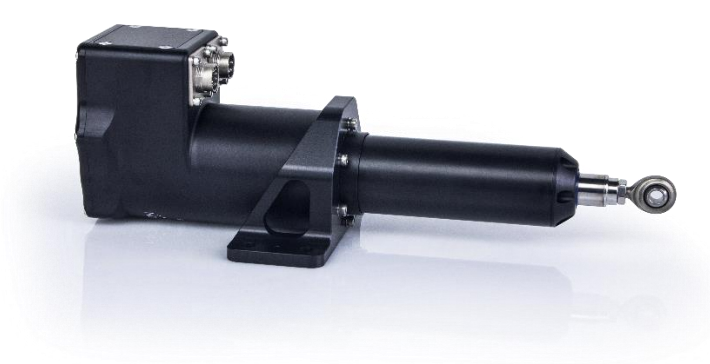
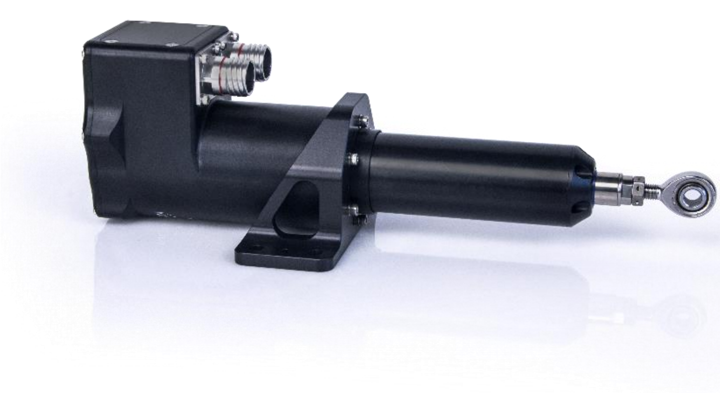
CAN Firmware v1.0

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*

Revision History

Revision	Date	Details
A.01	3/26/2021	Initial Release

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. <u>All life related data is provided as reference only</u> 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

Servo Cylinder CAN Firmware v1.0 Details

The following reference document reviews Ultra Motion's standard CAN 2.0B firmware v1.0

For details on the Servo Cylinder performance, configuration, wiring, etc. please reference the Servo Cylinder manual UM711293.

Servo Cylinder CAN Behavior/Configuration

This section provides an overview of the Servo Cylinder CAN firmware configuration options and behavior.

Operating Mode

The operating modes for this firmware include command line interface control to aid integration, and CAN control. The operating mode can be set via the opMode variable in the CONFIG.TXT file, or via the RS-485 interface with the "OP" command.

opMode = 0 enables motion control through the RS-485 serial Command Line Interface

opMode = 1 enables motion control through CAN position command messages

Command Line Interface Lock

The Servo Cylinder will start-up with the CLI locked, which means no received commands will be executed. This is to protect the actuator from executing erroneous commands due to noise or user error. To use the command line interface the special command "LK" with the argument "unlock" is required (i.e. "LK unlock\r"). The actuator's CLI will lock after every power cycle.

CAN Baud Rate

The Servo Cylinder's CAN baud rate is definable with the CANspd variable in the CONFIG.TXT file or via RS-485 with the "CS" command. Values of 0 to 5 are acceptable and defined as follows:

CANspd =

0: 1 Mbps
1: 500 Kbps
2: 250 Kbps
3: 125 Kbps
4: 100 Kbps
5: 50 Kbps

CAN Message Type

The firmware will support either 11-bit (standard) or 29-bit (extended) identifiers. The CANext variable can be set in the CONFIG.TXT file or via RS-485 with the "CE" command.

CANext = 0: Standard message type (11-bit identifier)

CANext = 1: Extended message type (29-bit identifier)

Unit ID

unitID is the actuator's network address and is used along with IDmask to filter incoming CAN command messages. For acceptance, the message's identifier must match unitID where each corresponding bit in IDmask is set to '1'. If CANext = 0, only the lower 11 bits of unitID and IDmask are examined (standard message type). If CANext is not 0, the lower 29 bits of unitID and IDmask are examined (extended message type). Configuration is accomplished with the unitID variable in the CONFIG.TXT file or via RS-485 with the “ID” command. unitID can be any value from 0x00000000 to 0x1FFFFFFF.

ID Mask

IDmask is used along with unitID to filter incoming CAN command messages. For acceptance, the message’s identifier must match unitID where each corresponding bit in IDmask is set to ‘1’. If a bit in IDmask is set to ‘0’, the corresponding bit of an incoming can message is matched for acceptance regardless of its value. If IDmask is set to 0x00000000, CAN messages with any identifier will be accepted. If IDmask is set to 0x1FFFFFFF, only CAN messages with an identifier that exactly matches the relevant bits of unitID will be accepted. If CANext = 0, only the lower 11 bits of unitID and IDmask are examined (standard message type). If CANext is not 0, the lower 29 bits of unitID and IDmask are examined (extended message type). Configuration is accomplished with the IDmask variable in the CONFIG.TXT file or via RS-485 with the “MA” command. IDmask can be any value from 0x00000000 to 0x1FFFFFFF.

Command Message Data Format

The Servo Cylinder can receive up to 8 data bytes in each position command message from the master controller. The rxData variable is a string with a maximum length of 8 characters where each character designates a byte of data in the command message. The actuator must be configured to receive a position command.

The data format for received CAN position command messages is configured with the rxData variable in the CONFIG.TXT file or via RS-485 with the “DR” command.

