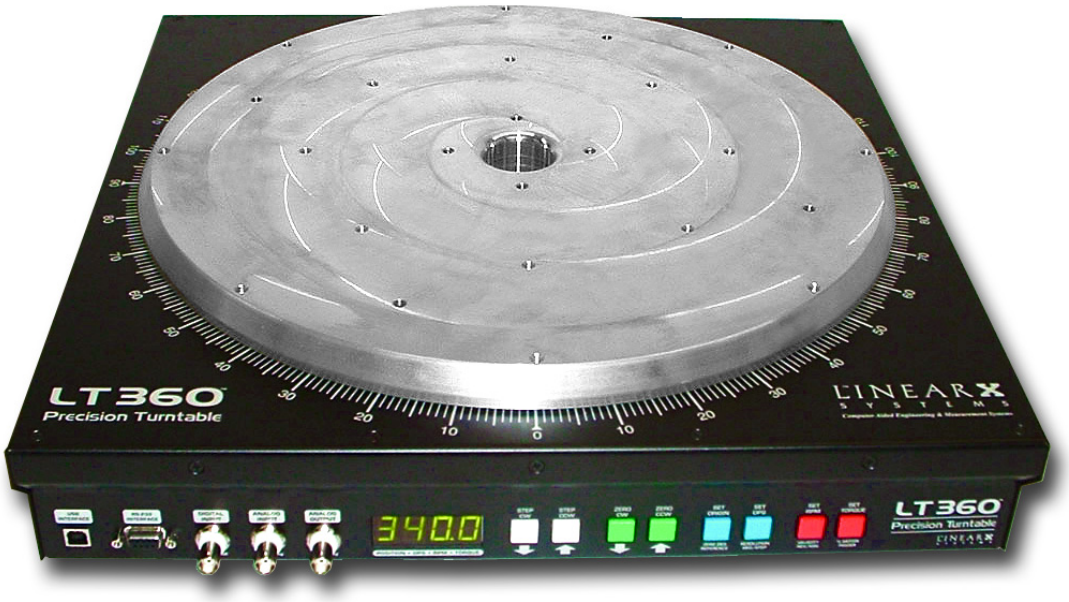
LT360™

Precision Turntable



RS-232 Programming Manual

LINEARX™
SYSTEMS

LT360 Precision Turntable
RS-232 Programming Manual

Tel: (503) 612-9565
Fax: (503) 612-9344

Printed in the United States of America.
April 15, 2011.

This document was produced on a Pentium-4 / 2GHz PC with Win2K using Adobe PageMaker 7.0, Adobe Illustrator 10.0, Adobe PhotoShop 7.0, MathType 4.0 for mathematics typography, and SnagIt 5.2 for screen captures. Final masters were produced using an Xerox Docutech image setter. Help files were composed and compiled using Windows Help Designer 3.1.

License Agreement and Limited Warranty

Carefully read all of the following terms and conditions of this agreement before opening and using the contents of this package. The opening of this package indicates your acceptance of the terms and conditions of this license agreement. If you are not willing to accept the terms and conditions of this agreement, then you should return the entire product, with the package seal unbroken, to the place of purchase for a full refund of the purchase price.

■ Copyright Ownership

Both the program and the documentation are protected under applicable copyright laws. LinearX is the holder of this copyright. Your right to use the program and the documentation are limited to the terms and conditions described herein. Use of the software unless pursuant to the terms and conditions of this license, or as otherwise authorized by law, is an infringement of the copyright.

■ Limited Non-Exclusive License

You may: (a) use the enclosed program on a single computer, (b) physically transfer the program from one computer to another provided that the program is used on only one computer at a time, and that you remove any copies of the program from the computer from which the program is being transferred, (c) make copies of the program solely for backup or archival purposes. You must reproduce and include the copyright notice and label any backup copy.

You may not: (a) distribute copies of the program or the documentation to others, (b) lease, rent, grant sublicenses, or other rights to the program, (c) provide use of the program in a computer service business, network, time-sharing multiple CPU or multiple users arrangement without the prior written consent of LinearX, (d) translate or otherwise alter the program or related documentation without the prior written consent of LinearX.

■ Terms

Your license to use the program and the documentation will automatically terminate if you fail to comply with the terms of this agreement. Your license terminates in the event that you receive a license for an updated version of the product that replaces this product. If a license expiration date is printed on your documentation, or provided through other means such as a time limited electronic or software key, your license expires on the day as shown in the documentation, or on the day that the electronic or software key expires. If this license is terminated you agree to destroy all copies of the program and documentation.

■ Limited Warranty

LinearX warrants to the original licensee that the disk(s) and or electronic key(s) on which the program is recorded will be free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of purchase as evidenced by a copy of your receipt. If failure of the product components has resulted from accident, abuse, or misapplication of the product, then LinearX or third party licensors shall have no responsibility to replace the disk(s) or key(s) under this limited warranty.

This limited warranty and right of replacement is in lieu of, and you hereby waive, any and all other warranties, both expressed and implied, including but not limited to warranties of merchantability and fitness for a particular purpose. The liability of LinearX or third party licensors pursuant to this limited warranty shall be limited to the replacement of the defective disk(s) or key(s), and in no event shall LinearX or third party licensors be liable for incidental, indirect, punitive, or consequential damages, including but not limited to loss of use, loss of profits, loss of data or data being rendered inaccurate, or losses sustained by third parties even if LinearX or third party licensors have been advised of the possibility of such damages. This warranty gives you specific legal rights which may vary from state to state. Some states do not allow the limitation or exclusion of liability for consequential damages, so the above limitation may not apply to you.

In addition to the foregoing, you should recognize that all complex software systems and their documentation contain errors and omissions. LinearX, its distributors, and dealers shall not be responsible under any circumstances for providing information on or corrections to errors and omissions discovered at any time in the product, whether or not they are aware of the errors or omissions. LinearX does not recommend the use of this product in applications in which errors or omissions could result in loss of life, injury, or other significant loss.

This license agreement shall be governed by the laws of the state of Oregon and shall inure to the benefit of LinearX, its successors, administrators, heirs and assigns or third party licensors.

