

# ULTRAMOTION

## servocylinder

### Information & User Manual

**A-Series:** Electromechanical Linear Actuator with Fully Integrated Phase Index™ Control System



**A1 Series:** Standard Servo Cylinder



**A2 Series:** Industrial Servo Cylinder



**AM Series:** Ruggedized Servo Cylinder



**AU Series:** Subsea Servo Cylinder

The most recent version of this document can be found here:  
[www.ultramotion.com/servo\\_cylinder/ultra-motion-servo-cylinder-manual.pdf](http://www.ultramotion.com/servo_cylinder/ultra-motion-servo-cylinder-manual.pdf)

# Important Information When Using Our Products

Please note that Ultra Motion's commercial off-the-shelf (COTS) products are not intended for use in critical applications where failure of the product may cause bodily harm or death. Please consider the following information when designing our products into your system.

## Performance

All commercial off-the-shelf products manufactured by Ultra Motion are designed to meet the performance specifications we publish in the product's manual. All life related data is provided as reference only and does not take into account application specific factors that can have significant impacts to the overall life of the product. Application specific factors can include: design loads, transient loads (shock, vibration, inertia), speed, environmental stresses (temperature, contamination), etc. Due to the fact that application specific factors can greatly affect the product's life, it is not possible to provide a generalized Mean Time Before Failure (MTBF). It is the customer's responsibility to determine the suitability of the product for their particular application.

## Software

A-Series actuators have a built-in controller and are shipped with the latest release of controller firmware. The controller firmware is changed from time to time to add features, fix undesired behavior, or change how the controller operates. We do our best to thoroughly test each firmware release, but we do not guarantee that the controller firmware will be free of software problems that may cause undesirable or unpredictable behavior. It is extremely important that you test the actuator for your application and do not use the actuator in applications where the failure or unpredictable operation of the actuator may result in injury or death.

## Change Control

Commercial off the shelf products are subject to changes that do not affect form, fit, or function. These changes can include the use of different PCB components, internal part revisions, suppliers, firmware, coloration, etc. Ultra Motion has the ability to track and manufacture version locked designs if your project has specific change control requirements. In a version locked design, the customer will be notified before any changes are made to their product.

## Quality Control

Ultra Motion actuators are manufactured under our internal quality management system. 100% of the product we manufacture goes through a complete performance QC inspection before leaving our facility. Documented results of QC records are available to all customers.

## Safety Information

**IMPORTANT:** Read this manual before installing and operating the Ultra Motion Servo Cylinder. Failure to read this section can result in personal harm or damage to the product.

### Safety Disclaimer

The Servo Cylinder is intended to be a subcomponent of a larger piece of machinery or automated system. This section is not intended to provide the safety guidelines for the entire machine or system that the Servo Cylinder is installed into. It is the responsibility of the purchaser or system designer to assess the risks and safety requirements of the end application they are designing.

### Safety Warnings

- Once powered, the Servo Cylinder is capable of rapid motion and can produce large amounts of force. Always ensure that safe clearances from people and equipment are maintained before applying power.
- While the Servo Cylinder operates on low voltage (8 to 36 VDC recommended), you must still use caution when handling and working around the actuator to avoid electrical shock.
- The motor of the actuator can become very hot, especially at high current draws. Take adequate time to cool before handling, and provide adequate ventilation for cooling of this device.

### Safety Notifications



As you read through the manual, you will notice certain safety notifications that indicate other important safety related information.

# Contents

<b>Important Information When Using Our Products</b> .....	<b>2</b>
Safety Information .....	2
Revision History .....	5
<b>Introduction</b> .....	<b>6</b>
<b>Product Information</b> .....	<b>7</b>
Compatible Control Signals & Operating Modes .....	7
Command Line Interface (CLI) Control Over Serial .....	8
CANopen (CiA DS301/CiA DS402) Control .....	8
CAN 2.0B Control .....	8
Analog Voltage (-10 VDC to +10 VDC) Control .....	8
Analog Current (4-20 mA) Control .....	8
Isolated Analog Current (4-20 mA) Output (Configurable Data Source) .....	8
1-2 ms Pulse (RC PWM Servo) Control .....	9
Step & Direction (Simulated Stepper) Control .....	9
A/B Quadrature Input Control .....	9
CW/CCW (Simulated Stepper) Control .....	9
Toggle (Simple On/Off for Extend/Retract) Control .....	9
Pre-set Position Control .....	9
Integrated Field-Oriented (FOC) Motor Control System with Phase Index™ Feedback .....	10
What is Phase Index™? .....	10
How is Phase Index™ used in Servo Cylinder? .....	10
Advantages of Field-Oriented (FOC) Motor Commutation .....	10
High Performance Electronics .....	10
Mechanical Design .....	11
Mechanical Interface & Dimensions .....	11
<b>Available Models</b> .....	<b>12</b>
Model Numbers and Ordering Codes .....	12
Ordering Code Structure .....	12
Series .....	13
Comparison of Series Features .....	13
Controllers .....	14
Comparison of Controller Options .....	14
Drive Screws .....	14
Comparison of Screw Options .....	16
PV-Limit for Acme Screws .....	16
Motor .....	16
Stroke .....	17
Mounting Options .....	17
Lubrication .....	17
Cabling .....	18
A1N Cables .....	18
<b>Requirements and Specifications</b> .....	<b>21</b>
Design Guidelines and Best Practices for Electromechanical Linear Actuators .....	21
Mechanical Specifications .....	22
Operating Environment .....	22
Overview of Environmental Ratings .....	22
Solid Particle Ingress Protection Ratings .....	22
Liquid Ingress Protection Ratings .....	23
Temperature Rating and Derating .....	24
Electrical Interface and Pinouts .....	25
A1 Series Actuator Electrical Interface .....	25
A2 Series Actuator Electrical Interface .....	26
AM Series Actuator Electrical Interface .....	27
AU Series Actuator Electrical Interface – Standard SubConn Connectors .....	28
AU Series Actuator Electrical Interface – Custom Teledyne Connectors .....	29
Cable Requirements and Specifications .....	30
Using Ultra Motion Standard Cable Assemblies .....	30
Using Custom Cable Assemblies .....	30
Electrical Input and Output Specifications .....	32

Power Bus (V+).....	32
Analog Current Input (ANI).....	34
Analog Voltage Input (ANV).....	35
Optically Isolated Digital Inputs (IN1+, IN1-, IN2+, IN2-, IN3+, and IN3-).....	35
Serial Communication Inputs and Outputs (TX, TX+, TX-, RX, RX+, RX-, TRX+, TRX-).....	36
CAN Connections (CAN_H and CAN_L).....	37
General Purpose Input / Output (GPIO) (DIO1 and DIO2).....	37
Analog Current Output (ANIO+ and ANIO-).....	38
<b>Using the Servo Cylinder .....</b>	<b>39</b>
Initial Setup .....	39
Getting Started.....	39
Setting up the Servo Cylinder Hardware .....	39
Configuration .....	41
How to Configure with Serial Commands.....	41
How to Configure with USB Mass Storage .....	41
Default Factory Settings.....	43
Determining Configuration Variables Relevant to Your Control Mode and Features .....	43
Configuration Setting Definitions .....	44
Installation and Use .....	61
Integrating Servo Cylinder into Your System .....	61
Using the Command Line Interface (CLI).....	61
Using the Status Register Feature.....	64
Tuning Performance.....	67
Updating the Firmware .....	71
Troubleshooting.....	75
Status Light Definitions N and P Controllers .....	75
<b>Appendix A: Serial Command Line Interface (CLI) List of Commands .....</b>	<b>76</b>
Read Commands .....	77
Set Commands .....	82
Trajectory Commands .....	87
Serial Data Streaming .....	89
System Commands .....	89
<b>Appendix B: Converting Between Sensor Counts and Physical Units .....</b>	<b>92</b>
Conversion Tables .....	92
Conversion Tables .....	93
Converting Torque to Force.....	94
<b>Contact Information .....</b>	<b>95</b>

## Revision History

Revision	Date	Details
B.03	11/5/2019	Internal release
B.04	1/27/2020	Approved for external release
B.05	3/3/2020	Added RV Table for CAN, added notes on lubrication, mounting arrangements, and AU receptacle orientation.
B.06	5/20/2020	Adjusted ballscrew dynamic load ratings
C.01	8/31/2021	Changes throughout including new features for -P- and -C- controllers
C.02	10/18/2021	Revised language

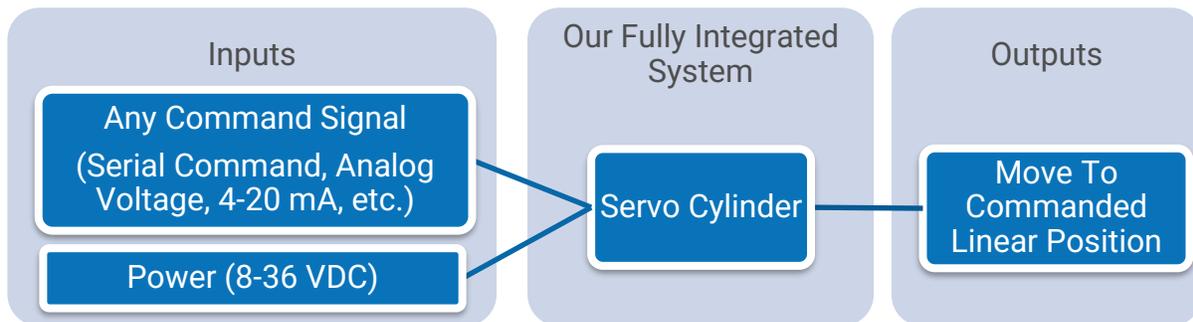
## Introduction

The Ultra Motion Servo Cylinder is a high performance linear servo system comprised of a robust rod-style actuator, a configurable brushless DC (BLDC) motor controller, and the cutting edge multi-turn absolute feedback technology Phase Index™. All components of the system have been engineered to seamlessly work together and provide a simple and extremely reliable linear positioning device.

Users must only provide power and a command input and the Servo Cylinder handles the rest. The built-in actuator-specific controller can be set to work with many common command inputs including CANopen, serial, analog, and stepper signals. It includes USB configurable settings for defining many attributes including performance characteristics, software limits, output information, and much more.

Servo Cylinder advantages include:

- Closed Loop position and current control directly out of the box
- Complete command language optimized for linear motion applications
- Absolute positioning – Never requires homing, even with loss of power.
- Fault tolerance and extensive diagnostic information available
- Precision in rugged package – Extremely long life
- Completely configurable
- Drop-in Replacement for many existing systems including stepper-motor driven actuators



## Product Information

The Servo Cylinder is an Electromechanical **Linear Actuator** with a fully integrated **Phase Index™** Field-Oriented (FOC) Brushless DC (BLDC) motor control and **positioning system**.

### Features:

- **Integrated Controller:** *Simply input a compatible control signal for fully closed-loop operation.*
- **Absolute Positioning:** *No homing/battery required.*
- **Full Command Line (CLI) language:** *For control, feedback, and communication.*
- **Easy Configuration:** *Configured via editable text file in USB mass storage, or with serial commands*
- *For more information visit the Product Web Page*

### Ratings:

Rating	Value
Force (Peak)	Up to 530 lbf [2.36 kN]
Force (Cont.)	Up to 270 lbf [1.20 kN]
Speed	Up to 14 in/s [356 mm/s]
Stroke Length	Up to 7.75 in [197 mm]
Resolution	Up to 0.000061in [1.55 μm]
Operating Voltage	8 to 36 VDC
Ingress Protection	IP50 to IP68 (Varies by Model)
Temp. Range	-40°F to 176°F [-40°C to 80°C]



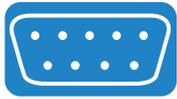
## Compatible Control Signals & Operating Modes

See the Controller section on page 14 for more information on which models support which control modes

\*Contact Ultra Motion to discuss preset position mode options for the CAN Controller

Feature	"N" <i>Classic Controller</i>	"P" <i>Industrial Controller</i>	"C" <i>CAN Controller</i>
RS232 Serial w/ Command Line Interface (CLI)	✓		
RS422 Serial (Full-Duplex, 4-wire) w/ Command Line Interface (CLI)		✓	
RS485 Serial (Half-Duplex, 2-wire) w/ Command Line Interface (CLI) *Does Not Command Motion			✓
CAN (Configurable Protocol)			✓
CANopen (CiA DS301/CiA DS402) Control			✓
Analog Voltage (-10 VDC to +10 VDC) Control	✓	✓	
Analog Current (4-20 mA) Control	✓	✓	
Isolated Analog Current (4-20 mA) Output (Configurable Data Source)		✓	
1-2 ms Pulse (RC PWM Servo) Control	✓		
Toggle (Simple On/Off for Extend/Retract) Control	✓		
Pre-set Position Control	✓		
Isolated Digital Inputs	2X	1X	3X
Transient Voltage Protection on Signal Lines		✓	✓

## Command Line Interface (CLI) Control Over Serial



The Servo Cylinder's Command Line Interface (CLI) offers the user complete control over all motion commands, configuration settings, and diagnostic information through a direct serial connection. You can connect to a programmable logic controller (PLC), a computer with any terminal program such as PuTTY or programmed with Python, C++, etc. (using a serial-to-USB adapter), or any platform that can transmit over a serial communication port such as an Arduino, LabVIEW, etc.

Servo Cylinder models are available which support the following serial protocols: RS232, RS422 (Full-Duplex, 4-wire), and RS485 (Half-Duplex, 2-wire). The CAN controller cannot be used to command motion via RS485 and is only meant for diagnostics and field updating of the firmware.

The Servo Cylinder can be configured to operate in "Human mode" (where detailed information regarding commands and asynchronous error messages are output to the terminal window) or "Machine Mode" (where asynchronous messages are turned off, and checksums are sent to ensure communication integrity).

The CLI may also be used in any other operating mode for diagnostic and non-motion commands.

## CANopen (CiA DS301/CiA DS402) Control



The CANopen controller is compliant to CiA DS301 and DS402 with support for cyclic synchronous position and profile position modes, allowing for sophisticated and synchronized control of up to 127 Servo Cylinders per CAN bus. The CANopen controller has a half-duplex RS-485 serial connection for diagnostics, configuration, and field updating of the firmware.

Contact Ultra Motion for details on CANopen operation including the Servo Cylinder .eds file and reference document UM711507

## CAN 2.0B Control

Ultra Motion's CAN 2.0B protocol is highly configurable allowing for simple integration with a variety of automation systems, commercial autopilots, test setups, and J1939 systems. The controller supports 11-bit or 29-bit identifiers and includes detailed telemetry as well as full control over the actuator's position and motor current.

Contact Ultra Motion for our CAN 2.0B protocol specification document UM711574

## Analog Voltage (-10 VDC to +10 VDC) Control



In Analog Voltage mode, the actuator moves proportionally to a range of input voltages from -10 VDC to +10 VDC. The upper and lower limits of this voltage range can be configured to any arbitrary voltage between -10 VDC and +10 VDC. Additionally, the range of positions over which the actuator moves proportionally to the input signal can be configured independently.

## Analog Current (4-20 mA) Control



In Analog Current mode, the actuator moves in proportion to a range of input currents between 4 mA and 20 mA. The upper and lower limits of this current range can be configured to any arbitrary current between 4 mA and 20 mA. Additionally, the range of positions over which the actuator moves proportionally to the input signal can be configured independently.

This is a control mode popular with industrial controllers and PLC's, since this analog current control is relatively noise resistant and enables the actuator to respond gracefully to a cut-wire/loss of signal (0 mA).

## Isolated Analog Current (4-20 mA) Output (Configurable Data Source)



For applications requiring closed loop feedback from the actuator, Servo Cylinder models are available with an analog current output feature with a configurable data source. This can be used to output an analog current signal (4-20 mA) proportional to position, force, velocity, and several other values.

## 1-2 ms Pulse (RC PWM Servo) Control

---



In this mode, the actuator can be easily configured to respond to a standard RC PWM signal with 50 Hz or 333 Hz update rate.

## Step & Direction (Simulated Stepper) Control

---



A Servo Cylinder in this mode operates with incremental closed loop movements in response to direction signal and step pulses. This allows the Servo Cylinder to directly replace a stepper motor that operates under this command signal.

## A/B Quadrature Input Control

---



A Servo Cylinder in this mode operates with A/B Quadrature, which consists of two square wave code signals spaced 90° out of phase, allowing for both speed and direction to be tracked. This allows the Servo Cylinder to directly replace a stepper motor that operates under this command signal.

## CW/CCW (Simulated Stepper) Control

---



A Servo Cylinder in this mode operates with incremental movements in response to stepper motor CW/CCW control signals. This allows the Servo Cylinder to directly replace a stepper motor that operates under this command signal.

## Toggle (Simple On/Off for Extend/Retract) Control

---



In Toggle mode, the Servo Cylinder operates with the simplicity of a brushed DC control system with the added performance and long-life of a BLDC motor. In this mode, the actuator simply extends for as long as it is given an extend signal to its digital input, and retracts likewise. The absolute position capability of the Servo Cylinder eliminates the need for external limit switches, and also allows for configurable acceleration and velocity for smooth, controlled motion. Output speed and torque can either be preset, or controlled by using analog inputs.

## Pre-set Position Control

---



Preset position mode provides the user with an easy way to perform a wide variety of positioning tasks. This mode is especially useful for replacing pneumatic pistons.

The user selects one of up to four (4) pre-programmed positions by toggling the state (00, 01, 10, or 11) of the two optically isolated digital inputs (IN1/IN2), causing the actuator to move to the selected position at a defined speed and acceleration using the Servo Cylinder's built-in trajectory generator.

# Integrated Field-Oriented (FOC) Motor Control System with Phase Index™ Feedback

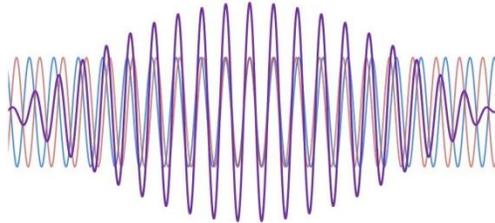
## What is Phase Index™?

---

Phase Index™ is Ultra Motion's innovative position sensor technology for electromechanical actuators. The Phase Index™ position sensor is a digital, high-speed, high-resolution, non-contacting actuator position sensor that works over a wide temperature range and is resistant to vibration, shock, particulate debris, and condensing moisture. The sensor is self-calibrating, and retains absolute position with full accuracy without power or backup power. No homing routine is needed to establish an accurate absolute position.



The Phase Index™ position sensor works by using the phase relationship between two cyclical signals with different periods to determine absolute position within a larger interference cycle of the combined signals. When implemented with state-of-the-art magnetic sensor technology, this technique allows for a high-speed, high-resolution, digital actuator position sensor that is always accurate and that works across a range of harsh environmental conditions.



The first implementation of Ultra Motion's Phase Index™ position sensor technology which was used in a highly demanding human spaceflight application, has performed flawlessly through extensive and rigorous environmental testing and field operations. It is now implemented in a wide variety of applications, from high-performance aerospace actuators to low-cost industrial motion control systems.

## How is Phase Index™ used in Servo Cylinder?

---

Phase Index is seamlessly integrated into the Field-Oriented Control system used to operate the Servo Cylinder's Brushless DC (BLDC) motor. Specifically, it provides reliable absolute position of the BLDC rotor for commutation and the absolute linear position of the actuator shaft. The onboard digital signal processor (DSP) handles all user inputs, configuration, trajectory generation, Field-Oriented Control, and position control loop calculations.

## Advantages of Field-Oriented (FOC) Motor Commutation

---

Field-Oriented Control (also known as Vector Control) is generally considered to be the best performing method of BLDC motor commutation when compared to other common methods such as trapezoidal or sinusoidal commutation.

FOC provides smooth movement at all speeds, as opposed to trapezoidal commutation which is subject to significant torque ripple (which gets magnified at lower rotor speeds) and sinusoidal commutation which has lower efficiency at high speeds due to limitations in bandwidth of P.I.D. controllers. Additionally, FOC provides increased acceleration performance, and allows for direct user control of torque and actuator force.

## High Performance Electronics

---

Modern performance components used in the motor control system allow for robust operation over a wide range of supply voltages and temperatures. The Servo Cylinder's low on-state resistance, high current capacity, and rapid switching time results in cool and efficient control over the load. High-speed current feedback from all three phases allows for maximum flexibility of Field-Oriented Control.

## Mechanical Design

The Servo Cylinder actuator has numerous mechanical design features

### Shock and Vibration Resilient Linear Position Sensor

- The linear position of the actuator is encoded using the Phase Index™ rotary position sensor. Rotary position encoders exhibit superior resilience to shock and vibration when compared to linear position sensors such as potentiometers or LVDTs. Further shock and vibration hardening is available with the AM series actuators.

### Directly Coupled Drivetrain

- The linear assembly is directly coupled to the BLDC rotor, eliminating drivetrain backlash and providing maximum position accuracy and acceleration performance.

### Various Drive Screws

- The Servo Cylinder is offered with a wide variety of different type of drive screws for different performance characteristics to optimize to your application. See “Drive Screws” on page 14.

### Environmental Protection

- Sealing levels available IP50 through IP68, depending on the needs demanded by the application.

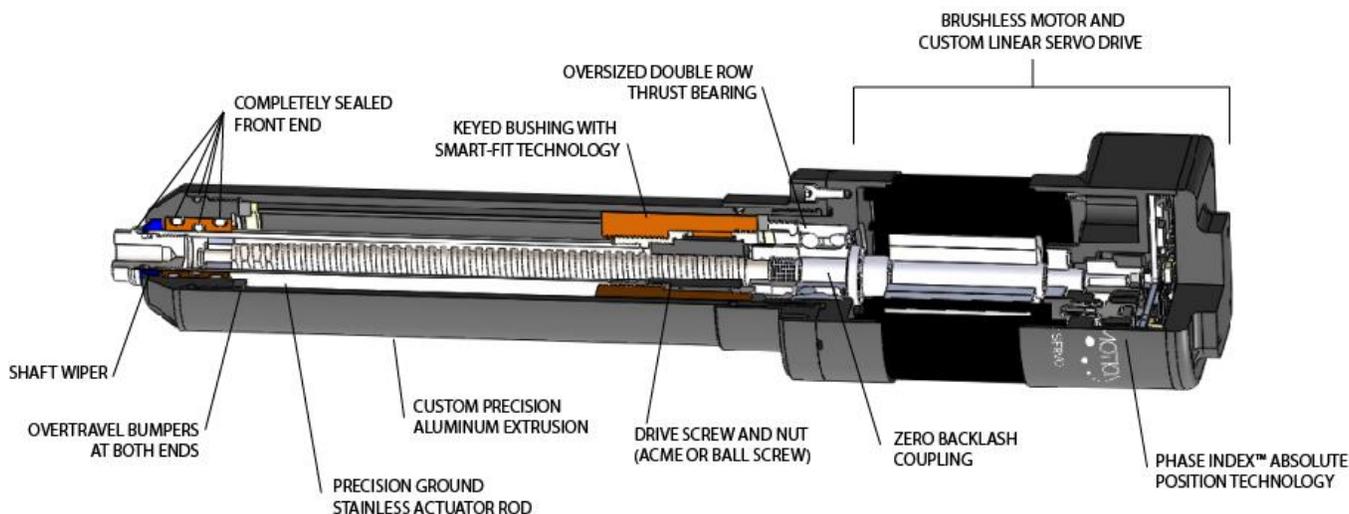


Figure 1: Anatomy of an A1 series Servo Cylinder

## Mechanical Interface & Dimensions

Our full line of A-series Servo Cylinder actuators is offered in a wide variety of sizes and types for different applications. For an offering of all models and features, with information on mechanical interface and dimensions including 2D and 3D CAD models, as well as datasheets and performance information, please visit the Ultra Motion online configurator:



[Servo Cylinder Web Configurator](http://www.ultramotion.com/servo-cylinder-configurator/)  
www.ultramotion.com/servo-cylinder-configurator/

## Available Models

All of the information available in this section as well as the performance related information for each specific model can be found online via our Servo Cylinder configurator:



[Servo Cylinder Web Configurator](http://www.ultramotion.com/servo-cylinder-configurator/)  
www.ultramotion.com/servo-cylinder-configurator/

### Model Numbers and Ordering Codes

The ordering code of the servo cylinder, also known as a model number, represents all options selected for that specific servo cylinder model. You may determine the ordering code of a servo cylinder model based on the information in this section, or by using our web configurator. The ordering code utilizes characters which represent the series, controller, screw, motor, stroke, and options for the Servo Cylinder actuator you are specifying. The characters and options available are elaborated upon further in this section.

### Ordering Code Structure

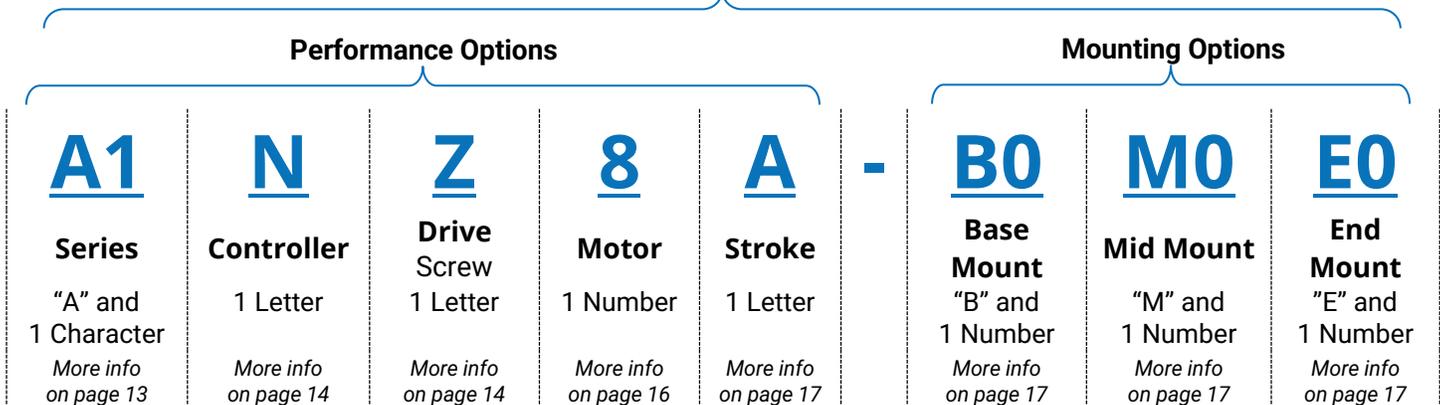
Use the following diagram to help determine the individual features called out within the ordering code. A full description of all available options can be found further in this section.

This example ordering code given here describes an A1-Series “Standard Servo Cylinder” with the “N” classic controller, a standard 0.125 inch lead ballscrew, a 180W motor, 1.75 inch stroke, and no additional mounting options or equipment.

**Example Ordering Code:**

# A1NZ8A-B0M0E0

Ordering Code / Model Number



#### Note: Legacy Ordering Codes

Until 2018 a similar ordering code system was utilized which was identical except that it did not feature a character for the Controller option. The legacy ordering codes contain 11 letters/numbers and a dash (12 total characters). Current ordering codes utilize 12 letters/numbers and a dash (13 total characters). All Servo Cylinder actuators with the legacy ordering code utilize the “N” – Classic Controller (code “N”). These codes can be made to adhere to the current ordering code system by adding an “N” for the controller code.

## Series

The actuator series predominantly defines the external features of the actuator. This includes the housing, ingress protection, and electrical interface of the actuator. It also includes the level of internal vibration hardening and ruggedization.

### Comparison of Series Features

---



#### A1 Series: Standard Servo Cylinder

The A1 Series actuator is our standard Servo Cylinder model. It is best suited to cleaner indoor environments, and is available at a lower cost than our other sealed models.

- IP50 Environmental protection
- Molex Micro-Fit 3.0™ Connector

#### A2 Series: Industrial Servo Cylinder

The A2 Series actuator is our industrialized Servo Cylinder model. It is sealed to a level suitable for wash-down, outdoor, or more debris laden industrial environments.

- IP65 dynamic environmental protection
- Robust metal housing



#### AM Series: Ruggedized Servo Cylinder

The AM series actuator is our most heavily ruggedized Servo Cylinder model. It features high performance mil-spec connectors, and has been hardened to a higher level of mechanical shock and vibration. This actuator is suitable for harsh environments.

- Hard-coat anodized aluminum housing
- Sealed to IP67 Static, IP65 Dynamic
- Mil-Spec (MIL-D-38999) electrical connectors
- Vibration hardened solder joints and electrical connections throughout
- Conformal coated PCBs
- Continuously conductive housing & components

#### AU Series: Subsea Servo Cylinder

The AU series actuator is our underwater Servo Cylinder model. It has been designed with an oil-filled internal pressure compensation system for operation in subsea environments of both salt and fresh water to a depth of 250 feet (~76 meters).

- Type III Hard-coat anodized aluminum housing
- Fully sealed, oil filled, IP68 rated enclosure.
- Wet-Mate electrical connectors
- Pressure balanced, rated to 250 ft (~76 m) depth
- Sacrificial anodes for galvanic corrosion resistance
- Continuously conductive housing & components
- *Note:* No USB port available. Actuator must be configured over serial with opMode configured by factory.

## Controllers

The controller selection will define which operating modes and electrical features the Servo Cylinder has.



**Warning:** Each controller model has a different electrical interface, and must be wired correctly to prevent damage to the actuator or equipment

### Comparison of Controller Options

Feature	<b>"N"</b> <i>Classic Controller</i>	<b>"P"</b> <i>Industrial Controller</i>	<b>"C"</b> <i>CAN Controller</i>
RS232 Serial w/ Command Line Interface (CLI)	✓		
RS422 Serial (Full-Duplex, 4-wire) w/ Command Line Interface (CLI)		✓	
RS485 Serial (Half-Duplex, 2-wire) w/ Command Line Interface (CLI) *Does Not Command Motion			✓
CAN (Configurable Protocol)			✓
CANopen (CiA DS301/CiA DS402) Control			✓
Analog Voltage (-10 VDC to +10 VDC) Control	✓	✓	
Analog Current (4-20 mA) Control	✓	✓	
Isolated Analog Current (4-20 mA) Output (Configurable Data Source)		✓	
1-2 ms Pulse (RC PWM Servo) Control	✓		
Toggle (Simple On/Off for Extend/Retract) Control	✓		
Pre-set Position Control	✓		
Isolated Digital Inputs	2X	1X	3X
Transient Voltage Protection on Signal Lines		✓	✓

Table 1: Feature matrix for all controllers available on Servo Cylinder Actuators

#### Note: Legacy Ordering Codes

Until 2018 a similar ordering code system was utilized which was identical except that it did not feature a character for the Controller option. The legacy ordering codes contain 11 letters/numbers and a dash (12 total characters). Current ordering codes utilize 12 letters/numbers and a dash (13 total characters). All Servo Cylinder actuators with the legacy ordering code utilize the "N" – Classic Controller (code "N"). These codes can be made to adhere to the current ordering code system by adding an "N" for the controller code.

## Drive Screws

The drive screw of the actuator is used to translate motor rotation and torque into linear displacement and force. While fundamentally similar, there are many subtle differences between the different types of screw systems offered with the Servo Cylinder, and it is important to understand them. To learn more about some of these important performance characteristics, see Table 2: Definitions of Terms Describing Screw Performance Characteristics (on page 15).

