models, the goal position value at 1.06" is 3686 due to the long stroke limit setting. It can be extended to 1.18" (4095) if desired.

19. Present Position [0~4095]

- Refers to the current position value that is being monitored.
- Range is between 0~4095 and the motor can normally readjust even after stopping at the desired position.

20. Present Motor Operating Rate [0~2047]

- Refers to the Current Motor Operating Rate that is being monitored. This can be affected by goal current, goal speed, acceleration/deceleration adjustment.
- Range is between 0~2047.
- The value 0 indicates the motor has stopped.
- The Motor Operating Rate on short stroke direction (retracting) is between 1~1023.
- The Motor Operating Rate on long stroke direction (extending) is between 1024~2047.

21. Present Voltage

- Refers to the current input voltage. Unit is in 0.1V (For example, 74 means 7.4V).

22. Moving

- Refers to the moving status of the motor

| Value | Description |
|-------|-----------------|
| 0 | Motor Stops |
| 1 | Motor Operating |

5) COMMAND EXAMPLE PACKET

1) Echo

Refers to the receiving of Feedback Packets (refer to the "Load Data" parameter for feedback on position values, etc.)

Command Packet

| HEADER | ID | Size | Command | Checksum |
|------------|------|------|---------|----------|
| 0xFFFFFFFF | 0x00 | 0x02 | 0xF1 | 0x0C |

- This is the command packet for recognizing the status of the actuator connection.

| HEADER | ID | Size | Error | Checksum |
|------------|------|------|-------|----------|
| 0xFFFFFFFF | 0x00 | 0x02 | 0x00 | 0xFD |

Feedback Packet

- This is the feedback packet to inform the status of the actuator connection (including error information)

2) Factory Reset

Refers to resetting the factory default parameter values.

- Basic memory parameters will be reset to their default values.
- If the concerned bit is "1", it means Reset. If it is "0", it means Hold.

| Option | Bit | Description |
|-------------|-----|-------------------------|
| Actuator ID | 0 | Reset actuator ID to 0 |
| Baud rate | 1 | Reset to 32 (57600 bps) |

Command Packet

| HEADER | ID | Size | Command | Factor Option | Checksum |
|------------|------|------|---------|------------------|----------|
| 0xFFFFFFFF | 0x01 | 0x03 | 0xF6 | 0x01 | 0x04 |

- Actuator ID is to be reset to 0 while baud rate maintains its current status.

Feedback Packet

| HEADER | ID | Size | Error | Checksum |
|------------|------|------|-------|----------|
| 0xFFFFFFFF | 0x01 | 0x02 | 0x00 | 0xFC |

- This is the feedback packet to inform the receipt of the factory reset.

3) Restart

Refers to the actuator system restart

Command Packet

| HEADER | ID | Size | Command | Checksum |
|------------|------|------|---------|----------|
| 0xFFFFFFFF | 0x00 | 0x02 | 0xF8 | 0x05 |

- This is the command packet to reboot the actuator system.

Feedback Packet

| HEADER | ID | Size | Error | Checksum |
|------------|------|------|-------|----------|
| 0xFFFFFFFF | 0x00 | 0x02 | 0x00 | 0xFD |

- This is the feedback packet to inform the receipt of a restart command for the actuator system (respond in the Feedback Return Mode 2 only).

4) Store Data

Refers to the stored data after sending address and data to set ID, Goal Position, Force Limit, Stroke Limit, Speed, Force-On/Off, etc.

- **ID change: Change ID from 0 to 1 (0x01)**

| HEADER | ID | Size | Command | Factor #1 Address | Factor #2 Data | Checksum |
|------------|------|------|---------|----------------------|-------------------|----------|
| 0xFFFFFFFF | 0x00 | 0x04 | 0xF3 | 0x03 | 0x01 | 0x04 |

- ID: Refers to the actuator motor ID
- Command: Save data at the respective address in order
- Address: The address at which the actuator motor ID is saved (see (3) Data Map)
- Data: Desired Actuator ID (put 0x01 at address 0x03)