Command Message Data Format

Char	Description
<	Position command low byte
>	Position command high byte
(	Max torque low byte
)	Max torque high byte
X	Ignore byte
x	Ignore byte

The default value of rxData = “<>” requires the user to send a CAN message with 2 data bytes containing the low byte of the position command as the first data byte in the message, and the high byte of the position command as the second data byte.

Position Interpolation

A position interpolation function provides smooth motion between each position update. The smoother motion provided by the interpolator increases mechanical reliability and actuator response.

Interpolation can be enabled/disabled with the inEna configuration variable or via RS-485 with the “EI” command.

inEna = 0 disables interpolation

inEna = 1 enables interpolation

Interpolation Period

The interpolation period can be set with the intvl variable in the CONFIG.TXT file, or via RS-485 with the “IL” command. This variable is set in 800 µs increments and should be set equal to the expected position update interval from the master controller. If the update rate cannot be matched exactly, setting the interpolation interval slightly larger than the expected update interval is recommended. The default value of intvl is 50. This value is equal to a 40ms update interval (25Hz update frequency).

Default Start Position

The Servo Cylinder can be configured to drive to a default position on startup before receiving a valid CAN message, or on failure of the CAN bus leading to a timeout event. These events would execute a trajectory move to the position defined by defPos with trajectory parameters defined by the maxSpeed and accel configuration variables. As an example, the defPos variable could be set to a control surface's neutral position.

The default position can be set in the CONFIG.TXT file with the defPos variable, or via RS-485 with the "DP" command. The units are in Phase Index absolute position encoder counts

Received Message Timeout

The received message timeout period can be set with the rxTO configuration variable or via RS-485 with the "NT" command. This variable is set in 800 μs increments. The default value is 1250 (1 second) When a timeout occurs, the appropriate status register bit will become active and the actuator's timeout behavior (set by TOact) will initiate.

Startup Action

The Servo Cylinder can be configured for different behaviors at startup before receiving the first valid CAN message.

SUact = 0 holds the current position

SUact = 1 executes a trajectory move to the default position defPos

The startup action can be set with the SUact configuration variable or via RS-485 with the "AS" command.

Timeout Action

The Servo Cylinder can be configured for different behaviors after a CAN timeout event

TOact = 0 holds the current position

TOact = 1 executes a trajectory move to the default position defPos

The timeout action can be set with the TOact configuration variable or via RS-485 with the "AR" command.

Servo Cylinder Telemetry

The Servo Cylinder will report back user configurable telemetry via the CAN bus with a configurable telemetry message ID.

Telemetry Enable

The txEna configuration variable is used to turn on/off the Servo Cylinder CAN telemetry transmit functionality. This can be configured by editing the CONFIG.TXT file or via RS-485 with the "ET" command.

txEna = 0 disables telemetry

txEna = 1 enables telemetry

Telemetry Interval

The rate at which the Servo Cylinder transmits the telemetry CAN message is configurable and set by editing the txIvl variable in the CONFIG.TXT file or via RS-485 with the "IT" command. The setting is in milliseconds and has a valid range of 16 to 65535. Care must be taken to not overload the CAN bus by setting the interval value too low. The default value is 1000 (1 second).

Telemetry Message ID

The txID variable sets the identifier for the transmitted telemetry CAN message. This represents the destination of the telemetry message. Either the lower 11 bits (standard) or the lower 29 bits (extended) of txID are used depending on CAN message type setting (CANext). The txID variable can be set via the CONFIG.TXT file or via RS-485 with the "NI" command. The user can specify the ID in decimal or hex if preceded with "0x". txID can be any value from 0x00000000 to 0x1FFFFFFF, the default value for txID is 0x0000007F.

Telemetry Message Data Format

The data bytes of the telemetry message are configurable by setting the txData variable via the CONFIG.TXT file or via RS-485 with the "DT" command. The txData variable is a string where each character designates a byte of data in the telemetry message. txData may be 1 to 8 characters in length.

Telemetry Message Data Format

Char	Description
A	Status word byte 0 (LSB)
B	Status word byte 1
C	Status word byte 2
D	Status word byte 3 (MSB)
E	Average motor current over telemetry interval (0 to 32767) byte 0 (LSB)
F	Average motor current over telemetry interval (0 to 32767) byte 1 (MSB)
G	Servo Cylinder position, absolute encoder value (0 to 65535) byte 0 (LSB)
H	Servo Cylinder position, absolute encoder value (0 to 65535) byte 1 (MSB)
I	Position converted to input range (pMin to pMax) byte 0 (LSB)
J	Position converted to input range (pMin to pMax) byte 1 (MSB)
K	Latched high copy of status word byte 0 (LSB)
L	Latched high copy of status word byte 1
M	Latched high copy of status word byte 2
N	Latched high copy of status word byte 3 (MSB)
O	Latched low copy of status word byte 0 (LSB)
P	Latched low copy of status word byte 1
Q	Latched low copy of status word byte 2
R	Latched low copy of status word byte 3 (MSB)
S	8-bit position between physical stops (rPos to ePos)
T	8-bit motor current 16-sample average of last 16 ms (0 to 255)
U	8-bit bus voltage 0 VDC to +50 VDC (0 to 255)
V	8-bit average motor current over telemetry interval (0 to 255)
W	8-bit max motor current over telemetry interval (0 to 255)
X	8-bit signed integer PCB temp sensor °C (-50 to +127)
Y	8-bit unsigned PCB temp sensor in °C where 0 = -50°C and 200 = +150°C (0 to 200)
Z	8-bit PCB relative humidity % (0 to 100)
m	Max motor current over telemetry interval (0 to 32767) byte 0 (LSB)
c	Max motor current over telemetry interval (0 to 32767) byte 1 (MSB)
p	unitID byte 0 (LSB)
q	unitID byte 1
r	unitID byte 2
s	unitID byte 3 (MSB)
t	Target position, absolute encoder value byte 0 (LSB)
u	Target position, absolute encoder value byte 1 (MSB)