Term	Definition
<b>Acme Screw</b>	Acme screws operate by threading a screw through a nut to generate linear motion. The screw uses the Acme thread-form which offers improved performance over ordinary V-threads and is available in a wide variety of leads. All standard Acme screws currently offered are self-locking.
<b>Ballscrews</b>	Ballscrew systems operate similarly to a traditional screw and nut assembly except that the load is carried by a continuous circuit of rolling ball bearings which flow through the nut and screw threads. This allows for overall better performance including higher efficiency, load capacity, and life. All standard ballscrews currently offered are back-drivable.
<b>Lead</b>	The screw lead is the linear distance traveled per single rotation of the screw.
<b>Max Dynamic Load</b>	<p>The max dynamic load is the amount of force that the actuator is capable of safely exerting while in a dynamic (moving) state. The maximum dynamic load is related to the screw/motor pair because it is driven by heat generation in the motor. There are different ratings for both continuous usage (indefinite time period) and for peak usage (for brief applications of load with suitable cool-down time in between).</p> <p>This is not to be confused with Dynamic Load Rating (<math>C_r</math>) which is used for life calculation.</p>
<b>Max Static Load</b>	<p>The max static load is the amount of force that the actuator is capable of safely exerting while in a static (non-moving) state.</p> <p><i>For back-drivable screws, the maximum static load is related to the screw/motor pair because it is driven by heat generation in the motor. For this type of screw, there are different ratings for both continuous usage (indefinite time period) and for peak usage (brief applications of load with suitable cool-down time in between).</i></p> <p><i>For self-locking screws, the maximum static load is driven by the load rating of the thrust bearing. For this type of screw, the actuator has the same rating for continuous and peak static load ratings. Note that the Servo Cylinder must be commanded to a low current or power removed in order to make use of the screw's self-locking characteristics.</i></p>
<b>Dynamic Load Rating (<math>C_r</math>)</b>	<p>The dynamic load rating (<math>C_r</math>) represents the continuous load at which the actuator will have an <math>L_{10}</math> life of 1,000,000 linear inches. The <math>L_{10}</math> life is the amount of travel after which approximately 10% of actuators will have failed (90% of actuators fully operational). The life of the actuator is heavily dependent upon the loads exerted upon the actuator.</p> <p>The dynamic load rating only applies to ballscrews, as the life calculation for Acme screws is based on PV and other operating conditions.</p>
<b>Resolution</b>	The smallest increment of linear position that the actuator is capable of moving by.
<b>Backlash</b>	Backlash is the largest allowed amount of play (linear motion) that can be seen in the actuator shaft while the actuator motor is being held still. This can be thought of as a hysteresis of motion that can be measured when switching between tensile and compressive loads.
<b>Self-Locking</b>	A screw system is considered "self-locking" when it is designed such that no amount of linear force will cause the screw to turn. In a self-locking screw, linear loads generate rotary friction which exceeds back-driving torque. Consequently, only motor torque on the screw will allow the actuator to move. The practical advantage of this is that a self-locking actuator only needs power to move between positions. It does not require any power to hold large static loads indefinitely.
<b>Back-Drivable</b>	A screw system is considered "back-drivable" if the screw assembly is capable of moving in response to a linear load when the motor is unpowered. Back-drivable screws such as ballscrews have generally better performance than self-locking screws. However, back-drivable screws must be powered in order to carry any load, whether it is static or dynamic.

Table 2: Definitions of Terms Describing Screw Performance Characteristics

## Comparison of Screw Options

Code	Description	Lead, in [mm]	Maximum Speed, in/s [mm/s]	Max Dynamic Load [Continuous], lbf (N)	Max Dynamic Load (Peak), lbf [N]	Max Static Load (Continuous), lbf [N]	Max Static Load (Peak), $C_0$ , lbf [N]	Dynamic Load Rating ( $C_r$ ), lbf [N]	Resolution, $\mu$ in [ $\mu$ m]	Typical Backlash, in [mm]	Self-Locking (SL) or Back-Drivable (BD)	Min Temperature, °F [°C]
Z	Standard Ballscrew	0.125 [3.175]	14 [356]	170 [756]	340 [1512]	117 [756]	530 [2358]	136 [605]	122 [3.1]	.005 [0.13]	BD	-40 [-40]
Y	Extended-Life Ballscrew	0.125 [3.175]	14 [356]	170 [756]	340 [1512]	170 [756]	530 [2358]	272 [1210]	122 [3.1]	.005 [0.13]	BD	-40 [-40]
X	Metric Pitch Ballscrew	0.079 [2.000]	9 [229]	270 [1200]	530 [2358]	270 [1200]	530 [2358]	255 [1134]	77 [1.9]	.005 [0.13]	BD	-40 [-40]
W	Standard Acme Screw	0.0625 [1.6]	7 [178]	115 [510]	225 [1000]	530 [2358]	530 [2358]	N/A	61 [1.5]	.005 [0.13]	SL	-4 [-20]
V	Extreme Temp. Acme Screw	0.0625 [1.6]	7 [178]	115 [510]	225 [1000]	530 [2358]	530 [2358]	N/A	61 [1.5]	.005 [0.13]	SL	-40 [-40]
U	High Force & Extreme Temp. Acme Screw	0.0625 [1.6]	7 [178]	115 [510]	225 [1000]	530 [2358]	530 [2358]	N/A	61 [1.5]	.005 [0.13]	SL	-40 [-40]

Table 3: Feature matrix for all screws available (COTS) on Servo Cylinder Actuators. All values for performance assume ideal assembly component selection (e.g. 180W motor) and operating conditions (e.g. room temperature, 36 VDC voltage, etc.)

## PV-Limit for Acme Screws

The Pressure-Velocity Limit for Acme screws is determined by the Acme nut material. Continuous application speeds/forces should be kept below the limit for optimal operation. The PV-Limit is a thermal limitation and can be exceeded for brief periods of time, but excessive wear or premature failure can occur if the PV-Limit is severely exceeded or exceeded continuously. PV-Limits can be seen on all Acme performance graphs on Ultra Motion's online configurator.

[Servo Cylinder Web Configurator](https://www.ultramotion.com/servo-cylinder-configurator/)

[www.ultramotion.com/servo-cylinder-configurator/](https://www.ultramotion.com/servo-cylinder-configurator/)

## Motor

The motor power rating and the screw are the two key factors affecting the dynamic performance of the actuator. It is a good practice to select a motor that will have good headroom for dynamic performance to avoid overheating.

Code	Motor	Characteristics
9	100W Brushless-DC (BLDC) Motor	<ul style="list-style-type: none"> <li>• Standard Performance                             <ul style="list-style-type: none"> <li>◦ Actual performance ratings depend upon other assembly components.</li> </ul> </li> <li>• Lower cost than 180W model</li> <li>• Shorter overall length</li> </ul>
8	180W Brushless-DC (BLDC) Motor	<ul style="list-style-type: none"> <li>• Enhanced Performance                             <ul style="list-style-type: none"> <li>◦ Actual performance ratings depend upon other assembly components.</li> </ul> </li> <li>• Higher cost than 100W model</li> <li>• Longer overall length</li> </ul>

Table 4: Feature matrix for all motors available on Servo Cylinder Actuators

## Stroke

The stroke of the actuator is the range of travel between the fully retracted and the fully extended states. Actuators with higher stroke will have longer overall lengths. Contact Ultra Motion if a custom stroke length may be necessary. Due to a constraint imposed by the Phase Index absolute position sensor where only 64 revolutions can be encoded, certain stroke lengths are limited to certain specific screw types.

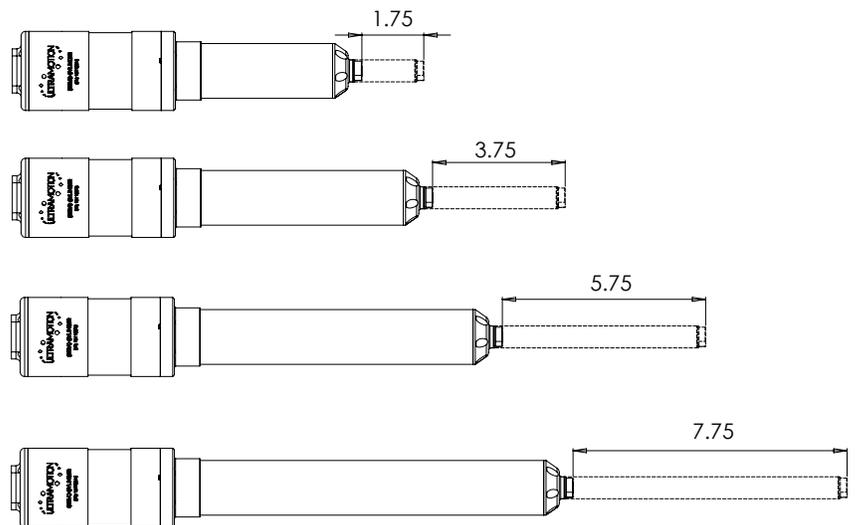
### Note: Stroke vs. Software-Limited Travel Range

The stroke length is defined as the full distance between the hard end-stops (physical travel limits) of the actuator. The positions of these limits are defined by the positions rPos (retracted end-stop) and ePos (extended end-stop). For safety and to account for trajectory overshoot, the actuators are shipped from the factory by default with the travel range limited to be slightly smaller than this full stroke. By default, spMin (the software retracted travel limit) is set 1024 counts (one screw rotation) higher than rPos, and spMax (the software extended travel limit) is set 1024 counts lower than ePos.

In general, it is possible to safely reclaim some of this stroke by reducing the size of both gaps between the hard end-stop and the software travel limit. Before doing this, we highly recommend that you measure any amount by which your actuator overshoots trajectory as part of normal operation. To safely re-define your software travel limits, include the full overshoot distance as well as a safe buffer into your gap between the hard end-stop and the software travel limit. Understand that hitting the hard end-stop can damage or destroy the actuator. 3D models are displayed in the full mechanically retracted position by default.

Stroke Length	Stroke Code	Compatible Screw Codes
1.75 in	A	Z, X, W, U, V
3.75 in	B	Z, X, W, U, V
5.75 in	C	Z
7.75 in	D	Z
1.00 in	E	Y
3.00 in	F	Y
5.00 in	G	Y
7.00 in	H	Y

Table 5: Feature matrix for all stroke lengths available (COTS) on Servo Cylinder actuators.



## Mounting Options

Proper mounting is critical for successful operation of an actuator. The appropriate degrees of freedom are required to account for assembly motion and for structural tolerances in the application. Failure to properly account for all required degrees of freedom can lead to reduced life, performance, and/or failure of an actuator. For a full list of options including all mounting hardware, cables, and power supplies, please visit the Servo Cylinder configurator or contact Ultra Motion engineering:



[Servo Cylinder Web Configurator](http://www.ultramotion.com/servo-cylinder-configurator/)  
www.ultramotion.com/servo-cylinder-configurator/

## Lubrication

Proper lubrication of the actuator is critical for achieving the maximum amount of life in a particular application. Lubricant degradation can occur over time and due to application loads, speeds, vibration, temperature, etc. If your application requires a high number of cycles it is recommended to discuss relubrication/service intervals with Ultra Motion. The Servo Cylinder's dynamic load ratings do not take into account lubricant degradation.

## Cabling

Using the Servo Cylinder requires appropriate cables to electrically connect the actuator to both power and signal connections. Additionally, you may need a USB mini B cable if you wish to use USB configuration. The exact cable necessary depends upon the actuator series.

To purchase a pre-fabricated cable from Ultra Motion, use the table of part numbers below. Alternatively, it is possible to build a cable custom for your application. To do so, see the sections detailing each type of electrical connector in the sections. For pricing or for information on any cables not in this table, please contact Ultra Motion.

Part Number	Description
<b>CBL-USB</b>	Cable, USB mini-B, 6ft
<b>CBL-X-XX-X</b>	Piecewise A1N Cable, See A1N Cable below for full details
<b>CBL-DEV</b>	A1 Series Cable Assembly, Fully Populated, 10ft
<b>CBL-A2</b>	Kit of 2x A2 Cable Assemblies, Fully Populated, 10ft (Both Power and Signal Cables)
<b>CBL-A2-RT</b>	Kit of 2x A2 Cable Assemblies, Right Angle, Fully Populated, 10ft (Both Power and Signal Cables)
<b>CBL-AM-DEV</b>	Kit of 2x AM Cable Assemblies, Fully Populated, 10 ft, PVC jacket, non-sealed, M85049/38-11N
<b>CBL-AU</b>	Contact Ultra Motion for details on AU cable options

Table 6: All available (off-the-shelf) cable assemblies for Servo Cylinder actuators.

### A1N Cables

If you are purchasing for an A1 actuator with an “N” - Classic Controller, an A1N piecewise cable is available for purchase which has been populated with conductors only for the features you will be using. If the options are selected, these cables are shipped with spade terminals for power connections, a DB9 connector for RS232 serial, and flying leads for all other signal wires.

Alternatively, you can purchase with a fully populated connector/cable assembly (**CBL-DEV**). This is necessary for all controllers besides the “N” - Classic Controller.

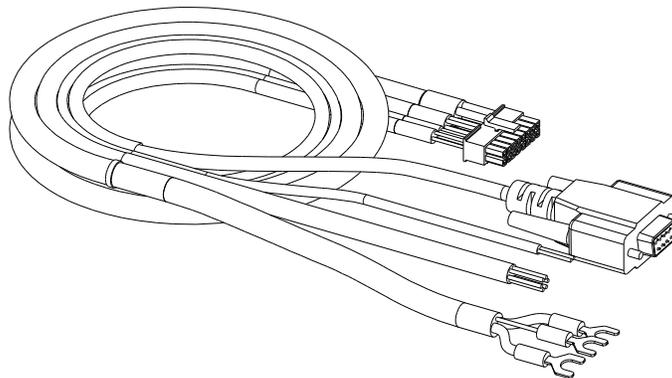


Figure 2: A1 series Power & Signal Cable (CLB-S-AV-D)

Use the information below to specify a cable part number based on what functions you would like. A complete part number **CBL-S-AV-D** (with analog voltage) or **CBL-S-AC-D** (with analog current) will come with all wires populated.

1. Start with Base Power Cable:
  - Base Power Cable: **CBL** (The base cable is a connector with power conductors only)
2. Choose additional functions as required in this order, adding their dash code to the end of the code.
  1. For Serial compatibility: **-S** (See serial cable section below for details.)
  2. For Analog compatibility: **-AC** or **-AV** (For analog current or analog voltage respectively, only one may be selected unless you use CBL-DEV, see analog cable section for details.)
  3. For Digital I/O compatibility: **-D** (See digital cable section below for details.)

#### Cable Part Number Example:

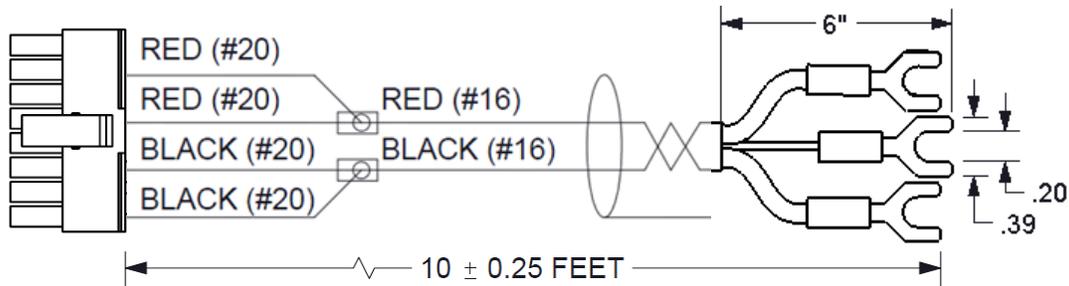
A user planning to control the actuator with an analog voltage signal (-AV) who also wants access to the Servo Cylinder's GPIO (-D) would specify the following cable assembly part number: CBL-AV-D.

### Base Power Cable (CBL)

Four (4) 20 AWG conductors spliced to two (2) 16 AWG conductors in a twisted, shielded cable terminated with 5 mm stud, crimp spade terminals.

#### Use:

- Supply power to the Servo Cylinder

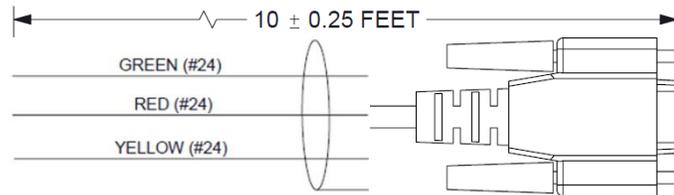


### Serial Cable (-S)

**Cable Specification:** Shielded cable with three (3) 24 AWG conductors terminated in an over molded DB9 connector

#### Uses:

- Command Line Interface (CLI) Control Over Serial
- Serial diagnostic information and data streaming
- Updating firmware

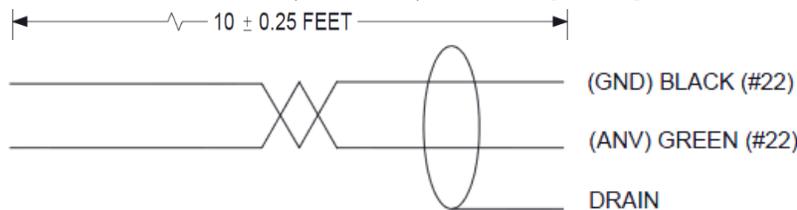


### Analog Voltage Cable (-AV)

Two (2) 22 AWG conductors in a shielded twisted pair terminated with 6" flying leads.

#### Uses:

- Analog Voltage (-10 VDC to +10 VDC) Control
- Toggle (Simple On/Off for Extend/Retract) Control (with analog voltage control of speed or force).

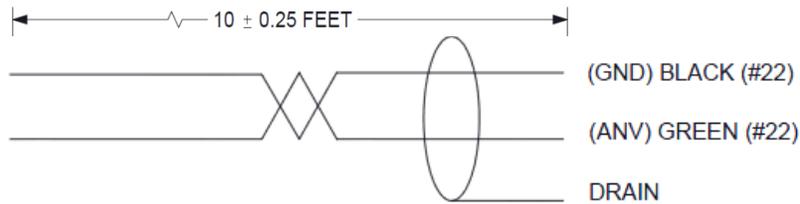


### Analog Current Cable (-AC)

Two (2) 22 AWG conductors in a shielded twisted pair terminated with 6" flying leads.

#### Uses:

- Analog Current (4-20 mA) Control
- Toggle (Simple On/Off for Extend/Retract) Control (with analog current control of speed or force).

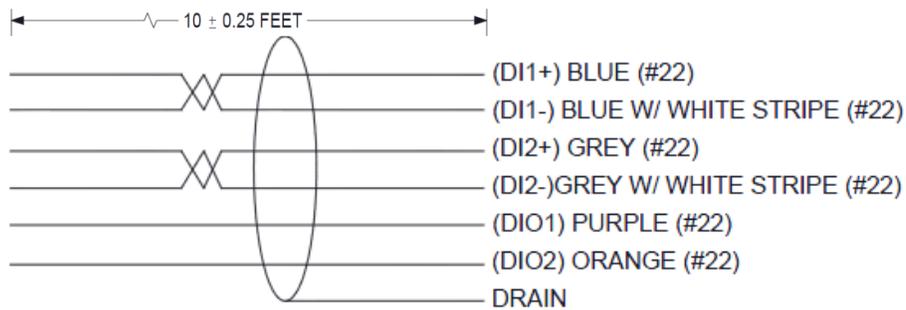


### Digital Cable (-D)

**Cable Specification:** Six (6) 22 AWG conductors in a shielded cable, terminated with 6" flying leads.

**Uses:**

1. 1-2 ms Pulse (RC PWM Servo) Control
2. Step & Direction (Simulated Stepper) Control
3. A/B Quadrature Input Control
4. CW/CCW (Simulated Stepper) Control
5. Four Preset Position Mode
6. Toggle (Simple On/Off for Extend/Retract) Control
7. General Purpose Input / Output (GPIO) functionality



# Requirements and Specifications

This section contains all information about the electrical and mechanical specifications and requirements of the Servo Cylinder actuator.

## Design Guidelines and Best Practices for Electromechanical Linear Actuators

This section aims to provide some helpful advice for implementing Servo Cylinder and electromechanical linear actuators in general. The information on this page will help get the most out of your Servo Cylinder while also avoiding the most common pitfalls seen by the application engineers at Ultra Motion.

### [Include Dynamic Forces in your Specified Load](#)

If using an actuator to lift a specific amount of mass, remember that the forces being exerted by the actuator will include the mass and inertial loads to accelerate the object.

### [Would a "Self-Locking" actuator be better for you?](#)

"Self-Locking" is a characteristic of a screw system where the thread pitch and friction force is such that no amount of applied axial load to the actuator will cause the screw to rotate.

An actuator powered by a self-locking leadscrew has the following performance tradeoffs:

Pros	Cons
<ul style="list-style-type: none"><li>• No power necessary to statically support a large load indefinitely.</li><li>• Lower Cost</li><li>• Lower Noise</li></ul>	<ul style="list-style-type: none"><li>• Lower force, speed, and efficiency.</li><li>• Increased stiction, precise micro-motions more difficult to achieve.</li></ul>

### [Increase Maximum Speed with a Higher Bus Voltage](#)

In BLDC motors, torque/force is independent of bus voltage. However, maximum actuator speed is proportional to bus voltage.

### [Use an Unregulated Power Supply if Possible](#)

For the reasons outlined in Power Supply Requirements and Wiring Details (Page 33), unregulated supplies are generally superior to regulated supplies for motion control applications.

### [Motor Back-EMF Causes Unstable Bus Voltage](#)

For the reasons outlined in Back-EMF (Voltage Fluctuations) (Page 32), motors will inject energy into a power bus and cause spikes in bus voltage. Use a Power Shunt (Page 34), and be wary of any equipment that shares a power bus with an actuator.

### [Shield All Signal Inputs and Outputs](#)

Wires leading to the actuator (especially analog inputs and GPIOs) should be shielded to prevent

electromagnetic interference (EMI) and cross-talk. This is the most common cause for unexpected behavior in analog control modes. Be sure to properly ground provided cable shields.

### [Never "Side-Load" an Actuator Not Rated for Side-Load](#)

Rod-style linear actuators such as the Servo Cylinder are typically not rated for bending moments, loads perpendicular to the output shaft axis, or any force loads other than axial tension or compression. Significant "side-loads" often occur inadvertently with over-constrained actuator mounting. Side-loads result in accelerated wear, reduced performance, or failure of the actuator. Mounting methods must be designed with adequate rotational degrees of freedom, such as pivoting joints (like a clevis or trunnion), or spherical bearings on both ends of the actuator, to ensure the actuator is only loaded in the axial direction when accounting for assembly motion and tolerances in your structure.

### [Beware Running an Actuator into a Hard-Stop](#)

When using an actuator with a DC motor, take care that you do not run the actuator into a physical "hard-stop", such as a machine element or any physical body which the actuator impacts on its travel and which disallows movement. If this happens, the control system will attempt to obey position commands by powering through the physical block, capping the motor torque command to the maximum. Often, this results in the motor over-heating and burning out, especially with higher maximum torque settings.

The Servo Cylinder is programmed by default with travel limits to prevent collision with its own internal hard-stops, however care must be taken to protect the actuator from external hard-stops. It also features errMode – Error Mode (Actuator Hardware Protection Feature) which will prevent the actuator from exerting damaging amounts of force or torque. This is especially useful when training and testing an application.

## Mechanical Specifications

Since there are a wide variety of different Servo Cylinder models with various performance characteristics, please refer to the Servo Cylinder Datasheet and test reports for information that pertains to your specific actuator. If you do not yet have your datasheet, it will be automatically generated by completing the Servo Cylinder Configurator on the Ultra Motion website. Each datasheet includes:

- Max Load vs. Speed (Continuous and Peak)
- Actuator life vs Load (for Ballscrew Models)
- PV Limits (for Acme Screw Models)
- Mechanical Interface Control Drawing (ICD) for the specific actuator model



[Servo Cylinder Web Configurator](http://www.ultramotion.com/servo-cylinder-configurator/)  
www.ultramotion.com/servo-cylinder-configurator/

## Operating Environment

### Overview of Environmental Ratings

The type of environment that the Servo Cylinder is designed to operate in varies by series. See Table 7 below. Note that all ingress protection ratings assume the actuator is in the fully assembled state, with all access covers installed. Additionally, the cables on the actuator used must be sealed to an ingress protection rating which meets or exceeds that of the actuator.

Series	Ingress Protection (Mated Connectors)	Ingress Protection (Unmated Connectors)	Max Operating Temp. Range	Other Environmental Ratings
A1	IP50	IP50	-40°F to 176°F [-40°C to 80°C]	-
A2	IP65	IP50	-40°F to 176°F [-40°C to 80°C]	-
AM	IP67 Static, IP65 Dynamic	IP50	-40°F to 176°F [-40°C to 80°C]	Shock and Vibration Hardened
AU – Standard	IP68 250 ft [76 m] Depth	IP64	-40°F to 140°F [-40°C to 60°C]	Saltwater Submersion
AU – Custom	IP68 250 ft [76 m] Depth	IP64	-4°F to 158°F* [-20°C to 70°C*] (*See Warning below)	Saltwater Submersion

Table 7: Operating Environment Ratings, by series

In the table above, the ingress protection (IP) rating of the actuators are defined in both the mated (with connectors properly installed) and unmated (no connectors). To properly install a connector to achieve ingress protection, see [Installing Cables on Sealed Servo Cylinders](#) on page 40.

Additionally, in the table above, the Max Operating Temperature Range is defined as the highest temperature range available within that series. Certain configurable options may further restrict the temperature range, as well as appropriate temperature derating.



**WARNING:** For the AU series actuators, the connectors must be mated in the +23°F to +104°F [-5°C to +40°C] temperature range, and cannot be un-mated outside this range without risking loss of sealing. The full specified operating temperature range only applies to mated connectors.

### Solid Particle Ingress Protection Ratings

Depending on the series, the Servo Cylinder is rated to be either dust-protected (A1 series) or dust-tight (A2, AM, and AU series). If the actuator is being used in an environment where a significant amount of dust or abrasive material can accumulate on the actuator shaft, it might be necessary to add a shaft bellows. A shaft bellows will help prevent

the accumulating dust from degrading the actuator's seal. Speak to one of Ultra Motion's Applications Engineers for more guidance.

Additionally, while mating connectors, no significant presence of solid bodies such as dirt, dust, or ice may be present within the connector, as this may interrupt seal integrity during the connector mating process.

### **Liquid Ingress Protection Ratings**

---

Certain models of Servo Cylinder including the A2, AM, and AU series are protected against liquid ingress. To preserve the ingress protection ratings of these actuators, you must follow the following requirements:

- The cable must be properly installed, see *Installing Cables on Sealed Servo Cylinders* on page 40 for sealing requirements for cables and required coupling torques on connectors.
- Access covers must be properly in place.

## Temperature Rating and Derating

The maximum continuous load rating of the Servo Cylinder related to the rate of heat generation by the motor and the maximum operating temperature of the actuator components. As such, when environmental temperature increases, the maximum continuous load rating of the actuator must decrease accordingly. A derating factor  $C_T$  is applied to the value for the nominal temperature maximum continuous load rating at ( $F_{nom}$ ) to determine the maximum continuous load rating ( $F_T$ ) for the actual operational ambient temperature:

$$F_T = F_{nom} \times C_T$$

The derating factor  $C_T$  is a function of the nominal temperature at which the maximum continuous load rating actuator is defined ( $T_{nom} = 25^\circ\text{C}$  for Servo Cylinder), the maximum operating environmental temperature of the actuator ( $T_{max} = 80^\circ\text{C}$  for Servo Cylinder), and the ambient temperature of the application to which you are derating ( $T_{ambient}$ ).

$$C_T = \sqrt{\frac{T_{max} - T_{ambient}}{T_{max} - T_{nom}}} = \sqrt{\frac{80^\circ\text{C} - T_{ambient} (^\circ\text{C})}{80^\circ\text{C} - 25^\circ\text{C}}} = \sqrt{\frac{176^\circ\text{F} - T_{ambient} (^\circ\text{F})}{176^\circ\text{F} - 77^\circ\text{F}}}$$

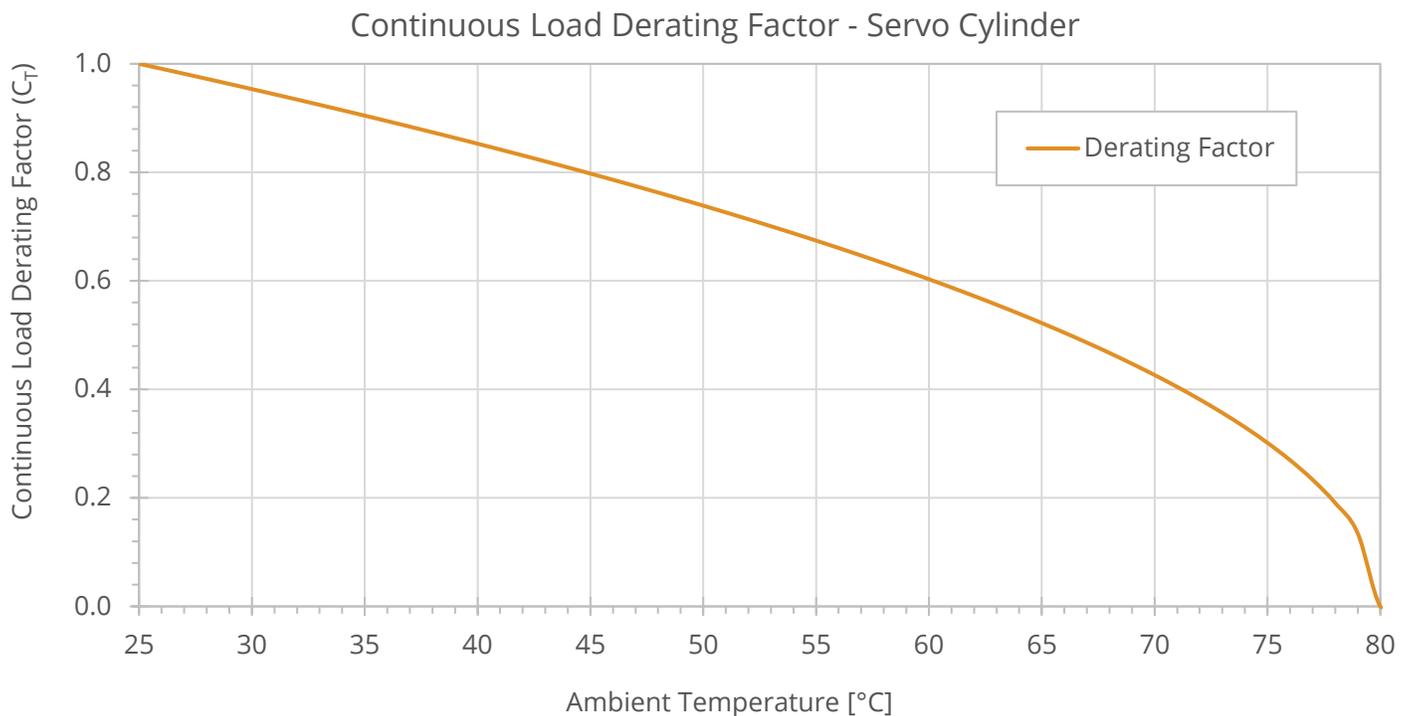


Figure 3: Derating factor for maximum continuous load  $C_T$  as a function of temperature.

### Self-Heating Functionality

The Servo Cylinder's -P- and CAN 2.0B -C- controllers have a configurable heating functionality to maintain an acceptable PCB operating temperature when the measured temperature drops below a configurable threshold. This allows for an expanded operation envelope to temperatures  $<-55^\circ\text{C}$ . The actuator must start up in an ambient environment of  $>-40^\circ\text{C}$  in order to guarantee operation.

The user has control over the threshold temperature and the amount of current to be used during heating. The heating functionality can be disabled if not used, monitored via status bits, and will not impact the performance of the actuator while active.

Contact Ultra Motion engineering for setting recommendations based on your application's requirements.

## Electrical Interface and Pinouts

### A1 Series Actuator Electrical Interface



**WARNING:** The electrical interface and pinouts vary between the actuator controller models. Using an improperly wired cable **will** damage to the actuator. Always be aware of your controller and its electrical interface before using the actuator.