■ United States Federal Government Restrictions

If this software is acquired by or on behalf of the U.S. Federal government or its agencies, this provision applies. Use, duplication, or disclosure of this software is subject to restrictions set forth in the appropriate FAR 52.227-19 and DFAR 252.227-7013 documents, as applicable. The software is "commercial computer software" and is licensed only with "Restricted Rights". Other Federal restrictions may also apply.

LinearX Systems Inc.
9500 SW Tualatin-Sherwood Rd.
Tualatin, OR 97062-8586 USA

TEL:(503) 612-9565 FAX:(503) 612-9344 WEB: www.linearx.com

Technical Support

LinearX provides detailed printed manuals and on-line help within the program as the primary source for user information and assistance regarding the use of this product. If these sources do not contain the answers to your questions, contact LinearX via any of the following methods:

Internet Forums: **www.linearx.com/forums**
Internet Email: **support@linearx.com**
Internet Web: **www.linearx.com**
Fax: **(503) 612-9344**
Tel: **(503) 612-9565**

Technical support is free and unlimited at this time, however we reserve the right to charge for this service in the future as conditions, overhead, and support personnel requirements dictate.

When contacting us regarding a technical support issue, PLEASE follow these steps to aid us in understanding and solving your problem:

(1) If your question involves specific details or parameters unique to your project and problem, please include a copy of your design files with the necessary data so that we can reproduce your problem. This is only possible if you are communicating via an electronic means such as Email or uploading files directly to our web site.

(2) If the issue regards error messages from the program, please include an exact description of the error message and/or address information that the program reports.

(3) If there are specific steps involved to reproduce the issue, please note these exact steps required so that we can reproduce the problem.

Note: Technical support *does not* include programming assistance. It is assumed that the reader has sufficient experience and knowledge to incorporate the RS-232 interface into their own application.

Technical support hours are: Monday-Friday 9:00AM to 5:00PM Pacific Standard Time.

Contents

Contents

1.1 Overview of RS-232

The RS-232 industry standard serial interface provides a universal means to control the LT360 by any program in any operating system environment. RS-232 still has many advantages over the newer bus formats such as USB. The length of a cable run for RS-232 is not restricted, and the commands can be sent to the LT360 from any program, device, or operating system.

Note: If you are using the Windows OS, then the Win32 LT360LIB.dll provides an easier means for custom programming support. Please refer to the DLL Programming Manual for more details.

1.2 Electrical Connection & Wiring

The LT360 provides a female DB-9 connector on the front panel for connection to the RS-232 serial host. If your host/computer has a DB-25, you will need to utilize a DB-25F to DB-9M adapter or cable.

The LT360 is configured as a true DCE (Data Communications Equipment) device and therefore only requires a straight-through cable connection. A null modem adapter/cable is *not* required and should not be used.

Within the DB-9, only 2 signals are used besides Ground: *RX* and *TX*. None of the handshaking control lines are used. The LT360 serial interface does not require either hardware or software handshaking.

1.3 Communication Parameters

The LT360 is configured with the following serial parameters:

- Baud Rate 9600 (factory default)
- Data Bits 8
- Parity None
- Stop Bits 1
- Handshaking None

The user can change the baud rate to higher values as desired. The maximum baud rate is 57600. The following are some typical baud rate values:

9600
14400
19200
28800
38400
57600

1.4 Termination Characters

Each and every command string that is sent to the LT360 must be terminated with a special character. Either with a Carriage Return Chr(13) or Null Chr(0). The LT360 accepts either character for string termination.

Examples are:

- Goto CCW -45.0 [13]
- Set DisplayPolarity UNIPOLAR[0]

The termination character tells the LT360 that the command string is complete and that it can now be processed. After the command string is processed, it is cleared and the command buffer is empty.

The LT360 employs a time-out of 10 seconds after the reception of the first character. If a termination character is not received within 10 seconds, then an error message is displayed on the front panel and the command buffer/string is cleared. This prevents corrupt data from building up in the command buffer.

Therefore, command strings must be sent to the LT360 complete within 10 seconds. This is of little concern when the data is being sent from a computer program which will send the data at line speed. However, if you are typing commands manually at a terminal or communication program, you must type in the command string within 10 seconds along with the [CR].

1.5 Command Buffer & Timing Issues

The LT360 contains a single command buffer. Each command must be processed by the LT360 before another command is sent by the host. Otherwise a command overflow would occur resulting in a corrupted command.

For this reason the LT360 *always* returns a string from *all* commands. The host must wait for this response string before sending another command. This command/response sequence automatically ensures that every command is processed before another is sent.

There are two classes of commands: *Set* commands and *Get* commands. The *Set* class of commands also includes the *Step* and *Goto* commands and these only send information to the LT360. However, the LT360 returns an '*Ok*' acknowledgement response string to all Set commands.

The *Get* commands cause the LT360 to return actual data strings to the host.

All commands are self timing due to the fact that the host must wait for data or 'Ok' strings to be returned from the LT360. Generally the host cannot send another command until it has received the last response from the previous command, and this means that the LT360 has completed processing the previous command.

1.6 Example RS-232 Commands

All commands sent to the LT360 can be either upper or lower case. Numeric parameters when needed for Set commands are contained in the string at the end, and then followed by a termination character.

There must be a [space] delimiter between the Set or Get word and the command name, and again between the parameter if used. The final termination character must follow to inform the LT360 that the command string is now complete and can be processed.

The following are some LT360 command examples, where [CR]=Chr(13) and [NULL]=Chr(0) are equivalent. All strings returned by the LT360 have a [NULL]=Chr(0) as the last character.

■ **Goto CCW -45.0[CR]** **(LT360 returns: *Ok[0]*)**
Move the platter counter clockwise to the -45.0 (315.0) degree position.

■ **Step CCW[CR]** **(LT360 returns: *Ok[0]*)**
Step the platter counter clockwise by the current step size.

■ **Set InputPolarity BIPOLAR[NULL]** **(LT360 returns: *Ok[0]*)**
Set the Analog Input for bipolar voltage input mode: 0..±1.800

■ **Get StepSize[CR]** **(LT360 returns: *5.00[0]*)**
Get the current step size, returned as an ASCIIZ string such as *5.00* in degrees.

■ **Get PulseInput[NULL]** **(LT360 returns: *OFF[0]*)**
Get the current status for the Pulse Input, returned as an ASCIIZ string either *OFF* or *ON*.