The telemetry message may include anywhere from 1 to 8 bytes of data and they can be any combination of the above options. Latched status bytes are reset after they are transmitted. Latched low bytes are reset high, and latched high bytes are reset low. The default value of txData is "KLMGHEFY"

Status Word Definition

The status word contains details of the Servo Cylinder's health and state. The configurable telemetry includes latched and non-latched versions of the individual status word bytes.

Status Word

Bit	Description
0	Position at or beyond retracted physical stop rPos
1	Position at or beyond extended physical stop ePos
2	Position beyond retracted software limit spMin
3	Position beyond extended software limit spMax
4	Supply voltage low, motor in COAST (<6.75 VDC, 1 V hysteresis)
5	Supply voltage high, motor in dynamic brake (>44.0 VDC, 2 V hysteresis)
6	Torque output greater than ovTorq limit
7	Torque command at maxTorq limit
8	Speed below "stop" threshold
9	Direction is extend
10	Position at target (position near target within posWin for posTime)
11	Following error (position error larger than fErrWin for time period fErrTime)
12	Command RX error (message not received in [rxTO * 800 μs])
13	Telemetry TX error (message not sent over full telemetry interval)
14	CAN position command input capped at low limit pMin
15	CAN position command input capped at upper limit pMax
16	Trajectory move active
17	Heating active
18	Temperature at PCB greater than ovTemp value
19	Temperature at PCB less than unTemp value
20	Relative humidity at PCB greater than ovHumi value
21	Fatal error in CONFIG.TXT or HARDWARE.TXT
22	Fault output bit of DRV8323RS (Bridge Driver)
23	Erroneous warm reset of the CPU has occurred
24	opMode (CLI = 0, CAN = 1)
25	Interpolation enabled
26	Heating enabled
27	CAN bus module in passive mode
28	USB connected
29	Opto input 1
30	Opto input 2
31	Opto input 3

Servo Cylinder Heater Functionality

The Servo Cylinder has a configurable heating functionality to maintain an acceptable PCB temperature when the measured temperature drops below a configurable threshold. This section outlines the configuration variables used to define the behavior. The Servo Cylinder is rated for -40°C low temperature operation. The actuator must be initially powered on at a minimum temperature of -40°C.

Heater Enable

The heating functionality is enabled/disabled with the heatEna variable in the CONFIG.TXT file, or via RS-485 with the "EH" command.

heatEna = 0 disables heating function

heatEna = 1 enables heating function

Heater Temperature Threshold

The heatTmp variable sets the temperature threshold below which the heater will activate. This value is set in the CONFIG.TXT file or via RS-485 with the "SH" command. As an example, "SH -35.0" will set the heater activation threshold to -35.0 °C.

Heater Current Setting

The heater current setting defines the quiescent motor current that is applied when heating is activated. The Servo Cylinder controller will ensure the minimum current by increasing a non-torque producing magnetic field in the stator when the torque command drops below the minimum quiescent current setting. The controller will trade torque-producing and non-torque-producing magnetic fields as the load changes in order to ensure the minimum current setting is met. This quiescent current will produce heat in the PCB and stator when the PCB temperature falls below the heatTmp threshold. The heatCurr variable can be set in the CONFIG.TXT file or via RS-485 with the "HC" command. The units are the same as the motor current feedback.

Status LED

On Servo Cylinder A1 and A2 series actuators, a status LED indicates the operational state.

Status LED

LED	Description
Solid Dim White	Startup
Blinking 1Hz Bright Red	Error in CONFIG.TXT or HARDWARE.TXT or problem with bridge driver IC, **MOTOR OFF**
Blinking 1Hz Dim Green	Operating in CLI mode
Solid Dim Green	Operating in CAN mode
Solid Dim Red	CAN mode, Timeout waiting for CAN command message
Solid Yellow	CAN mode, 0 command messages received since startup

Servo Cylinder Configuration Variables

This section details configuration variables used by this firmware. These variables are stored in a text file named CONFIG.TXT which is located in the root directory of the controller's internal flash filesystem. The controller reads this file upon startup and loads the configuration information. Most of these variables can be updated through the serial command line interface (CLI). The CONFIG.TXT file can be read, written, or deleted through the CLI, and the CLI can be used to back up the file system to flash storage or restore the file system from backup. The internal filesystem can also be accessed through the USB port as a mass storage device.

opMode – Operating Mode

Variable Type: Integer
Valid Range: 0 to 1
Default: 1
Serial Command: OP

This setting determines whether the actuator's motion is controlled via the RS-485 command line interface (opMode = 0), or via the CAN bus (opMode = 1)

cliMode – Serial Interface mode

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 please reference UM711293 page 61

Setting	Mode	Description
0	Human Mode	All commands are followed by a verbose acknowledgement in the serial prompt. Detailed information regarding actuator commands is written to the serial prompt. Echo is enabled.
1	Machine Mode 1 (MACHINE1)	All commands are followed by a simplified acknowledgement (ACK/NACK). 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). 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. Both MACHINE1 and MACHINE2 modes return a checksum with the response.

cliBaud – Serial Baud Rate

Variable Type: Integer
Valid Range: 2400 to 256000
Default: 115200
Serial Command: -

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 diagnostics, configuration, and initial setup. Default Baud rate is 115200. Lower baud rates will be more tolerant to noise and crosstalk at the expense of data bandwidth.