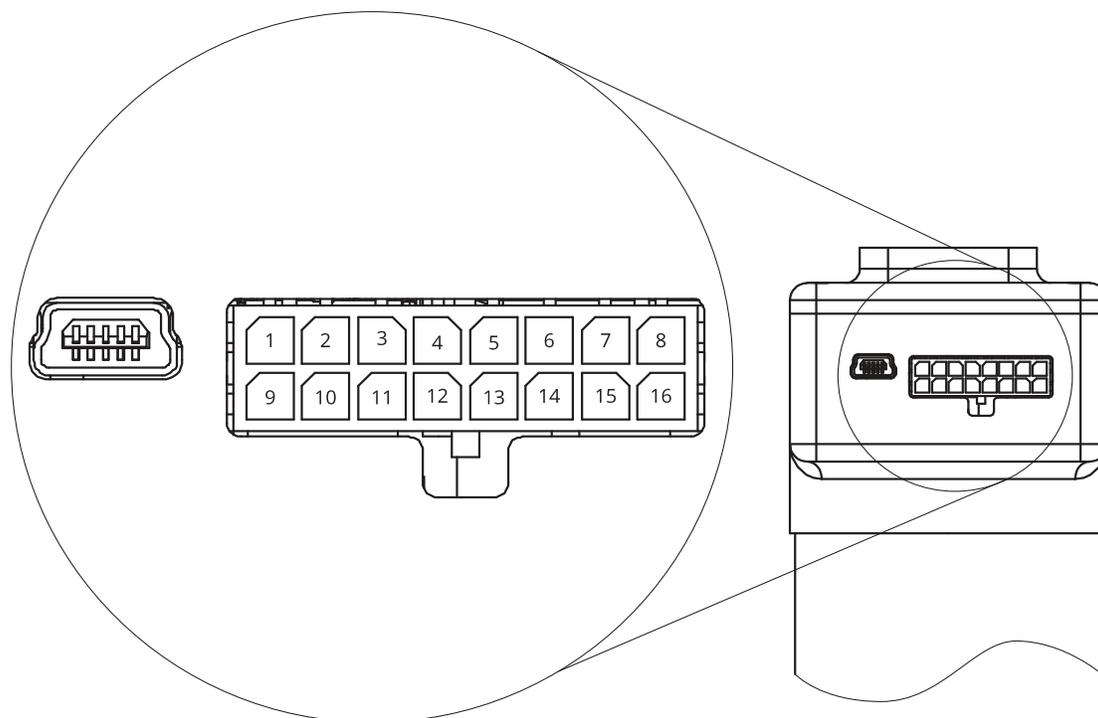


Figure 4: Pinout for A1 series Actuators

No.	Pin for "N" Controller	Pin for "P" Controller	Pin for "C" Controller	Flying Lead Color (CBL-S-AV/AC-D) For "N" Controller only	Flying Lead Color (CBL-A1DEV)
1	ANI	ANI OUT-	CAN_L	GREEN	WHITE
2	GND	ANI OUT+	GND	BLACK	BROWN
3	TX	TX+	TRX-	GREEN (DB9)	GREEN
4	DIO1	TX-	IN1+	PURPLE	VIOLET
5	IN2+	RX-	IN2+	GREY	GRAY
6	IN1+	IN1+	IN3+	BLUE	BLUE
7	V+	V+	V+	RED (Spade Terminal)	RED (Spade Terminal)
8	V+	V+	V+	RED (Spade Terminal)	RED (Spade Terminal)
9	ANV	ANI IN	CAN_H	GREEN	RED
10	GND	GND	GND	RED (DB9)	BLACK
11	RX	ANV	TRX+	YELLOW (DB9)	YELLOW
12	DIO2	GND	IN1-	ORANGE	ORANGE
13	IN2-	RX+	IN2-	GREY W/ WHITE	PINK
14	IN1-	IN1-	IN3-	BLUE W/ WHITE	TAN
15	GND	GND	GND	BLACK (Spade Terminal)	BLACK (Spade Terminal)
16	GND	GND	GND	BLACK (Spade Terminal)	BLACK (Spade Terminal)

Table 8: Pin numbers, functionality, and standard cable wire colors for A1 series actuators.

## A2 Series Actuator Electrical Interface



**WARNING:** The electrical interface and pinouts vary between the actuator controller models. Using an improperly wired cable **will** damage to the actuator. Always be aware of your controller and its electrical interface before using the actuator.

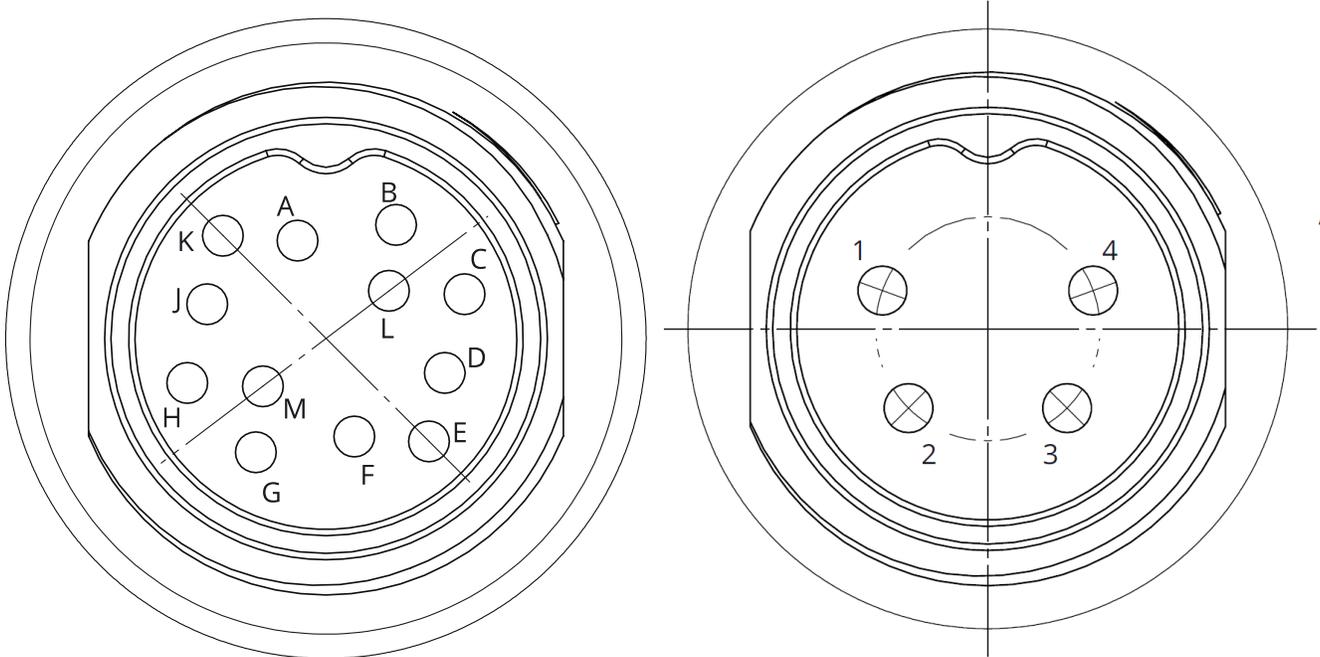


Figure 5: Pinouts for Signal (left) and Power (right) connectors on A2 series actuators.

	No.	Pin for "N" Controller	Pin for "P" Controller	Pin for "C" Controller	Flying Lead Color (CBL-A2 and CBL-A2-RT)
Signal	A	IN1+	IN1+	IN3+	BLUE
	B	IN1-	IN1-	IN3-	TAN
	C	IN2+	RX-	IN2+	GRAY
	D	IN2-	RX+	IN2-	PINK
	E	DIO1	TX-	IN1+	VIOLET
	F	DIO2	GND	IN1-	ORANGE
	G	ANV	ANI IN	CAN_H	RED
	H	GND	GND	GND	BLACK
	I	(REMOVED)	(REMOVED)	(REMOVED)	(REMOVED)
	J	ANI	ANI OUT-	CAN_L	WHITE
	K	TX	TX+	TRX-	GREEN
	L	GND	ANI OUT+	GND	BROWN
	M	RX	ANV	TRX+	YELLOW
Power	1	V+	V+	V+	RED
	2	V+	V+	V+	GREEN
	3	GND	GND	GND	BLACK
	4	GND	GND	GND	WHITE

Table 9: Pin numbers, functionality, and standard cable wire colors for A2 series actuators.

## AM Series Actuator Electrical Interface



**WARNING:** The electrical interface and pinouts vary between the actuator controller models. Using an improperly wired cable **will** damage to the actuator. Always be aware of your controller and its electrical interface before using the actuator.

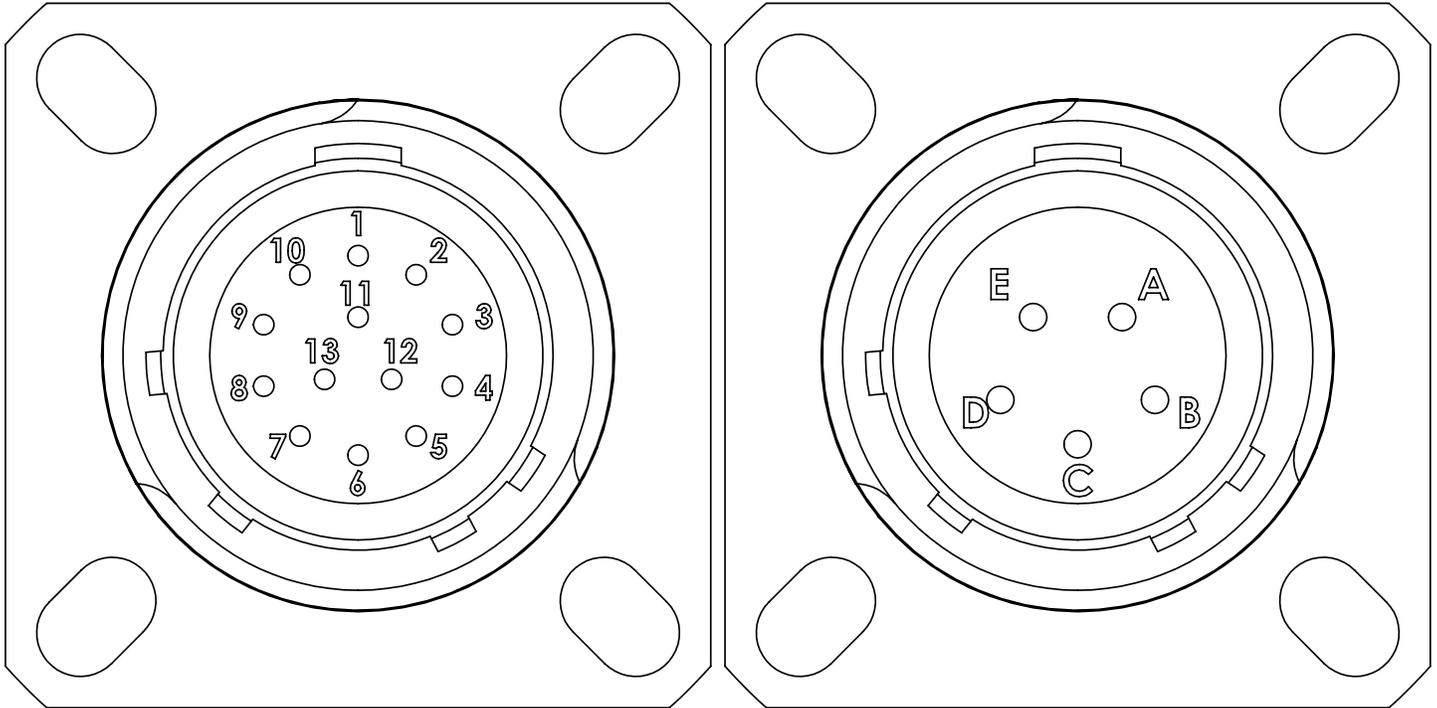


Figure 6: Pinouts for Signal (left) and Power (right) connectors on AM series actuators.

	No.	Pin for "N" Controller	Pin for "P" Controller	Pin for "C" Controller	Flying Lead Color (CBL-AM)
Signal	1	IN1+	IN1+	IN3+	BLUE
	2	IN1-	IN1-	IN3-	TAN
	3	IN2+	RX-	IN2+	SLATE
	4	IN2-	RX+	IN2-	PINK
	5	DIO1	TX-	IN1+	VIOLET
	6	DIO2	GND	IN1-	ORANGE
	7	ANV	ANI IN	CAN_H	RED
	8	GND	GND	GND	BLACK
	9	ANI	ANI OUT-	CAN_L	WHITE
	10	TX	TX+	TRX-	GREEN
	11	GND	ANI OUT+	GND	BROWN
	12	RX	ANV	TRX+	YELLOW
	13	N/C	N/C	N/C	N/C
	SHELL	CHASSIS	CHASSIS	CHASSIS	DRAIN
Power	A	V+	V+	V+	RED
	B	V+	V+	V+	GREEN
	C	GND	GND	GND	BLACK
	D	GND	GND	GND	WHITE
	E	N/C	N/C	N/C	N/C
	SHELL	CHASSIS	CHASSIS	CHASSIS	DRAIN

Table 10: Pin numbers, functionality, and standard cable wire colors for AM series actuators. Note that the "SHELL" pin refers to the metal connector casing.

## AU Series Actuator Electrical Interface – Standard SubConn Connectors

For information about the AM or AU series electrical interface, please contact Ultra Motion directly.

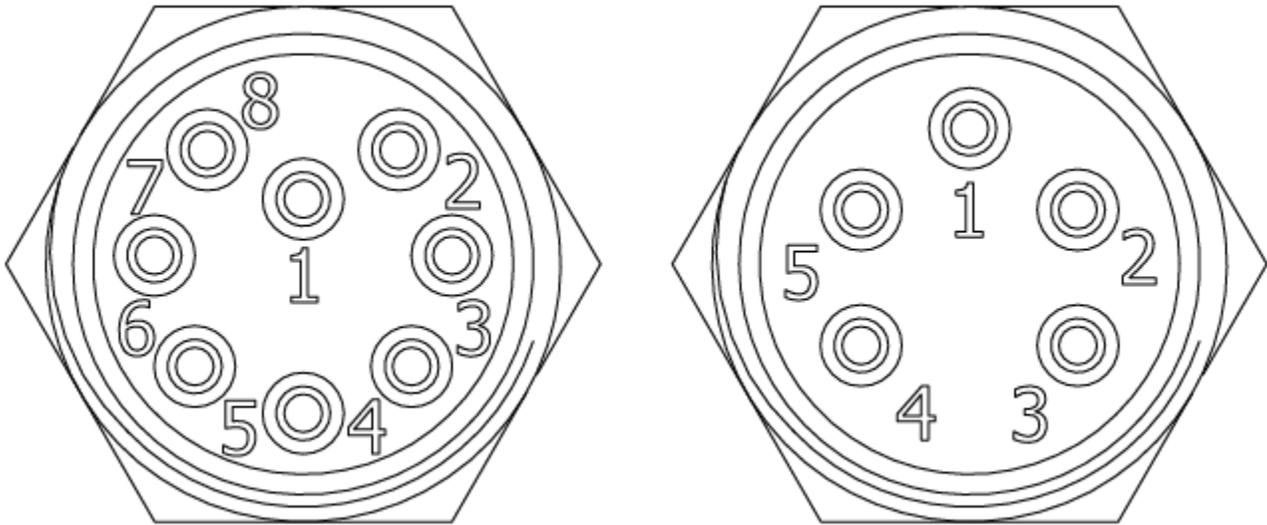


Figure 7: Pinouts for Signal (left) and Power (right) connectors on Standard AU series actuators.

	No.	Pin for "N" Controller	Pin for "P" Controller	Pin for "C" Controller	Flying Lead Color (CBL-AU)
Signal	1	GND	GND	GND	BLACK
	2	IN2+	ANI OUT+	IN1+	WHITE
	3	IN1+	RX+	CAN_H	RED
	4	TX	TX+	TRX+	GREEN
	5	RX	TX-	TRX-	ORANGE
	6	DIO1	ANI IN	GND	BLUE
	7	IN2-	ANI OUT-	IN1-	WHITE/BLACK
	8	IN1-	RX-	CAN_L	RED/BLACK
Power	1	GND	GND	GND	BLACK
	2	GND	GND	GND	WHITE
	3	V+	V+	V+	RED
	4	V+	V+	V+	GREEN
	5	CHASSIS	CHASSIS	CHASSIS	ORANGE

Table 11: Pin numbers, functionality, and cable wire colors for Standard AU series actuators.

### AU – Standard Cable Installation

The standard AU SubConn connectors must be lubricated with Molykote 44 Medium on every installation. A layer of grease corresponding to a minimum of 1/10 socket depth should be applied to the female connector in dry-mate conditions. A layer of grease corresponding to 1/3 socket depth should be applied to the female connector in wet-mate situations. Angular loads on the cable and mated cable assembly must be minimized. Debris and mud on the male connectors and within the female connector sockets must be cleaned thoroughly before mating with liquid soap, hot water, and isopropyl alcohol. The orientation of the AU receptacles on the housing will be as shown with a tolerance of  $\pm 30^\circ$ .

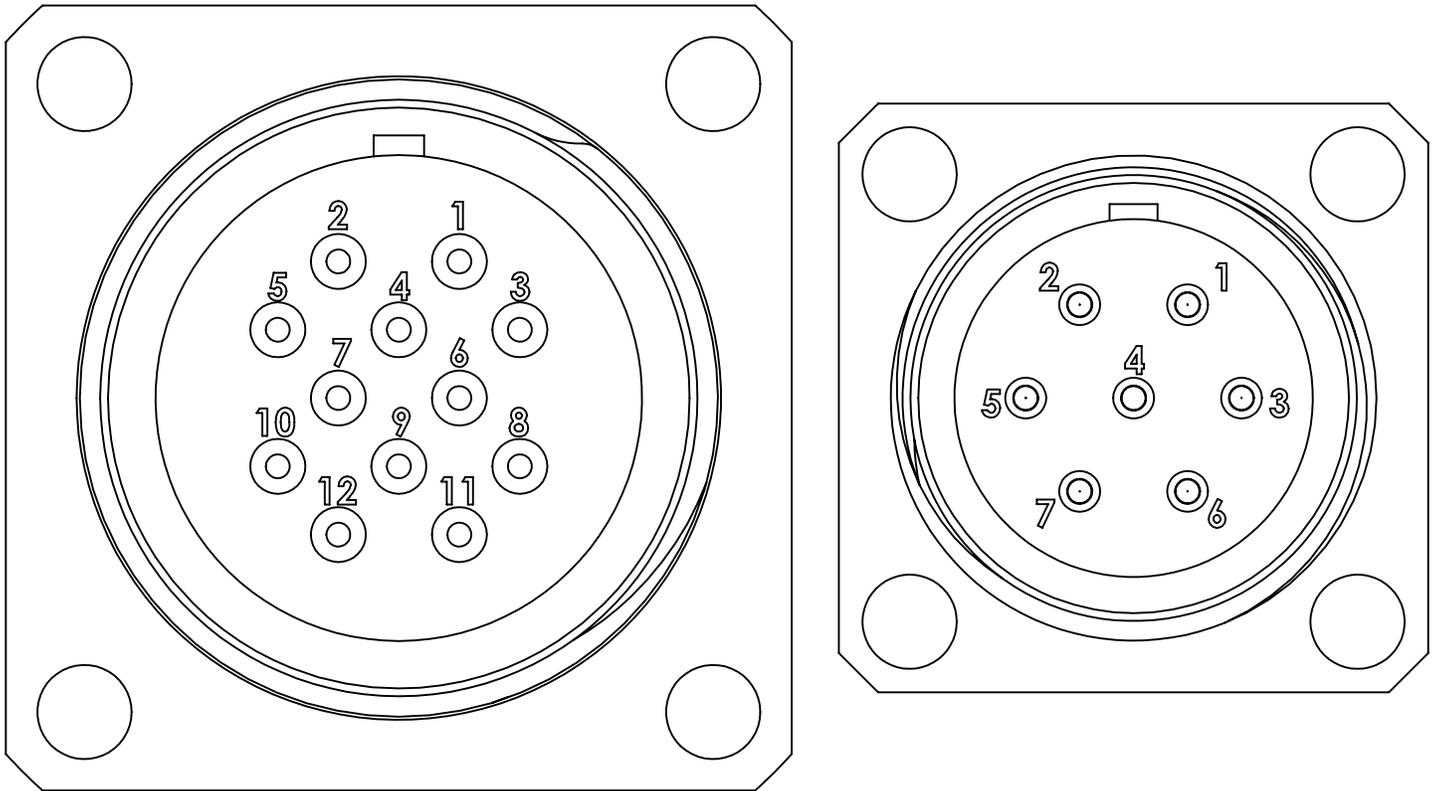


Figure 8: Pinouts for Signal (left) and Power (right) connectors on Custom AU series actuators.

	No.	Pin for "N" Controller	Pin for "P" Controller	Pin for "C" Controller	Flying Lead Color (CBL-AU*)
<b>Signal</b>	1	IN1+	IN1+	IN3+	-
	2	IN1-	IN1-	IN3-	-
	3	IN2+	RX-	IN2+	-
	4	IN2-	RX+	IN2-	-
	5	DIO1	TX-	IN1+	-
	6	DIO2	GND	IN1-	-
	7	ANV	ANI IN	CAN_H	-
	8	GND	GND	GND	-
	9	ANI	ANI OUT-	CAN_L	-
	10	TX	TX+	TRX-	-
	11	GND	ANI OUT+	GND	-
	12	RX	ANV	TRX+	-
<b>Power</b>	1	V+	V+	V+	-
	2	V+	V+	V+	-
	3	N/C	N/C	N/C	-
	4	CHASSIS	CHASSIS	CHASSIS	-
	5	N/C	N/C	N/C	-
	6	GND	GND	GND	-
	7	GND	GND	GND	-

Table 12: Pin numbers, functionality, and standard cable wire colors for AU series actuators.

## Cable Requirements and Specifications



**WARNING:** The electrical interface and pinouts vary between the actuator controller models. Using an improperly wired cable will damage to the actuator. Always be aware of your controller and its electrical interface before using the actuator.

### Using Ultra Motion Standard Cable Assemblies

---

All Ultra Motion manufactured cables are designed specifically for the actuator and should fulfill all electrical requirements. If for any reason our off-the-shelf cable does not work for your application, please see Using Custom Cable Assemblies below.

If using a sealed Servo Cylinder, such as A2, AM, or AU series, carefully read and follow the requirements of each series to ensure proper sealing of the connectors. This section covers critical requirement including proper coupling torque of the connector and proper sealing of the flying lead side of the cable.

The CBL-AM-DEV cables are not meant to be used in applications requiring liquid protection.

### Using Custom Cable Assemblies

---

#### *General Requirements for Custom Cable Assemblies*

---

If you choose to develop your own cable solution, please follow the requirements and considerations in this section.

Requirements:



**Use all power pins.** Due to maximum current ratings, power input to the Servo Cylinder requires the use of *all* power pins. Ensure that all four (4) pins are used, two (2) power in (V+) pins and two (2) power ground (GND) pins.



**Use appropriate wire gauges.** We recommend 20 AWG or heavier for power, 24 AWG for serial connections, and 22 AWG for all other signal connections.



**Do not daisy chain.** Servo Cylinders should not be strung together by their power pins, or “daisy chained”.



**Properly seal the cable assembly.** For ingress protected applications, it is necessary that the cable assembly is sealed properly, since the connectors themselves can be a point of ingress. This includes:

- Using an appropriate cable jacket material for the sealed environment.
- Proper sealing between the cable jacket and the signal/power sources.
- Proper sealing between the cable body and cable’s connector plug. Proper sealing between the cable’s connector plug and the Servo Cylinder’s connector receptacle. This can be achieved by following “Installing Cables on Sealed Servo Cylinders” on page 40.

## Connectors by Series

This section details the type of connector used on each actuator. Use this information to determine the connector which you will use to electrically couple to your cable. To purchase a pre-fabricated cable, see page the cabling section on page 18.

Series	Power Connector	Signal Connector
<b>A1</b>	Molex Micro-Fit 3.0, Dual-Row, 16-Pin (Single connector carries both Power & Signal connections) Part No: 0430451600	
<b>A2</b>	Amphenol C091 M16 flange-mount male metal receptacle, 4-pin Part No: C091 11S004 000 2	Amphenol C091 M16 flange-mount male metal receptacle, 12-pin Part No: C091 11S012 000 2
<b>AM</b>	MIL-D-38999 Wall Receptacle, Electroless Nickel, B-Shell, 5-pin, Normal Keying Part No: D38999/20FB5PN	MIL-D-38999 Wall Receptacle, Electroless Nickel, B-Shell, 13-pin, Normal Keying Part No: D38999/20FB35PN
<b>AU - Standard</b>	SubConn Micro-Circular, 5-pin Part No: MCBH5MAS	SubConn Micro-Circular, 8-pin Part No: MCBH8MAS
<b>AU - Custom</b>	Teledyne Impulse Metal Key Titan Series, 7-pin Part No: MKS(W)-307-FCR	Teledyne Impulse Metal Key Titan Series, 12-pin Part No: MKS(W)-3XL12-FCR

Table 13: Connector receptacles and respective part numbers used on the Servo Cylinder, by series.

## Assembling Custom Connectors for A1 Series Actuators

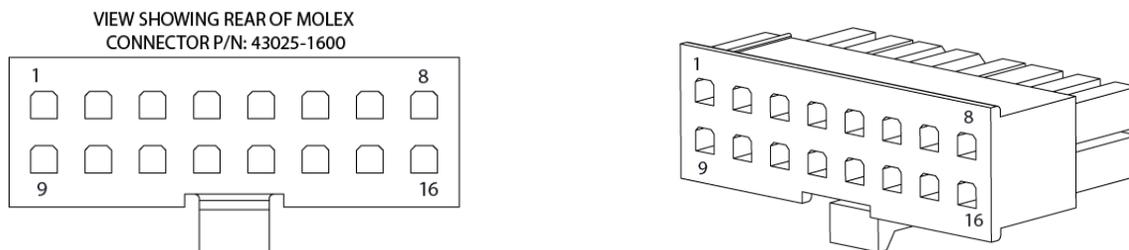
Use the information in this section to assemble your own cable and connector. The part numbers for the required Molex parts are listed below:

Part	Molex Part No
Cable-Side Connector	43025-1600
Crimping Tool	63819-000
Crimp Pins (20-24 AWG)	43030-0008
Pin Extraction Tool	11-03-0043

Table 14: Molex part numbers for components necessary to assemble a custom connector for A1 Series actuators

The illustrations below provide the mating connector pinout and show proper orientation when inserting pins into a connector. Be sure that high current power connections (PWR and GND) are both used in parallel to avoid excessive current on the power input pin.

**Note:** The pin barbs should always be on the side of pins as they are inserted into the socket



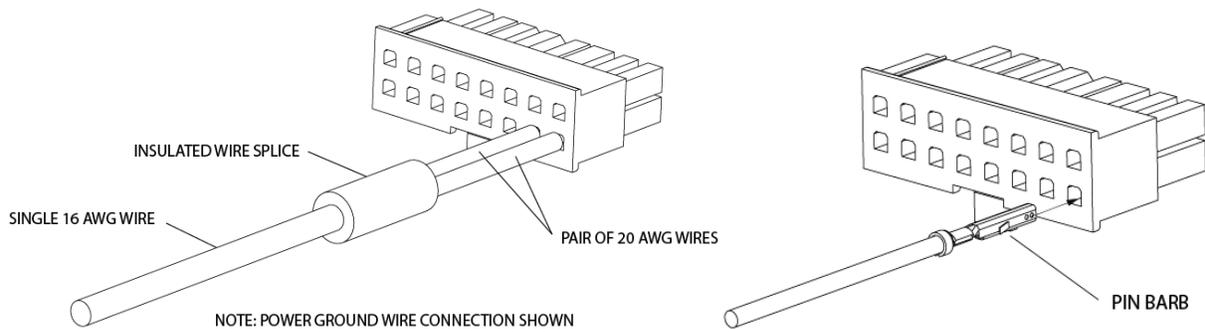


Figure 9: Diagrams of A1 series connector assembly

Be sure to properly insulate your wire splices. The illustration above shows an example of a proper wire splice going into the two power GND pins.

## Electrical Input and Output Specifications

### Power Bus (V+)

**Controller:** All Controllers

**Modes:** All Modes

**Rating:** 8 VDC to 36 VDC Recommended (43 VDC absolute max)

**Ground To:** GND (Two (2) grounding connections required). Must not use GPIO (DIO1/DIO2) grounds.

**Max Current:** See product datasheets for continuous and peak current ratings for your model.

The Servo Cylinder should be powered with anywhere between 8 VDC and 36 VDC, preferably sourced from an unregulated power supply (Refer to Power Supply Requirements and Wiring Details below for further recommendations). Using higher voltages will enable the actuator to achieve higher maximum speed. Maximum force is not related to supply voltage.



**WARNING:** Due to maximum current ratings, power input to the Servo Cylinder requires the use of a total of four (4) inputs (two (2) V+ inputs and two (2) GND connections). It is required that current is carried on all of these inputs, such that both V+ pins and respective grounds carry current equally in parallel.



**WARNING:** Controller electronics will be damaged if your bus voltage ever exceeds 51 VDC, and the actuator will enter a dynamic braking mode if voltage exceeds 44 VDC. Note that this includes fluctuations due to self-induced Back-EMF. Read section “Back-EMF (Voltage Fluctuations) and Back-Driving” below to completely understand voltage fluctuation before operating.



**WARNING:** Do not reverse polarity on the power input. If poles are reversed, internal protection circuitry will result in the two poles being shorted, potentially causing damage to the actuator and power supply.



**WARNING:** Never “Hot Plug” the actuator. Hot plugging is connecting or disconnecting the connector to the actuator when the any of the wires in the cable are powered.

### Back-EMF (Voltage Fluctuations) and Back-Driving

During normal operation it is typical to see fluctuations or spikes in your power bus voltage which are the result of voltage draws and regenerations from the motor. The most likely drivers of severe voltage spikes are:

1. Rapid acceleration and deceleration
2. Back-driving the actuator. Back-driving occurs when the actuator has sufficient load applied to it to move the shaft and spin the motor. A scenario in which this might occur is if a non- self-locking actuator is lifting a mass in vertical direction while electrical power is lost, causing the lifted weight to force down on the actuator shaft, resulting in back-driving.

The Servo Cylinder has an automatic dynamic braking feature that will attempt to clip voltage spikes at 44 VDC. When this happens, the energy is dissipated in the motor windings as heat. When this feature engages, you will notice the

actuator stutter as the feature flutters on and off. If normal operation results in this feature engaging, we strongly recommend you remedy the issue by any of the following means:

1. Adding a power shunt to your system (see Using a Power Shunt on page 34).
2. Using an unregulated power supply (see Power Supply Requirements and Wiring Details below).
3. Using an Ultra Motion PS-1X series Power Supply Unit, which is an unregulated power supply that features an integrated power shunt. See Power Supply Requirements and Wiring Details below.

### *Power Supply Requirements and Wiring Details*

---

**Voltage Rating:** 8 VDC to 36 VDC Recommended (43 VDC Absolute Max)

**Power rating:** >180W, smaller supplies are suitable for less demanding applications

We strongly recommend using an Ultra Motion PS-1X series Power Supply Unit to power the Servo Cylinder. The PS-1X Series PSUs have been specifically designed for use with Servo Cylinder and other intensive motion control applications. Benefits of the PS-1X Series PSUs over other PSUs include:

- Unregulated power conversion (greatly preferred over regulated for motion control applications, see below).
- Integrated Over-Voltage Protection (OVP) device eliminates the need for a power shunt.
- High Voltage (36 VDC), gets highest speed performance out of Servo Cylinder.
- High-efficiency regenerating supply, >80% efficient at draws 10-100% of capacity, quiescent power <5 W.
- AC- and DC-side fuses: DC-side fuse isolates the capacitor when blown, removing need for external fuse.
- Robust design, dust resistant (IP50), EPO Ready, all in integrated package.