1.7 Error Messages

The LT360 can display a variety of error codes on the front panel display. The error message will be displayed over a 4 second interval with a format such as *Err0* or *Er13*.

The following list describes some of the error codes which can appear:

```
//-------------------------------------------------
// Show Error Display Codes
//-------------------------------------------------
//       0      -       Motor Home Position Error
//       1      -       RS232-OE Overrun Error
//       2      -       RS232-PE Parity Error
//       3      -       RS232-FE Framing Error
//       4      -       RS232-BI Break Int Error
//       5      -       RS232 Unknown Command
//       6      -       RS232 Invalid Parameter
//       7      -       RS232 Invalid Return
//       8      -       USB   Unknown Command
//       9      -       RS232 Command Timeout
//      10      -
//      11      -       Display Cntrlr Adr Failure
//      12      -       Switch Cntrlr Adr Failure
//      13      -       Torque Cntrlr Adr Failure
//      14      -       Output Cntrlr Adr Failure
//      15      -
//      16      -
//      17      -
//      18      -
//      19      -
//      20      -
//-------------------------------------------------
```

Most errors produced by the LT360 will be RS-232 command or parameter errors, especially if the user is programming via RS-232. The USB processing routines also rely on the RS-232 command decoders, so it is very possible to see RS-232 error codes while using the USB interface.

The basic LT360 serial RS-232 communication parameters are: 9600 baud, 8 data bits, no parity, 1 stop bit, and no flow control.

Error - 0 : Motor Home Position Error

The stepper motor produces a specific number steps for a given amount of platter rotation. The LT360 utilizes microstepping for additional precision control. One particular phase of the motor is designated as the *Home* position. For any given platter rotation, the motor should always end in its Home position. This is checked by the firmware continuously, if the *MotorHomeChk* option is enabled. If the motor position is not in the Home phase at the end of a movement, this error will be shown.

In normal operation this error should never occur. If it does, the unit must be powered off, or the *MotorHomeChk* option disabled, to clear the error. It is possible for this error to be produced by noise in the stepper motor controller, or by the load exceeding the torque capability thereby causing the motor to skip steps.

If this error is shown repeatedly, either there is a noise problem internal in the LT360 or the torque load is too great for the drive system.

Error - 1 : RS232-OE Overrun Error

This error indicates problems with the serial port communications. It can be caused by differences in the communication parameters between the computer and the LT360. Make sure that the computer serial port setup is 9600 baud, 8 data bits, no parity, and 1 stop bit. If the LT360 baud rate has been changed, then the PC must be set the same.

Error - 2 : RS232-PE Parity Error

This error indicates problems with the serial port communications. It can be caused by differences in the communication parameters between the computer and the LT360. Since the LT360 does not use parity, this error is unlikely. Make sure that the computer serial port setup is 9600 baud, 8 data bits, no parity, and 1 stop bit. If the LT360 baud rate has been changed, then the PC must be set the same.

Error - 3 : RS232-FE Framing Error

This error indicates problems with the serial port communications. It can be caused by differences in the communication parameters between the computer and the LT360, or invalid parameters in the computer UART of the computer. Make sure that the computer serial port setup is 9600 baud, 8 data bits, no parity, and 1 stop bit. If the LT360 baud rate has been changed, then the PC must be set the same. You may also need to reboot the computer to clear and reinitialize the UART in the computer.

Error - 4 : RS232-BI Break Int Error

This error indicates problems with the serial port communications. It can be caused by differences in the communication parameters between the computer and the LT360. Make sure that the computer serial port setup is 9600 baud, 8 data bits, no parity, and 1 stop bit. If the LT360 baud rate has been changed, then the PC must be set the same.

Error - 5 : RS232 Unknown Command

This error indicates that the character string sent to the LT360 did not contain a valid command. This error will generally occur if you are writing your own RS-232 program, and there are bugs in your program. Check the command strings being sent to verify that they contain valid LT360 commands. Case is not important, but spaces in the right or wrong places can make a difference.

Error - 6 : RS232 Invalid Parameter

This error indicates that the LT360 was expecting a parameter value for this command, but did not find one in the command string. Some commands need parameters and some do not. This error will generally occur if you are writing your own RS-232 program, and there are bugs in your program. Check the command strings being sent to verify that they contain valid LT360 commands and parameters. Case is not important, but spaces in the right or wrong places can make a difference.

Error - 7 : RS232 Invalid Return

This error indicates that the LT360 was expecting to produce a return value for a Get command, but the return value was an empty string. This error will generally occur if you are writing your own RS-232 program, and there are bugs in your program. Check the command strings being sent to verify that they contain valid LT360 commands and/or parameters. Case is not important, but spaces in the right or wrong places can make a difference.

Error - 8 : USB Unknown Command

This error indicates that the LT360 did not understand the command index sent via USB. This error is very uncommon. It probably indicates faulty communication over USB, such as a cable being disconnected or dirty connections. Possibly some other unusual problem.

Error - 9 : RS232 Command Timeout

This error indicates that the LT360 received some characters, but never received a termination character [Null(0) or CR(13)] before the 10 second timeout expired. When the first character is sent to the LT360, a timer is started. If a termination character is not received within 10 seconds, then this error is produced and the command buffer is cleared. Receiving all of the characters for a command string should take very little time, on the order of micro or milli seconds, so if a termination charcater does not arrive within 10 seconds, something is wrong. The LT360 uses this timeout to clear the command buffer every 10 seconds if characters are received with no termination character. The termination character is mandatory to notify the LT360 that the entire command string has been sent and it can now process it.

Error - 11: Display Controller Address Failure

This error indicates that the CPU in the LT360 cannot communicate with the display controller. This error should never occur. If it does, a failure of a component in the LT360 is indicated.

Error - 12: Switch Controller Address Failure

This error indicates that the CPU in the LT360 cannot communicate with the switch controller. This error should never occur. If it does, a failure of a component in the LT360 is indicated.

Error - 13: Torque Controller Address Failure

This error indicates that the CPU in the LT360 cannot communicate with the stepper motor torque DAC controller. This error should never occur. If it does, a failure of a component in the LT360 is indicated.

Error - 14: Output Controller Address Failure

This error indicates that the CPU in the LT360 cannot communicate with the analog output DAC controller. This error should never occur. If it does, a failure of a component in the LT360 is indicated.