- **Goal Position command 1: Command packet to assign Goal Position to 2047 (0x07FF)**

| HEADER | ID | Size | Command | Factor #1 Address | Factor #2 Data #1 | Factor #3 Data #2 | Checksum |
|------------|------|------|---------|----------------------|----------------------|----------------------|----------|
| 0xFFFFFFFF | 0x01 | 0x05 | 0xF3 | 0x86 | 0xFF | 0x07 | 0x7A |

- ID: Refers to the actuator motor ID
- Command: Save data at the respective address in order
- Address: The address at which the goal position value is saved (see (3) Data Map)
- Data #1: Desired goal position's lower byte (address 0x86: 0xFF)
- Data #2: Desired goal position's upper byte (address 0x87: 0x07)
- Goal position value hex change (decimal to hexadecimal): 2047 → 0x07FF

- **Goal Position command 2: Command packet to assign Goal Position to 1000 (0x03E8)**

| HEADER | ID | Size | Command | Factor #1 Address | Factor #2 Data #1 | Factor #3 Data #2 | Checksum |
|------------|------|------|---------|----------------------|----------------------|----------------------|----------|
| 0xFFFFFFFF | 0x01 | 0x05 | 0xF3 | 0x86 | 0xE8 | 0x03 | 0x95 |

- ID: Refers to the actuator motor ID
- Command: Save data at the respective address in order
- Address: The address at which the goal position value is saved (see (3) Data Map)
- Data #1: Desired goal position's lower byte (address 0x86: 0xE8)
- Data #2: Desired goal position's upper byte (address 0x87: 0x03)
- Goal position value hex change (decimal to hexadecimal): 1000 → 0x03E8

- **Motor Operating Rate 1: Command packet to assign Motor Operating Rate to 512 (0x0200)**

| HEADER | ID | Size | Command | Factor #1 Address | Factor #2 Data #1 | Factor #3 Data #2 | Checksum |
|------------|------|------|---------|----------------------|----------------------|----------------------|----------|
| 0xFFFFFFFF | 0x01 | 0x05 | 0xF3 | 0x0E | 0x00 | 0x02 | 0xF6 |

- ID: Refers to the actuator motor ID
- Command: Save data at the respective address in order
- Address: The address at which the moving speed value is saved (see (3) Data Map)
- Data #1: Desired Motor Operating value's lower byte (address 0x0E: 0x00)
- Data #2: Desired Motor Operating value's upper byte (address 0x0F: 0x02)
- Motor Operating value hex change (decimal to hexadecimal): 512 → 0x0200

- **Motor Operating Rate 2: Command packet to assign Motor Operating Rate to 400 (0x0190)**

| HEADER | ID | Size | Command | Factor #1 Address | Factor #2 Data #1 | Factor #3 Data #2 | Checksum |
|------------|------|------|---------|----------------------|----------------------|----------------------|----------|
| 0xFFFFFFFF | 0x01 | 0x05 | 0xF3 | 0x0E | 0x90 | 0x01 | 0x67 |

- ID: Refers to the actuator motor ID
- Command: Save data at the respective address in order
- Address: The address at which the Motor Operating value is saved (see (3) Data Map)
- Data #1: Desired Motor Operating value's lower byte (address 0x0E: 0x90)
- Data #2: Desired Motor Operating value's upper byte (address 0x0F: 0x01)
- Motor Operating value hex change (decimal to hexadecimal): 400 → 0x0190

- **Stroke Limit 1: Command packet to assign Short Stroke Limit to 100 (0x0064)**

| HEADER | ID | Size | Command | Factor #1 Address | Factor #2 Data #1 | Factor #3 Data #2 | Checksum |
|------------|------|------|---------|----------------------|----------------------|----------------------|----------|
| 0xFFFFFFFF | 0x01 | 0x05 | 0xF3 | 0x06 | 0x64 | 0x00 | 0x9C |

- ID: Refers to the actuator motor ID
- Command: Save data at the respective address in order
- Address: The address at which the Short Stroke Limit value is saved (see (3) Data Map)
- Data #1: Desired Short Stroke Limit value's lower byte (address 0x06: 0x64)
- Data #2: Desired Short Stroke Limit value's upper byte (address 0x07: 0x00)
- Stroke limit value hex change (decimal to hexadecimal): 100 → 0x0064