A third-party PSU or existing power system may also be used if it meets the power supply requirements. When considering power systems, take into consideration the tradeoffs of regulated vs unregulated power supplies:

- Unregulated Power Supplies (Preferred): An unregulated power supply is preferred for servo applications because of its ability to supply bursts of energy during acceleration, and its ability to absorb energy during deceleration/back-driving events. The large smoothing capacitor of a properly sized unregulated supply maintains safe actuator voltage levels when absorbing/supplying large amounts of electrical energy. The use of an unregulated power supply reduces but does not eliminate the need for a power shunt.
- Regulated Power Supplies: Many popular benchtop power supplies are regulated, which is acceptable for applications that don't require high accelerations. Regulated power supplies cannot absorb the energy created during deceleration or back-driving events as effectively as unregulated supplies, which can cause the supply voltage to quickly rise to unsafe levels. The power supply can also fall below required voltage levels due to high current demands during acceleration or high force events. A power shunt can reduce the risk of over-voltage in a regulated power supply system.

The power system should also have following safety features, shown in the circuit below:

- Emergency Power-Off (EPO) switch: Generally, this is a "push-lock" type NC switch or relay which serves the purpose of immediately removing power from the system. An AC power cut-off is generally inadequate for rapidly removing power to the actuator due to stored energy in power supply capacitors.
- External Fuse: A slow burning fuse should be used on the positive leg of the DC power supply to isolate the Servo Cylinder from the DC power stored in capacitors in an overloading condition.
- Power Shunt: A power shunt, described above, must not be isolated by a blown fuse or EPO switch.

The Ultra Motion Power Supply Unit is has an integrated power shunt, fuse, and is EPO switch ready. Please ensure you have properly designed your power system and addressed all safety concerns.

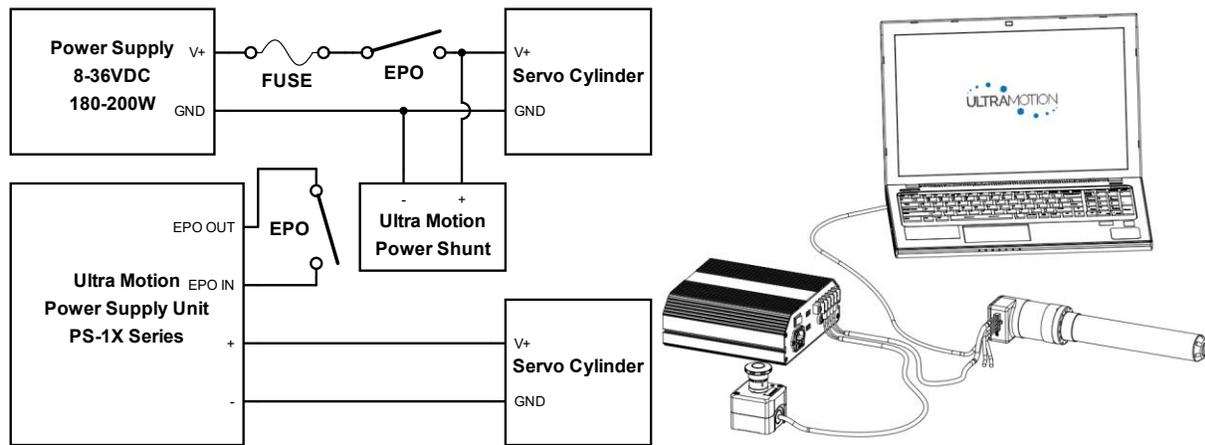


Figure 10: Power Wiring Diagrams. The top circuit is a diagram of a complete power supply setup, featuring a fuse, and Emergency Power-Off (EPO) switch, and a power shunt. The bottom circuit is an Ultra Motion Power Supply Unit configuration.

### Using a Power Shunt

A Power Shunt, such as the Ultra Motion Power Shunt, is a type of Over-Voltage Protection (OVP) device. They protect a power bus by engaging at a predefined voltage level and dissipating energy across a high-power shunt resistor until voltage is restored to nominal levels. This protects the Servo Cylinder and other equipment on the same power bus from damage due to excessive voltage spikes and back-EMF. The use of a power shunt is always recommended for safety, but especially important in the following situations:

- Applications which require hard/fast acceleration and deceleration.
- Applications where the load is assisting the actuator's motion (spring, gravity in a vertical application, etc.).
- Applications which require the use of a regulated power system.
- Applications where the actuator must share a power bus with other sensitive equipment.

Power shunts must be wired in parallel with the output of the DC power supply. In order to be effective, power shunts must maintain connectivity to the actuator in the event of a loss of power, blown fuse, or engaged Emergency Power-Off (EPO) switch. See Figure 10: Power Wiring Diagram.

NOTE: A power shunt is not necessary when using an Ultra Motion PS-1X series PSU, which has integrated OVP.

### Analog Current Input (ANI)

**Controller:** "N" – Classic Controller, "P" – Industrial Controller

**Modes:** Analog Current (4-20 mA) Control, Toggle (Simple On/Off for Extend/Retract) Control

**Rating:** 4 mA to 20 mA recommended

**Ground To:** Any GND

Used to receive a 4-20 mA analog current signal, which can be used to control actuator position or control either speed or torque in toggle mode.

Care should be taken to ensure analog signals are properly shielded against EMI, and are routed separately from power wires and other sources of interference. When possible, a 100% foil shield should be used and the shield electrically connected to chassis ground.

 The analog current input should be connected to a suitable analog current transmitter. Connecting a voltage directly across this input will damage the actuator, the input must be sent a controlled 4 to 20 mA signal.

## Analog Voltage Input (ANV)

**Controller:** "N" – Classic Controller, "P" – Industrial Controller

**Modes:** Analog Voltage (-10 VDC to +10 VDC) Control, Toggle (Simple On/Off for Extend/Retract) Control

**Rating:** -10 VDC to +10 VDC recommended

**Ground:** Any GND

Used to receive an analog voltage signal, which can be used to control actuator position or control either speed or torque in toggle mode.

For the best position accuracy results, we recommend that each actuator be calibrated individually to the analog voltage signal due to small differences which may exist in the signal processing instrumentation from actuator to actuator.

Care should be taken to ensure analog signals are properly shielded against EMI, and are routed separately from power wires and other sources of interference. When possible, a 100% foil shield should be used and the shield electrically connected to chassis ground.

## Optically Isolated Digital Inputs (IN1+, IN1-, IN2+, IN2-, IN3+, and IN3-)



**WARNING:** These electrical connections have different ratings and functionality depending upon which controller is being used. Always be aware of your controller and your electrical interface before setup.

**Controller:** "N" – Classic Controller, "P" – Industrial Controller, "C" – CAN Controller

**Modes:** Most or All, depending upon controller and features used

For "N" – Classic Controller: 3.3 VDC to 7 VDC, or more when using an appropriate current limiting resistor (See Using Higher Voltages for Digital Inputs on "N" – Classic Controllers below)

**Rating:** For "P" – Industrial Controller: 12 VDC to 24 VDC recommended, 5 V absolute min to 36 V absolute max

For "C" – CAN Controller: 12 VDC to 24 VDC recommended, 5 VDC absolute min to 36 VDC absolute max

**Ground:** INX- for each INX+.

The Servo Cylinder's digital inputs are optically isolated, meaning that they are electrically isolated from the rest of the circuit and therefore do not share a common ground. This is to provide optimal noise rejection. It is acceptable to connect IN1- or IN2- to actuator ground in order to simplify wiring, but this will remove the benefit of the noise rejection from optical isolation.

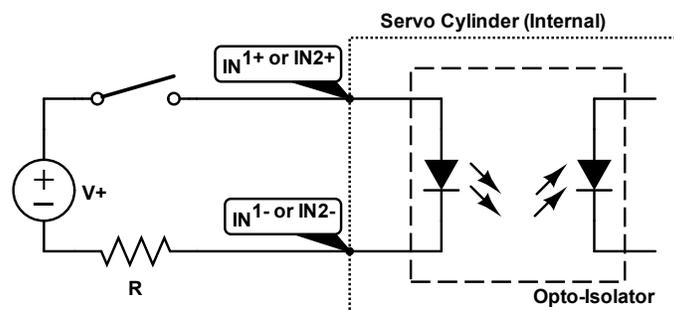


Figure 11: Schematic for Optically Isolated Digital Inputs. "R" is a current limiting resistor, which is necessary for using voltages in excess of 7 volts on the "N" – Classic Controller.

## Using Higher Voltages for Digital Inputs on "N" – Classic Controllers

For the "N" control board, the digital inputs need to be energized with between 5 mA and 15 mA of forward current to be activated. This can be achieved with a voltage between 3.3 and 7 volts (DC). A voltage in excess of 7 volts can be used if it is in conjunction with an appropriately sized current limiting resistor. The range of resistor sizes can be calculated as a function of voltage using the equations on page 36:

$$R_{min}[\text{ohms}] = \frac{V[\text{volts}] - 1.8}{.015} - 285, \quad R_{max}[\text{ohms}] = \frac{V[\text{volts}] - 2}{.005} - 315$$

Equation 1 and 2: Upper and Lower bounds for sizing the current limiting resistor when using voltages in excess of 7V with the "N" – Classic Controller

A voltage of +5 V and/or a ground can be supplied by the Servo Cylinder's on-board GPIO pins. Suggested wiring for the optically isolated inputs is shown in Figure 11 on page 35.

Forward Voltage [VDC]	Resistor size [ $\Omega$ ]
9	400
12	700
24	1900
28	2300
36	3100

Table 15: Current limiting resistor sizes for common DC voltages

## Serial Communication Inputs and Outputs (TX, TX+, TX-, RX, RX+, RX-, TRX+, TRX-)



**WARNING:** These electrical connections have different ratings and functionality depending upon which controller is being used. Always be aware of your controller and your electrical interface before setup.

"N" – Classic Controller (For TX and RX)

**Controller:** "P" – Industrial Controller (For TX+, TX-, RX+, RX-)

"C" – CAN Controller (For TRX+ and TRX-)

**Modes:** Command Line Interface (CLI) Control Over Serial, and serial communication in all other modes

**Baud Rate:** 2400 to 256000 (Default: 115200)

**Ratings:** Per RS232, RS422, and RS485 standards

**Ground:** Any GND

Depending upon the controller model, the serial connection to the actuator uses one of three different serial protocols, RS232 for the "N" Controller, Full-Duplex RS422 for the "P" controller, and Half-Duplex RS485 for the "C" controller. To function, all serial connections must share a common ground.

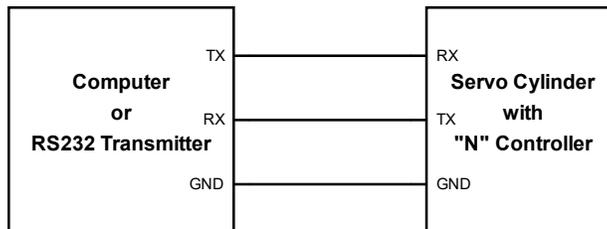


Figure 12: Circuit Diagram for RS232 Transmitter

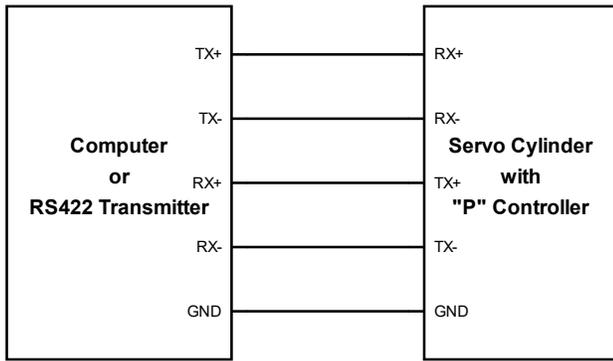


Figure 13: Circuit Diagram for RS422 Transmitter in full-duplex, as used by the Servo Cylinder "P" – Industrial Controller

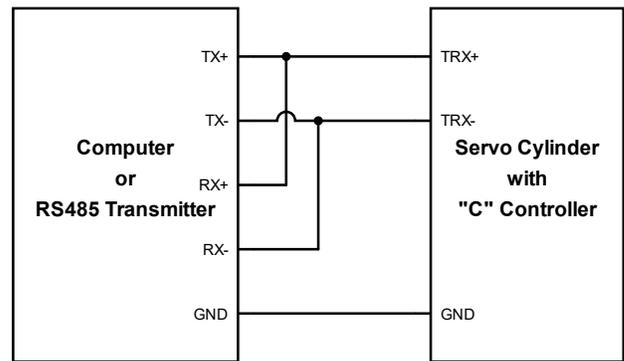


Figure 14: Circuit Diagram for RS485 Transmitter in half-duplex, as used by the Servo Cylinder "C" – CAN Controller

## CAN Connections (CAN\_H and CAN\_L)

**Controller:** "C" – CAN Controller

**Modes:** CANopen (CiA DS301/CiA DS402) Control, Configurable CAN 2.0B protocol

**Ratings:** Per CANopen (CiA DS301/CiA DS402) standards

**Ground:** Any GND

The CAN control connections CAN\_H and CAN\_L follow the CANopen specifications (CiA DS301/CiA DS402).

## General Purpose Input / Output (GPIO) (DIO1 and DIO2)



**WARNING:** Damage will occur if input voltage exceeds 5 VDC or if 5 VDC output is shorted to GND

**Controller:** "N" – Classic Controller

**Modes:** None, used only for additional features

*As an input:* 0 to 5 VDC

**Rating:** *As an output:* +5 VDC, up to 250 mA

*As a ground:* Up to 250 mA

**Ground:** Any GND

The Servo Cylinder with the "N" – Classic Controller has two (2) GPIO pins which can be used for a wide variety of tasks. It is important to note that DIO1 and DIO2 are not optically isolated, meaning that they may be more prone to inject inadvertent noise into the system.

The GPIO pins can be configured to be used as a high impedance digital input (no standard functions as an input), or to output any of the following:

- +5 VDC supply (up to 250 mA)
- Ground (sink up to 250 mA)
- 1 kHz PWM waveform proportional to a variable set by ioPWM1 and ioPWM2 – PWM Output Signal Type for GPIO pin 1 and 2 (Respectively). Output variables include position, velocity, force, or bus voltage
- Limit switch signals (Triggers at specified travel distances). To use this feature, use the masked status register, see Reading the Status Register over GPIO Pins Using a Register Mask on page 65.
- Output the result of the masked status register (1 kHz refresh rate). See Reading the Status Register over GPIO Pins Using a Register Mask on page 65.

## Analog Current Output (ANIO+ and ANIO-)



**WARNING:** The power supply used to power the analog current output loop **MUST** be isolated from any other voltages used by the actuator. If the bus voltage (the voltage used to power the actuator) or a non-isolated voltage is used to power the analog current loop, the actuator **WILL** be damaged. Read more below.

**Controller:** "P" – Industrial Controller

**Modes:** None, used only for Analog Current Output and feedback

**Ratings:** For Loop Voltage: 8 VDC to 36 VDC

**Ground:** See below

Servo Cylinders with the "P" – Industrial Controller have the ability to provide an isolated 4-20 mA current feedback for a wide variety of data sources such as absolute position feedback (see xOutSrc – Analog Current Output Data Source on page 55). To function, an isolated power supply must be used to provide a loop voltage. If your Analog current receiver has an integrated isolated loop voltage an external loop supply is not needed.

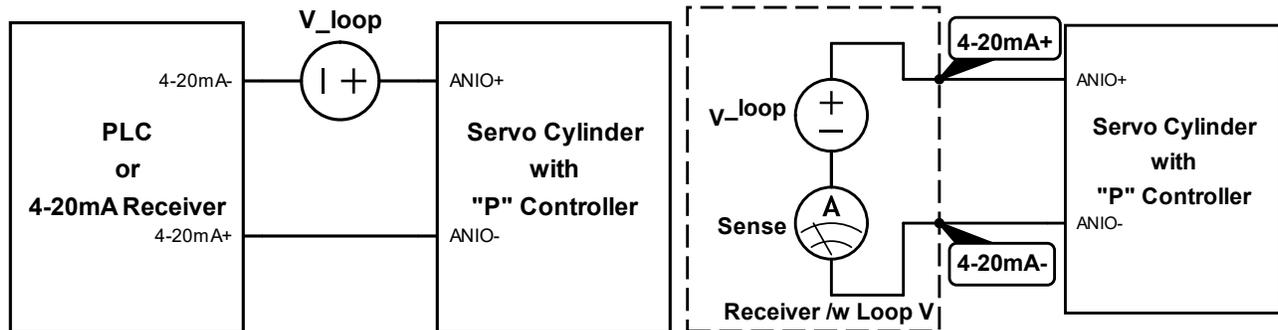


Figure 15: Schematics of Analog Voltage Output circuits. In the left schematic, an external isolated loop power supply is present. In the right schematic, a 4-20 mA analog current receiver is present with an integrated loop voltage supply.

# Using the Servo Cylinder

## Initial Setup

This section will cover how to start set up the Servo Cylinder, and power it for the first time.

### Getting Started

---

You must determine what control mode you wish to operate in before getting started. You must provide the Servo Cylinder with power and a compatible command signal. If you need assistance in determining which control mode is right for your application, please review the control modes section or contact one of our application engineers.

#### You will need:

- Servo Cylinder (A-series) Actuator
- Signal Cable
- Power Cable
- USB Mini-B Cable (only if configuring using USB Mass-Storage)
- Power Supply Unit (PSU) capable of 8-36 VDC ( $\geq 180$  W suggested, see PS-1X Series PSUs)
- A method of issuing a command signal (Such as serial, analog voltage, or analog current, etc.)

### Setting up the Servo Cylinder Hardware

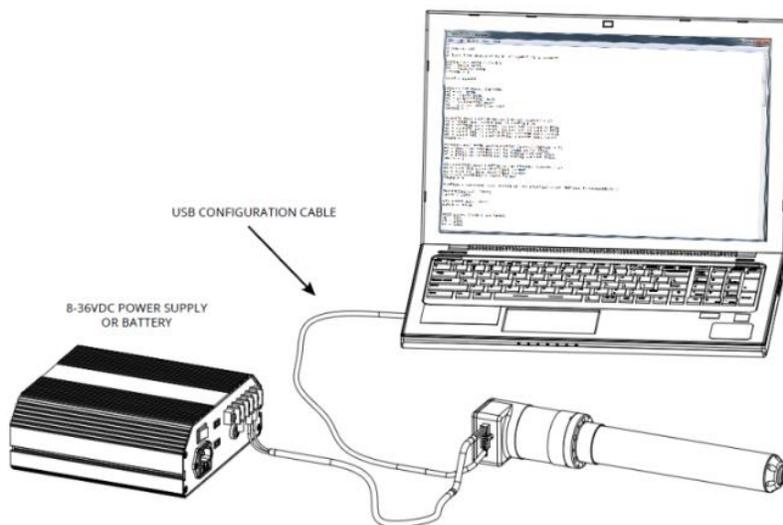
---



**WARNING:** Never “Hot Plug” the actuator. Hot plugging is connecting or disconnecting the connector to the actuator when the any of the wires in the cable are powered.

Begin by connecting all of the wires and cables first, and then turning on the power. To do this:

1. Connect the Signal cable to the system that will be issuing commands (in this example, we will connect to a Windows PC with an RS232-to-USB Converter).
2. Connect the Power Cable to the Power Supply Unit.
3. Connect the Power and Signal Cables to the Servo Cylinder:
  - a. *If using a Sealed Servo Cylinder (such as an A2, AM, or AU):* Read Installing Cables on Sealed Servo Cylinders on page 40.
  - b. *If using an A1 Series Servo Cylinder:* Connect the Power & Signal cable with retaining clip engaged.
4. *If configuring using USB Mass-Storage:*
  - a. USB configuration is not available for AU actuators. The AU must be preconfigured at the factory for your operating mode.
  - b. *If using an A1 Series Servo Cylinder:* Connect the USB Mini-B cable between the Servo Cylinder and your computer.
  - c. *If using an A2 or AM Servo Cylinder:* Follow the instructions on page 40
5. Turn on the Power Supply.



### Installing Cables on Sealed Servo Cylinders

This section only pertains to sealed actuators, such as A2, AM, and AU.

The full ingress protection rating of the actuator only applies if the actuator is fully and properly assembled, with all connectors and access panels in place. To install the connector, it is necessary to use an appropriate amount of coupling torque to ensure sealing, especially for the A2 series. See the table “Table 16: Connector coupling torques, by series” below.

Series	Power Connector Torque	Signal Connector Torque
<b>A2</b>	13.3 in·lbf [1.5 N·m]	13.3 in·lbf [1.5 N·m]
<b>AM</b>	12 in·lbf [1.35 N·m]	12 in·lbf [1.35 N·m]
<b>AU - Custom</b>	< 50 in·lbf [< 5.6 N·m]	< 50 in·lbf [< 5.6 N·m]
<b>AU – Standard</b>	See Below	See Below

Table 16: Connector coupling torques, by series

### AU – Standard Cable Installation

The standard AU SubConn connectors must be lubricated with Molykote 44 Medium on every installation. A layer of grease corresponding to a minimum of 1/10 socket depth should be applied to the female connector in dry-mate conditions. A layer of grease corresponding to 1/3 socket depth should be applied to the female connector in wet-mate situations. Angular loads on the cable and mated cable assembly must be minimized. Debris and mud on the male connectors and within the female connector sockets must be cleaned thoroughly before mating with liquid soap, hot water, and isopropyl alcohol. The orientation of the AU receptacles on the housing will be as shown with a tolerance of  $\pm 30^\circ$ .

Please contact Ultra Motion for additional details on handling the standard AU Servo Cylinder.

### Accessing USB Port for A2 and AM Servo Cylinders

#### Note: Configuration using Serial Commands

Configuration of the Servo Cylinders can be performed using specific serial commands as an alternative to USB access, with the exception of changing the operating mode. When re-seating any O-ring seal, there is always a risk that the O-ring will not seal properly. If it is practical to do so, we recommend configuration over serial to eliminate the risk of compromising the cover seal.

To access the Mini-USB port on A2 and AM series actuators, you must first remove the Access Port Cover, and replace it after you have finished configuration. *Cleanliness is critical*, and care must be taken to not compromise the seal in access port installation and removal.



**IMPORTANT:** Be sure to minimize the amount of time for which the USB port cover is removed, and avoid having it off for more than 1 to 2 hours. Both the A2 and AM series actuators contain a desiccant which becomes less effective at protecting the actuators when exposed to open atmosphere for an excessive period of time.

#### You will need:

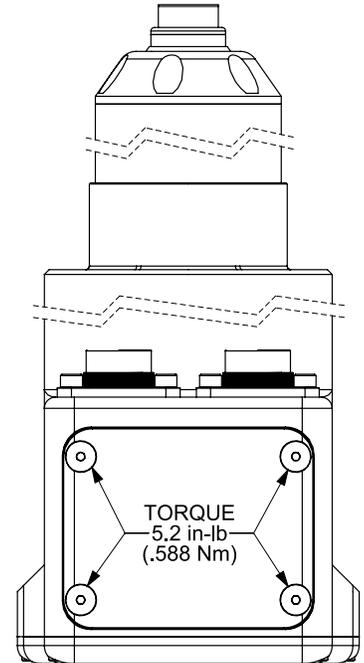
- A torque wrench capable of measuring 5.2 in·lbf [.588 N·m]
- A 1/16" hex bit for a Torque wrench

#### To Remove the Access Port Cover:

1. Remove the four (4) USB Cover screws
2. Ensure that the O-ring is kept clean of dust and debris while the panel is removed. Also ensure that the cover or connector covers are not removed for too long. The O-ring should have a light coat of grease on it (from factory).

#### To Re-Install the Access Port Cover:

1. Inspect the O-ring, ensuring it is clean and intact. If dirty, clean the O-ring, groove, and the faying surface on the cover with acetone. Then re-grease the O-ring with nitrile/Buna-N compatible grease such as Super Lube® High Temperature, Extreme Pressure Grease.
2. Place the access port cover. Ensure that the O-ring has not been pinched.
3. Install all four screws finger tight. *Do not apply significant torque yet.*
4. In a cross pattern (e.g. top left, then bottom right), carefully torque each of the four (4) screws to 5.2 in·lbf [0.588 N·m].



## Configuration

All parameters in the servo cylinder are set by editing configuration variables, which can be performed by using the serial commands, or by directly editing the CONFIG.TXT file using USB mass storage.

### How to Configure with Serial Commands

Every configuration setting used by the Servo Cylinder, besides the chosen operating mode, can be configured using a specific serial command, all of which are detailed in Appendix A: Serial Command Line Interface (CLI) List of Commands. If USB access is not possible in your application, or an AU is specified, please contact Ultra Motion engineering so that your operating mode can be preconfigured at the factory.

To learn how to use the Command Line Interface (CLI) Control Over Serial, see Using the Command Line Interface (CLI) on page 60. This method of editing and saving configuration variables via serial can greatly facilitate setup.

### How to Configure with USB Mass Storage

The Servo Cylinder has a small 10 kB section of non-volatile flash memory that is used to store configuration data. Do not use this memory to store backup copies of CONFIG.TXT and HARDWARE.TXT. Storing any files other than CONFIG.TXT and HARDWARE.TXT will exceed available storage space.

#### **Note: HARDWARE.TXT and CONFIG.TXT must be present to operate**

The Servo Cylinder's microcontroller needs both HARDWARE.TXT and CONFIG.TXT to operate. Deleting, removing, or renaming these files or deleting variables within the files will result in a fault. We recommend creating a backup of these files to your local drive before starting to edit them. Do not back up on Servo Cylinder.

#### *Hardware Specific Variables (within HARDWARE.TXT) – Not Intended for Editing*

HARDWARE.TXT contains hardware specific information (such as FOC gains [which are different than the actuator PID gains in CONFIG.TXT], sensor calibration information, etc.). This file is not intended to be edited by the user unless

suggested by Ultra Motion application engineers after a consultation. This file is specific to each actuator and cannot be copied between multiple Servo Cylinders. If for some reason this file is lost, it can be automatically generated by running a self-calibration routine through the Command Line Interface, or requested from Ultra Motion. Thoroughly read the section on running the calibration routine for important safety considerations

```

rotor=164
adc1z=512
adc2z=512
adc3z=513
encRef=11206933
rPos=1024
ePos=25317
qP=2000
qI=200
dP=2000
dI=200
sNum="10107287"

```

Figure 16: Example of a HARDWARE.TXT file. Several of these values are specific to each individual actuator.

### Editing User Configuration Variables (Within CONFIG.TXT)

This file contains user configurable variables that set operating behavior (operating mode, position limits, input ranges, etc.). CONFIG.TXT can be edited manually through a text file editor like Notepad. Many of the configuration variables that exist within CONFIG.TXT can be edited over serial using the suite of serial commands detailed in Appendix A: Serial Command Line Interface (CLI) List of Commands.

```

# CONFIG.TXT
# Each line prefaced by a "#" symbol is a comment

#Interface Mode (ifMode)
#0 = Human Mode
#1 = Machine Mode
ifMode = 0

baud = 115200

#Operating Modes (opMode)
#0 = CLI mode
#1 = toggle mode
#2 = proportional mode
#3 = incremental mode
#4 = pre-set position mode
opMode = 4

#Toggle mode configuration (tmCFG) (opMode = 1)
#0 = speed and torque set in config file

```

```

# CONFIG.TXT
# Ultra Motion
# Servo Cylinder
# Firmware: 0.956
# Serial#: 10106326
#
ifMode=0
baud=256000
opMode=0
tmCFG=0
pmCFG=0
imCFG=0
spMin=2048
spMax=15550
kp=1200
ki=250000
kd=10000
maxTorg=28000
mtMin=8000
maxSpeed=50000000
nsMin=200000
accel=50000
stopMode=0
stepSize=1
pMin=66000
pMax=115000

```

Figure 17 and 18: CONFIG.TXT file, as default (left) and after running ZW which removes most comments (right)

## How to configure variables within CONFIG.TXT (Windows)

1. Once the Servo Cylinder is connected via Mini-USB and powered, the computer will automatically recognize the Servo Cylinder as a Mass Storage device (similar to a flash drive).
2. Browse to “My Computer” and open the new storage drive labeled “Ultra Motion”.
3. Locate the text file “CONFIG.TXT” and open it in a text editor.
4. Edit configuration variables to set the actuator to behave as desired. Refer to the following sections for details
5. Save the file locally to your windows machine, and then copy it to the Ultra Motion drive to OVERWRITE the CONFIG.TXT file currently on the Servo Cylinder. If there is no existing CONFIG.TXT file, simply copy this new file onto the drive.
6. Once your CONFIG.TXT file is properly configured for your application, this file can be copied across multiple Servo Cylinders. If you need to replace a Servo Cylinder, simply copy the existing CONFIG.TXT file to the new Servo Cylinder and it is ready to go.

## Default Factory Settings

---

The Servo Cylinder is pre-configured to operate in Command Line Interface (CLI) by default. We recommend you configure the actuator for your application, but you can skip these setup steps if you want to just run the actuator in Command Line Interface (CLI) with default settings and travel limits.

These default settings can be changed once you are comfortable using the product and understand the repercussions of changing these values.

Default setting values can be found in the description for each setting.

## Determining Configuration Variables Relevant to Your Control Mode and Features

---

In order to use the Servo Cylinder, it is necessary for it to be configured properly for your application. To help determine which configuration variables are necessary for proper operation of the actuator, please use the following guidelines

1. Determine exactly which operating mode (such as Command Line Interface (CLI) Control Over Serial, Analog Voltage (-10 VDC to +10 VDC) Control, etc.) you will be using.
2. Select exactly which features you will be using (Analog Current (4-20 mA) Output, Data Streaming, etc.).
  - When first getting started, we strongly recommend using Error Mode to reduce the risk of accidental damage to the actuator. Read more about this feature in [errMode – Error Mode \(Actuator Hardware Protection Feature\)](#) on page 53.
3. Properly configure all General Settings within “General Configuration Settings” on page 45.
4. Using Table 17, determine which configuration variables are critical for your control mode, and then configure them in Control Mode Specific Configuration Settings which starts on page 47
5. Determine which configuration variables are critical for features being used in Table 17, and then configure them in Actuator Feature Configuration Settings which starts on page 53.

	Control Mode or Feature	Controller	Required Settings (in addition to General)
Control Modes	Command Line Interface (CLI) Control Over Serial *C-Controller Does Not Command Motion	All	baud, maxSpeed, accel, ifMode
	Analog Voltage (-10 VDC to +10 VDC) Control	N, P	<u>For all controllers:</u> vMin, vMax, vFilter <u>"N" controller only:</u> pmCFG
	Analog Current (4-20 mA) Control	N, P	<u>For all controllers:</u> cMin, cMax, cFilter <u>"N" controller only:</u> pmCFG <u>"P" controller only:</u> revSgnl
	1-2 ms Pulse (RC PWM Servo) Control	N	pMin, pMax, pFilter, pmCFG
	Step & Direction (Simulated Stepper) Control	N	imCFG, stepSize
	A/B Quadrature Input Control	N	imCFG, stepSize
	CW/CCW (Simulated Stepper) Control	N	imCFG, stepSize
	Toggle (Simple On/Off for Extend/Retract) Control	N	maxSpeed, Accel, stopMode, tmCFG, mtMin, msMin
	Pre-set Position Control	N	maxSpeed, Accel, stopMode, posTime, Pos1, Pos2, Pos3, Pos4
Actuator Features	Serial Communication (for non-motion commands)	All	baud
	Error Mode	All	errMode, ovErrP, ovTorq
	Status Bit Thresholds	All	<u>For all controllers:</u> atTarg, ovErrP, ovSpeed, ovTorq, posGrtr, posLess <u>"P" controller only:</u> ovHumid
	Isolated Analog Current (4-20 mA) Output (Configurable Data Source)	P	xOutSrc xOutMin, xOutMax, cOutMin, cOutMax, revCout
	Data Streaming	All	baud, sFlag, sTime, strx1, strx2, strx3
	General Purpose Input / Output (GPIO)	N	ioPin1, ioPin2, ioPWM1, ioPWM2, ioBit1, ioBit2

Table 17: All configuration settings which pertain to specific control modes and actuator features. These settings must be configured in addition to all General Configuration Settings (on page 45).