2.1 Goto CCW

■ Class	Set (out)
■ RS-232 String	GoTo CCW ±*nnn.n* [CR]
■ Parameter	Position (degrees)
■ Returns	*Ok[0]*

Description
This command causes the LT360 to move to the desired location specified by the parameter value using counter clockwise rotation. If the LT360 is already at this location, no movement occurs.

The parameter value should have tenths of a degree precision, and may be either unipolar (0..+360.0) or bipolar (0..±180.0).

The Goto 0 command can also be issued from the front panel of the unit, or the LR360 remote using the switches on the front panel.

2.2 Goto CW

■ Class	Set (out)
■ RS-232 String	GoTo CW ±*nnn.n* [CR]
■ Parameter	Position (degrees)
■ Returns	*Ok[0]*

Description
This command causes the LT360 to move to the desired location specified by the parameter value using clockwise rotation. If the LT360 is already at this location, no movement occurs.

The parameter value should have tenths of a degree precision, and may be either unipolar (0..+360.0) or bipolar (0..±180.0).

The Goto 0 command can also be issued from the front panel of the unit, or the LR360 remote using the switches on the front panel.

2.3 Step CCW

■ Class	Set (out)
■ RS-232 String	Step CCW [CR]
■ Parameter	(none)
■ Returns	*Ok[0]*

Description
This command causes the LT360 to move from its current position by the current step size using counter clockwise rotation. No parameter is required.

The step command can also be issued from the front panel of the unit, the LR360 remote, or by the TTL *Pulse* input on the front panel.

2.4 Step CW

■ Class	Set (out)
■ RS-232 String	Step CW [CR]
■ Parameter	(none)
■ Returns	*Ok[0]*

Description
This command causes the LT360 to move from its current position by the current step size using clockwise rotation. No parameter is required.

The step command can also be issued from the front panel of the unit, the LR360 remote, or by the TTL *Pulse* input on the front panel.

The parameter value should have tenths of a degree precision, and may be either unipolar (0..360.0) or bipolar (0..±180.0).

2.5 Set Smart Torque

■ Class	Set (out)
■ RS-232 String	Set SmartTorque *xxx* [CR]
■ Parameter	'OFF' or 'ON'
■ Returns	*Ok[0]*

Description
This command controls the behavior of the stepper motor. When *SmartTorque* is set ON, the motor is powered down to half the driving power 2 seconds after each movement stops. Motor power is returned to full (as controlled by the Maximum Torque setting) when the motor starts again.

This feature takes advantage of the fact that the worm gear drive system employed by the LT360 is irreversible. Meaning, the platter is self-locking at its current position. The load cannot rotate the platter itself. Therefore, it is not necessary to maintain full power on the motor merely to hold its current position when the platter is not in movement.

The use of *SmartTorque* greatly reduces the thermal heating in the motor and driver circuitry, and produces a very efficient drive system. Under most conditions and use, this feature should always be kept ON.

If a particular application demands that the motor power be kept constant at all times, even when stationary (SmartTorque=OFF), then the maximum torque should be to 70% or less to prevent excessive heating. It is difficult to envision what kinds of applications would require this behavior, but the option is available if needed.

2.6 Set Baud Rate

■ Class	Set (out)
■ RS-232 String	Set BaudRate *nnnnn* [CR]
■ Parameter	Baud rate, eg. '9600'
■ Returns	*Ok[0]*

Description
This command sets the baud rate for RS-232 communications. Once the new baud rate is set, all further RS-232 communication with an LT360 must be at the new baud rate.

Legal baud rate values are:

9600
14400
19200
28800
38400
57600

Note: While increasing the baud rate will provide faster communication, it will not necessarily produce a direct ratio effect on the speed of communication and commands with the LT360. The LT360 has a response time for commands which is dependent on many factors internally. The CPU must perform many other tasks so there can be a latency of several mSec before the commands are processed regardless of the baudrate.

2.7 Set Origin

■ Class	Set (out)
■ RS-232 String	Set Origin [CR]
■ Parameter	(none)
■ Returns	*Ok[0]*

Description
This command resets the rotational origin to that of the current platter position. No parameters are required, and the platter does not rotate. All further goto commands will be referenced to this new origin (0.0).

2.8 Set Pulse Direction

■ Class	Set (out)
■ RS-232 String	Set PulseDir *xxx* [CR]
■ Parameter	'CCW' or 'CW'
■ Returns	*Ok[0]*

Description
This command controls the behavior to a TTL step pulse at the *Pulse* input BNC connector. The parameter string can be either CCW or CW, meaning counter clockwise or clockwise.

When a pulse arrives at the *Pulse* BNC input, the platter will rotate either CCW or CW as defined by this setting. The size of the step is controlled by the *Set StepSize* command.

Note: The TTL pulse should be a minimum of 10uS, or longer. The triggering edge can be controlled by the Set PulseEdge command.

2.9 Set Pulse Edge

■ Class	Set (out)
■ RS-232 String	Set PulseEdge *xxxx* [CR]
■ Parameter	'RISE' or 'FALL'
■ Returns	*Ok[0]*

Description
This command controls the behavior to a TTL step pulse at the *Pulse* input BNC connector. The parameter string can be either RISE or FALL, meaning triggering occurs on the rising or falling edge of the pulse.

When a pulse arrives at the *Pulse* BNC input, the platter will rotate either at the rising or falling edge as defined by this setting. The size of the step is controlled by the *Set StepSize* command.

Note: The TTL pulse should be a minimum of 10uS, or longer. The direction can be controlled by the Set PulseDir command.

2.10 Set Step Size

■ Class	Set (out)
■ RS-232 String	Set StepSize *nn.n* [CR]
■ Parameter	Step Size (degrees)
■ Returns	*Ok[0]*

Description
This command sets the angular step size by which the platter will rotate to either a step command or TTL pulse. The parameter string must contain a numeric value such as '15.0' with tenth of a degree precision.

The minimum step size is 0.1 degrees. Step size is always positive. Direction is controlled by the CCW/CW commands listed else where.

2.11 Set Velocity

■ Class	Set (out)
■ RS-232 String	Set Velocity *n.nn* [CR]
■ Parameter	Velocity (RPM)
■ Returns	*Ok[0]*

Description
This command sets the angular velocity by which the platter will rotate. The value is set in RPM (revolutions per minute). The allowable range for the parameter is 0.01 to 3.00 RPM.