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

Variable Type: Integer
Valid Range: 0 to 268435455 (for practical limits, see Tuning Performance on page 66 of UM711293)
Default: kp = 1200, ki =250000, kd = 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. Internal scaling of these gains is unique to this PID algorithm. Default factory values used represent typical stable gains.

spMin - Software Position Minimum

Variable Type: Integer
Valid Range: rPos to ePos (Note: rPos and ePos are the physical travel limits, found in HARDWARE.TXT)
Rules: rPos ≤ spMin < spMax ≤ 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 of the motor and screw.

spMax - Software Position Maximum

Variable Type: Integer
Valid Range: rPos to ePos (Note: rPos and ePos are the physical travel limits, found in HARDWARE.TXT)
Rules: rPos ≤ spMin < spMax ≤ 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.

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. Understand that hitting the hard end-stop can damage or reduce the operating life of the actuator.

maxTorq - Max Torque

Variable Type: Integer
Valid Range: 0 to 32767
Default value: 10000
Set Command: MT

This setting limits the torque demand signal that commands the FOC current loop, thereby limiting the force produced by the Servo Cylinder. The value represents a percentage of full force output where 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.

maxSpeed - Maximum Speed

Variable Type: Integer
Valid Range: 0 to 50000000 (Practical upper limit ~8000000)
Default: 1500000
Set Command: SP

This variable sets the top speed for trapezoidal profile trajectory moves. Note that it is possible to set the max speed higher than what the motor can move at. The maximum speed that the motor can move at depends on the operating voltage, load, and other factors.

accel - Acceleration and Deceleration Rate

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 profile trajectory moves. Note: the acceleration and deceleration will be equal.

CANspd – CAN Baud Rate

Variable Type: Integer
Valid Range: 0 to 5
Default: 0
Serial Command: CS

This variable defines the baud rate of the CAN interface. There are 6 options designated by the integers 0 to 5.

0 = 1 Mbps, 1 = 500 Kbps, 2 = 250 Kbps, 3 = 125 Kbps, 4 = 100 Kbps, 5 = 50 Kbps

CANext – CAN Message Type

Variable Type: Integer
Valid Range: 0 to 65535
Default: 1
Serial Command: CE

This variable defines whether the Servo Cylinder is configured for CAN2.0B standard message type (11-bit identifiers), or extended message type (29-bit identifiers). CANext = 0 configures standard message type (11-bit ID). A non-zero value in CANext sets the actuator to extended message type (29-bit ID).

unitID – CAN Identifier

Variable Type: Integer
Valid Range: 0x00000000 to 0x1FFFFFFF
Default: 0x00000003
Serial Command: ID

unitID is the actuator's network address and is used along with IDmask to filter incoming CAN command messages. For acceptance, the message's identifier must match unitID where each corresponding bit in IDmask is set to '1'. If CANext = 0, only the lower 11 bits of unitID and IDmask are examined (standard message type). If CANext is not 0, the lower 29 bits of unitID and IDmask are examined (extended message type).

IDmask –Mask for CAN Identifier

Variable Type: Integer
Valid Range: 0 to 0x1FFFFFFF
Default: 0x1FFFFFFF
Serial Command: MA

IDmask is used along with unitID to filter incoming CAN command messages. For acceptance, the message's identifier must match unitID where each corresponding bit in IDmask is set to '1'. If a bit in IDmask is set to '0', the corresponding bit of an incoming can message is matched for acceptance regardless of its value. If IDmask is set to 0x00000000, CAN messages with any identifier will be accepted. If IDmask is set to 0x1FFFFFFF, only CAN messages with an identifier that exactly matches the relevant bits of unitID will be accepted. If CANext = 0, only the lower 11 bits of unitID and IDmask are examined (standard message type). If CANext is not 0, the lower 29 bits of unitID and IDmask are examined (extended message type).

rxData – Actuator Command Data Configuration

Variable Type: String
Valid Range: <>()Xx
Default: <>
Serial Command: DR

This variable configures the data format of received position command messages. rxData is a string with a length from 1 to 8 characters. Each character in rxData designates a function for the corresponding data byte in the position command message. The number of data bytes in the position command message must match the number of characters in rxData. For example, if the master is sending out command messages with 8 data bytes, then rxData should be padded with ‘X’ or ‘x’ like this: “xxxx<>xx”, where ‘<’ = position low byte, ‘>’ = position high byte, and ‘x’ = ignore byte. The characters ‘(‘ and ‘)’ can also be used to set the maxTorq value, similar to running the “MT” command through the CLI. ‘(‘ is maxTorq low byte, and ‘)’ is maxTorq high byte. The acceptable range for the position command value is pMin to pMax. The acceptable range of the maxTorq value is 0 to 32,767. maxTorq is not updated if ‘(‘ or ‘)’ characters are not included in rxData.

pMin and pMax – Position command range

Variable Type: Integer
Valid Range: 0 to 65535
Default: pMin = 0, pMax = 65535
Set Command: PN, PX

This variable defines the valid command range that will be sent to the actuator via CAN messages. The value of pMin/pMax will be mapped to the actuator’s travel range spMin/spMax.

sigInv – Inversion of Response to Input Signal

Variable Type: Integer
Valid Range: 0 or 1
Default: 0
Set Command: SI

Inverts the response of the actuator with respect to the input signal. A value of 0 does not invert. A value of 1 inverts.