## Configuration Setting Definitions

This section completely outlines and defines all of the configuration settings. It is divided into 3 sections.

1. **General Configuration Settings:** This section details the settings used in all control modes, and which must be set properly to achieve desired operation and performance.
2. **Control Mode Specific Configuration Settings:** This section details the configuration settings which pertain to one or more specific to certain control modes. For the settings in this section, only the settings used by your chosen control mode must be configured.
3. **Actuator Feature Configuration Settings:** This section details the configuration settings which are used by the features offered for additional functionality. These settings are necessary to configure depending on which additional features you plan on using.

## General Configuration Settings

Configuration variables in this section are used in all control modes of the actuator, and must all be properly configured. Refer to Table 17 to determine which additional settings need to be configured in addition to the settings in this section.

### opMode – Operating Mode



**WARNING:** This setting has different functionality depending upon which controller is being used. Always be aware of your controller and your electrical interface before setup or use of the Servo Cylinder.

**Controllers:** All Controllers

**Modes:** All Modes

**Variable Type:** Integer

**Valid Range:** For “N” – Classic Controller: 0 to 4  
For “P” – Industrial Controller: 0 to 2

**Default:** 0

This setting sets the Servo Cylinder’s operating mode. Available operating modes vary based upon the actuator controller. Use the following table for setting definitions

#### “N” Classic Controller:

Setting	Control Mode(s)	Description
0	Command Line Interface (CLI) Control Over Serial	Control of actuator CLI using command language, which includes a command set for position moves, trajectory moves, acceleration, speed, torque, etc.
1	Toggle (Simple On/Off for Extend/Retract) Control	Extend/Retract control with adjustable acceleration, speed, and force. See “tmCFG – Toggle Mode Configuration” for additional configuration.
2	Proportional Modes: <ul style="list-style-type: none"> <li>Analog Current (4-20 mA) Control</li> <li>Analog Voltage (-10 VDC to +10 VDC) Control</li> <li>1-2 ms Pulse (RC PWM Servo) Control</li> </ul>	Actuator moves proportionally to the command signal. See “pmCFG - Proportional Mode Configuration” for additional configuration options.
3	Incremental Modes: <ul style="list-style-type: none"> <li>Step &amp; Direction (Simulated Stepper) Control</li> <li>A/B Quadrature Input Control</li> <li>CW/CCW (Simulated Stepper) Control</li> </ul>	Actuator moves as it receives each incremental command signal. See “imCFG - Incremental Mode Configuration” for additional configuration options.
4	Pre-set Position Control	Actuator has up to four pre-set positions. See “Pos1, Pos2, Pos3, and Pos4 – Preset Positions 1, 2, 3, and 4” to define the positions.

#### “P” Industrial Controller:

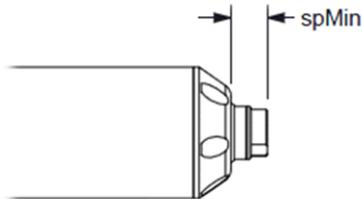
Setting	Control Mode(s)	Description
0	Command Line Interface (CLI) Control Over Serial	Control of actuator CLI using command language, which includes a command set for position moves, trajectory moves, acceleration, speed, torque, etc.
1	Analog Current (4-20 mA) Control	Actuator moves proportionally to a 4-20 mA analog current signal.
2	Analog Voltage (-10 VDC to +10 VDC) Control	Actuator moves proportionally to an analog voltage signal.

Table 18: All operating modes available to the Servo Cylinder.

## spMin - Software Position Minimum

- Controllers:** All Controllers  
**Modes:** All Modes  
**Variable Type:** Integer  
**Valid Range:** rPos to ePos (Note: rPos and ePos are the physical travel limits, found in HARDWARE.TXT)  
**Rules:**  $rPos \leq spMin < spMax \leq ePos$   
**Default:** 2048  
**Set Command:** LN

This configuration variable sets the minimum allowable position of the Servo Cylinder. The variable spMin can be treated as a retracted software limit switch. The value is expressed in Phase Index sensor counts and there are 1024 counts per revolution.



### Note: Stroke vs. Software-Limited Travel Range

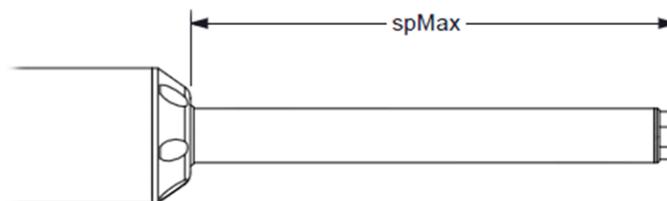
The stroke length is defined as the full distance between the hard end-stops (physical travel limits) of the actuator. The positions of these limits are defined by the positions rPos (retracted end-stop) and ePos (extended end-stop). For safety and to account for trajectory overshoot, the actuators are shipped from the factory by default with the travel range limited to be slightly smaller than this full stroke. By default, spMin (the software retracted travel limit) is set 1024 counts (one screw rotation) higher than rPos, and spMax (the software extended travel limit) is set 1024 counts lower than ePos.

In general, it is possible to safely reclaim some of this stroke by reducing the size of both gaps between the hard end-stop and the software travel limit. Before doing this, we highly recommend that you measure any amount by which your actuator overshoots trajectory as part of normal operation. To safely re-define your software travel limits, include the full overshoot distance as well as a safe buffer into your gap between the hard end-stop and the software travel limit. Understand that hitting the hard end-stop can damage or destroy the actuator.

## spMax - Software Position Maximum

- Controllers:** All Controllers  
**Modes:** All Modes  
**Variable Type:** Integer  
**Valid Range:** rPos to ePos (Note: rPos and ePos are the physical travel limits, found in HARDWARE.TXT)  
**Rules:**  $rPos \leq spMin < spMax \leq ePos$   
**Default:**  $spMax = (ePos - 1024)$   
**Set Command:** LX

This setting sets the maximum allowable position of the Servo Cylinder. spMax can be treated as an extended software limit switch. The value is expressed in Phase Index sensor counts and there are 1024 counts per revolution.



## maxTorq - Max Torque

**Controllers:** All Controllers  
**Modes:** All Modes  
**Variable Type:** Integer  
**Valid Range:** 0 to 32767  
**Default value:** 10000  
**Set Command:** MT

This setting sets the limit for the current controlled by the FOC current loop, thereby limiting the maximum force produced by the Servo Cylinder. The value represents a percentage of force output whereas 32767 equals 100%. The relationship is linear with a slight offset do to unloaded running friction of the system. Contact Ultra Motion engineering for more detailed information.

## kp, ki, and kd – PID Gains $k_p$ , $k_i$ , and $k_d$ (Respectively)

**Controllers:** All Controllers  
**Modes:** All Modes  
**Variable Type:** Integer  
**Valid Range:** 0 to 268435455 (for practical limits, see Tuning Performance on page 66)  
**Default:**  $k_p = 1200$ ,  $k_i = 250000$ ,  $k_d = 10000$   
**Set Command:** KP, KI, KD

These three values represent the gains for the proportional, integral, and derivative terms of the position PID control loop. See the Tuning Performance section on page 66 for more information about what these values represent. Default factory values used represent nominal, stable gains.

## Control Mode Specific Configuration Settings

Configuration variables in this section are relevant to several (but not all) operating modes and must be set for proper operation of the Servo Cylinder in these modes.

### baud – Serial Baud Rate

**Controllers:** All Controllers  
**Modes:** Command Line Interface (CLI) Control Over Serial,  
All modes (*for non-motion serial commands and data streaming*)  
**Variable Type:** Integer  
**Valid Range:** 2400 to 256000  
**Default:** 115200

This setting sets the Baud rate for serial communication. Setting this is only necessary if you plan to use a serial connection to the actuator. Serial is accessible in all control modes and is used for streaming output information and diagnostics. Default Baud rate is 115200. Lower baud rates will be more tolerant to noise and crosstalk at the expense of data bandwidth.

## maxSpeed - Maximum Speed

**Controllers:** All Controllers  
**Modes:** Command Line Interface (CLI), Preset Position Mode, Toggle Mode  
**Variable Type:** Integer  
**Valid Range:** 0 to 50000000 (Practical upper limit ~8000000)  
**Default:** 2000000  
**Set Command:** SP

This variable limits the maximum speed the Servo Cylinder will reach when executing trajectory moves. Note that it is possible to set the max speed higher than what the motor is capable of producing. The default value used here represents approximately 25% of maximum speed depending on the operating voltage, load, and other factors.

To convert units, see Appendix B: Converting Between Sensor Counts and Physical Units on page 90.

### Note: Proportional Modes

The maxSpeed and accel variables do not control the top speed and acceleration of the actuator when in proportional modes, which include Analog Current, Analog Voltage, and 1-2 ms Pulse Modes. Maximum speed, acceleration and deceleration are controlled in proportional modes by controlling the rate of change (slew rate) of the command signal.

## Accel - Acceleration and Deceleration Rate

**Controllers:** All Controllers  
**Modes:** Command Line Interface (CLI), Preset Position Mode, and Toggle Mode  
**Variable Type:** Integer  
**Valid Range:** 0 to 131071  
**Default:** 5000  
**Set command:** AC

This setting defines the acceleration and deceleration the Servo Cylinder will use in trajectory moves. Note: the acceleration and deceleration must be equal to one another. To convert units, see Appendix B: Converting Between Sensor Counts and Physical Units on page 90.

### Note: Proportional Modes

The maxSpeed and accel variables do not control the top speed and acceleration of the actuator when in proportional modes, which include Analog Current, Analog Voltage, and 1-2 ms Pulse Modes. Maximum speed, acceleration and deceleration are controlled in proportional modes by controlling the rate of change (slew rate) of the command signal.

## stopMode – End-of-Trajectory Stopping mode

**Controllers:** “N” – Classic Controller  
**Modes:** Preset Position mode, and Toggle Mode  
**Variable Type:** Integer  
**Valid Range:** 0 to 2 (see table below)  
**Default:** 0  
**Set Command:** SM

This setting defines the behavior of the actuator when stopping at a preset position.

Setting	Behavior
0	Snap to position
1	Decelerate at rate set by “accel” variable
2	Dynamically brake for 100 ms, then snap to position

Table 19: End of Trajectory behaviors

## ifMode – Serial Interface mode

**Controllers:** All Controllers

**Modes:** Command Line Interface (CLI)

**Variable Type:** Integer

**Valid Range:** 0 to 2 (see table below)

**Default:** 0

**Serial Command:** IM

This setting changes the behavior of the Serial Command Line Interface to be either human mode or machine mode. For more information on how to use machine mode (which allows the use of checksums and acknowledgements, read Using Machine Mode for Acknowledgements and Checksum Validation on page 61).

Setting	Mode	Description
0	Human Mode	All commands are followed by a verbose acknowledgement in the serial prompt, including asynchronous error messages. Detailed information regarding actuator commands is written to the serial prompt
1	Machine Mode 1 (MACHINE1)	All commands are followed by a simplified acknowledgement (ACK/NACK). Asynchronous error messages are disabled. No detailed information is sent to the serial prompt. Echo is disabled
2	Machine Mode 2 (MACHINE2)	All commands followed by a simplified acknowledgement (ACK/NACK). Asynchronous error messages are disabled. No detailed information is sent to the serial prompt. Echo is disabled Every command must be followed by a valid checksum to be acknowledged. Note: If you are in Machine Mode 2, you can use the command "im0 26" (case sensitive) to set interface back to Human Mode.

Table 20: Serial Interface Modes

The primary difference between Machine Mode 1 and Machine Mode 2 is that Machine Mode 2 requires a checksum is sent with *all* commands (see Using Machine Mode for Acknowledgements and Checksum Validation on page 61). Both MACHINE1 and MACHINE2 modes return a checksum with the response, and a checksum is also added to all lines of streamed data.

### Checksums

A checksum is a single byte represented by two hexadecimal characters. It is added to the end of each line, including both commands and streaming data. It must be preceded by a delimiter (space, '\t', or comma) except when a command takes no argument. The checksum is the least significant byte of the sum of all bytes received in the line before the checksum.

## pmCFG - Proportional Mode Configuration

**Controllers:** "N" – Classic Controller

**Modes:** Analog Voltage (-10 VDC to +10 VDC) Control, Analog Current (4-20 mA) Control, 1-2 ms Pulse (RC PWM Servo) Control

**Variable Type:** Integer

**Valid Range:** 0 to 2 (See below for details)

**Default:** 1

This setting selects the type of proportional mode behavior

Setting	Mode
0	1-2 ms Pulse (RC PWM Servo) Control
1	Analog Voltage (-10 VDC to +10 VDC) Control
2	Analog Current (4-20 mA) Control

Table 21: Proportional Modes

### vMin and vMax – Analog Voltage Min and Analog Voltage Max (Respectively)

- Controllers:** “N” – Classic Controller, “P” – Industrial Controller  
**Modes:** Analog Voltage (-10 VDC to +10 VDC) Control  
**Variable Type:** Integer  
**Valid Range:** 0 to 65535\* (Note: this integer corresponds to sensor count, not volts)  
**Rules:**  $0 \leq vMin < vMax \leq 65536$   
**Default:** vMin = 33114 (~0 VDC), vMax = 63402 (~ +10 VDC)  
**Set Command:** VN, VX

These settings define the usable range of voltages used by the analog voltage control mode that will relate proportionally to the Servo Cylinder travel range (spMin to spMax).

#### Note: Analog Voltage Calibration

For optimal results, the user should calibrate vMin and vMax for each individual actuator.

### vFilter – Analog Voltage Filter

- Controllers:** “N” – Classic Controller, “P” – Industrial Controller  
**Modes:** Analog Voltage (-10 VDC to +10 VDC) Control  
**Variable Type:** Integer  
**Valid Range:** 0 to 65535  
**Default:** 10000  
**Set Command:** FV

The analog voltage input can be filtered in software to reduce the effect of noise from the transmitter. The larger the value the less filtering will occur, allowing for more dynamic movements. A smaller value will increase the level of filtering, leading to smoothed motion, less dithering, and slower response.

### cMin and cMax – Analog Current Input Minimum and Maximum (Respectively)

- Controllers:** All Controllers  
**Modes:** Analog Current (4-20 mA) Control  
**Variable Type:** Integer  
**Valid Range:** 0 to 65535 (Note: this integer corresponds to sensor count, not amps)  
**Rules:**  $cMin < cMax$   
**Default:** cMin = 3143 (~4 mA), cMax = 64106 (~20 mA)  
**Set Command:** CN and CX

These settings define the usable range of currents used by the analog current control mode that will relate proportionally to the Servo Cylinder travel range (spMin to spMax).

For optimal results the user should calibrate the minimum and maximum current range to each actuator, due to small differences in each actuator and signal system.

### cFilter - Current filter

- Controllers:** All Controllers  
**Modes:** Analog Current (4-20 mA) Control  
**Variable Type:** Integer  
**Valid Range:** 0 to 65535  
**Default:** 2000  
**Set Command:** FC

The analog current input can be filtered in software to reduce the effect of noise from the transmitter. The larger the value the less filtering will occur, allowing for more dynamic movements. A smaller value will increase the level of filtering, leading to smoothed motion, less dithering, and slower response.

## pMin and pMax – 1-2 ms Pulse (RC PWM Servo) Control Pulse Width Min and Max (Respectively)

**Controllers:** "N" – Classic Controller

**Modes:** 1-2 ms Pulse (RC PWM Servo) Control

**Variable Type:** Integer

**Valid Range:** 48000 to 70000 (for pMin) and 108000 to 132000 (for pMax)

**Default:** pMin = 66000, pMax = 115000

**Set Command:** PN, PX

This variable defines the usable range of the 1-2 ms Pulse (RC PWM Servo) Control inputs that will relate to the Servo Cylinder travel range (spMin to spMax). The Servo Cylinder can accept a 1 to 2 ms pulse input at both standard update frequencies of 50 Hz and 333 Hz. The variables for pMin/pMax use units of 1/60,000<sup>th</sup> of a millisecond. See Appendix B: Converting Between Sensor Counts and Physical Units on page 90, or use the below table for common pulse width values.

Pulse width [ms]	Value for pMin/pMax	Note
1.0000	60000	Exactly 1 ms
1.1000	66000	A common lower limit for some PWM controllers
1.9166	115000	A common upper limit for some PWM controllers
2.000	120000	Exactly 2 ms

Table 22: Commonly used pulse widths and associating pMin/pMax variables

## pFilter – 1-2ms Pulse (RC Servo) filter

**Controllers:** "N" – Classic Controller

**Modes:** 1-2 ms Pulse (RC PWM Servo) Control

**Variable Type:** Integer

**Valid Range:** 0 to 65535

**Default:** 3000

**Set Command:** FP

The 1 to 2 ms pulse input can be filtered in software to reduce the effect of noise from the transmitter. The larger the value the less filtering will occur, allowing for more dynamic movements. A smaller value will increase the level of filtering, leading to smoothed motion, less dithering, and slower response.

## revSgnl – Inversion of Response to Analog Current Input Signal

**Controllers:** "P" – Industrial Controller

**Modes:** Analog Current (4-20 mA) Control

**Variable Type:** Integer

**Valid Range:** 0 or 1

**Default:** 0

**Set Command:** IR

Inverts the response of the actuator with respect to the analog current input signal.

## imCFG - Incremental Mode Configuration

**Controllers:** "N" – Classic Controller

**Modes:** Step & Direction (Simulated Stepper) Control, A/B Quadrature Input Control, CW/CCW (Simulated Stepper) Control

**Variable Type:** Integer

**Valid Range:** 0 to 2 (See below for details)

**Default:** 0

This variable sets the behavior in incremental mode

Variable	Mode
0	Step & Direction
1	Step up, Step down input format
2	A/B Quadrature

Table 23: imCFG configuration settings

## stepSize – Incremental Step Size

**Controllers:** "N" – Classic Controller

**Modes:** Step & Direction (Simulated Stepper) Control, A/B Quadrature Input Control, CW/CCW (Simulated Stepper) Control

**Variable Type:** Integer

**Valid Range:** 1 to 65535

**Default:** 1

**Set Command:** IC

The value of the stepSize variable defines the amount of actuator motion (Phase Index steps taken) per incremental command.

## tmCFG – Toggle Mode Configuration

**Controllers:** "N" – Classic Controller

**Modes:** Toggle (Simple On/Off for Extend/Retract) Control

**Variable Type:** Integer

**Valid Range:** 0 to 4

**Default:** 1

This variable sets behavior in toggle mode

Variable	Description	Notes
0	Speed and Torque set in CONFIG.TXT	Torque and speed are static settings
1	Voltage sets speed, torque set in CONFIG.TXT	Dynamically controllable torque
2	Current sets speed, torque set in CONFIG.TXT	Dynamically controllable torque
3	Speed set in CONFIG.TXT, voltage sets torque	Dynamically controllable speed
4	Speed set in CONFIG.TXT, current sets torque	Dynamically controllable speed

Table 24: tmCFG configuration settings

## mtMin - maxTorq Minimum

**Controllers:** "N" – Classic Controller

**Modes:** Toggle (Simple On/Off for Extend/Retract) Control

**Variable Type:** Integer

**Valid Range:** 0 to 32766

**Default:** 8000

Sets the lower limit for maximum output motor torque when using the actuator in toggle mode with an analog control signal to control torque. The full range of the analog signal being used to control torque will linearly control the torque value to be within mtMin and maxTorq.

## msMin – Max Speed Minimum

**Controllers:** “N” – Classic Controller

**Modes:** Toggle (Simple On/Off for Extend/Retract) Control

**Variable Type:** Integer

**Valid Range:** 0 to 32766

**Default:** 200000

Sets the lower limit for maximum speed when using the actuator in toggle mode with an analog control signal to control speed. The full range of the analog signal being used to control speed will linearly control the speed value to be within msMin and maxSpeed.

## posTime – Preset Position Response Time

**Controllers:** “N” – Classic Controller

**Modes:** Pre-set Position Control

**Variable Type:** Integer

**Valid Range:** 0 to 65535

**Default:** 25

This variable defines the amount of continuous time in milliseconds it takes for a state change on the optically isolated digital inputs (IN1 and IN2) to register as a valid command.

## Pos1, Pos2, Pos3, and Pos4 – Preset Positions 1, 2, 3, and 4 (Respectively)

**Controllers:** “N” – Classic Controller

**Modes:** Pre-set Position Control, Command Line Interface (CLI) Control Over Serial

**Variable Type:** Integer

**Valid Range:** spMin to spMax

**Default:** Pos1 = 9175, Pos2 = 5120, Pos3 = 7168, Pos4 = 9216

These variables define the absolute position that corresponds to each state of the digital inputs.

Position	Input 1 (IN1) Status	Input 2 (IN2) Status
Position 1 (Pos1)	0 (Off)	0 (Off)
Position 2 (Pos2)	1 (On)	0 (Off)
Position 3 (Pos3)	0 (Off)	1 (On)
Position 4 (Pos4)	1 (On)	1 (On)

Table 25: All four (4) possible positions and the corresponding states of the optically isolated inputs (IN1 and IN2)

## Actuator Feature Configuration Settings

---

### errMode – Error Mode (Actuator Hardware Protection Feature)

**Controllers:** All Controllers

**Modes:** All Modes

**Variable Type:** Integer

**Valid Range:** 0 to 8 (see table below)

**Default:** 0

**Set Command:** EM

Error mode is a feature designed to help protect the actuator hardware in the event of accidental misuse. This feature is useful when training, or safely testing the actuator for an application. The actuator must be power cycled after errMode is triggered in order to return to normal operation.

The setting errMode defines the conditions which will cause the actuator during error conditions such as over torque (which will occur if the actuator hits a hard-stop), or a large position error. Error Mode can be set to any of the following:

<b>errMode Setting</b>	<b>Motor Disables if:</b>
<b>0</b>	Never (Error Mode Disabled)
<b>1</b>	Absolute Position Error > ovErrP for 1/2 second
<b>2</b>	Absolute Position Error > ovErrP for 1 second
<b>3</b>	Absolute Position Error > ovErrP for 2 seconds
<b>4</b>	Absolute Position Error > ovErrP - no delay
<b>5</b>	Absolute motor current > ovTorq and speed = 0 for 1/2 second
<b>6</b>	Absolute motor current > ovTorq and speed = 0 for 1 second
<b>7</b>	Absolute motor current > ovTorq and speed = 0 for 2 second
<b>8</b>	Absolute motor current > ovTorq and speed = 0 - no delay

Table 26: All available protection behaviors when using the errMode feature.

#### atTarg – At Target Position Threshold

**Controllers:** All

**Variable Type:** Integer

**Valid Range:** 0 to 4095

**Default:** 63

Defines the position error threshold that causes the “At Target Position” status bit to go active. The “at target position” bit will go active when the actuator position falls within Setpoint  $\pm$  atTarg.

#### ovErrP – Position Error Threshold

**Controllers:** All

**Variable Type:** Integer

**Valid Range:** 0 to 65535

**Default:** 8191

**Set Command:** OE

Defines the position error threshold that causes the “Over Position Error” status bit to go active (Position error > ovErrP).

#### ovHumid – Over Humidity Threshold

**Controllers:** “P” – Industrial Controller

**Variable Type:** Integer

**Valid Range:** 0 to 65535

**Default:** 32765

Defines the humidity threshold that causes the “Over Humidity” status bit to go active. Please note that this will not provide useful data in the AU series actuators.

#### ovSpeed – Over Speed Threshold

**Controllers:** All

**Variable Type:** Integer

**Valid Range:** 0 to 127

**Default:** 8

Defines the speed error threshold that causes the “Over Speed” status bit to go active (when Speed > ovSpeed).

## ovTorq – Over Torque Threshold

**Controllers:** All  
**Variable Type:** Integer  
**Valid Range:** 0 to 32767  
**Default:** 8000  
**Set Command:** OT

Defines the torque threshold that causes the “Over Torque” status bit to go active (when Torque > ovTorq).

## ovTemp – Over Temperature Threshold

**Controllers:** All  
**Variable Type:** Integer  
**Valid Range:** 0 to 32767  
**Default:** 32765  
**Set Command:**

Defines the torque threshold that causes the “Over Temperature” status bit to go active (when Temp > ovTemp). Note that the

## posGrtr – Position Greater Than Threshold

**Controllers:** All  
**Variable Type:** Integer  
**Valid Range:** 0 to 65535  
**Default:** 32767

Defines the threshold that causes the “Position Greater Than” status bit to go active (when Position > posGrtr). For example, setting posGrtr = 35000 will cause an active bit if the actuator position is 35001 and an inactive bit if the actuator position is 34999.

## posLess – Position Less Than Threshold

**Controllers:** All  
**Variable Type:** Integer  
**Valid Range:** 0 to 65535  
**Default:** 32768

Defines the threshold that causes the “Position Less than” status bit to go active (when Position < posLess). For example, setting posLess = 5000 will cause an active bit if the actuator position is 4999 and an inactive bit if the actuator position is 5001.

## xOutSrc – Analog Current Output Data Source

**Controllers:** “P” – Industrial Controller  
**Variable Type:** Integer  
**Valid Range:** 0 to 6  
**Set Command:** MS

Defines which data source the analog current output is providing. For xOutSrc = 6, the analog current output is directly controlled using serial commands, which is helpful for calibration, development, and debugging.

Setting	Data Source	Setting	Data Source
0	Position	4	Temperature

Setting	Data Source	Setting	Data Source
1	Torque	5	Humidity
2	Velocity	6	CLI command (CO)
3	Bus Voltage		

Table 27: Data sources for proportional 4-20mA Output

#### xOutMin and xOutMax – Data Range for Analog Current Position Feedback (Min and Max Respectively)

**Controllers:** “P” – Industrial Controller  
**Variable Type:** Integer  
**Valid Range:** See Description  
**Default:** xOutMin = spMin and xOutMax = spMax  
**Set Command:** MN and MX

These settings define the range of sensor counts between which the Servo Cylinder will output a proportional analog current signal between cOutMin and cOutMax. The count unit used by xOutMin and xOutMax is the count unit of the data source selected with xOutSrc. For example, if outputting position (xOutSrc=0), xOutMin and xOutMax are in position counts. Alternatively, if outputting torque (xOutSrc=1), xOutMin and xOutMax are in torque counts.

The values for xOutMin and xOutMax must be within the possible data range of that particular data type (e.g. when in position output (xOutSrc=0),  $rPos \leq xOutMin < xOutMax \leq ePos$ ).

#### cOutMin and cOutMax – Current range for Analog Current Position Feedback (Min and Max Respectively)

**Controllers:** “P” – Industrial Controller  
**Variable Type:** Integer  
**Valid Range:** 0 to 65535  
**Default:** cOutMin = 1064, cOutMax = 61975  
**Set Command:** YN and YX

Defines the analog current output range between which the output will vary. Default values correspond to ~4 mA and ~20 mA.

#### revCout – Inversion of Analog Current Output Signal Response

**Controllers:** “P” – Industrial Controller  
**Variable Type:** Integer  
**Valid Range:** 0 or 1  
**Default:** 0  
**Set Command:** YR

Inverts the response of the analog current output with respect to the selected data source.

#### ioPin1 and ioPin2 – GPIO Behavior for Pins 1 and 2 (Respectively)

**Controllers:** “N” – Classic Controller  
**Variable Type:** Integer  
**Valid Range:** 0 to 4  
**Default:** 0  
**Set Command:** D1 and D2

The variables ioPin1, ioPin2 set the operating mode for General Purpose Input / Outputs (GPIO) pin 1 and pin 2 respectively.

Variable	Description	Notes
0	High speed digital input	Used for Custom Applications
1	Ground	Sink up to 250 mA max
2	+5 V Output	Source up to 250 mA max
3	Output the result of the masked status register	1 kHz refresh rate
4	1 kHz, +5 V PWM Output	Proportional to a variable set by <i>ioPWM1</i> or <i>ioPWM2</i>

Table 28: GPIO pin functions

#### ioPWM1 and ioPWM2 – PWM Output Signal Type for GPIO pin 1 and 2 (Respectively)

**Controllers:** “N” – Classic Controller

**Variable Type:** Integer

**Valid Range:** 0 to 7 (See below for details)

**Default:** 0

**Set Command:** H1 and H2

The variables *ioPWM1* and *ioPWM2* set the variable to be output via a 1 kHz PWM waveform on General Purpose Input / Outputs (GPIO) pin 1 and 2 respectively.

Variable	Description	Output Range
0	Actuator Position	spMin to spMax
1	Actuator Velocity	-512 to 511
2	Actuator Torque	-32768 to 32767
3	Bus Voltage	0 to 1023
4	Commanded position	spMin to spMax
5	1-2 ms pulse input	pMin to pMax
6	Analog voltage input	vMin to vMax
7	Analog current input	cMin to cMax

Table 29: GPIO output signal functions

#### ioBit1 and ioBit2 – Masked Status Register Bit Output to GPIO Pin 1 and 2 (Respectively)

**Controllers:** “N” – Classic Controller

**Variable Type:** Unsigned Long

**Valid Range:** 0 to 4294967295

**Set Command:** W1 and W2

The decimal value of *ioBit1* and *ioBit2* in will be compared to the value of the status register, and a high bit will be signaled by the DIO1 or DIO2 pin respectively if the two are equal. Otherwise, the GPIO will be low.

For more information on utilizing this setting with the status register, see “Reading the Status Register over GPIO Pins Using a Register Mask” on page 65.

#### DinMode – Digital Input Disable feature

**Controllers:** “P” – Industrial Controller

**Variable Type:** Integer

**Valid Range:** 0 or 1

**Default:** 0

This setting is used to control the digital input disable feature. When set to 1, the actuator will only function when there is power applied to the optically isolated digital input IN1. This is useful for operation in conjunction with a safety mechanism.

### sFlag – Data Streaming on Startup

**Controllers:** “N” – Classic Controller and “P” – Industrial Controller

**Variable Type:** Integer

**Valid Range:** 0 or 1 (See below for details)

**Default Value:** 0

This variable defines the start-up behavior of the data streaming functionality. Note that streaming is not available on the CANopen control board. Streaming functionality should be handled with TPDO's with the CANopen controller.

### sTime – Data Stream Time Interval

**Controllers:** “N” – Classic Controller and “P” – Industrial Controller

**Variable Type:** Integer

**Valid Range:** 1 to 65535 (See below for details)

**Default** 65535

**Set Command:** XT

This variable defines the rate at which data is streamed from the actuator. This will be limited by the pre-selected baud rate. 65535 is the minimum data transfer rate (~3.576 Hz), the maximum data transfer rate is a function of the baud rate.

$$\text{Stream rate (Hz)} = \frac{234375}{XT}$$

$$XT = 2344 \approx 100 \text{ Hz}$$

$$XT = 23438 \approx 10 \text{ Hz}$$

Baud Rate	Maximum Streaming Frequency
9600	24.0 Hz
19200	48.0 Hz
38400	96.0 Hz
57600	144.0 Hz
115200	287.9 Hz

Table 30: Streaming frequency limitations for common baud rates

### heatEna – Heater Enable Flag

**Controllers:** “P” – Industrial Controller, “C” – CAN Controller with CAN 2.0B Firmware

**Variable Type:** Integer

**Valid Range:** 0 to 65535

**Default:** 0

**Serial Command:** EH

Setting heatEna to 0 disables the heating function, setting to a non-zero value enables the heating function. The recommended active setting is 7

### heatTmp – Heater Temperature Threshold

**Controllers:** “P” – Industrial Controller, “C” – CAN Controller with CAN 2.0B Firmware

**Variable Type:** Float

**Valid Range:** -50.0 to 149.0

**Default:** -20.0

**Set Command:** SH

Defines the temperature threshold that causes the heater to activate. Units are in °C

## heatCur – Heater Quiescent Current

**Controllers:** “P” – Industrial Controller, “C” – CAN Controller with CAN 2.0B Firmware

**Variable Type:** Integer

**Valid Range:** 0 to 15000

**Default:** 5000

**Set Command:** HC

This command sets the quiescent motor current value used when heating is activated. The units are the same as the maxTorq variable. The value required to maintain the actuator’s PCB temperature  $>-40^{\circ}\text{C}$  will be a function of your ambient environment’s temperature and heat transfer coefficient.

strx1, strx2, and strx3 – Data Stream for Columns 1, 2, and 3 (Respectively)



**WARNING:** This setting has different functionality depending upon which controller is being used. Always be aware of your controller and your electrical interface before setup.