The parameter string must contain a numeric value such as '1.00' with hundreths precision.

2.12 Set Torque

■ Class	Set (out)
■ RS-232 String	Set Torque *nnn.n* [CR]
■ Parameter	Torque (%)
■ Returns	*Ok[0]*

Description
This command sets the stepper motor torque for the platter rotation. The value is set in percent of maximum torque. The allowable range for the parameter is 10.0 to 100.0 %.

The parameter string must contain a numeric value such as '100.0' with tenths of a degree precision.

For most typical applications your will want the torque set to 100%. If you intend to have the platter in constant motion, than you may wish to set the torque to a lower value such as 70%. This will reduce heating in the motor and drive circuitry.

The *SmartTorque* feature of the LT360 automatically reduces the motor torque to 1/2 its normal rotational power when it is not moving. Thus providing the advantages of maximum torque when in motion and reduced power consumption when stationary.

2.13 Set Acceleration Function

■ Class	Set (out)
■ RS-232 String	Set AccelFunc *n* [CR]
■ Parameter	Function Number 0..4
■ Returns	*Ok[0]*

Description
This command sets the acceleration function which the LT360 will use during step or goto commands. The acceleration profile is an important feature of the LT360 which enables heavy loads to be moved with minimum disturbance.

The parameter string must contain a numeric value such as '1' with allowable values from 0..4. There are five acceleration functions numbered 0,1,2,3,4. The names of these profiles are:

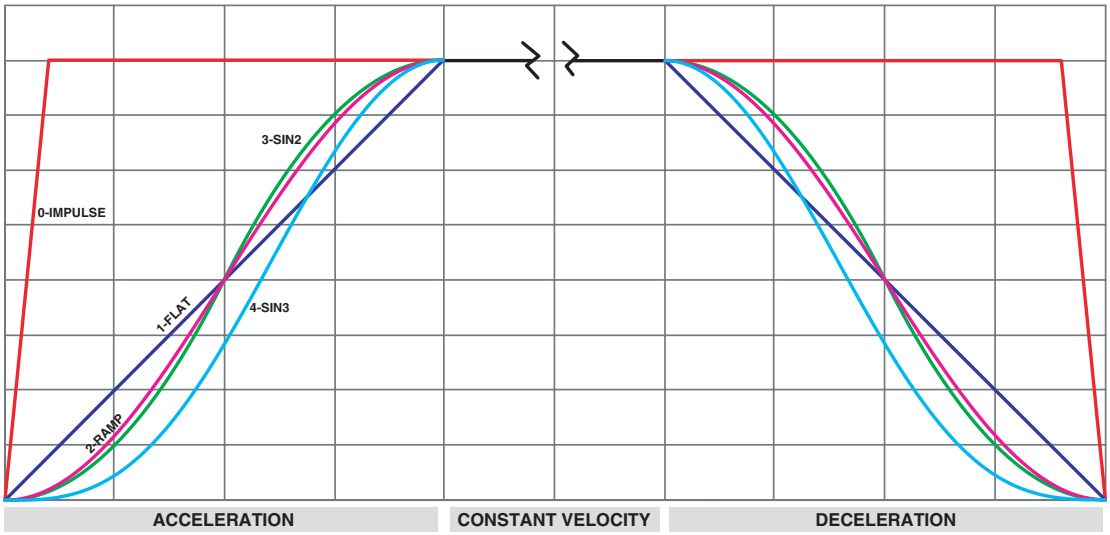
0 - Impulse
1 - Flat
2 - Ramp
3 - Sin2
4 - Sin3

For most typical applications acceleration function 1 - FLAT will provide excellent performance. Because the LT360 controls the acceleration and deceleration so precisely, very heavy loads can be rotated with ease.

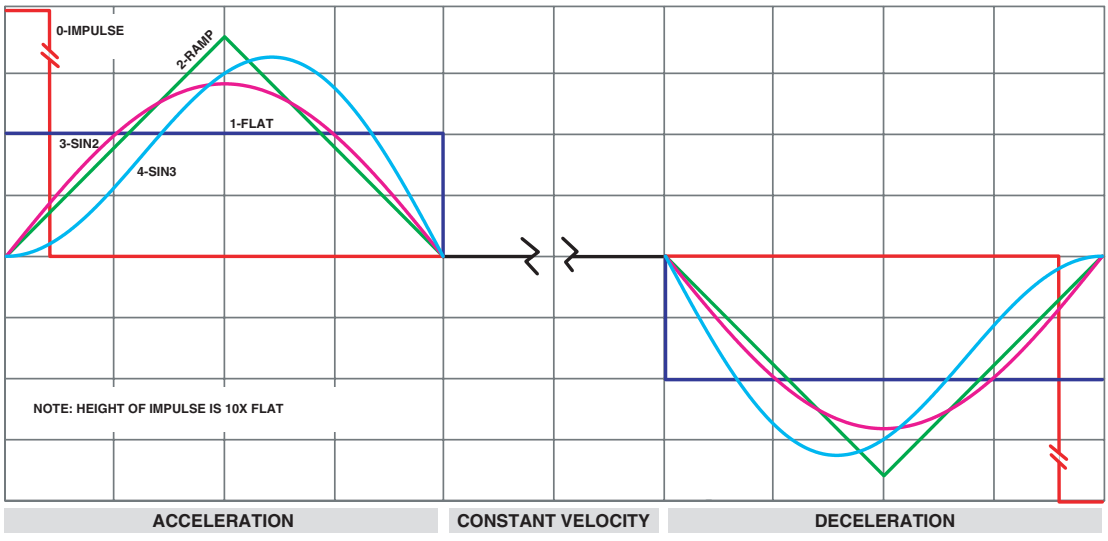
The 0 - IMPULSE function provides the fastest rise time from zero to constant velocity. However it also greatly reduces the maximum load which the LT360 can rotate, due to its higher acceleration. The acceleration of this function is about 10X that of the others.

The graphs on the following page show the velocity and acceleration profile curves for the five different LT360 functions.

LT360 VELOCITY PROFILES



LT360 ACCELERATION PROFILES



2.14 Set Name

■ Class	Set (out)
■ RS-232 String	Set Name *xxxxxxxxxx* [CR]
■ Parameter	Name
■ Returns	*Ok[0]*

Description
This command sets the Name for the LT360 unit. This is an arbitrary user name which can be assigned to the unit. This can be useful for applications which utilize multiple units to distinguish between the units, for example Horz, Vert, etc. This name is stored in the LT360 and is non volatile.