SUact – Startup Action

Variable Type: Integer
Valid Range: 0 to 65535
Default: 0
Serial Command: AS

This variable defines the behavior of the actuator on start up before receiving a valid CAN command message. A value of 0 causes the actuator to hold position and a value of 1 executes a trajectory move to defPos

defPos – Default Position

Variable Type: Integer
Valid Range: spMin to spMax
Default: (spMin + spMax)/2
Serial Command: DP

defPos is the absolute encoder value of the default position. The actuator can move to defPos upon startup if no valid position command messages have been received, or in the event of a position command message not being received in the rxTO timeout period. The behavior of the actuator in these situations is defined by the SUact and TOact variables.

inEna –Interpolation Enable Flag

Variable Type: Integer
Valid Range: 0 to 65535
Default: 7
Serial Command: EI

Setting inEna to 0 disables position interpolation, setting to a non-zero value enables position interpolation.

intvl –Interpolation Interval

Variable Type: Integer
Valid Range: 10 to 65535
Default: 50
Serial Command: IL

This variable defines the interpolation period in units of 800 µs. The default value is 50 (40 milliseconds), which is equivalent to a 25 Hz position update rate.

rxTO – Interpolation Timeout Period

Variable Type: Integer
Valid Range: 0 to 65535
Default: 1250
Serial Command: NT

This variable defines the interpolation timeout period in increments of 800 µs.

TOact – Timeout Action

Variable Type: Integer
Valid Range: 0 to 65535
Default: 0
Serial Command: AR

This variable defines the behavior of the actuator after not receiving a valid CAN position command message within the configurable receive timeout period rxTO. A value of 0 causes the actuator to hold position and a value of 1 executes a trajectory move to defPos

txEna – Telemetry Enable Flag

Variable Type: Integer
Valid Range: 0 to 65535
Default: 7
Serial Command: ET

Setting txEna to 0 disables the telemetry being broadcast from the Servo Cylinder, setting to a non-zero value enables telemetry.

txIvl – Telemetry Interval

Variable Type: Integer
Valid Range: 0 to 65535
Default: 1000
Serial Command: IT

The telemetry interval defines the broadcast period of the telemetry message from the Servo Cylinder in milliseconds. Care must be taken to ensure the CAN bus is not overloaded by too rapid a transmission rate.

txData – Telemetry Data Selection

Variable Type: String
Valid Range: A-Z, c, m, p, q, r, s
Default: “KLMGHEFY”
Serial Command: DT

The specific telemetry data bytes to be broadcast are selected with this configuration variable. The string may have a length from 1 to 8 characters. The length of the txData string will determine how many data bytes are included in the telemetry message. Each character in txData designates one byte in the telemetry message.

txID – CAN Message ID

Variable Type: Integer
Valid Range: 0 to 0x1FFFFFFF
Default: 0x0000007F
Serial Command: NI

This variable is the 29-bit or 11-bit CAN 2.0B identifier for the telemetry message. This represents the destination address of the telemetry message. If CANext=0, only the lower 11 bits are used.

heatEna – Heater Enable Flag

Variable Type: Integer
Valid Range: 0 to 65535
Default: 7
Serial Command: EH

Setting heatEna to 0 disables the heating function, setting to a non-zero value enables the heating function.

heatTmp – Heater Temperature Threshold

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

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.

fErrWin – Following Error Window

Variable Type: Integer
Valid Range: 0 to 32767
Default: 512
Serial Command: FE

This variable defines the position error threshold in encoder counts that will trigger the “following error” status bit. This variable is only used to define the behavior of the related status bit.

fErrTime – Following Error Timeout Period

Variable Type: Integer
Valid Range: 0 to 10000
Default: 100
Serial Command: FT

This variable defines the position error threshold timeout in milliseconds that will trigger the following error status bit. The following error must exceed the fErrWin threshold for fErrTime milliseconds for the status bit to go high. This variable is only used to define the behavior of the related status bit.

posWin – Position At-Target Window

Variable Type: Integer
Valid Range: 0 to 32767
Default: 5
Serial Command: AT

This variable defines the window within the position setpoint that will be considered a completion of the position move and set the "position at target" status bit to high. This variable is only used to define the behavior of the related status bit

posTime – Position At-Target Time

Variable Type: Integer
Valid Range: 0 to 10000
Default: 10
Serial Command: AD

This variable defines the position at-target time in milliseconds that will trigger the position-at-target status bit. The position must be within the posWin window for posTime milliseconds for the status bit to go high. This variable is only used to define the behavior of the related status bit.

ovTorq – Over Torque Threshold

Valid Range: 0 to 32766
Default: 8200
Set Command: OT

Defines the torque threshold that causes the "Over Torque" status bit to go active (when Torque > ovTorq). This variable is only used to define the behavior of the related status bit.

stSpeed – Stopped Speed Threshold

Variable Type: Integer
Valid Range: 0 to 65534
Default 2
Set Command: ST

Defines the speed below which the "stopped" status bit is set high. This variable is only used to define the behavior of the related status bit.

ovTemp – Over Temperature Threshold

Variable Type: Float
Valid Range: -50.0 to 149.0
Default: 40.0
Set Command: LT

Defines the torque threshold that causes the "Over Temperature" status bit to go active (when Temp > ovTemp). This variable is only used to define the behavior of the related status bit. Units are in °C

unTemp – Under Temperature Threshold

Variable Type: Float
Valid Range: -50.0 to 149.0
Default: -40.0
Set Command: UT

Defines the torque threshold that causes the "Under Temperature" status bit to go active (when Temp < unTemp). This variable is only used to define the behavior of the related status bit. Units are in °C

ovHumi – Over Humidity Threshold

Variable Type: Float
Valid Range: 1.0 to 99.0
Default: 95.0
Set Command: LH

Defines the humidity threshold that causes the "Over Humidity" status bit to go active. This variable is only used to define the behavior of the related status bit.