- **Stroke Limit 2: Command packet to assign Long Stroke Limit to 3800 (0x0ED8)**

| HEADER | ID | Size | Command | Factor #1 Address | Factor #2 Data #1 | Factor #3 Data #2 | Checksum |
|------------|------|------|---------|----------------------|----------------------|----------------------|----------|
| 0xFFFFFFFF | 0x01 | 0x05 | 0xF3 | 0x08 | 0xD8 | 0x0E | 0x18 |

- ID: Actuator motor ID
- Command: Save data at the respective address in order
- Address: The address at which the Long Stroke Limit value is saved (see (3) Data Map)
- Data #1: Desired Long Stroke Limit value's lower byte (address 0x08: 0xD8)
- Data #2: Desired Long Stroke Limit value's upper byte (address 0x09: 0x0E)
- Stroke limit value hex change (decimal to hexadecimal): 3800 → 0x0ED8

- **Force On/Off : Command packet to cut off motor power while communication is alive.**

| HEADER | ID | Size | Command | Factor #1 Address | Factor #2 Data #1 | Checksum |
|------------|------|------|---------|----------------------|----------------------|----------|
| 0xFFFFFFFF | 0x01 | 0x04 | 0xF3 | 0x80 | 0x00 | 0x87 |

- ID: Refers to the actuator motor ID
- Command: Save data at the respective address in order.
- Address: The address at which the Force On/Off value is saved (see (3) Data Map)
- Data: Desired Force On/Off data byte (address 0x86: 0x00(Off)/ 0x01(On))
- After Force-Off, automatically Force-On when the next goal position command is made.

- **Feedback Return Mode 1: Command packet to send Feedback Packet for Load Data Command only.**

| HEADER | ID | Size | Command | Factor #1 | Factor #2 | Checksum |
|------------|------|------|---------|-----------|-----------|----------|
| | | | | Address | Data #1 | |
| 0xFFFFFFFF | 0x01 | 0x04 | 0xF3 | 0x10 | 0x01 | 0xF6 |

- ID: Refers to the actuator motor ID
- Command: Save data at the respective address in order
- Address: The address at which the Feedback Return Mode value is saved (see (3) Data Map)
- Data: Feedback Return Mode Data (address 0x10: 0x01)
(1: Send Feedback packet only to Load Data (0xF3) Command)

- **Feedback Return Mode 2: Command packet to send Feedback Packet for all commands**

| HEADER | ID | Size | Command | Factor #1 | Factor #2 | Checksum |
|------------|------|------|---------|-----------|-----------|----------|
| | | | | Address | Data #1 | |
| 0xFFFFFFFF | 0x01 | 0x04 | 0xF3 | 0x10 | 0x02 | 0xF5 |

- ID: Refers to the actuator motor ID
- Command: Save data at the respective address in order
- Address: The address at which the Feedback Return Mode value is saved (see (3) Data Map)
- Data: Feedback Return Mode Data (address 0x10: 0x02)
(2: Send Feedback packet to all)

5) Load Data

Send address and get data feedback

***Load data is data read. To avoid communication collision, at least 10ms communication delay is required when using 57600bps baud rate.**

- **Present Position: Command packet to read present position**

Command Packet

| HEADER | ID | Size | Command | Factor #1 | Factor #2 | Checksum |
|------------|------|------|---------|-----------|-----------|----------|
| | | | | Address | Length | |
| 0xFFFFFFFF | 0x00 | 0x04 | 0xF2 | 0x8C | 0x02 | 0x7B |

- ID: Refers to the actuator motor ID
- Command: Read bytes (equivalent to the length number) from the address
- Address: The address at which the present position value is saved (see (3) Data Map)
- Length: The number of bytes to read from the address (present position value consists of 2 bytes)

Feedback Packet

| HEADER | ID | Size | Error | Factor #1 | Factor #2 | Checksum |
|------------|------|------|-------|-----------|-----------|----------|
| 0xFFFFFFFF | 0x00 | 0x04 | 0x00 | 0xFF | 0x07 | 0xF5 |