The Name parameter string has a maximum length of 21 characters.

2.15 Set Pulse Input

■ Class	Set (out)
■ RS-232 String	Set PulseInput *xxx* [CR]
■ Parameter	'OFF' or 'ON'
■ Returns	*Ok[0]*

Description
This command enables the TTL Pulse Input feature. When *PulseInput* is set ON, the LT360 will respond to TTL pulses at the BNC connector. When it is set OFF, it will ignore any pulses.

Each pulse will trigger a step. The width of the pulse should be at least 10uSec, and the triggering can be set on the rising or falling edge. Any pulses which occur while the LT360 is moving will be ignored.

If you are not using the TTL pulse input, it is probably best to keep this feature OFF to prevent false triggering due to any noise pickup at the open connector.

2.16 Set Analog Input

■ Class	Set (out)
■ RS-232 String	Set AnalogInput *xxx* [CR]
■ Parameter	'OFF' or 'ON'
■ Returns	*Ok[0]*

Description
This command enables the Analog Input feature. When *AnalogInput* is set ON, the LT360 will respond to the DC voltage at the analog input BNC connector. When it is set OFF, it will ignore that input.

The LT360 has a scale factor of 10mV/deg for the analog input. The range can be either unipolar (0 ... +3.600V) or bipolar (0 ... ±1.800V). The input impedance of the analog input is 1M Ohm. Although the input is an unbalanced BNC connector, the actually circuitry is differential. This helps to reject ground noise.

The input source must be very stable and of low noise. The LT360 has a resolution of 0.1 degrees and therefore responds to changes of 1mV at the analog input. When this input is enabled, the LT360 will sample the input and move to the location as indicated by the voltage. After that move is completed, it then samples the analog input again and moves to the new position. Monitoring of the analog input is real time.

If you are not using the analog input, this input *must* be set OFF. Otherwise the LT360 will be controlled by any noise received at the input.

2.17 Set Display Polarity

■ Class	Set (out)
■ RS-232 String	Set DisplayPolarity *xxxxxxxx* [CR]
■ Parameter	'UNIPOLAR' or 'BIPOLAR'
■ Returns	*Ok[0]*

Description
This command changes how the LT360 represents the degree position on the front display of the chassis, and in the Win32 software. When *DisplayPolarity* is set to UNIPOLAR, the LT360 will display the position in the range of 0 to 360 degrees. All position values are positive. When it is set to BIPOLAR, the range will be 0 to ±180 degrees. Both positive and negative values will be shown.

2.18 Set Input Polarity

■ Class	Set (out)
■ RS-232 String	Set InputPolarity *xxxxxxxx* [CR]
■ Parameter	'UNIPOLAR' or 'BIPOLAR'
■ Returns	*Ok[0]*

Description
This command changes how the LT360 interprets the DC voltage received at the Analog Input connector. When *InputPolarity* is set to UNIPOLAR, the LT360 will accept voltages in the range of 0.000 to +3.600 volts. All position values are positive. When it is set to BIPOLAR, the range will be 0.000 to ±1.800 volts. Both positive and negative values are used.

Voltages higher than the allowable range for the selected mode are invalid.

The LT360 has a scale factor of 10mV/deg for the analog input. The range can be either unipolar (0 ... +3.600V) or bipolar (0 ... ±1.800V). The input impedance of the analog input is 1M Ohm. Although the input is an unbalanced BNC connector, the actually circuitry is differential. This helps to reject ground noise.

The input source must be very stable and of low noise. The LT360 has a resolution of 0.1 degrees and therefore responds to changes of 1mV at the analog input. When this input is enabled, the LT360 will sample the input and move to the location as indicated by the voltage. After that move is completed, it then samples the analog input again and moves to the new position. Monitoring of the analog input is real time.

If you are not using the analog input, this input *must* be set OFF. Otherwise the LT360 will be controlled by any noise received at the input.

2.19 Set Output Polarity

■ Class	Set (out)
■ RS-232 String	Set OutputPolarity *xxxxxxxx* [CR]
■ Parameter	'UNIPOLAR' or 'BIPOLAR'
■ Returns	*Ok[0]*

Description
This command changes how the LT360 generates the DC voltage output at the Analog Output connector. When *OutputPolarity* is set to UNIPOLAR, the LT360 will produce voltages in the range of 0.000 to +3.600 volts. All position values are positive. When it is set to BIPOLAR, the range will be 0.000 to ±1.800 volts. Both positive and negative values are produced.

The LT360 has a scale factor of 10mV/deg for the analog output. The range can be either unipolar (0 ... +3.600V) or bipolar (0 ... ±1.800V). The output impedance of the analog output is <200 Ohms. Although the output is an unbalanced BNC connector, it is highly recommended that the input on the other device be differential to reject ground noise. Since the smallest resolution of the LT360 is 0.1 degree, the equivalent resolution in the output is 1mV. The input impedance of the other device should be at least 100K Ohm to prevent loading errors.

2.20 Set Move Abort

■ Class	Set (out)
■ RS-232 String	Set MoveAbort [CR]
■ Parameter	(none)
■ Returns	*Ok[0]*

Description
This command can be used to terminate the rotation in progress. The LT360 will stop rotating as quickly as possible at any current location. This command can be used as an emergency stop feature if needed.

2.21 Set Motor Home Check

■ Class	Set (out)
■ RS-232 String	Set MotorHomeChk *xxx* [CR]
■ Parameter	'OFF' or 'ON'
■ Returns	*Ok[0]*

Description
This command controls checking of the stepper motor synchronization . When *MotorHomeChk* is set ON, the firmware checks that the motor returns to its home position after each movement. If disabled, no checking is performed. If checking is enabled and the motor does not end its operation in the home position, an error message is shown on the front panel display.

The stepper motor produces a specific number steps for a given amount of platter rotation. The LT360 utilizes microstepping for additional precision control. One particular phase of the motor is designated as the *Home* position. For any given platter rotation, the motor should always end in its Home position.