Read Commands

RV – Read Configuration Variable(s)

Argument: See Tabled 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 table below.

Argument	Setting	Argument	Setting	Argument	Setting	Argument	Setting
0	rotor	**14**	Kp	**28**	pMax	**42**	heatCur
1	encRef	**15**	Ki	**29**	sigInv	**43**	fErrWin
2	rPos	**16**	Kd	**30**	SUact	**44**	fErrTime
3	ePos	**17**	spMin	**31**	defPos	**45**	posWin
4	sType	**18**	spMax	**32**	inEna	**46**	posTIme
5	qP	**19**	maxTorq	**33**	intvl	**47**	ovTorq
6	qI	**20**	maxSpeed	**34**	rxTO	**48**	stSpeed
7	dP	**21**	accel	**35**	TOact	**49**	ovTemp
8	dI	**22**	CANspd	**36**	txEna	**50**	unTemp
9	sNum	**23**	CANext	**37**	txlvl	**51**	ovHumid
10	pNum	**24**	unitID	**38**	txData		
11	opMode	**25**	IDmask	**39**	txID		
12	cliMode	**26**	rxData	**40**	heatEna		
13	cliBaud	**27**	pMin	**41**	heatTmp		

FS – Read Actuator Information

Argument: None

This command returns the firmware version currently running on the Servo Cylinder as well as the unique product serial number.

SR – Read Status Register

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.

BV – Read Bus Voltage

Argument: None

This command returns bus voltage, which is the DC power supply voltage used to power the actuator.

RT – Read Temperature at Controller Circuit Board

Argument: 0 to 3, or none

This command returns the temperature measured by the temperature sensors on the controller circuit board (not within the motor windings). 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.

Argument	Return
0	Temperature of sensor 1 in °F
1	Temperature of sensor 1 in °C
2	Temperature of sensor 2 in °F
3	Temperature of sensor 2 in °C

RH – Read Relative Humidity at Controller Circuit Board

Argument: 0 or 1, or none

This command returns the Relative Humidity (%RH) measured by a sensor on the controller circuit board. The A2 and AM series Servo Cylinder incorporate an o-ring sealed chassis and a desiccant block to adsorb any moisture that gets into the actuator via the dynamic shaft seal. The RH sensor can be used to detect if the desiccant block needs to be replaced or if there is a problem with the actuator's seal integrity.

Argument	Return
0, none	Relative Humidity (%RH)
1	Raw sensor value

QP – Read Position Setpoint

Argument None

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

AP – Read Actuator Position

Argument: None

This command returns the absolute position of the Servo Cylinder.

TQ – Read Motor Torque

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

Argument: None

This command returns the velocity of the actuator as measured by the change in position over a time interval. Negative values represent motion towards the retracted direction; positive values represent motion towards the extended position.

CI – Read CAN bus status info

Argument: None

Returns details on CAN bus communications

HI – Read Heater status info

Argument: None

Returns details on the motor heater function

RI – Read Information about the last CPU reset

Argument: None

Returns details about the cause of the last CPU reset

RL – Read total distance travelled over actuator life

Argument: None

Returns the total encoder counts travelled and total revolutions of the screw over the life of the actuator

EC – Read Startup File Error

Argument: None

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

ES –Read critical errors with configuration files on power-up

Argument: None

Returns any critical errors associated with the CONFIG.TXT and HARDWARE.TXT files

ZD – Read Bridge Driver DIAG Register

Argument: None

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

FR – Read File from Internal File System

Argument: "File Name"

Display the contents of a file in the internal file system.

Set Commands

OP – Read/Set opMode - Operating mode of the actuator

Argument: See Setting Description

This command sets the operating mode opMode of the Servo Cylinder. Issue this command without an argument to read the current value of the configuration variable.

IM – Read/Set ifMode – Serial Interface mode

Argument: See Setting Description

This command sets the configuration variable "ifMode – Serial Interface mode" to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable. Details on human and machine mode can be found in the Servo Cylinder manual UM711293

LN – Read/Set spMin - Software Position Minimum

Argument: See Setting Description

This command sets the configuration variable "spMin - Software Position Minimum" to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

LX – Read/Set spMax - Software Position Maximum

Argument: See Setting Description

This command sets the configuration variable "spMax - Software Position Maximum" 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 – Read/Set kp, ki, and kd – PID Gains

Argument: See Setting Description

These commands set the configuration variables "kp, ki, and kd – PID Gains" to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable.

MT – Read/Set maxTorq - Max Torque

Argument: See Setting Description

This command sets the configuration variable "maxTorq - Max Torque" to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable. Torque demand output of the PID algorithm will be capped at this value before being passed to the FOC layer.