**Controllers:** "N" – Classic Controller and "P" – Industrial Controller

**Variable Type:** Integer

**Valid Range:** See Below

**Default:** 0

**Set Command:** X1, X2, X3

This variable defines the variable that will be streamed in columns 1, 2, and 3 respectively. There are differences between

<b>Variable</b>	<b>Streamed Data</b> <i>(For "N" – Classic Controller)</i>	<b>Streamed Data</b> <i>(For "P" – Industrial Controller)</i>
0	Column off	Column Off
1	Position feedback	Actuator Position
2	Position set-point	Position Command Setpoint
3	Motor current feedback	Motor torque feedback
4	Motor current set-point	Motor torque command setpoint
5	Motor Phase 1 Current	Motor phase A current
6	Motor Phase 2 Current	Motor phase B current
7	Motor Phase 3 Current	Motor phase C current
8	Velocity at 1.25 kHz	Encoder counts per 10 ms
9	Velocity at 156.25 Hz	60 MHz counter increments between encoder counts
10	Encoder interval timer	Power supply voltage
11	Supply Voltage	Status register
12	Status register	Analog voltage input value
13	1-2 ms pulse input	Filtered analog voltage input value
14	Filtered 1-2 ms pulse input	Analog current input value
15	Analog Voltage Input	Filtered analog current input value
16	Filtered Analog Voltage Input	Temperature at circuit board in °F
17	Analog Current Input	Temperature at circuit board in °C
18	Filtered Analog Current Input	Temperature sensor on circuit board (raw)
19	Temperature sensor input	Relative Humidity at circuit board (%RH)
20	Position Error	Relative Humidity sensor on circuit board (raw)
21	Quadrature Input Errors	Position error
22	1-2 ms pulse input errors	32-bit counter with 100 µs increments
23	32-bit counter with 100 µs increments	N/A

Table 31: Settings for all available streamable data sources

## Installation and Use



We highly recommend running the servo cylinder unloaded on a benchtop to gain familiarity with the system before using it to drive a load. Please use Error Mode until the system is fully understood to protect the hardware and prevent motor burn outs.

## Integrating Servo Cylinder into Your System



**WARNING:** Do not apply torque to actuator rod when torquing hardware into the rod end. Use the rod-end wrench flats to apply torque here. The rod end wrench flats are not indexable and can be shipped in any orientation. These should not be relied upon for locating or orienting parts in your assembly.

### Note: Understanding “Hard-Stop” Behavior

If a Servo Cylinder is commanded to run into a “Hard-stop” such as a machine element, physical body, or an excessive exerted force which disallows movement, the controller will attempt to obey the command and power through the hard-stop, which will very likely overheat and cause damage to the motor windings.

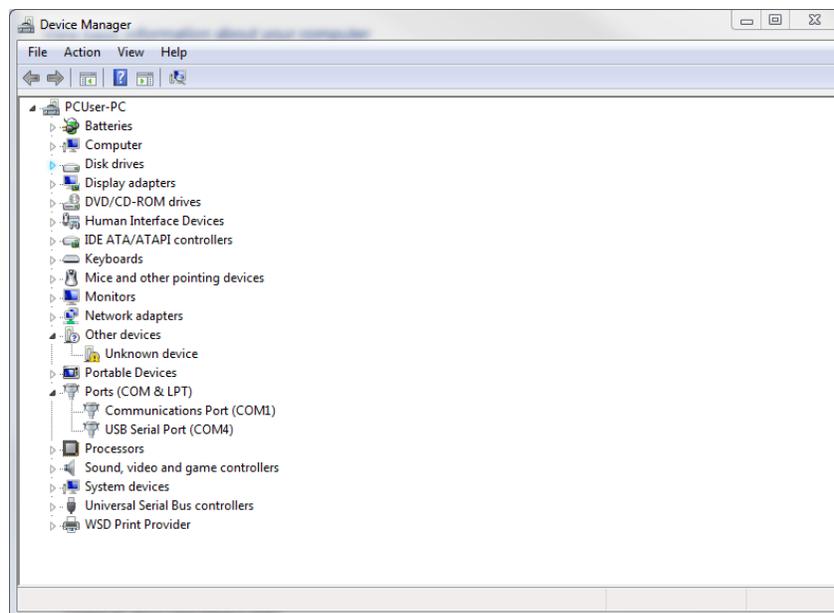
With default settings, the Servo Cylinder will be unable to be commanded to hit its own internal physical “hard-stop” travel limits. However, it will still be possible to hit these limits with either excessive force or motion while the actuator is unpowered. Additionally, it is important to understand that if you have installed the Servo Cylinder into a system with its own hard-stop (such as a machine element or other travel limit), you should configure spMin and spMax accordingly to prevent running into the hard-stops of your mechanism.

## Using the Command Line Interface (CLI)

To use the command line interface, connect via the serial port and open a serial terminal.

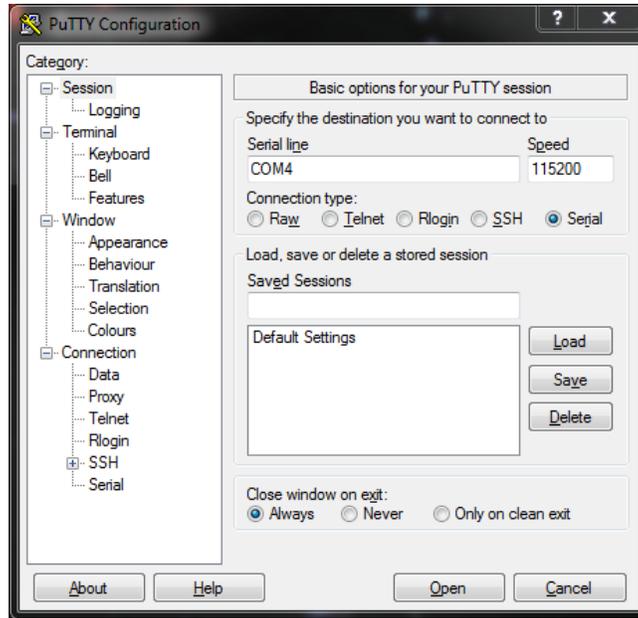
### Setting up a PuTTY Terminal (in Windows)

1. (If not done already) Install VCP (Virtual COM Port) drivers, which can be found on the FTDI website.
2. (If not done already) Install PuTTY, which can be found on the PuTTY website.
3. (If not done already) Follow the wiring procedure outlined in Setting up the Servo Cylinder Hardware on page 39. The USB mini B cable is not necessary if the Servo Cylinder is already configured.
4. In Windows, determine which COM port number was assigned to the Servo Cylinder. This can be done by opening Device Manager. In Device manager, expand the “Ports (COM & LPT)” menu, and make a note of the number of the COM port for the USB Serial Port (COM4 in the window below)

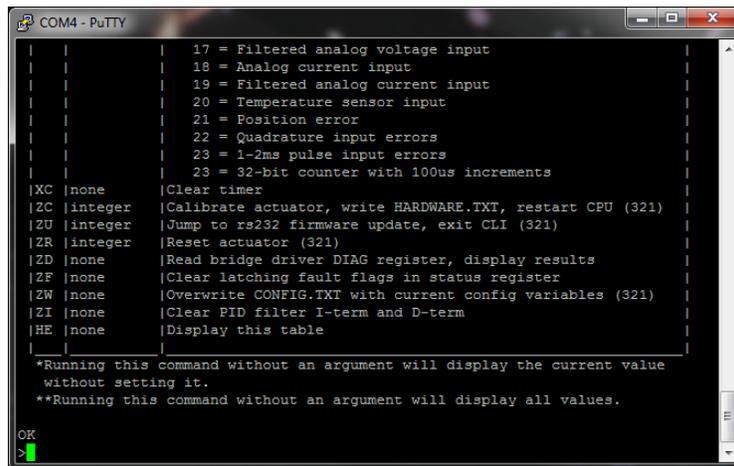


5. Open PuTTY.

- Set up PuTTY to run a serial command line interface. To do this, click the “Serial” Radio Button. Under Serial Line, enter the correct COM port for the Servo Cylinder (COM4 in this example). Under “Speed”, enter the Baud Rate set up in CONFIG.TXT, which is 115200. You may now save these as default settings for PuTTY.



- Click “Open” to open the command line interface. A blank terminal window will open.
- As a test, issue the Help command (HE) by typing HE and pressing enter. If correctly set up, the Servo Cylinder should return a table of serial commands (as seen in the screenshot below).



- The Servo Cylinder is now set up and ready to take commands.

### [Information about the Command Line Interface](#)

The Servo Cylinder will respond to single line commands. All commands consist of two letter characters (not case sensitive) and may or may not be followed by an argument and/or a checksum. Command is executed when followed by a carriage-return (CR) character (newline characters are ignored).

For a complete list of serial commands, see Appendix A: Serial Command Line Interface (CLI) List of Commands on page 75.

### [Using Machine Mode for Acknowledgements and Checksum Validation](#)

The Servo Cylinder’s Machine mode is an advanced mode generally preferred over human mode when there is a computerized machine (such as a PLC or program) controlling the actuator with RS232 serial communication. Machine mode is less verbose, allowing for faster and more robust communications over serial while still retaining acknowledgement and negative-acknowledgement (ACK/NACK) functionality. Additionally, the use of checksums for

validation of TX and RX serial communications is available in Machine Mode 2 (see Using Checksums for Signal Integrity on page 63)

There are two distinct machine modes, Machine Mode 1 (Machine1) and Machine Mode 2 (Machine2). Both modes will remove verbose acknowledgement messages and replace them with simple acknowledgement (ACK) and negative-acknowledgement (NACK) responses. In Machine Mode 2 the Servo Cylinder will only acknowledge a command with a valid checksum. To enable machine mode, either use either the config variable “ifMode – Serial Interface mode” (on page 49) or the IM serial command. The command “im0 26” (case sensitive) sets the interface back to Human Mode from machine mode 2.

### Acknowledgements and Negative-Acknowledgements (ACKs/NACKs) in Machine Mode

In any mode, the servo cylinder will respond with either an acknowledgement (ACK) or negative-acknowledgement (NACK) in response to every command. In human mode, these will be verbose statements intended to be interpreted by a human issuing direct serial commands via a command line interface (CLI) terminal. In machine mode, these verbose ACK/NACKs are simplified in order to reduce the amount of data being transferred.



Figure 19: A command line terminal of a Servo Cylinder in Machine Mode 1 displaying a valid command followed by an ACK, and an invalid command followed by a NACK. Both acknowledgements are followed by a two-character checksum.

In machine mode, an ACK consists of the letter “A” alongside the command response (if applicable) followed by a space and a checksum. A NACK consists of the letter “N” alongside a single hex character which describes the NACK, and is also followed by a space and a checksum. A complete table of definitions of the NACKs is in the table below.

“NACK” Code	Definition	Description
1	ERROR_INVALID	The two characters at the start of the command line do not match a valid Servo Cylinder serial command.
2	ERROR_LEN_ZRO	The command line has a length of 0
3	ERROR_LEN_ONE	The command has a length of only 1 character
4	ERROR_ARG_TOOMANY	Too many arguments are present in the command line
5	ERROR_ARG_TOOFEW	The command requires an argument and no argument is found
6	ERROR_ARG_INVALID	The argument is in an invalid format
7	ERROR_ARG_RANGE	The numerical argument is outside of an acceptable range
8	ERROR_ARG_LONG	The argument is longer than 63 characters
9	ERROR_CKSM_MISSING	Checksum required and not found
A	ERROR_CKSM_INVALID	Checksum characters not valid hexadecimal
B	ERROR_CKSM_LEN	Checksum is not 2 characters
C	ERROR_CKSM_BAD	Computed checksum does not match checksum received
D	ERROR_COND	Condition required by the command to execute has not been met

Table 32: Negative Acknowledgement (NACK) codes and definitions.

## Using Checksums for Signal Integrity

A checksum is a small set of information which is mathematically derived from a data block, such as a serial command line. If any of the characters in the data block become corrupted due to a signal transmission error, it is unlikely that the corrupted data will compute the same checksum as the original data, thus allowing for error detection and a corrective action to be taken by the master program.

The Servo Cylinder uses an 8-bit modulo 256 checksum, which consists of a single byte (8-bits, represented by two hexadecimal characters) which is calculated as the remainder of the sum of all numerical ASCII character codes for each character in the command when divided by 256 (also known as modulo 256).

$$checksum = \left( \sum (Char_1, Char_2, Char_3 \dots) \right) \text{mod}(256)$$

For reference, if represented in hexadecimal numbers the checksum can be computed more easily, as it is the last two digits of the sum of all ASCII character codes in the command.

## Using Checksums in Servo Cylinder's Machine Mode 2

To use the Servo Cylinder with Checksums on both received and transmitted serial data, first enable Machine Mode 2. Once machine mode 2 is enabled, the Servo Cylinder will only acknowledge commands which are sent with a valid checksum.

In machine mode 2, Servo Cylinder serial command consists of a two-letter command, which may or may not be followed by an argument, and which must be followed by a two-character checksum. A delimiter such as a space or a tab character is required between the argument and the checksum in order to delineate the end of the argument and the beginning of the checksum. See Figure 20: The structure of a valid TA command. The command used is TA, which commands a trajectory move to an arbitrary absolute position commanded in the argument (5000 in this case). The space between the command and the argument, as well as the type of checksum delimiter will each have an impact on the checksum calculation. See Figure 20 below for more information on the structure of a command and how to calculate a valid checksum.

### Note: Spaces and Delimiters in the Checksum Calculation

The checksum encompasses all characters in line except the actual checksum (two hexadecimal characters). This includes optional spaces between the command and the argument, as well as the delimiter before the checksum. Be sure to include all of these characters in the checksum calculation.

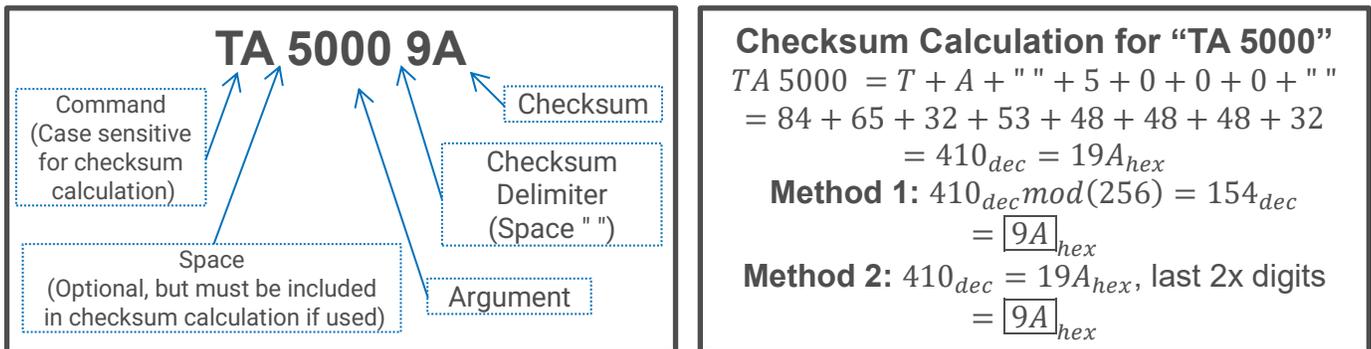


Figure 20: The structure of a valid TA command. The command used is TA, which commands a trajectory move to an arbitrary absolute position commanded in the argument (5000 in this case). The space between the command and the argument, as well as the type of checksum delimiter will each have an impact on the checksum calculation.

## Using the Status Register Feature

The Servo Cylinder updates a 32-bit status register to provide the user with an indication of actuator health and behavior. The entire status register can be read over serial (all controllers), or can be masked and output over the General-Purpose Input / Output (GPIO) pins at a refresh rate of 1 kHz ("N" – Classic Controller only).

Several of the bits require a user defined threshold that defines when they switch between being active and inactive. The configuration variables which define these thresholds are detailed in Actuator Feature Configuration Settings on page 53.

## Definition of Status Bits

Some of the status bits are latching, meaning that once the activation criteria are met, they will remain on until the user resets them manually, resets the actuator, or power-cycles the actuator.

<b>Bit</b>	<b>Status Register</b> <i>for "N" – Classic Controller</i>	<b>Latching?</b>
<b>0</b>	State of optically isolated input 1	No
<b>1</b>	State of optically isolated input 2	No
<b>2</b>	Temperature sensor > ovTemp	No
<b>3</b>	Motor Disabled due to errMode criteria having been met	<b>Yes</b>
<b>4</b>	Over-voltage on Vbus, dynamic brake enabled	No
<b>5</b>	Motor Control Enabled	No
<b>6</b>	Trajectory Active	No
<b>7</b>	Direction (extend)	No
<b>8</b>	Position < spMin	No
<b>9</b>	Position > spMin	No
<b>10</b>	Command input below min value, capped at min	No
<b>11</b>	Command input above max value, capped at max	No
<b>12</b>	Position Error < atTarg	No
<b>13</b>	Position error > ovErrP	No
<b>14</b>	Speed > ovSpeed	No
<b>15</b>	Torque > ovTorq	No
<b>16</b>	Position > posGrtr	No
<b>17</b>	Position < posLess	No
<b>18</b>	Serial Connected	No
<b>19</b>	USB Connected	No
<b>20</b>	Retracted physical stop has been hit	<b>Yes</b>
<b>21</b>	Extended physical stop has been hit	<b>Yes</b>
<b>22</b>	Supply voltage has been too low	<b>Yes</b>
<b>23</b>	Supply voltage has been too high	<b>Yes</b>
<b>24</b>	Bridge drive fault condition was reported	<b>Yes</b>
<b>25</b>	Over-current detected on motor phase	<b>Yes</b>
<b>26</b>	1 to 2 ms pulse input error	<b>Yes</b>
<b>27</b>	A/B Quadrature input error	<b>Yes</b>
<b>28</b>	File system has been modified	<b>Yes</b>
<b>29</b>	HARDWARE.TXT file contains errors	<b>Yes</b>
<b>30</b>	CONFIG.TXT file contains some errors	<b>Yes</b>
<b>31</b>	Critical error(s) in configuration file, MOTOR OFF	<b>Yes</b>

Table 33: Status Bit Definitions "N" – Classic Controller

Bit	Status Register for "P" – Industrial Controller	Latching?
0	Optically isolated digital input	No
1	Relative Humidity sensor value greater than threshold "ovHumi"	No
2	Temperature sensor value greater than threshold "ovTemp"	No
3	Emergency shutdown condition met, motor disabled	<b>Yes</b>
4	Supply voltage is greater than upper limit	No
5	Motor control is enabled	No
6	Trajectory generator is active	No
7	Direction of motion is extending	No
8	Actuator position is beyond retracted limit "spMin"	No
9	Actuator position is beyond extended limit "spMax"	No
10	Input control signal is below minimum, capped at minimum value	No
11	Input control signal is above maximum, capped at maximum value	No
12	Position error absolute value is below threshold "atTargWin"	No
13	Position error absolute value is above threshold "ovErrP"	No
14	Actuator speed is above threshold "ovSPeed"	No
15	Actuator torque absolute value is above threshold "ovTorq"	No
16	Actuator position is above threshold "posGrtr"	No
17	Actuator position is below threshold "posLess"	No
18	Bridge driver fault indication is active	No
19	USB is connected	No
20	Actuator has run against retracted physical stop	<b>Yes</b>
21	Actuator has run against extended physical stop	<b>Yes</b>
22	Supply voltage has been below the lower limit	<b>Yes</b>
23	Supply voltage has been above the upper limit	<b>Yes</b>
24	Bridge driver fault has occurred	<b>Yes</b>
25	Bridge current feedback ADC has been saturated	<b>Yes</b>
26	4-20mA input is below the lower limit	No
27	4-20 mA output signal source is outside of range "xOutMin, xOutMax"	No
28	Internal disk drive has been modified	<b>Yes</b>
29	HARDWARE.TXT contained errors upon startup	<b>Yes</b>
30	CONFIG.TXT contained errors upon startup	<b>Yes</b>
31	Critical errors found in configuration files, motor disabled	<b>Yes</b>
32	Supply voltage is lower than lower limit	No
33	Error caused CPU warm reset	<b>Yes</b>
34	Error calibrating ADC offset values – using defaults	<b>Yes</b>
35	Heating is active	No

Table 34: Status Bit Definitions "P" – Industrial Controller

### [Reading the Status Register over Serial](#)

The status register can be read using the "SR" command which is detailed in "SR – Read Status Register" on page 80. Using this command, the entire register can be called upon, which will return in the form of a hexadecimal number. Individual status register bits can be read by issuing the command with the appropriate argument.

Sending the SR command without an argument will return the value of the status register in hex format. Calling the SR command followed by a value from 0 to 31 will return the value of the corresponding status register bit.

### [Reading the Status Register over GPIO Pins Using a Register Mask](#)

Unlike the serial connection, the DIO pins can only output a single bit at a 1 kHz update rate. In order to provide more useful information over this single bit, the status register must be masked. For this to be possible, you must first configure the GPIO pins (1 and/or 2) to output the masked status register. See the GPIO section for information on configuring the GPIO pins.

Once the GPIO pins are properly set to output the masked status register value, you must define one (or both) mask values. These values are ioBit1 and ioBit2.

### Status Register Masking Example

Let's assume that your complete status register may read as 0x2008070 (in hexadecimal format). This is the value that would be returned to you if you ran the "SR – Read Status Register" command. Converting to binary to see the status of the individual bits, you get the value 0b0000001000000000100000001110000, which is represented in the table below to clarify bit positions:

Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0

Table 35: Example status register output, in binary

This indicates that Bits 4, 5, 6, 15, and 25 are active.

### Masking Example 2:

Let's say you want to monitor if the actuator is exerting excessive torque or if a trajectory move is active by having the GPIO bit go high when either of these two behaviors are occurring. This means you must monitor the status of Bit 6 (Trajectory Active) and bit 15 (Torque > ovTorq) on GPIO pin 1. Therefore, you need to set the configuration variable ioBit1 to the decimal value that represents these two bits: 32832

The value of 32832 is obtained by adding the integer representation of the bits you want to mask together. In this case:

$$Bit6 = 01000000_2 = 64_{10}, \quad Bit15 = 01000000000000000_2 = 32768_{10}, \quad Bit6 + Bit15 = 64 + 32768 = 32832$$

With ioBit1 set to 32832, DIO1 will output high when *either* Bit6 OR Bit15 are equal to 1. It will output low if both of those bits are equal to zero.

## Tuning Performance

A proportional-integral-derivative controller (PID Controller) is a very common industrial control algorithm that operates by establishing a control loop in which a process variable is adjusted by a control variable ( $u(t)$ ) calculated as a function of a continuously calculated error function ( $e(t)$ ). The PID control variable has gain values for the proportional ( $K_p$ ), integral ( $K_i$ ), and derivative ( $K_d$ ) terms, each of which can be adjusted or "tuned" to create the desired system response.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

In the case of the Servo Cylinder, the control variable  $u(t)$  is the duty cycle that controls the current through the motor coils. The Servo Cylinder cascades a high frequency current control loop to increase efficiency and provide control over actuator force. The details of the PID operation is handled by the Servo Cylinder's control electronics, but is presented to the user to help create a better understanding of the control structure.

This section will provide a brief introduction to PID controllers and demonstrate the effects of the three gain values on the Servo Cylinder's performance. All graphs were generated from data streamed by an unloaded A1W9A Servo Cylinder via RS232. "PO2048" was sent to the actuator to generate the step command, and the actuator was powered by a 24 VDC supply. Note that the steady state errors shown in this graph are for an unloaded actuator and will therefore not be as large as a loaded actuator. An integral term will be required to bring the steady state error to an acceptable level when the actuator is loaded.

# PID Response - A1NW9A

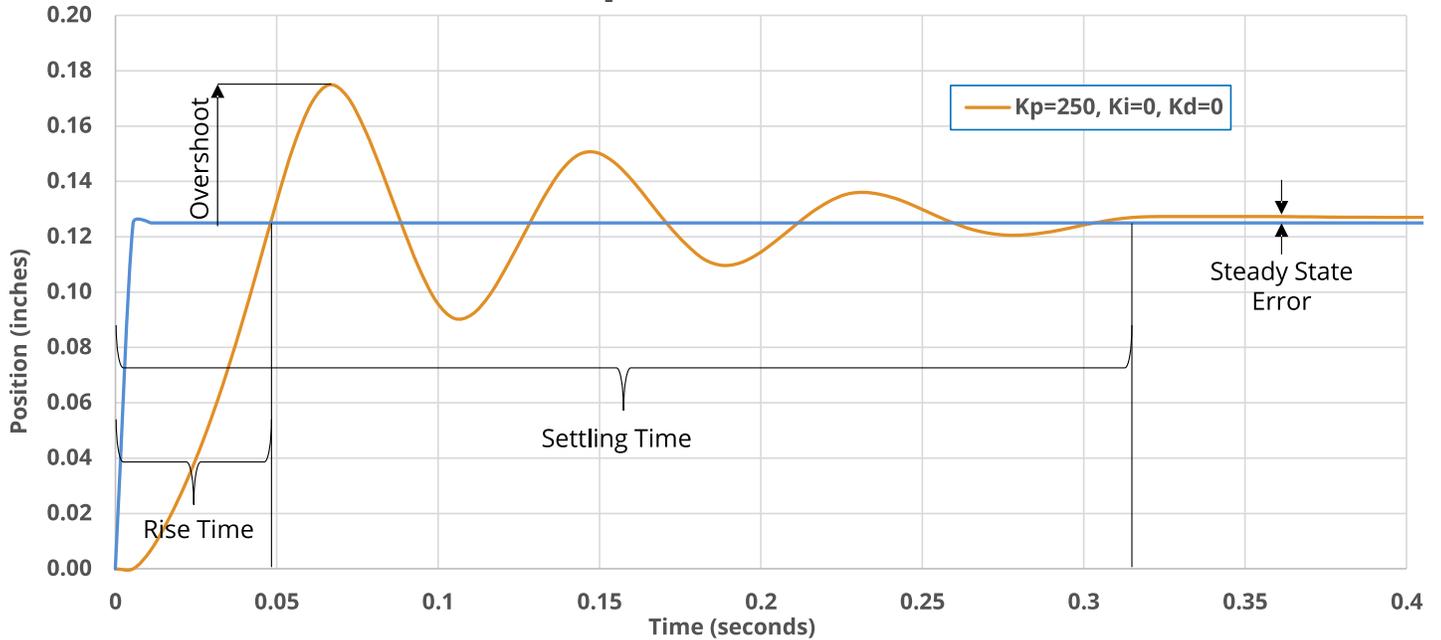


Figure 21: A control loop response of an A1W9A Servo Cylinder via RS232 which only has a proportional term, and which has not yet been properly tuned. The blue line represents the commanded signal, and the yellow line represents the position response. The different response characteristics such as rise time, overshoot, settling time, and steady state error are demonstrated.

## Proportional Term

The proportional term multiplies  $K_p$  by the current position error and uses this product as the correction value to the motor.

$$\text{Proportional Correction} = K_p e(t)$$

Using the proportional term alone will not generate an ideal system response in the majority of cases. There will always be a steady state error due to the lack of integral term, and the response cannot be as stable as one that includes a derivative term. As  $K_p$  increases, oscillations will take longer to diminish unless damped by the derivative term. For this reason, it is possible to reach an unstable state by increasing  $K_p$  too much.

## PID Response - A1NW9A

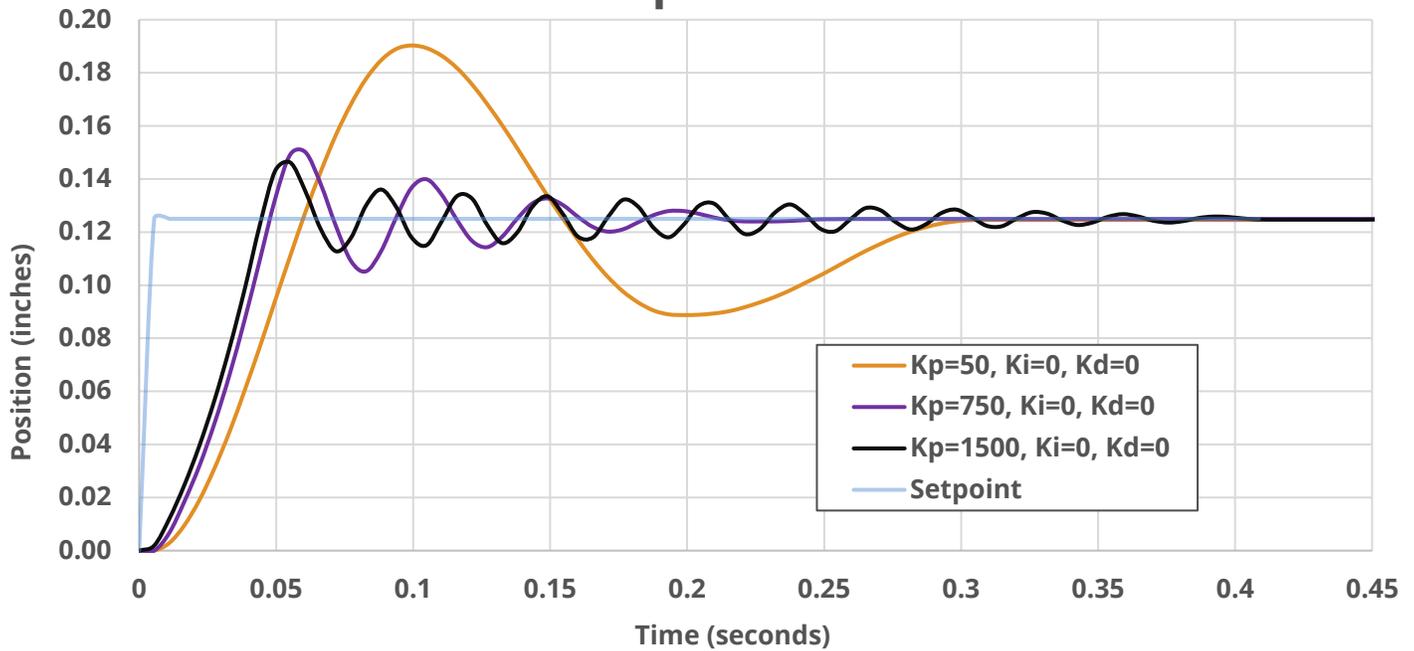


Figure 22: Control loop response with various  $K_p$  values. Note that the rise time, overshoot, and steady state error are inversely proportional to  $K_p$ . The oscillation frequency and the settling time have increased proportionally with  $K_p$ .

### Integral Term

The integral term is a time-based correction that uses the sum of the past errors to create a motor command. Unlike the proportional term, the integral term will accumulate to large values over time even when the instantaneous error is very small. The Integral term is necessary to reduce steady state error.

$$\text{Integral Correction} = K_i \int_0^t e(\tau) d\tau$$

The integral term should always be used in conjunction with a proportional and derivative term. The use of just the integral term ( $K_p = 0, K_d = 0$ ) will lead to instability and potential damage to the actuator and associated mechanism.

### Derivative Term

The derivative term acts by reducing the rate of change of the error. As shown in the chart below, the addition of the derivative term provides a damping action to the system response, which has the effect of reducing oscillations and overshoot, and increasing the rise time.

$$\text{Derivative Correction} = K_d \frac{de(t)}{dt}$$

The graph below depicts a response that limits the overshoot by eliminating the integral term. A PD-Controller (which has a zero integral term,  $K_i = 0$ ) like that shown below will have non-zero steady state error.