In normal operation this error should never occur. If it does, the unit must be powered off to clear the error, or this feature disabled. It is possible for this error to be produced by noise in the stepper motor controller, or by the load exceeding the torque capability thereby causing the motor to skip steps.

If this error is shown repeatedly, either there is a problem internal in the LT360 or the torque load is too great for the drive system.

2.22 Set Output Mode

■ Class	Set (out)
■ RS-232 String	Set OutputMode *xxxxx* [CR]
■ Parameter	*CONT* or *START* or *STOP*
■ Returns	*Ok[0]*

Description
This command selects the operating mode for the Analog Output. The parameter string value is either CONT or START or STOP.

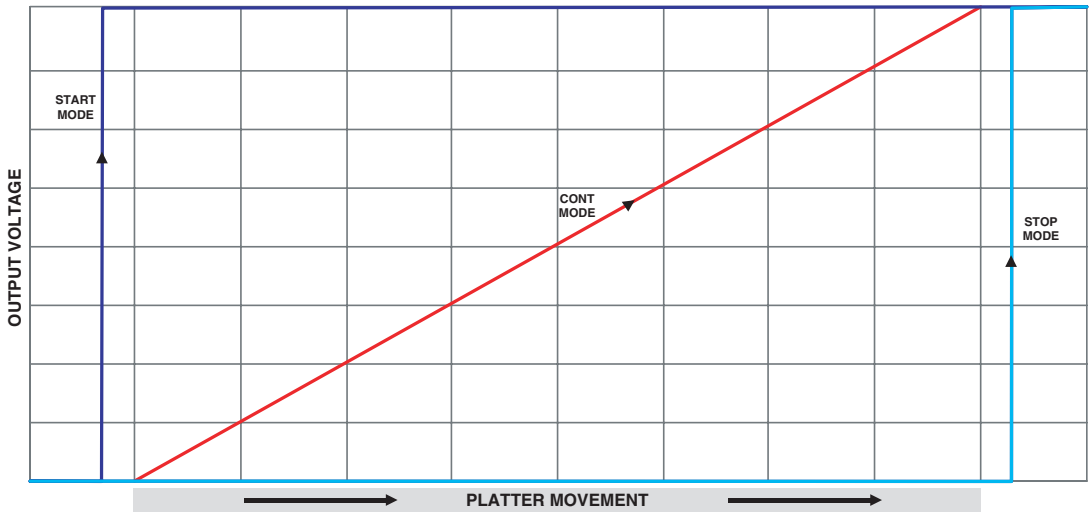
When *CONT* is the mode, the Analog Output will continuously follow the platter position as it is moving. This is a real time output mode where the analog output voltage always represents the current platter position.

When *START* is the mode, the Analog Output will only produce a single step change output voltage representing the final destination position at the start of the movement. For example, if the current position is 0.0 degrees and the destination is 100.0 degrees, the Analog Output voltage will change from 0.000V to 1.000V at the start of the movement.

When *STOP* is the mode, the Analog Output will only produce a single step change output voltage representing the final destination position after the movement stops. For example, if the current position is 0.0 degrees and the destination is 100.0 degrees, the Analog Output voltage will remain at 0.000V during the movement and then change to 1.000V when the movement stops. The final output voltage changes 2 seconds after the platter movemet stops.

If the movement is aborted before it reaches the destination, the output voltage will be updated to the correct final value in all modes.

LT360 ANALOG OUTPUT MODES



2.23 Set Revolution

■ Class	Set (out)
■ RS-232 String	Set Revolution *nnn* [CR]
■ Parameter	Revolution (count)
■ Returns	*Ok[0]*

Description
This command sets the internal Revolution Counter value. The integer parameter string value is an integer (eg. -1, 0, 1 etc).

This command is typically used to set the Revolution count to Zero, but it can be any positive or negative integer value as well.

The Revolution Counter is internally incremented by +1 each time the LT360 rotates CCW across zero degrees, and decremented by -1 each time it rotates CW across zero degrees.

The Revolution Counter is useful for keeping track of the total number of circular rotations when issues such as cable wrapping are involved.

The count value will be stored and maintained during power Off, and restored to the previous value during power On.

2.24 Set EnableControls

■ Class	Set (out)
■ RS-232 String	Set EnableControls [CR]
■ Parameter	(none)
■ Returns	*Ok[0]*

Description
This command will enable the front panel switch button controls on the LT360. Typically it is used after previous use of the *DisableControls* set command.

If the LT360 is used under software programmable control, then it may be desirable in some cases to disable the front panel controls to prevent users from altering the parameters manually. This command will re-enable them.

Note: The front panel controls will always be enabled during power up, even if the unit was previously turned off with the controls disabled.

2.25 Set DisableControls

■ Class	Set (out)
■ RS-232 String	Set EnableControls [CR]
■ Parameter	(none)
■ Returns	*Ok[0]*

Description
This command will disable the front panel switch button controls on the LT360. Typically it is used to eliminate manual changes from the front panel.

If the LT360 is used under software programmable control, then it may be desirable in some cases to disable the front panel controls to prevent users from altering the parameters manually.

Note: The front panel controls will always be enabled during power up, even if the unit was previously turned off with the controls disabled.

3.1 Get Name

■ Class	Get (in)
■ RS-232 String	Get Name [CR]
■ Returns	*NameString[0]*

Description
This command returns the Name for the LT360 unit. This is an arbitrary user name which can be assigned to the unit. This can be useful for applications which utilize multiple units to distinguish between the units, for example Horz, Vert, etc. This name is stored in the LT360 and is non volatile.

The Name parameter string has a maximum length of 21 characters.

3.2 Get Title

- Class — Get (in)
- RS-232 String — Get Title [CR]
- Returns — *LT360 Precision Turntable[0]*

Description
This command returns the Title of the LT360 unit, which always is *LT360 Precision Turntable*. This command can be useful if you are writing your own RS-232 program to control the LT360. This command can be used as a test to verify that you are communicating with an LT360 unit.

3.3 Get Firmware Version

■ Class	Get (in)
■ RS-232 String	Get FirmwareVersion [CR]
■ Returns	*n.nn[0]*

Description
This command returns the version code of the firmware within the LT360. For example: *1.50* The numeric value is returned as a string.

3.4 Get Firmware Date

■ Class	Get (in)
■ RS-232 String	Get FirmwareDate [CR]
■ Returns	*mmm-dd-yyyy[0]*

Description
This command returns the version date of the firmware within the LT360. For example: *JAN-01-2006* The date value is returned as a string.