SP – Read/Set maxSpeed - Maximum Profile Speed

Argument: See Setting Description

This command sets the configuration variable "maxSpeed - Maximum Speed" to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable. This variable sets the max speed at the top of a trapezoidal profile trajectory move.

AC – Read/Set accel - Profile Acceleration and Deceleration Rate

Argument: See Setting Description

This command sets the accel configuration variable. Issue this command without an argument to read the current value of the configuration variable. This variable sets the acceleration and deceleration of a trapezoidal profile trajectory move.

CS – Read/Set CANspd - CAN Bus Baud Rate

Argument: See Setting Description

This command sets the CANspd configuration variable. This variable selects the baud rate of the CAN bus. Issue this command without an argument to read the current value of the configuration variable.

CE – Read/Set CANext - CAN Bus Message Type

Argument: See Setting Description

This command sets the CANext configuration variable. This variable selects the CAN bus message type. 0 selects standard message type with an 11-bit identifier, 1 selects the extended message type with a 29-bit identifier. Issue this command without an argument to read the current value of the configuration variable.

ID – Read/Set unitID - CAN Bus Identifier

Argument: See Setting Description

This command sets the unitID configuration variable. This variable is used along with IDmask to filter incoming CAN messages. Issue this command without an argument to read the current value of the configuration variable.

MA – Read/Set IDmask - CAN Bus ID mask

Argument: See Setting Description

This command sets the IDmask configuration variable. This variable is used along with unitID to filter incoming CAN messages. Issue this command without an argument to read the current value of the configuration variable.

DR – Read/Set rxData - Actuator Command Message Data Format

Argument: See Setting Description

This command sets the rxData configuration variable. This variable defines the format for data bytes in the CAN position command message. Issue this command without an argument to read the current value of the configuration variable.

PN and PX – Read/Set pMin and pMax – Position Command Input Min and Max (Respectively)

Argument: See Setting Description

These commands set the configuration variable "pMin and pMax –position command input range Min and Max (Respectively)" to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable. These variables set the allowable range of the position command input values. They are also used to scale the input signal to the desired full travel of the actuator (spMin and spMax)

SI – Read/Set sigInv the invert input signal flag

Argument: See Setting Description

This command sets the sigInv configuration variable. This variable inverts the relationship between spMin/spMax and pMin/pMax. Issue this command without an argument to read the current value of the configuration variable.

AS – Read/Set SUact the startup action

Argument: See Setting Description

This command sets the startup behavior before the actuator receives a valid CAN message. Issue this command without an argument to read the current value of the configuration variable.

DP – Read/Set defPos the default position

Argument: See Setting Description

This command sets the default position (defPos) in Phase Index encoder counts. Issue this command without an argument to read the current value of the configuration variable.

NT – Read/Set rxTO the interpolation timeout period

Argument: See Setting Description

This command defines the received packet timeout period in increments of 800 µs. This value should be set to slightly less than two times the nominal message transmission rate. Issue this command without an argument to read the current value of the configuration variable. This variable is used to define the behavior of the related status bit and timeout action functionality. Default value is 125 (100 milliseconds)

AR – Read/Set TOact the timeout action

Argument: See Setting Description

This command sets the actuator behavior when a CAN message timeout occurs. Issue this command without an argument to read the current value of the configuration variable.

EI – Read/Set inEna the enable interpolation signal flag

Argument: See Setting Description

This command enables (intEna = NONZERO) or disables (intEna = 0) position interpolation. Issue this command without an argument to read the current value of the configuration variable.

IL – Read/Set intvl the interpolation period

Argument: See Setting Description

This command defines the interpolation period in increments of 800 µs. This value is set to 100 (80 milliseconds) by default, which is recommended for a 16 Hz position update rate.

ET – Read/Set txEna the Telemetry Enable flag

Argument: See Setting Description

This command sets the telemetry enable flag of the Servo Cylinder. Setting to a NONZERO value enables telemetry to be broadcast over the CAN bus. Issue this command without an argument to read the current value of the configuration variable.

IT – Read/Set txIvl the telemetry interval

Argument: See Setting Description

This command sets the telemetry broadcast interval in milliseconds. Issue this command without an argument to read the current value of the configuration variable.

DT – Read/Set txData the telemetry data to broadcast

Argument: See Setting Description

This command sets the txData configuration variable. This variable allows the user to configure each data byte of the telemetry message. The argument to this command is a string with a length of 1-8 characters where each character designates a data byte in the telemetry message. The number of data bytes in the telemetry message will be equal to the number of characters in the txData string. See the Servo Cylinder Telemetry section for details. Issue this command without an argument to read the current value of the configuration variable.

NI – Read/Set txId the telemetry message ID

Argument: See Setting Description

This command sets the telemetry message 29-bit or 11-bit CAN 2.0B identifier. Issue this command without an argument to read the current value of the configuration variable.

EH – Read/Set heatEna the heater enable flag

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.

SH – Read/Set heatTmp the heater threshold temperature

Argument: See Setting Description

This command sets the threshold temperature at which lower measured temperatures activates the heating functionality. Issue this command without an argument to read the current value of the configuration variable.

HC – Read/Set heatCurr the heater current value

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.

FE – Read/Set fErrWin the following error window

Argument: See Setting Description

This command defines the position error threshold in encoder counts that will trigger the following error status bit. Issue this command without an argument to read the current value of the configuration variable. This variable is only used to define the behavior of the related status bit.