### PID Response - A1NW9A

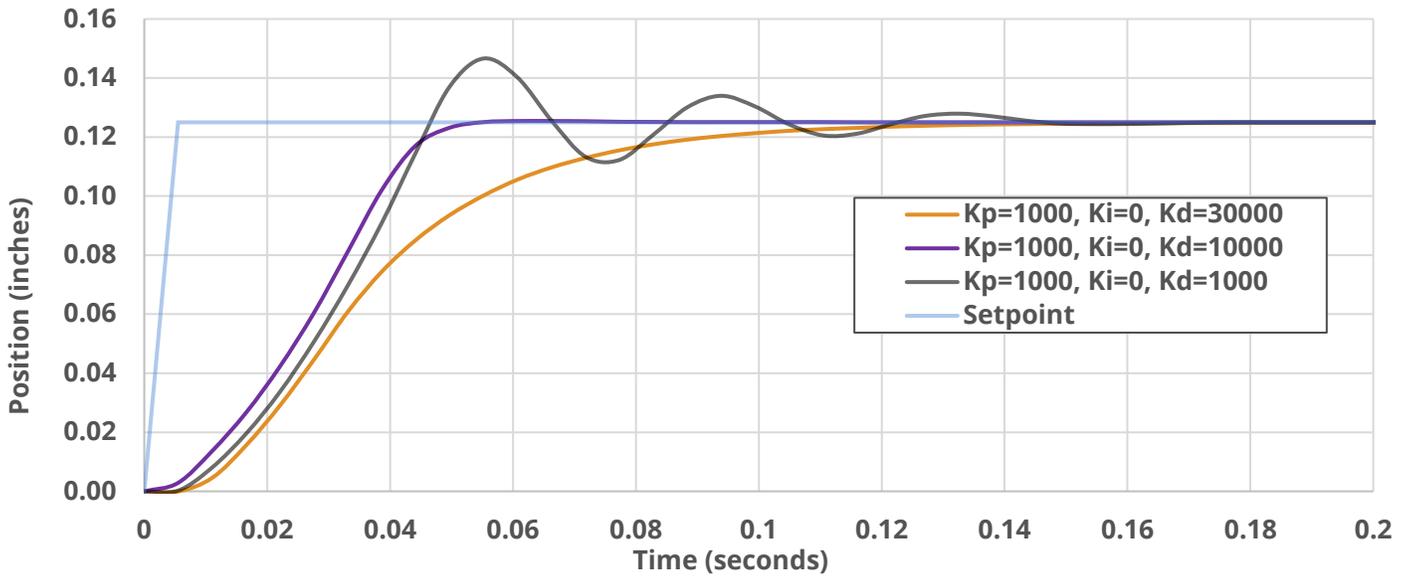


Figure 23: Three different responses acquired with three different values for  $K_d$ . The top line (orange) with a value of  $K_d = 30000$  is significantly overdamped, whereas the bottom line (black) with  $K_d = 1000$  is significantly underdamped

#### Stability

Unstable, diverging, and non-decaying oscillations such as those seen in the graph below, are possible with PID control and must be avoided. These can occur when the three gain values make the control variable function  $u(t)$  non-convergent on zero. This unstable state will result in uncontrolled motions, and can cause damage to the actuator and the attached mechanism if left unchecked.

### PID Response - A1NW9A

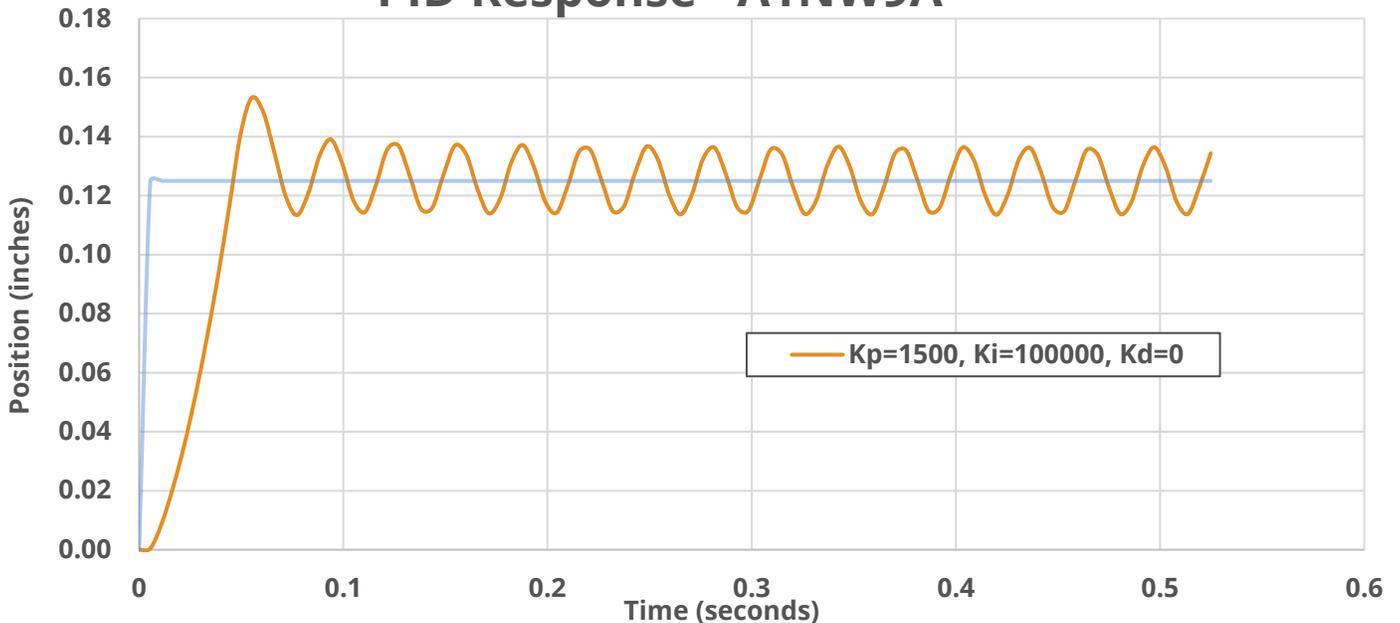


Figure 24: Unstable, non-decaying oscillations, which is the result of an excessive  $K_i$  term.

All Servo Cylinders come with default gain values that provide an optimized (critically damped) no load response.

Gain Variable	Value
$K_p$	1200
$K_i$	250000
$K_d$	10000

Table 36: Default Servo Cylinder PID Gains.

The large  $K_i$  value ensures negligible steady state error. Tuning is necessary if the application's required settling time, response time, overshoot, etc. is not achieved by the default gain variables. When tuning, it is recommended to plot actuator setpoint and position feedback vs. time. This can be done with external feedback or by streaming time, commanded position, and absolute position feedback from the actuator via RS232 (which is how the graphs in this section were generated).

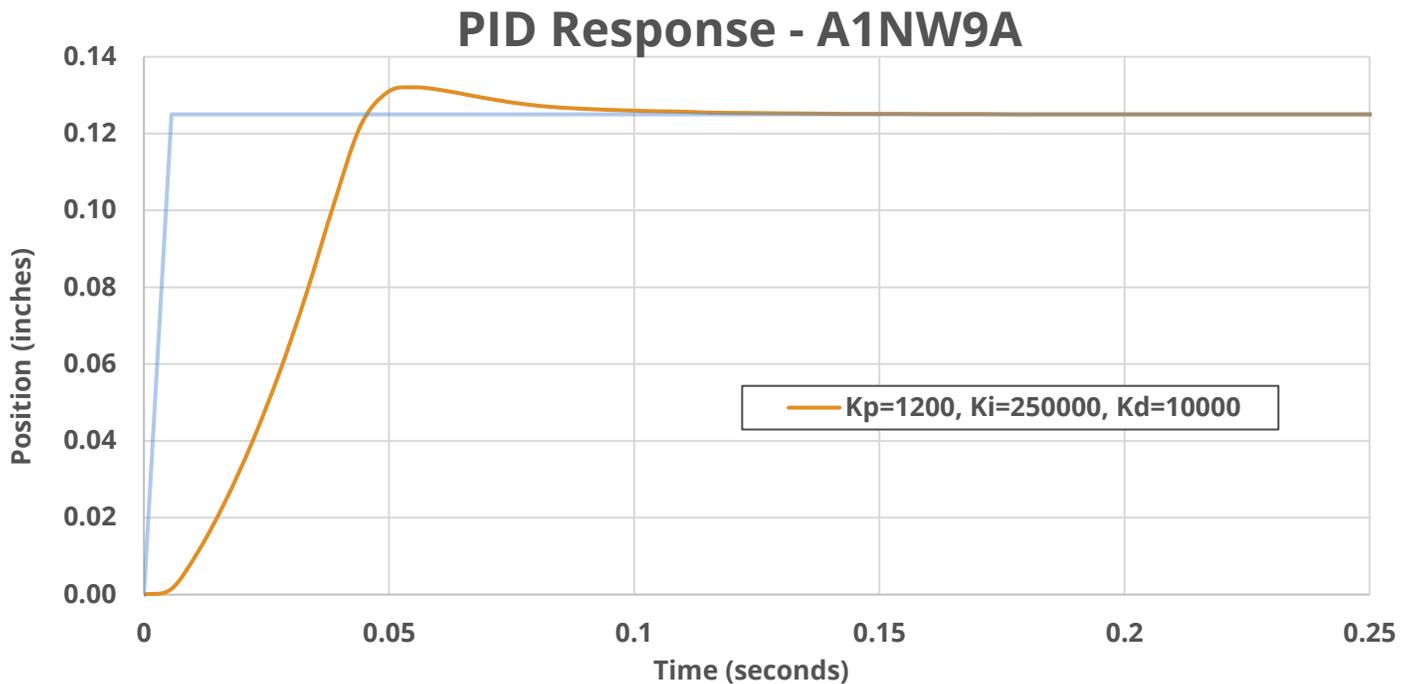


Figure 25: A well-tuned response. Note the greatly improved settling time, reduced overshoot, and lack of steady state error.

$$\text{Motor Correction} = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

Contact Ultra Motion engineering support for additional PID tuning assistance.

## Updating the Firmware

The firmware on the Servo Cylinder is field upgradeable through a serial connection. The update process uses a windows based utility called PC Loader.exe to load the new version of the firmware onto your Servo Cylinder. Contact Ultra Motion to obtain a copy of the most recent firmware and the windows update utility. To update the Servo Cylinder, you will need:

- A cable assembly to provide power AND a serial connection
- A USB mini-B cable
- The PC Loader.exe utility (contact Ultra Motion)
- The firmware .HEX file (contact Ultra Motion)
- A PC running Windows

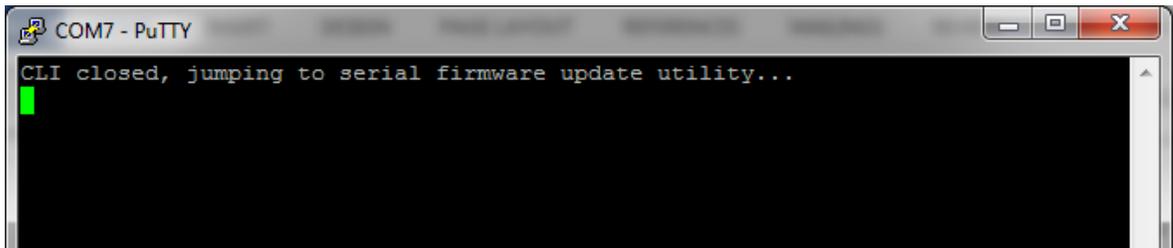


WARNING: While updating your firmware the Servo Cylinder will be put into an inoperable state. Make sure there is no load applied to the Servo Cylinder before starting this process. It is a good practice to remove the Servo Cylinder from the machine or installed location before initiating the update.

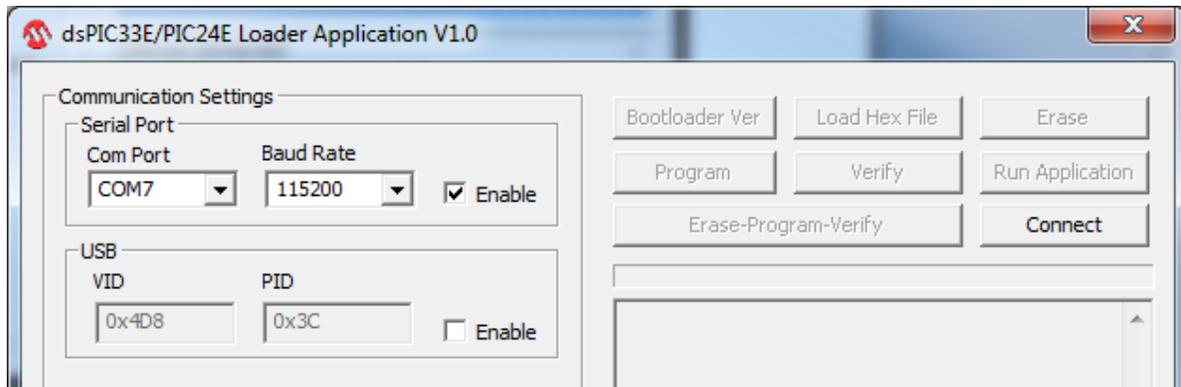


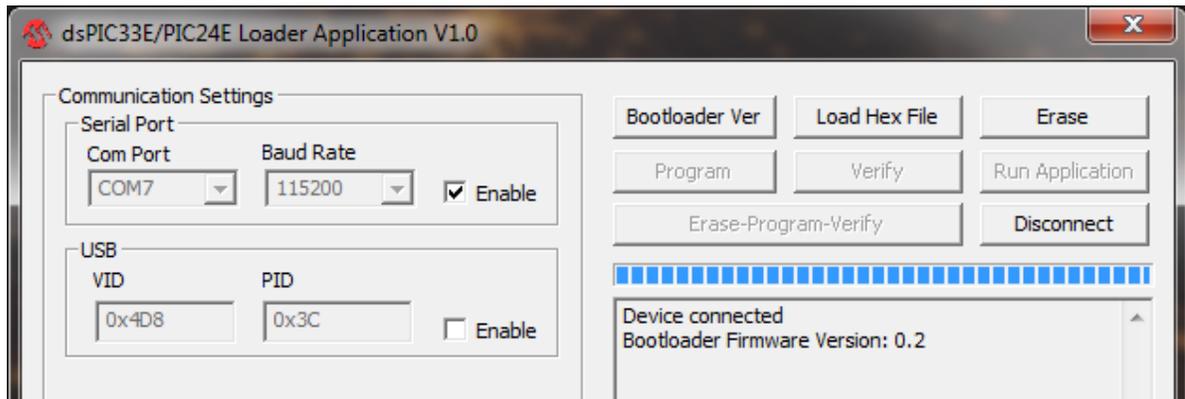
WARNING: Do not unplug or power cycle the Servo Cylinder until the update process has completed.

1. Connect the Servo Cylinder to the serial port on the windows computer.
2. Connect mini-B cable between the Servo Cylinder and your computer.
3. Connect the Servo Cylinder to a power supply
4. Turn on the power supply.
5. Browse to the USB Mass storage drive labeled "UltraMotion".
6. Back up the CONFIG.TXT and HARDWARE.TXT files by dragging them to a location on your computer. **This is critical** because you will need to replace these files after the update.
  - a. For "C" –CAN Controllers and "P" – Industrial Controllers, the CONFIG.TXT and HARDWARE.TXT files can be backed up to the Servo Cylinder's on-board persistent memory with the "SB321" command. After the firmware update is complete, the backup can be restored with the "BR321" command.
7. Open a PuTTY terminal using the instructions in Setting up a PuTTY Terminal (in Windows) on page 60.
8. Enter the command ZU321. The updater will start (indicated by flashing blue or blue/green STATUS LED and/or the message shown below).

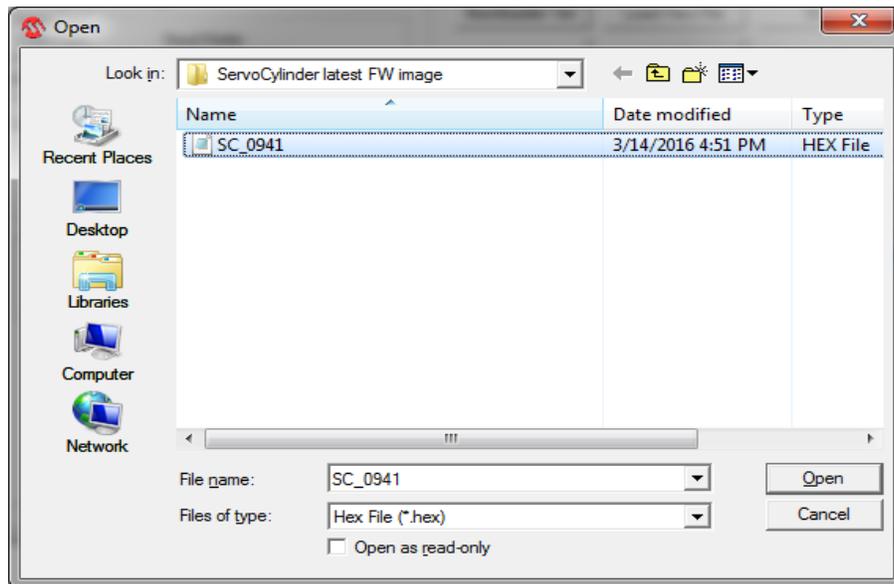


9. Close the PuTTY terminal window. **This is critical.**
10. Launch the PC Loader.exe program.
11. Select the correct COM port and baud rate (these are the same settings used in the PuTTY terminal setup).
12. Click "connect" button as shown to establish connection. Once connected you will see confirmation in the window. **It must say Bootloader Firmware Version 0.2** to indicate a successful connection.

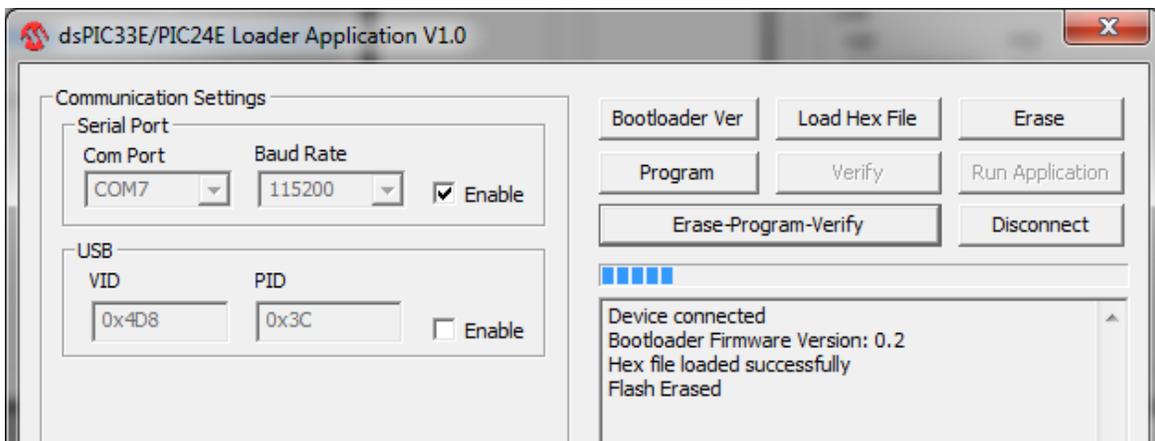




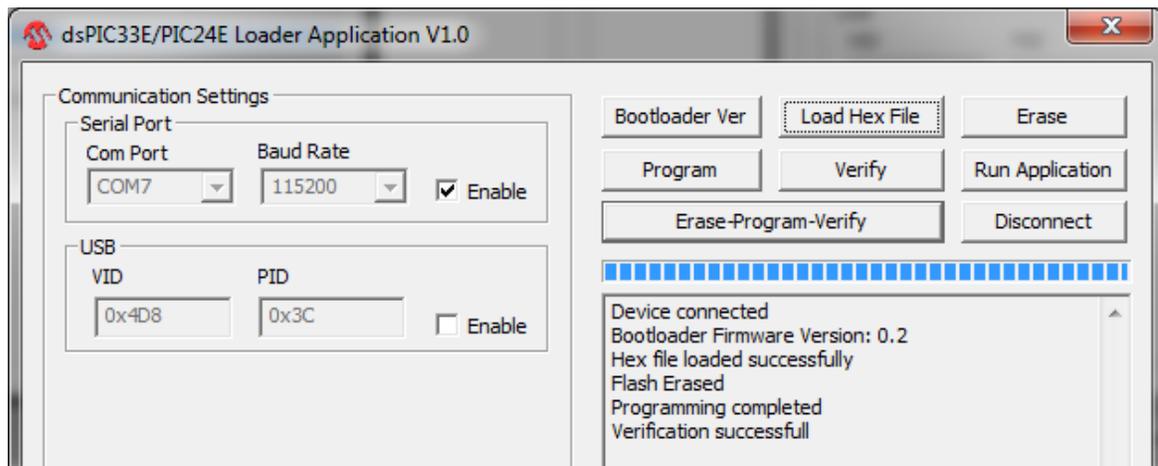
13. Once connected, click “Load Hex File”, and then navigate to the firmware file you would like to use for the update (.hex file format). Click “Open”



14. Click on the “Erase-Program-Verify” button to start the update.



15. Once complete. You will see “programming complete” and “verification successful”. See screenshot below.



16. Click “Run Application” to run the new firmware image. Since CONFIG.TXT and HARDWARE.TXT file are were overwritten by the firmware update, the Servo Cylinder will start in Safe Mode with the motor disabled and the STATUS LED solid red.
17. Disconnect and Close the PC Loader application.
18. Restore the CONFIG.TXT and HARDWARE.TXT files by browsing to the drive labeled “Ultra Motion” on the windows computer and dragging the previously backed up HARDWARE.TXT and CONFIG.TXT files onto the drive. Note: The firmware update will leave an empty CONFIG.TXT on the drive. Overwrite this empty file with the backed-up version of CONFIG.TXT.
19. Restart Servo Cylinder. This can be done by cycling the power, or issuing the command “ZR321” to the CLI. After restart, the STATUS LED should be solid green.
20. If the STATUS LED is red after restart, establish a serial connection via putty or similar program. Power cycle the actuator and note the error message(s) displayed in the terminal window. Contact Ultra Motion for assistance troubleshooting any error.

## Troubleshooting

Please read Design Guidelines and Best Practices for Electromechanical Linear Actuators on page 21 for an overview of best practices for using electromechanical actuators, including solutions to common problems.

### Status Light Definitions N and P Controllers

LED Activity	Indicated Status
<b>Green (solid)</b>	Normal Operation
<b>Red (solid)</b>	Safety Self-Shutdown (Several possible causes): Critical CONFIG.TXT error, critical HARDWARE.TXT error, errMode has triggered a shutdown, dip switches set both ON at startup <ul style="list-style-type: none"><li>• Motor disabled (coast)</li></ul>
<b>Green Flash</b>	USB connected (1x flash) or disconnected (2x flash)
<b>Blue Flash</b>	Serial connected (1x flash) or disconnected (2x flash)
<b>Red Flash</b>	One (1) Flash per fault that has occurred. Faults that cause a red flash: <ul style="list-style-type: none"><li>• rStopFault: Actuator has hit retracted physical stop (as defined in HARDWARE.TXT)</li><li>• eStopFault: Actuator has hit extended physical stop</li><li>• vBusHFault: Excessive supply voltage (&gt;44.0 VDC, 42.0 VDC hysteresis), dynamic braking was triggered</li><li>• vBusLFault: Supply voltage has been too low (&lt; 6.75 VDC), motor coast was triggered</li><li>• drvFault: Bridge driver IC has indicated a fault</li><li>• cpuFault: CPU error trap was hit</li><li>• adcFault: Current feedback was saturated on one or more of the motor phases (&gt;25 amps)</li><li>• uartFault: UART error</li><li>• pulseFault: Pulse was received on outside of the valid PWM range (1-2ms Pulse mode only)</li><li>• quadFault: A logical error was detected on A/B quadrature input (A/B quadrature mode only)</li></ul>

# Appendix A: Serial Command Line Interface (CLI) List of Commands

<b>PV-Limit for Acme Screws</b> .....	<b>16</b>
<b>Read Commands</b> .....	<b>77</b>
AN – Read Analog Inputs .....	77
AP – Read Actuator Position .....	77
BV – Read Bus Voltage .....	77
DI – Read Digital Input .....	78
EC – Read Startup File Error .....	78
ES – Read Startup System Error .....	78
FS – Read Actuator Information .....	78
HI – Read Heater status info .....	78
HE – Display Help Tables .....	79
QP – Read Position Setpoint .....	79
RC – Read 1 to 2 ms Pulse (RC PWM) Input .....	79
RH – Read Relative Humidity at Controller Circuit Board .....	79
RL – Read Total Distance Traveled over Actuator Life .....	79
RT – Read Temperature at Controller Circuit Board .....	80
RV – Read Configuration Variable(s) .....	80
SR – Read Status Register .....	81
TQ – Read Motor Torque .....	82
VL – Read Actuator Velocity .....	82
<b>Set Commands</b> .....	<b>82</b>
AC – Set Accel - Acceleration and Deceleration Rate .....	82
CG – Coast On .....	82
CO – Set CLI Driven 4-20 mA Output value .....	82
CN and CX – Set cMin and cMax – Analog Current Input Minimum and Maximum (Respectively) .....	82
CS – Coast Off .....	82
D1 and D2 – Set ioPin1 and ioPin2 – GPIO Behavior for Pins 1 and 2 (Respectively) .....	83
EH – Read/Set heatEna the heater enable flag .....	83
EM – Set errMode – Error Mode (Actuator Hardware Protection Feature)t .....	83
FC – Set cFilter - Current filter .....	83
FP – Set pFilter – 1-2ms Pulse (RC Servo) filter .....	83
FV – Set vFilter – Analog Voltage Filter .....	83
H1 and H2 – Set ioPWM1 and ioPWM2 – PWM Output Signal Type for GPIO pin 1 and 2 (Respectively) .....	84
HC – Read/Set heatCur the heater current value .....	84
IC – Set stepSize – Incremental Step Size .....	84
IR – Set revSgnl – Inversion of Response to Analog Current Input Signal .....	84
IM – Set ifMode – Serial Interface mode .....	84
KP, KI, and KD – Set kp, ki, and kd – PID Gains .....	84
LN – Set spMin - Software Position Minimum .....	84
LX – Set spMax - Software Position Maximum .....	85
MN and MX – Set xOutMin and xOutMax – Data Range for Analog Current Position Feedback .....	85
MT – Set maxTorq - Max Torque .....	85
MS – Set xOutSrc – Analog Current Output Data Source .....	85
OE – Set ovErrP – Position Error Threshold .....	85
OT – Set ovTorq – Over Torque Threshold .....	85
PA – Set Absolute Target Position .....	85
PC – Set Target Position to Current Position .....	86
PO – Set Target Position with an Offset .....	86
PN and PX – Set pMin and pMax – 1-2 ms Pulse (RC PWM Servo) Control Pulse Width Min and Max (Respectively) .....	86
SH – Read/Set heatTmp the heater threshold temperature .....	86
SM – Set stopMode – End-of-Trajectory Stopping mode .....	86
SP – Set maxSpeed - Maximum Speed .....	86
VN and VX - Set vMin and vMax – Analog Voltage Min and Analog Voltage Max (Respectively) .....	87
W1 and W2 – Set ioBit1 and ioBit2 – Masked Status Register Bit Output to GPIO Pin 1 and 2 (Respectively) .....	87
YN and YX – Set cOutMin and cOutMax – Current range for Analog Current Position Feedback (Min and Max Respectively) .....	87
YR – Set revCout – Inversion of Analog Current Output Signal Response .....	87
ZI – Clear PID Filter I-term and D-Term .....	87
<b>Trajectory Commands</b> .....	<b>87</b>

T1, T2, T3, and T4 – Trajectory Move to positions Pos1, Pos2, Pos3, and Pos4 (Respectively) .....	87
TA – Trajectory Move to Absolute Position .....	88
TE – Trajectory Move to Fully Extended Position (spMax).....	88
TK – Interrupt Current Trajectory Move.....	88
TM – Trajectory Move to Midpoint .....	88
TO – Trajectory Move to Offset (Incremental Trajectory Move) .....	88
TR – Trajectory Move to Fully Retracted Position (spMin).....	88
<b>Serial Data Streaming .....</b>	<b>89</b>
X1, X2, and X3 – Set strx1, strx2, and strx3 – Data Stream for Columns 1, 2, and 3 (Respectively) .....	89
XC – Clear Timer.....	89
XG – Turn Data Streaming On.....	89
XR – Display Data Stream Setup.....	89
XS – Turn Data Streaming Off.....	89
XT – Set sTime – Data Stream Time Interval.....	89
<b>System Commands .....</b>	<b>89</b>
BR – Restore File System from Backup.....	89
CW – Write current settings to CONFIG.TXT (“P” – Industrial Controller, “C” – CAN controller).....	89
FR – Read File from Internal File System .....	90
SB – Back Up Internal File System.....	90
ZC – Run Calibration Routine.....	90
ZD – Read Bridge Driver DIAG Register .....	90
ZF – Clear Latching Fault Flags in Status Register.....	90
ZR – Reboot Actuator.....	90
ZU – Jump to Serial Firmware Updater.....	91
ZW – Write current settings to CONFIG.TXT (“N” – Classic Controller Only).....	91

## Read Commands

### *AN – Read Analog Inputs*

**Applies to:** "N" - Classic Controller

**Argument:** 1 to 3, or none

This command returns the value of the Servo Cylinder’s analog inputs. Sending this command with no argument will return the value for all inputs. To convert raw sensor counts to physical units, see Appendix B: Converting Between Sensor Counts and Physical Units on page 90.

Argument	Returned Input
1	Voltage
2	Current
3	Temperature

Table 37: Analog Inputs for "AN" Command

### *AP – Read Actuator Position*

**Applies to:** All Controllers

**Argument:** None

This command returns the absolute position of the Servo Cylinder. To convert raw sensor counts to physical units, see Appendix B: Converting Between Sensor Counts and Physical Units on page 90.

### *BV – Read Bus Voltage*

**Applies to:** All Controllers

**Argument:** None

This command returns bus voltage, which is the input power supply voltage used to power the actuator. To convert raw sensor counts to physical units, see Appendix B: Converting Between Sensor Counts and Physical Units on page 90.

## DI – Read Digital Input

---

**Applies to:** All Controllers

**Argument:** 1 to 4, or none

This command returns the value of the Servo Cylinder’s digital inputs, both optically isolated inputs (IN1/IN2) and GPIOs (DIO1/DIO2). Sending this command with no argument will return the value for all of the digital inputs.

*Note: Not to be confused with the command D1 and D2 – Set ioPin1 and ioPin2 – GPIO Behavior for Pins 1 and 2 (Respectively)*

### **Important: Digital Input States**

The digital state of OPTO1 and OPTO2 is inverted from convention. When IN1 or IN2 has voltage applied to it, OPTO1/OPTO2 switches from 1 to 0.

Argument	Digital Input
1	OPTO1 (IN1)
2	OPTO2 (IN2)
3	GPIO1 (DIO1)
4	GPIO2 (DIO2)

Table 38: Functionality of the DI command.

## EC – Read Startup File Error

---

**Applies to:** All Controllers

**Argument:** None

This command returns any errors that are associated with the startup file, CONFIG.TXT.

## ES – Read Startup System Error

---

**Applies to:** All Controllers

**Argument:** None

This command returns any errors that are associated with the system startup

## FS – Read Actuator Information

---

**Applies to:** All Controllers

**Argument:** None

This command returns the firmware version currently running on the Servo Cylinder as well as the unique product serial number. On certain models, this will also return the product number.

## HI – Read Heater status info

---

**Applies to:** “P” – Industrial Controller

**Argument:** None

Returns details on the motor heater function over the command line interface

## HE – Display Help Tables



**WARNING:** This command has different functionality depending upon which controller is being used. Always be aware of your controller and your electrical interface before setup.

**Applies to:** All Controllers

**Argument:** For “N” – Classic Controller: None  
For “P” – Industrial controller: None or 0 to 2

This command will print out a list of all Serial CLI commands.