- ID: Refers to the actuator motor ID
- Error: Refers to error indication during operation
- Factor 1: Present position value's lower byte (ex> 0xFF)
- Factor 2: Present position value's upper byte (ex> 0x07)
- Present position value hex change (hexadecimal to decimal): 0x07FF → 2047

- **Present Motor Operating Rate: Command packet to read present Motor Operating Rate.**

Command Packet

| HEADER | ID | Size | Command | Factor #1 | Factor #2 | Checksum |
|------------|------|------|---------|-----------|-----------|----------|
| | | | | Address | Length | |
| 0xFFFFFFFF | 0x00 | 0x04 | 0xF2 | 0x90 | 0x02 | 0x77 |

- ID: Refers to the actuator motor ID
- Command: Read bytes (equivalent to the Length number) from the address
- Address: The address at which the present motor operating rate value is saved (see (3) Data Map)
- Length: The number of bytes to read from the address (present operating rate value consists of 2 bytes)

Feedback Packet

| HEADER | ID | Size | Error | Factor #1 | Factor #2 | Checksum |
|------------|------|------|-------|-----------|-----------|----------|
| 0xFFFFFFFF | 0x00 | 0x04 | 0x00 | 0xFF | 0x03 | 0xF9 |

- ID: Refers to the actuator motor ID
- Error: Refers to error indication during operation
- Factor 1: Present Motor Operating Rate value lower byte (ex> 0xFF)
- Factor 2: Present Motor Operating Rate value upper byte (ex> 0x03)
- Present Motor Operating Rate value hex change (hexadecimal to decimal): 0x03FF → 1023

- **Present Voltage: Command packet to read present input voltage**

Command Packet

| HEADER | ID | Size | Command | Factor #1 | Factor #2 | Checksum |
|------------|------|------|---------|-----------|-----------|----------|
| | | | | Address | Length | |
| 0xFFFFFFFF | 0x00 | 0x04 | 0xF2 | 0x92 | 0x01 | 0x76 |

- ID: Refers to the actuator motor ID
- Command: Read bytes (equivalent to the Length number) from the address
- Address: The address at which the present voltage value is saved (see (3) Data Map)
- Length: The number of bytes to read from the address (present voltage value consists of 2 bytes)

Feedback Packet

| HEADER | ID | Size | Error | Factor #1 | Checksum |
|------------|------|------|-------|-----------|----------|
| 0xFFFFFFFF | 0x00 | 0x03 | 0x00 | 0x7B | 0x81 |

- ID: Refers to the actuator motor ID
- Error: Refers to error indication during operation
- Factor 1: Present Voltage value byte (ex> 0x7B)
- Present voltage value hex change (hexadecimal to decimal): 0x7B → 123 (12.3V)

6) Send Data

Send "Address" and "Data", then temporarily store it.

Command Packet

| HEADER | ID | Size | Command | Factor #1 | Factor #2 | Factor #3 | Checksum |
|------------|------|------|---------|-----------|-----------|-----------|----------|
| | | | | Address | Data #1 | Data #2 | |
| 0xFFFFFFFF | 0x01 | 0x05 | 0xF4 | 0x86 | 0xFF | 0x07 | 0x79 |

- Command packet for the temporary storage of goal position as 2047 (0x07FF)

Feedback Packet

| HEADER | ID | Size | Error | Checksum |
|------------|------|------|-------|----------|
| 0xFFFFFFFF | 0x01 | 0x02 | 0x00 | 0xFC |

- Feedback packet informing receipt of the temporary store for actuator goal position (respond in the Feedback Return Mode 2 only)

7) Execution

Execute temporarily stored data that is made by Send Data.

Command Packet

| HEADER | ID | Size | Command | Checksum |
|------------|------|------|---------|----------|
| 0xFFFFFFFF | 0x01 | 0x02 | 0xF5 | 0x07 |

- Command packet to execute all temporarily stored data at the same time

Feedback Packet

| HEADER | ID | Size | Error | Checksum |
|------------|------|------|-------|----------|
| 0xFFFFFFFF | 0x01 | 0x02 | 0x00 | 0xFC |

- Feedback packet informing receipt of execution command for temporarily stored data (respond in the Feedback Return Mode 2 only)

8) Symmetric Store

Save data in the same address of multiple actuators.