FT – Read/Set fErrTime the following error timeout

Argument: See Setting Description

This command defines the position error threshold timeout in milliseconds that will trigger the following error status bit. The following error must exceed the fErrWin threshold for fErrTime milliseconds for the status bit to go high. Issue this command without an argument to read the current value of the configuration variable. This variable is only used to define the behavior of the related status bit.

AT – Read/Set posWin the at-target position window

Argument: See Setting Description

This command defines the window within the position setpoint that will be considered a completion of the position move and set the "position at target" status bit to high. Issue this command without an argument to read the current value of the configuration variable. This variable is only used to define the behavior of the related status bit.

AD – Read/Set posTime the at-target time

Argument: See Setting Description

This command defines the position at-target time in milliseconds that will trigger the position-at-target status bit. The position must be within the posWin window for posTime milliseconds for the status bit to go high. Issue this command without an argument to read the current value of the configuration variable. This variable is only used to define the behavior of the related status bit.

OT – Read/Set ovTorq – over torque threshold

Argument: See Setting Description

This command sets the configuration variable "ovTorq – Over Torque Threshold" to be equal to the provided argument. Issue this command without an argument to read the current value of the configuration variable. This variable is only used to define the behavior of the related status bit.

ST – Read/Set stSpeed the stopped speed threshold

Argument: See Setting Description

This command defines the measured speed at which the actuator is to be considered stopped. Issue this command without an argument to read the current value of the configuration variable. This variable is only used to define the behavior of the related status bit.

LT – Read/Set ovTemp the over-temperature threshold value

Argument: See Setting Description

This command defines the temperature threshold at which the over-temperature status bit will activate when the measured temperature is greater than the threshold value. Issue this command without an argument to read the current value of the configuration variable. This variable is only used to define the behavior of the related status bit.

UT – Read/Set unTemp the under-temperature threshold value

Argument: See Setting Description

This command defines the temperature threshold at which the under-temperature status bit will activate when the measured temperature is less than the threshold value. Issue this command without an argument to read the current value of the configuration variable. This variable is only used to define the behavior of the related status bit.

LH – Read/Set ovHumi the over-humidity threshold value

Argument: See Setting Description

This command defines the relative-humidity threshold at which the over-humidity status bit will activate when exceeded. Issue this command without an argument to read the current value of the configuration variable. This variable is only used to define the behavior of the related status bit.

Trajectory Commands

Trajectory commands are used to smoothly move the actuator to a desired position using the user configurable max speed and acceleration. The actuator must be in CLI operating mode (opMode = 0) to execute trajectory moves.

TA – Trajectory Move to Absolute Position

Argument: spMin to spMax

Sending this command result in a trajectory move to the requested position equal to the argument. Trajectory moves obey the maxSpeed (maximum profile speed) and accel (profile acceleration) settings.

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

AC2000
SP200000
TA25000

TO – Trajectory Move to Offset (Incremental Trajectory Move)

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)

Argument: None

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

TM – Trajectory Move to Midpoint

Argument: None

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

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

TE – Trajectory Move to Fully Extended Position (spMax)

Argument: None

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

TD – Trajectory Move to the default Position

Argument: None

Trajectory move to "defPos – Default Position" with user defined speed and acceleration

TK – Interrupt Current Trajectory Move

Argument: None

Halt the current trajectory motion before completion.

PC – Set PID Target to Current Position

Argument: None

Sets the current position setpoint to the current absolute position value

System Commands

ZC – Run Calibration Utility

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 can result in erroneous operation of the motor.

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

Argument: 321

Consult with Ultra Motion engineers before using this command. 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. After calibration, the HARDWARE.TXT and CONFIG.TXT files are written and the Servo Cylinder is reset. The first 5 configuration variables in HARDWARE.TXT are updated, and spMin, spMax, and defPos are updated in CONFIG.TXT. spMin and spMax are set to one revolution inside of rPos and ePos. defPos is set to halfway between spMin and spMax. USB must be disconnected before running this command.

ZU – Jump to Serial Firmware Update Utility

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 damage and must be disconnected from USB.

The CONFIG.TXT and HARDWARE.TXT files can be backed up to the Servo Cylinder's on-board flash storage with the "SB321" command. After the firmware update is complete, the backup can be restored with the "BR321" command. Do not issue the "SB" command while the actuator is moving or the motion may be interrupted.

ZR – Reboot Controller

Argument: 321

CPU warm reset. When restarting, the Servo Cylinder will draw all configuration settings from current configuration files (CONFIG.TXT and HARDWARE.TXT). This must be done to enact any changes to that were made within CONFIG.TXT.

CW – Write current settings to CONFIG.TXT

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. USB must be disconnected before running this command.

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

SB – Back up Internal File System

Argument: 321

Backs up the current file system (CONFIG.TXT and HARDWARE.TXT) to flash storage. USB must be disconnected before running this command.

WARNING: Do not execute the "SB" command while the actuator is moving or the motion may be interrupted.

BR – Restore File System from Backup

Argument: 321

Restores the file system from backup in flash storage. USB must be disconnected before running this command.

BC – Check for Existing Backup

Argument: 321

Check if a backup of the file system exists in the actuator's flash storage.

LK – Lock or Unlock the CLI

Argument: "lock" or "unlock"

The actuator will startup with a locked CLI and must be issued an unlock command to be used "lk unlock".

HE – Display Help Tables

Argument: None

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

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
LIVE CHAT	Live Chat directly with one of our engineers: ultramotion.com
EMAIL	Email (to be replied to within one business day): 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