Argument	Return
0 or none	List of Serial CLI Commands
1	List of Data Stream Column Options
2	List of Bits of the Status Register

Table 39: Functionality of the HE command (with in the “P” – Industrial Controller only).

## QP – Read Position Setpoint

**Applies to:** All Controllers

**Argument** None

This command returns the position setpoint currently being commanded by the chosen operating mode.

## RC – Read 1 to 2 ms Pulse (RC PWM) Input

**Applies to:** “N” – Classic Controller

**Argument:** None

This command returns the value of the 1-2 ms pulse being read by the Servo Cylinder. This command is only valid in pulse mode. To convert raw sensor counts to physical units, see Appendix B: Converting Between Sensor Counts and Physical Units on page 90.

## RH – Read Relative Humidity at Controller Circuit Board

**Applies to:** “P” - Industrial Controller

**Argument:** 0 or 1

This command returns the Relative Humidity (%RH) as measured by the onboard temperature sensor of the control circuit board. Note that temperature of windings will typically be in excess of this figure due to localized heating. The units depend upon the argument provided with the command. To convert raw sensor counts to physical units, see Appendix B: Converting Between Sensor Counts and Physical Units on page 90.

Argument	Return
0	Relative Humidity (%RH)
1	Raw sensor value

Table 40: Functionality of the RH Command.

## RL – Read Total Distance Traveled over Actuator Life

**Applies to:** “P” - Industrial Controller, “C” – CAN Controller

**Argument:** none

This command returns the total distance traveled over the lifespan of the actuator. This is useful for assessing wear and remaining actuator lifespan.

## RT – Read Temperature at Controller Circuit Board

**Applies to:** "P" - Industrial Controller

**Argument:** 0 to 2

This command returns the temperature as measured by the onboard temperature sensor of the control circuit board. Note that temperature of windings will typically be in excess of this figure due to localized heating. The windings can very rapidly heat up at high loads before the PCB temperature sensor detects a change due to the large differences in thermal mass. The units depend upon the argument provided with the command. To convert raw sensor counts to physical units, see Appendix B: Converting Between Sensor Counts and Physical Units on page 90.

Argument	Return
0	Temperature in °F
1	Temperature in °C
2	Raw sensor value

Table 41: Functionality of the RT command

## RV – Read Configuration Variable(s)



**WARNING:** This command has different functionality depending upon which controller is being used. Always be aware of your controller and your electrical interface before setup.

**Applies to:** All Controllers (Table varies by controller)

**Argument:** Varies by controller, see tables below

This command returns the value of any of the configuration variables for all configuration settings. The argument numbers associated with each setting are defined in the tables below. The configuration settings present in each servo cylinder vary based upon the controller, so be sure you are looking at the correct table below

Argument	Setting	Argument	Setting	Argument	Setting	Argument	Setting
0	rotor	17	imCFG	34	vMax	51	oveErrP
1	adc1z	18	spMin	35	vFilter	52	atTarg
2	adc2z	19	spMax	36	cMin	53	ovTorq
3	adc3z	20	kp	37	cMax	54	ovSpeed
4	encRef	21	ki	38	cFilter	55	ovTemp
5	rPos	22	kd	39	posTime	56	posGrtr
6	ePos	23	maxTorq	40	Pos1	57	posLess
7	qP	24	mtMin	41	Pos2	58	sFlag
8	ql	25	maxSpeed	42	Pos3	59	sTime
9	dP	26	msMin	43	Pos4	60	strX1
10	dl	27	accel	44	ioPin1	61	strX2
11	sNum	28	stopMode	45	ioBit2	62	strX3
12	ifMode	29	stepSize	46	ioPWM1	-	-
13	baud	30	pMin	47	ioPin2	-	-
14	opMode	31	pMax	48	ioBit2	-	-
15	tmCFG	32	pFilter	49	ioPWM2	-	-
16	pmCFG	33	vMin	50	errMode	-	-

Table 42: All readable configuration variables for the "N" – Classic Controller.

Argument	Setting	Argument	Setting	Argument	Setting	Argument	Setting
0	rotor	14	spMin	28	revSgnl	42	ovHumid
1	encRef	15	spMax	29	xOutSrc	43	posGrtr
2	rPos	16	kp	30	xOutMin	44	posLess
3	ePos	17	ki	31	xOutMax	45	sFlag
4	sType	18	kd	32	cOutMin	46	sTime
5	qP	19	maxTorq	33	cOutMax	47	strX1
6	ql	20	maxSpeed	34	revCout	48	strX2
7	dP	21	accel	35	DinMode	49	strX3
8	dl	22	vMin	36	errMode	50	heatEna
9	sNum	23	vMax	37	ovErrP	51	heatTmp
10	pNum	24	vFilter	38	atTarg	52	heatCur
11	ifMode	25	cMin	39	ovTorq	-	-
12	baud	26	cMax	40	ovSpeed	-	-
13	opMode	27	cFilter	41	ovTemp	-	-

Table 43: All readable configuration variables for the "P" – Industrial Controller.

Argument	Setting	Argument	Setting	Argument	Setting	Argument	Setting
0	rotor	15	kd	30	abrtOp	45	ftTimeB
1	encRef	16	nodeID	31	haltOp	46	ftMaskC
2	rPos	17	canSpeed	32	posOP	47	ftTimeC
3	ePos	18	drvMode	33	lsMin	48	tpmSpd
4	sType	19	spMin	34	lsMax	49	tpmAcc
5	qP	20	spMax	35	posA	50	tpmDec
6	ql	21	maxTorq	36	posB	51	tpmHdec
7	dP	22	pSpeed	37	errMask	52	tpmTorq
8	dl	23	pAcc	38	ovTorq	53	tpmPol
9	sNum	24	pDec	39	stSpeed	-	-
10	pNum	25	qsDec	40	ovTemp	-	-
11	cliMode	26	fErrWin	41	ovHumid	-	-
12	cliBaud	27	fErrTime	42	ftMaskA	-	-
13	kp	28	posWin	43	ftTimeA	-	-
14	ki	29	posTime	44	ftMaskB	-	-

Table 44: All readable configuration variables for the "C" – CANopen Controller.

## SR – Read Status Register

**Applies to:** All Controllers

**Argument:** 0 to 31, or none

This command returns the value of the 32-bit Status Register, or request specific bits within the status register. Sending this command without an argument will return the value of the status register in hex format. Calling the SR command followed by a value from 0 to 31 will return the value of the status register bit which corresponds with the argument.

For some examples:

Input Command	Return
SR 4	bit4: "Emergency over-volt" = false
SR	0x020E10A3

Table 45: Example commands and returns for the SR command

Refer to "Using the Status Register Feature" on page 63 for more information of the status register.

## TQ – Read Motor Torque

---

**Applies to:** All Controllers

**Argument:** None

This command returns the transformed three phase current feedback, which is a measure of motor torque (-32768 to 32767).

## VL – Read Actuator Velocity

---

**Applies to:** All Controllers

**Argument:** None

This command returns the velocity of the actuator as measured by the change in Phase Index over a time interval. Negative values represent motion towards the retracted direction; positive values represent motion towards the extended position. To convert between sensor count, see Appendix B: Converting Between Sensor Counts and Physical Units on page 90.

## Set Commands

### AC – Set Accel - Acceleration and Deceleration Rate

---

**Applies to:** All Controllers

**Argument:** See Setting Description

This command sets the configuration variable “Accel - Acceleration and Deceleration Rate” (more info on page 48) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

### CG – Coast On

---

**Applies to:** All Controllers

**Argument:** 321

This command disables the Servo Cylinder motion.



**WARNING:** Enabling Coast electrically decouples the motor from the power stage, removing all power from the motor. Enabling coast is not recommended for applications in which back-driving is possible. **Care must be taken when enabling coast!**

### CO – Set CLI Driven 4-20 mA Output value

---

**Applies to:** “P” – Industrial Controller

**Argument:** 0 to 65535, or none

This command is used to set an arbitrary 4-20 mA current output. This feature only works when the setting “xOutSrc – Analog Current Output Data Source” (on page 55) is set to CLI driven output (xOutSrc=6). Generally, this is used for calibration and diagnostic purposes. To convert raw sensor counts to physical units, see Appendix B: Converting Between Sensor Counts and Physical Units on page 90.

### CN and CX – Set cMin and cMax – Analog Current Input Minimum and Maximum (Respectively)

---

**Applies to:** “N” – Classic Controller and “P” – Industrial Controller

**Argument:** See Setting Description

This command sets the configuration variable “cMin and cMax – Analog Current Input Minimum and Maximum (Respectively)” (more info on page 50) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable

### CS – Coast Off

---

**Applies to:** All Controllers

**Argument:** 321

This command re-enables the BLDC motor, allowing for normal operation of the Servo Cylinder.



**WARNING:** Re-energizing the motor will cause the Servo Cylinder to immediately move to the current commanded position at maximum speed and torque. **Care must be taken when disabling Coast!**

To avoid having the actuator immediately translate to whatever its target position may be, consider running, “ZI” and “PC” beforehand. This will clear the PID integral term and set the current position to target position, ensuring there is no movement upon coast being disabled.

#### *D1 and D2 – Set ioPin1 and ioPin2 – GPIO Behavior for Pins 1 and 2 (Respectively)*

---

**Applies to:** “N” – Classic Controller

**Argument:** See Setting Description

These commands set the configuration variables “ioPin1 and ioPin2 – GPIO Behavior for Pins 1 and 2 (Respectively)” (more info on page 56) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

*Note: Not to be confused with the command DI – Read Digital Input*

#### *EH – Read/Set heatEna the heater enable flag*

---

**Applies to:** “P” – Industrial Controller, “C” – CAN Controller with CAN 2.0B Firmware

**Argument:** See Setting Description

This command sets the heater enable flag to activate or deactivate the heating functionality. Setting to a NONZERO value enables heating. Issue this command without an argument to read the current value of the configuration variable.

#### *EM – Set errMode – Error Mode (Actuator Hardware Protection Feature)*

---

**Applies to:** All Controllers

**Argument:** See Setting Description

This command sets the configuration variable “errMode – Error Mode (Actuator Hardware Protection Feature)” (more info on page 53) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

#### *FC – Set cFilter - Current filter*

---

**Applies to:** “N” – Classic Controller and “P” – Industrial Controller

**Argument:** See Setting Description

This command sets the configuration variable “cFilter - Current filter” (more info on page 50) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

#### *FP – Set pFilter – 1-2ms Pulse (RC Servo) filter*

---

**Applies to:** “N” – Classic Controller

**Argument:** See Setting Description

This command sets the configuration variable “pFilter – 1-2ms Pulse (RC Servo) filter” (more info on page 51) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

#### *FV – Set vFilter – Analog Voltage Filter*

---

**Applies to:** “N” – Classic Controller and “P” – Industrial Controller

**Argument:** See Setting Description

This command sets the configuration variable “vFilter – Analog Voltage Filter” (more info on page 50) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

## *H1 and H2 – Set ioPWM1 and ioPWM2 – PWM Output Signal Type for GPIO pin 1 and 2 (Respectively)*

---

**Applies to:** “N” – Classic Controller

**Argument:** See Setting Description

These commands set the configuration variables “ioPWM1 and ioPWM2 – PWM Output Signal Type for GPIO pin 1 and 2 (Respectively)” (more info on page 57) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

## *HC – Read/Set heatCur the heater current value*

---

**Applies to:** “P” – Industrial Controller, “C” – CAN Controller with CAN 2.0B Firmware

**Argument:** See Setting Description

This command sets the quiescent motor current value used when heating is activated. The units are the same as the maxTorq variable. Issue this command without an argument to read the current value of the configuration variable.

## *IC – Set stepSize – Incremental Step Size*

---

**Applies to:** “N” – Classic Controller

**Argument:** See Setting Description

This command sets the configuration variable “stepSize – Incremental Step Size” (more info on page 52) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

## *IR – Set revSgnl – Inversion of Response to Analog Current Input Signal*

---

**Applies to:** “P” – Industrial Controller

**Argument:** See Setting Description

This command sets the configuration variable “revSgnl – Inversion of Response to Analog Current Input Signal” (more on page 51) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

## *IM – Set ifMode – Serial Interface mode*

---

**Applies to:** All Controllers

**Argument:** See Setting Description

This command sets the configuration variable “ifMode – Serial Interface mode” (more info on page 49) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

## *KP, KI, and KD – Set kp, ki, and kd – PID Gains*

---

**Applies to:** All Controllers

**Argument:** See Setting Description

These commands set the configuration variables “kp, ki, and kd – PID Gains” (more info on page 47) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

## *LN – Set spMin - Software Position Minimum*

---

**Applies to:** All Controllers

**Argument:** See Setting Description

This command sets the configuration variable “spMin - Software Position Minimum” (more info on page 46) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

### *LX – Set spMax - Software Position Maximum*

---

**Applies to:** All Controllers

**Argument:** See Setting Description

This command sets the configuration variable “spMax - Software Position Maximum” (more info on page 46) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

### *MN and MX – Set xOutMin and xOutMax – Data Range for Analog Current Position Feedback*

---

**Applies to:** “P” – Industrial Controllers

**Argument:** See Setting Description

This command sets the configuration variable “xOutMin and xOutMax – Data Range for Analog Current Position Feedback” (more on page 56) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

### *MT – Set maxTorq - Max Torque*

---

**Applies to:** All Controllers

**Argument:** See Setting Description

This command sets the configuration variable “maxTorq - Max Torque” (more info on page 47) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

### *MS – Set xOutSrc – Analog Current Output Data Source*

---

**Applies to:** “P” – Industrial Controller

**Argument:** See Setting Description

This command sets the configuration variable “xOutSrc – Analog Current Output Data Source” (more on page 55) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

### *OE – Set ovErrP – Position Error Threshold*

---

**Applies to:** All Controllers

**Argument:** See Setting Description

This command sets the configuration variable “ovErrP – Position Error Threshold” (more info on page 54) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

### *OT – Set ovTorq – Over Torque Threshold*

---

**Applies to:** All Controllers

**Argument:** See Setting Description

This command sets the configuration variable “ovTorq – Over Torque Threshold” (more info on page 55) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

### *PA – Set Absolute Target Position*

---

**Applies to:** “N” – Classic Controller and “P” – Industrial Controller

**Argument:** spMin to spMax, or none

Sending this command with a valid argument will set the current target position to that value, resulting in an immediate move to the newly position. This writes directly to the position register of the actuator, bypassing all trajectory generation. This leads to a full acceleration, full speed move to the new target position and ignoring maximum speed and maximum acceleration settings. Users will typically use this command for maximum dynamic performance in conjunction with an external trajectory generator.

Sending this command without an argument will return the current target position (the value of the target position register).

#### *PC – Set Target Position to Current Position*

---

**Applies to:** “N” – Classic Controller and “P” – Industrial Controller

**Argument:** None

This command sets the current position as the Servo Cylinder’s setpoint, causing the actuator to hold in its current place.

#### *PO – Set Target Position with an Offset*

---

**Applies to:** “N” – Classic Controller and “P” – Industrial Controller

**Argument:** See Description

Sending this command with a valid argument will offset the current target position by an amount equal to the argument, resulting in an immediate move to the newly requested position.

$$Position_{new} = Position_{old} + Argument$$

This command directly writes the position register of the actuator, bypassing all trajectory generation. This leads to a full acceleration, full speed move to the new target position, ignoring maximum speed and maximum acceleration settings that are features of trajectory generation. Users will typically use this command for maximum dynamic performance in conjunction with an external trajectory generator.

#### *PN and PX – Set pMin and pMax – 1-2 ms Pulse (RC PWM Servo) Control Pulse Width Min and Max (Respectively)*

---

**Applies to:** “N” – Classic Controller

**Argument:** See Setting Description

These commands set the configuration variable “pMin and pMax – 1-2 ms Pulse (RC PWM Servo) Control Pulse Width Min and Max (Respectively)” (more info on page 51) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

#### *SH – Read/Set heatTmp the heater threshold temperature*

---

**Applies to:** “P” – Industrial Controller, “C” – CAN Controller with CAN 2.0B Firmware

**Argument:** See Setting Description

Defines the temperature threshold that causes the heater to activate. Units are in °C

#### *SM – Set stopMode – End-of-Trajectory Stopping mode*

---

**Applies to:** “N” – Classic Controller and “P” – Industrial Controller

**Argument:** See Setting Description

This command sets the configuration variable “stopMode – End-of-Trajectory Stopping mode” (more info on page 48) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

#### *SP – Set maxSpeed - Maximum Speed*

---

**Applies to:** All Controllers

**Argument:** See Setting Description

This command sets the configuration variable “maxSpeed - Maximum Speed” (more info on page 48) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

### *VN and VX - Set vMin and vMax – Analog Voltage Min and Analog Voltage Max (Respectively)*

---

**Applies to:** “N” – Classic Controller and “P” – Industrial Controller

**Argument:** See Setting Description

These commands set the configuration variables “vMin and vMax – Analog Voltage Min and Analog Voltage Max (Respectively)” (more info on page 50) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

### *W1 and W2 – Set ioBit1 and ioBit2 – Masked Status Register Bit Output to GPIO Pin 1 and 2 (Respectively)*

---

**Applies to:** “N” – Classic Controller

**Argument:** See Setting Description

These commands set the configuration variables “ioBit1 and ioBit2 – Masked Status Register Bit Output to GPIO Pin 1 and 2 (Respectively)” (more info on page 57) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

### *YN and YX – Set cOutMin and cOutMax – Current range for Analog Current Position Feedback (Min and Max Respectively)*

---

**Applies to:** “P” – Industrial Controller

**Argument:** See Setting Description

This command sets the configuration variable “cOutMin and cOutMax – Current range for Analog Current Position Feedback (Min and Max Respectively)” (more on page 56) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

### *YR – Set revCout – Inversion of Analog Current Output Signal Response*

---

**Applies to:** “P” – Industrial Controller

**Argument:** See Setting Description

This command sets the configuration variable “revCout – Inversion of Analog Current Output Signal Response” (more on page 56) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

### *ZI – Clear PID Filter I-term and D-Term*

---

**Applies to:** All Controllers

**Argument:** None

Clears the values of the “I” term and the “D” term of the PID control loop. This feature may be useful to stop actuator dithering about the set position in self-locking actuator applications.

## Trajectory Commands

Trajectory commands are used to smoothly move the actuator to a desired position using the user configurable max speed and acceleration.

### *T1, T2, T3, and T4 – Trajectory Move to positions Pos1, Pos2, Pos3, and Pos4 (Respectively)*

---

**Applies to:** “N” – Classic Controller

**Argument:** None

Trajectory move to Pos1, Pos2, Pos3, or Pos4 with user defined speed and acceleration. See Pos1, Pos2, Pos3, and Pos4 – Preset Positions 1, 2, 3, and 4 (Respectively) on page 53.

Command	Position
T1	Pos1
T2	Pos2
T3	Pos3

Command	Position
T4	Pos4

Table 46: Commands and respective destination positions for T1, T2, T3, and T4

### *TA – Trajectory Move to Absolute Position*

**Applies to:** “N” – Classic Controller and “P” – Industrial Controller

**Argument:** spMin to spMax

Sending this command result in a trajectory move to the requested position equal to the argument. Trajectory moves obey the speed and acceleration settings.

Example: Move to absolute position 25,000 with acceleration value 2,000 and speed 200,000:

```
AC2000
SP200000
TA25000
```

### *TE – Trajectory Move to Fully Extended Position (spMax)*

**Applies to:** “N” – Classic Controller and “P” – Industrial Controller

**Argument:** None

Trajectory move to “spMax - Software Position Maximum” with user defined speed and acceleration.

### *TK – Interrupt Current Trajectory Move*

**Applies to:** “N” – Classic Controller and “P” – Industrial Controller

**Argument:** None

Halt the current trajectory motion before completion.

### *TM – Trajectory Move to Midpoint*

**Applies to:** “N” – Classic Controller and “P” – Industrial Controller

**Argument:** None

Trajectory move to midpoint with user defined speed and acceleration. The midpoint is defined by

$$\frac{spMin + (spMax - spMin)}{2}$$

### *TO – Trajectory Move to Offset (Incremental Trajectory Move)*

**Applies to:** “N” – Classic Controller and “P” – Industrial Controller

**Argument:** None

Trajectory move a relative distance from the Servo Cylinder’s current position. This command will not execute if the requested offset move will send the actuator beyond spMin or spMax.

### *TR – Trajectory Move to Fully Retracted Position (spMin)*

**Applies to:** “N” – Classic Controller and “P” – Industrial Controller

**Argument:** None

Trajectory move to “spMin - Software Position Minimum” with user defined speed and acceleration.

## Serial Data Streaming

### *X1, X2, and X3 – Set strx1, strx2, and strx3 – Data Stream for Columns 1, 2, and 3 (Respectively)*

---

**Applies to:** “N” – Classic Controller and “P” – Industrial Controller

**Argument:** See Setting Description

These commands set the configuration variables “strx1, strx2, and strx3 – Data Stream for Columns 1, 2, and 3 (Respectively)” (more info on page 59) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

### *XC – Clear Timer*

---

**Applies to:** “N” – Classic Controller and “P” – Industrial Controller

**Argument:** None

Reset the 32-bit timer used for synchronizing data to a time interval.

### *XG – Turn Data Streaming On*

---

**Applies to:** “N” – Classic Controller and “P” – Industrial Controller

**Argument:** None

Initiate data streaming.

### *XR – Display Data Stream Setup*

---

**Applies to:** “N” – Classic Controller and “P” – Industrial Controller

**Argument:** None

Display the current stream setup including what variables are being streamed and streaming frequency.

### *XS – Turn Data Streaming Off*

---

**Applies to:** “N” – Classic Controller and “P” – Industrial Controller

**Argument:** None

Stop the data streaming.

### *XT – Set sTime – Data Stream Time Interval*

---

**Applies to:** “N” – Classic Controller and “P” – Industrial Controller

**Argument:** See Setting Description

This command sets the configuration variable “sTime – Data Stream Time Interval” (more info on page 58) to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

## System Commands

### *BR – Restore File System from Backup*

---

**Applies to:** “P” – Industrial Controller, “C” – CAN controller

**Argument:** 321

Restores the configuration settings file from non-volatile persistent storage.

### *CW – Write current settings to CONFIG.TXT (“P” – Industrial Controller, “C” – CAN controller)*

---

**Applies to:** “P” – Industrial Controller, “C” – CAN controller

**Argument:** 321

This command will take all current operational settings and write them to CONFIG.TXT. This must be done if a setting was changed via a serial command, and you wish to retain this change after a reboot. Use “ZW321” for the “N” – Classic Controller.



WARNING: Running this command will strip CONFIG.TXT of all comments. Back up CONFIG.TXT before running this command to not lose this information.

### *FR – Read File from Internal File System*

---

**Applies to:** “P” – Industrial Controller, “C” – CAN controller

**Argument:** “File Name”

Reads the configuration settings file currently backed up in non-volatile persistent storage.

### *SB – Back Up Internal File System*

---

**Applies to:** “P” – Industrial Controller, “C” – CAN controller

**Argument:** 321

Backs up the current configuration settings to non-volatile persistent storage. This storage will persist through a firmware update.

### *ZC – Run Calibration Routine*

---



WARNING: Servo Cylinder must be at least one full motor rotation (1024 encoder counts) from either physical hard-stop before running this command. Running the calibration routine command when the actuator is less than one full motor rotation from an end stop will cause damage.



WARNING: Backup all configuration files before calibrating, as this command will overwrite CONFIG.TXT and HARDWARE.TXT



WARNING: The servo Cylinder must be disconnected from any external load and free to move the full stroke/travel length during the entire calibration routine.

**Applies to:** All Controllers

**Argument:** 321

This command should only be sent after consulting with Ultra Motion engineers. The Servo Cylinder must be disconnected from any external system and free to move for the entire stroke of the actuator. Additionally, it must be at a position at least one revolution from either hard-stop. The CONFIG.TXT file should be saved to a local location. After calibration, the HARDWARE.TXT file is written and the Servo Cylinder is reset, at this time CONFIG.TXT should be reloaded on to the Servo Cylinder.

For “C” –CAN Controllers and “P” – Industrial Controllers, the CONFIG.TXT and HARDWARE.TXT files can be backed up to the Servo Cylinder’s on-board persistent memory with the “SB321” command. After the firmware update is complete, the backup can be restored with the “BR321” command.

### *ZD – Read Bridge Driver DIAG Register*

---

**Applies to:** All Controllers

**Argument:** None

Read the detailed bridge driver diagnostic information from the Servo Cylinder.

### *ZF – Clear Latching Fault Flags in Status Register*

---

**Applies to:** All Controllers

**Argument:** None

Clear all latching fault flags in the status register.

### *ZR – Reboot Actuator*

---

**Applies to:** All Controllers

**Argument:** 321

Power cycle the actuator. When rebooting, the Servo Cylinder will draw all configuration settings from current configuration file (CONFIG.TXT). This must be done to enact any changes to that were made within CONFIG.TXT.

## *ZU – Jump to Serial Firmware Updater*

---

**Applies to:** All Controllers

**Argument:** 321

Suspend normal operation of the Servo Cylinder and launch the on-board bootloader to update the firmware. The actuator should be removed/disconnected from the machine to prevent potential damages and must be disconnected from USB.

## *ZW – Write current settings to CONFIG.TXT (“N” – Classic Controller Only)*

---

**Applies to:** “N” – Classic Controller

**Argument:** 321

This command will take all current operational settings and write them to CONFIG.TXT. This must be done if a setting was changed via a serial command, and you wish to retain this change after a reboot.



WARNING: Running this command will strip CONFIG.TXT of all comments. Back up CONFIG.TXT before running this command to not lose this information.

## Appendix B: Converting Between Sensor Counts and Physical Units

The sensor readings onboard the Servo cylinder, such as the position, force/torque, temperature, etc. are all given in the units of sensor counts. Sensor counts are in integer form, and must be converted in order to arrive at physical units. When units are entered into the actuator they must be rounded to the nearest integer.

To quickly convert between sensor counts and physical units, an excel spreadsheet tool is available for download from the Ultra Motion website, which can be found here:

<https://www.ultramotion.com/servo-cylinder/#downloads>

Alternatively, you can use the following two tables of to convert to and from physical units.

### Conversion Tables

Variable	Definition	Units	Note
$N$	Sensor Counts	–	$N$ denotes sensor counts for the specific data type in question
$P$	Position	$in$	Linear Position with respect to retracted endstop (rPos) of the actuator. The retracted endstop is always at $N = 1024$ .
$L$	Screw Lead	$in$	The lead of the drive screw, see “Drive Screw” section on page 14.
$a$	Acceleration	$in/s^2$	Linear acceleration, which is always positive (scalar) for setting ‘accel’
$v$	Velocity	$in/s$	Linear velocity of the actuator shaft
$f$	Stream Frequency	$Hz$	The frequency at which data is being streamed
$V_{AN}$	Analog Voltage Input	$V$	The voltage applied for analog voltage control.
$I_{AN}$	Analog Current Input	$mA$	The current applied for analog current control.
$V_{dc}$	Bus Voltage	$V$	Input voltage to the actuator from the power supply unit
$T$	Temperature	$^{\circ}F$	Internal temperature, as measured at the controller, not motor windings.
$H_R$	Relative Humidity	%	Relative humidity, actual values will be between 0 and 100%
$t_{pwm}$	Pulse width	$ms$	Duration of the pulse provided by 1-2 ms Pulse (RC PWM Servo) Control signal.
$I_{phase}$	Phase Current	$A$	Current currently passed through a specific motor phase
$I_{RMS}$	Equivalent Current	$A$	Equivalent RMS current passed through the entire motor.
$I_{ANO}$	Analog Current Output	$mA$	Analog output current, for feedback on “P” – Industrial Controllers

Table 47: Definition of variables used in Table 48: Conversions between Sensor Counts and Physical Unit

## Conversion Tables

Data Type	Range (sensor counts)	Range (physical units)	Conversion to physical units	Conversion to sensor counts
Position	rPos to ePos	Varies (See Equation)	$P = \frac{(N - 1024)L}{1024}$	$N = \frac{1024P}{L} + 1024$
Velocity*	0 to 50×10 <sup>6</sup>	Varies (See Equation)	$v = \frac{1250NL}{2^{16} \times 1024}$	$N = \frac{2^{16} \times 1024v}{1250L}$
Acceleration	0 to 130,071	Varies (See Equation)	$a = \frac{1250^2NL}{2^{16} \times 1024}$	$N = \frac{2^{16} \times 1024a}{1250^2L}$
Torque/Force	-32767 to 32767	See page 92	See page 92	See page 92
Analog Voltage Input	0 to 65535	-10.993 V to 10.704 V	$V_{AN} = \frac{21.637(N - 33114)}{65535}$	$N = \frac{65535V_{AN}}{21.637} + 33114$
Analog Current Input	0 to 65535	3.175 mA to 20.375 mA	$I_{AN} = \frac{17.2N}{65535} + 3.175$	$N = \frac{65535(I_{AN} - 3.175)}{17.2}$
Analog Current Output	0 to 65535	3.72 mA to 20.93 mA	$I_{ANO} = .0002637N + 3.721$	$N = \frac{I_{ANO} - 3.721}{.000263}$
Pulse Width	60000 to 120000	1 ms to 2 ms	$t_{pwm} = \frac{N}{60000}$	$N = 60000t_{pwm}$
Bus Voltage	0 to 1023	0 V to 50 V	$V_{dc} = \frac{50N}{1024}$	$N = \frac{1024V_{dc}}{50}$
Velocity (streamed)	-512 to 511	Varies (See Equation)	$v = \frac{fNL}{1024}$	$N = \frac{1024v}{fL}$
Temperature ("N" Controller)	0 to 65535	-58°F to 482°F	$T = \frac{N - 7037.344}{121.333}$	$N = 121.333T + 7037.344$
Temperature ("P" Controller)	0 to 65535	-40°F to 305°F	$T = .006N - 88.3$	$N = \frac{T + 88.3}{.006}$
Relative Humidity	0 to 65520	-12.5% to 112.5% <sup>†</sup>	$H_R = \frac{125N}{65520} - 12.5$	$N = \frac{65520H_R}{125} + 6552$
Phase Current	-512 to 511	-25 A to 24.95 A	$I_{phase} = \frac{25N}{512}$	$N = \frac{512I_{phase}}{25}$
Motor Current	-32768 to 32767	-17.7 A to 17.7 A	$I_{RMS} = \frac{N}{1848.43}$	$N = 1848.43I_{RMS}$

Table 48: Conversions between Sensor Counts and Physical Units

\*This velocity value is that which is used by maxSpeed and the SP command. The sensor counts from streamed data are scaled differently, and are dependent upon the stream frequency.

†Relative Humidity will only ever be measured between 0% and 100%. These RH% values which extend outside this range correspond to upper and lower limits of sensor counts, and are for calculation reference only.

## Converting Torque to Force

---

While there is a direct relationship between motor torque and output force, there are some complicating factors which can have a significant impact on this relationship, including seal friction, lubricant viscosity, and rotational inertial loads. In applications where it is critical that applied force is precisely known or controlled, we highly recommend using an external load cell in a control loop.

To perform this conversion most accurately, use the test data that is provided with your actuator. This data was collected on each actuator during the Acceptance Test Procedure (ATP) performed on every actuator with an “N” or “P” controller before leaving the factory.

## Contact Information

If you have any questions about the Servo Cylinder or any of our other products, contact us by one of the following methods:



### INQUIRY

Leave a web inquiry (to be replied to within one business day):

[ultramotion.com/contact](http://ultramotion.com/contact)



### LIVE CHAT

Live Chat directly with one of our engineers:

[ultramotion.com](http://ultramotion.com)



### EMAIL

Email (to be replied to within one business day):

[info@ultramotion.com](mailto:info@ultramotion.com)



### CALL

PH: 888-321-9178

Fax: 631-298-6593



### ADDRESS

Ultra Motion

22355 CR 48, #21

Cutchogue, NY 11935



### HOURS

Our Business Hours:

Monday-Friday

9AM – 5PM EST