- **Goal Position: Command packet to assign multiple actuator's goal positions.**

- Actuator ID 1: 1023 (0x03FF), Actuator ID 2: 2047 (0x07FF)

| HEADER | ID | Size | Command | Factor #1 | Factor #2 |
|------------|------|------|---------|-----------|-----------|
| | | | | Address | Length |
| 0xFFFFFFFF | 0xFE | 0x0S | 0x73 | 0x86 | 0x02 |

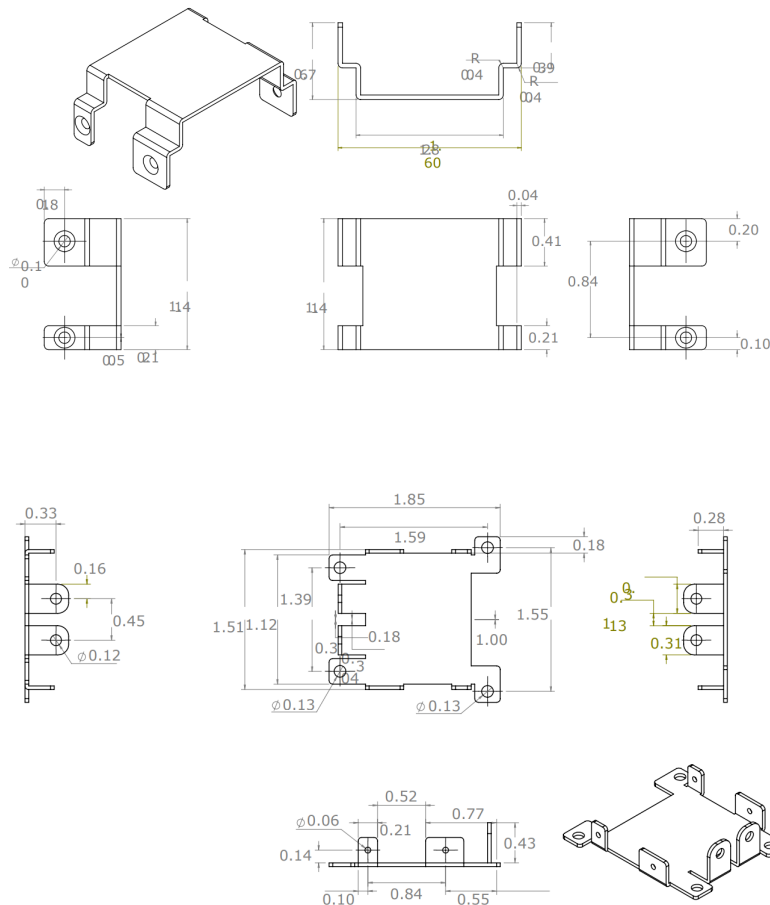
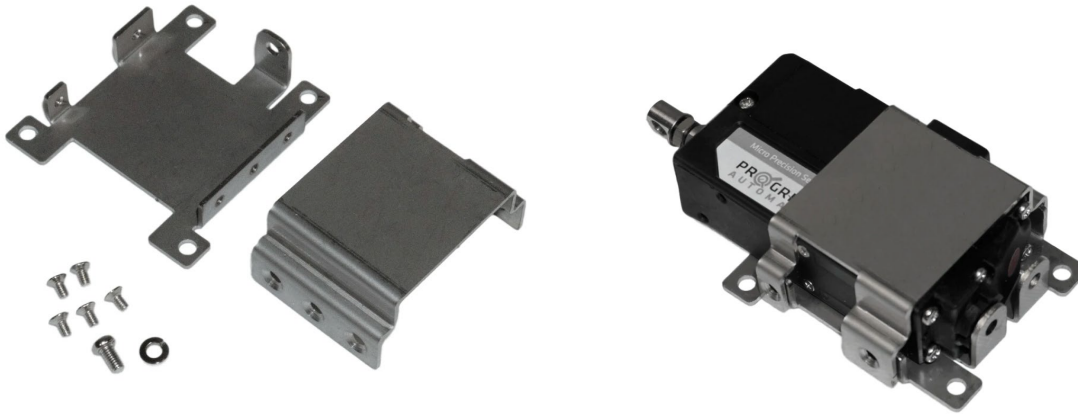
| Factor #3 | Factor #4 | Factor #5 | Factor #6 | Factor #7 | Factor #8 | Checksum |
|-----------|------------|-----------|-----------|------------|------------|----------|
| 1> ID | 1> Data #1 | 1>Data #2 | 2>ID | 2> Data #1 | 2> Data #2 | |
| 0x01 | 0xFF | 0x03 | 0x02 | 0xFF | 0x07 | 0xF1 |

- Command packet to assign respective goal position to multiple actuator motors at the same time
- Better synchronization without delay than the respective command is made for each actuator motor
- ID: Broadcast ID (command to all connected IDs)
- Command: Send data at the same time to the ID defined in Factor (1>ID, 2>ID ...)
- Address: The address at which the present position value is saved (see (3) Data Map)
- Length: The number of bytes to read from the Address (present position value consists of 2 bytes)
- Feedback Packet: Does not have feedback

5. Optional Accessories

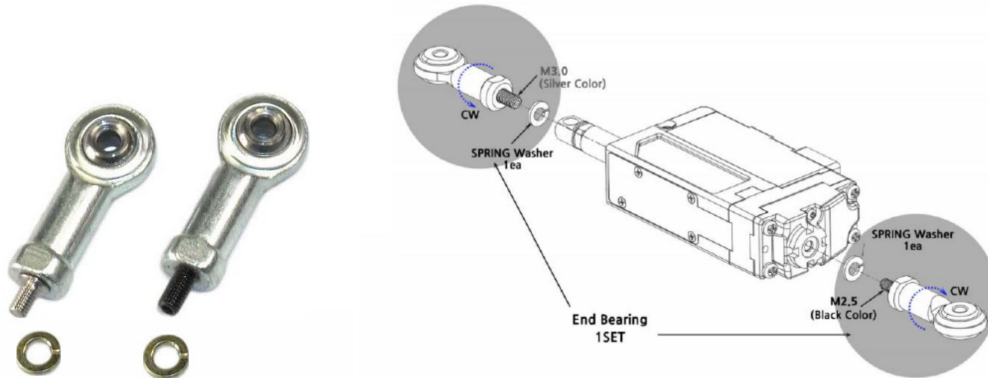
Mounting Bracket for PA-12 (BRK-12)

The BRK-12 is our steel mounting bracket set specifically designed to be mounted on our PA-12 Micro Precision Actuator with a 1.06" stroke. It will not fit models with a 2.20" stroke. The set comes with a top and bottom mounting piece that attaches to the actuator with the included screws. This bracket set provides stability as well as protection for the PA-12 actuator.



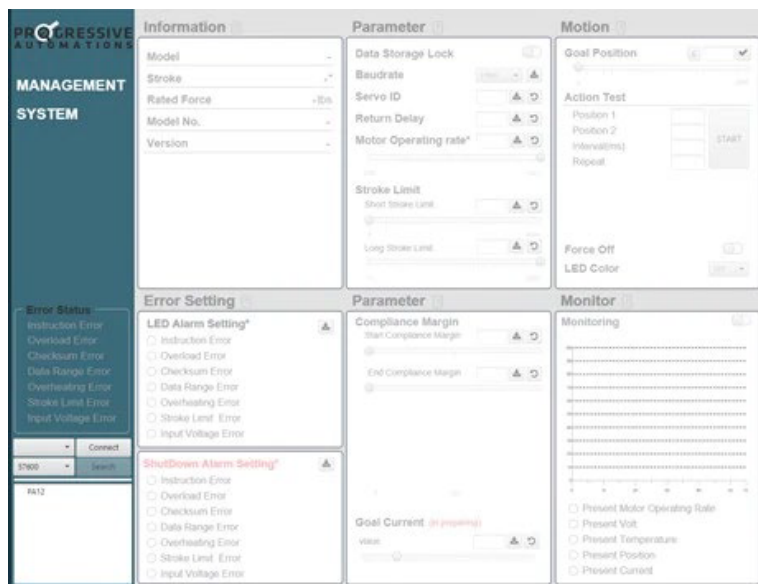
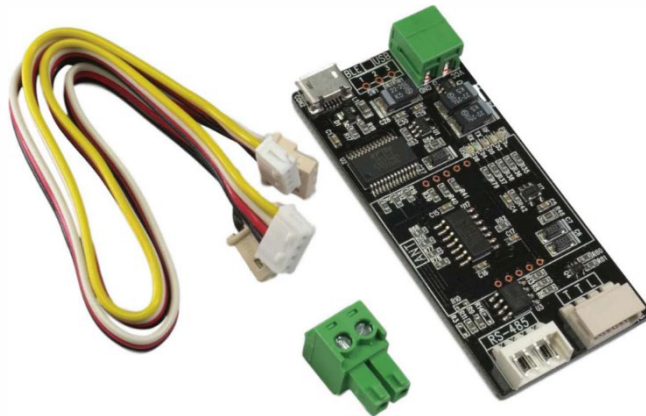
Fisheye Mounting Hole - PA-12 Models

The PRT-12 is an alternative fisheye steel mounting end for our PA-12 linear actuators. These are ideal for when you require a unique mounting option for the PA-12 actuator. It comes with two mounting holes, one for the motor end and one for the stroke end of the unit.



PC Interface Controller

The LC-12 is a PC USB interface controller that pairs together with our PA-12 Micro Precision Actuator. This interface allows for operation of the actuator and memory parameter settings, motion testing as well as voltage, load, temperature, and position monitoring of the linear actuator. To use the interface, you will need the program manager application.

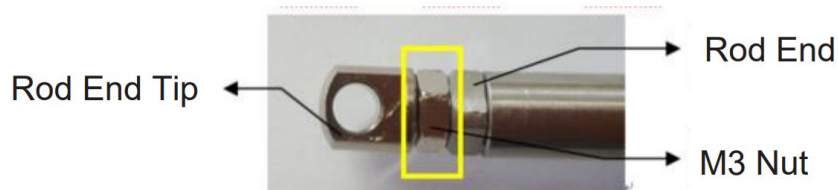


Included Accessories



- 1. Hinge Shaft (1pc)
- 2. Hinge Base (1pc)
- 3. Rod End Tip (1pc)
- 4. Hinge (1pc)
- 5. M2.5x6 Screws (3pcs)
- 6. M3 Nut (2pcs)
- 7. Wire (PF Version)
4Pin Molex to Molex (RS-485)
- 8. Wire (PT Version)
3Pin Molex to Molex
(TTL) & S-02 to Molex (PWM)

#6 M3 nut can be used to fix the hinge and hinge base. The M3 nut can also be used between rod-end and rod-end tip as a stopper.



6. Warranty

6.1. Warranty & Service

At Progressive Automations we take pride in the quality of our products. To meet our performance standards, all our products are thoroughly tested, checked, and undergo rigorous quality control procedures throughout the production process and prior to being shipped to the customer.

This warranty is subject to the proper storage, handling, transportation, installation, use, and maintenance of PA products by the customer.

Warranty Coverage

Progressive Automations warrants that the products sold are free from defects in materials and workmanship under normal and proper use and will conform to the manufacturer's published specifications.

If a product becomes defective during the warranty period, the customer will ship the product to Progressive Automations. We will repair or replace the product at our discretion and ship the product back to the customer free of charge.

Please note our PA-12 actuators have a warranty period of 18 months.

Warranty Exclusions

This Warranty does not cover any damage sustained from:

- Incorrect installation, handling, usage, and applications
- Use in extremely hazardous environments
- Improper assembly (other than described in the datasheet)

Warranty will be void if products have been altered or tampered with in any way. This warranty is limited to only the repair or replacement of the product and does not cover any additional costs.

Outside Warranty Coverage

Maintaining strong relationships with our existing customers is important to us at Progressive Automations. That is why we are pleased to offer special discounts on newer products to replace out-of-warranty items. Be sure to contact us and inquire about our discounts.

To obtain technical assistance for your product under warranty, please contact us at:

1-800-676-6123 | sales@progressiveautomations.com

Have any queries? Our expert engineers are here to help!



sales@progressiveautomations.com



1-800-676-6123



progressiveautomations.com