3.5 Get Production Date

■ Class	Get (in)
■ RS-232 String	Get ProductionDate [CR]
■ Returns	*mmm-dd-yyyy[0]*

Description
This command returns the date when the LT360 was manufactured. For example: *JAN-01-2006* The date value is returned as a string.

3.6 Get Calibration Date

■ Class	Get (in)
■ RS-232 String	Get CalibrationDate [CR]
■ Returns	*mmm-dd-yyyy[0]*

Description
This command returns the date when the LT360 was last calibrated. For example: *JAN-01-2006* The date value is returned as a string.

Calibration is performed using the Win32 application software. Calibration is performed originally at the factory, and in most cases will be permanent and not be required again. It is possible for a customer to performed the calibration, but only if the proper equipment is available. The computer must have a GPIB bus, and two 6-1/2 digit DMMs are required themselves with accurate recent calibration. See the LT360 Manual for further details.

3.7 Get Calibration Due

■ Class	Get (in)
■ RS-232 String	Get CalibrationDue [CR]
■ Returns	*mmm-dd-yyyy[0]*

Description
This command returns the date when the LT360 should be re-calibrated. For example: *JAN-01-2007* The date value is returned as a string.

Calibration is performed using the Win32 application software. Calibration is performed originally at the factory, and in most cases will be permanent and not be required again. However if your application requires NIST compliance, this is the date when the LT360 should be rechecked. It is possible for a customer or other lab to performed the calibration, but only if the proper equipment is available. The computer must have a GPIB bus, and two 6-1/2 digit DMMs are required themselves with accurate recent calibration. See the LT360 Manual for further details.

3.8 Get Serial Number

■ Class	Get (in)
■ RS-232 String	Get SerialNumber [CR]
■ Returns	*nnnnnn[0]*

Description
This command returns the serial number of the LT360 unit. The number is returned as a string value of 6 digits.

3.9 Get Baud Rate

■ Class	Get (in)
■ RS-232 String	Get BaudRate [CR]
■ Returns	*nnnnn[0]*

Description
This command returns the current baud rate the LT360 is using. The number is returned as a string value of 4-6 digits.

3.10 Get Position

- Class — Get (in)
- RS-232 String — Get Position [CR]
- Returns — *±nnn.n[0]*

Description
This command returns the current position of the LT360 in degrees. The value is returned as a string with tenths of degree precision. The value may be 0 to +360.0 or 0 to ±180.0 depending on the settings for the display polarity.

This command is very useful when you want to know the current position of the LT360, or if the LT360 has completed a movement. It can be used in a polling loop to see when the LT360 step or goto command has finished. Another alternative method would be to call the *Get Moving* command until the *NO* value is returned.

When using either method in a polling loop, the frequency of calls should be reasonable. Typically, testing for movement completion once every second is more than adequate.

3.11 Get Pulse Direction

■ Class	Get (in)
■ RS-232 String	Get PulseDir [CR]
■ Returns	*CCW[0]* or *CW[0]*

Description
This command returns the setting for the Pulse Direction. The value is returned as either CCW (counter clockwise) or CW (clockwise). This parameter controls the behavior to a TTL step pulse at the *Pulse* input BNC connector.

When a pulse arrives at the *Pulse* BNC input, the platter will rotate either CCW or CW as defined by this setting. The size of the step is controlled by the *Set StepSize* command.

Note: The TTL pulse should be a minimum of 10uS, or longer. The triggering edge can be controlled by the Set PulseEdge command.

3.12 Get Pulse Edge

■ Class	Get (in)
■ RS-232 String	Get PulseEdge [CR]
■ Returns	*RISE[0]* or *FALL[0]*

Description
This command returns the setting for the Pulse Edge triggering. The value is returned as either RISE (rising edge) or FALL (falling edge). This parameter controls the behavior to a TTL step pulse at the *Pulse* input BNC connector.

When a pulse arrives at the *Pulse* BNC input, the platter will rotate either at the rising or falling edge as defined by this setting. The size of the step is controlled by the *Set StepSize* command.

Note: The TTL pulse should be a minimum of 10uS, or longer. The direction can be controlled by the Set PulseDir command.

3.13 Get StepSize

■ Class	Get (in)
■ RS-232 String	Get StepSize [CR]
■ Returns	*nnn.n[0]*

Description
This command returns the step size value. The value is returned as a string with tenths of a degree precision, and is always positive. This value sets the angular step size by which the platter will rotate to either a step command or TTL pulse.

The minimum step size is 0.1 degrees. Step size is always positive. Direction is controlled by the CCW/CW commands listed else where.

3.14 Get Velocity

■ Class	Get (in)
■ RS-232 String	Get Velocity [CR]
■ Returns	*n.nn[0]*

Description
This command returns the velocity in RPM (revolutions per minute). The value is returned as a string with hundreths precision, and is always positive. The allowable range for the parameter is 0.01 to 3.00 RPM.

3.15 Get Torque

■ Class	Get (in)
■ RS-232 String	Get Torque [CR]
■ Returns	*nnn.n[0]*

Description
This command returns the torque in percent (%). The value is returned as a string with tenths precision, and is always positive. The allowable range for the parameter is 10.0 to 100.0 %.

This value sets the stepper motor torque for the platter rotation. The value is set in percent of maximum torque. For most typical applications your will want the torque set to 100%. If you intend to have the platter in constant motion, than you may wish to set the torque to a lower value such as 70%. This will reduce heating in the motor and drive circuitry.

The *SmartTorque* feature of the LT360 automatically reduces the motor torque to 1/2 its normal rotational power when it is not moving. Thus providing the advantages of maximum torque when in motion and reduced power consumption when stationary.

3.16 Get Acceleration Function

- Class — Get (in)
- RS-232 String — Get AccelFunc [CR]
- Returns — *n[0]*

Description
This command returns the acceleration function number (0 to 4). This value gives the acceleration function which the LT360 will use during step or goto commands. The acceleration profile is an important feature of the LT360 which enables heavy loads to be moved with minimum disturbance.

The parameter string contains a numeric value such as '1' with allowable values from 0..4. There are five acceleration functions numbered 0,1,2,3,4. The names of these profiles are:

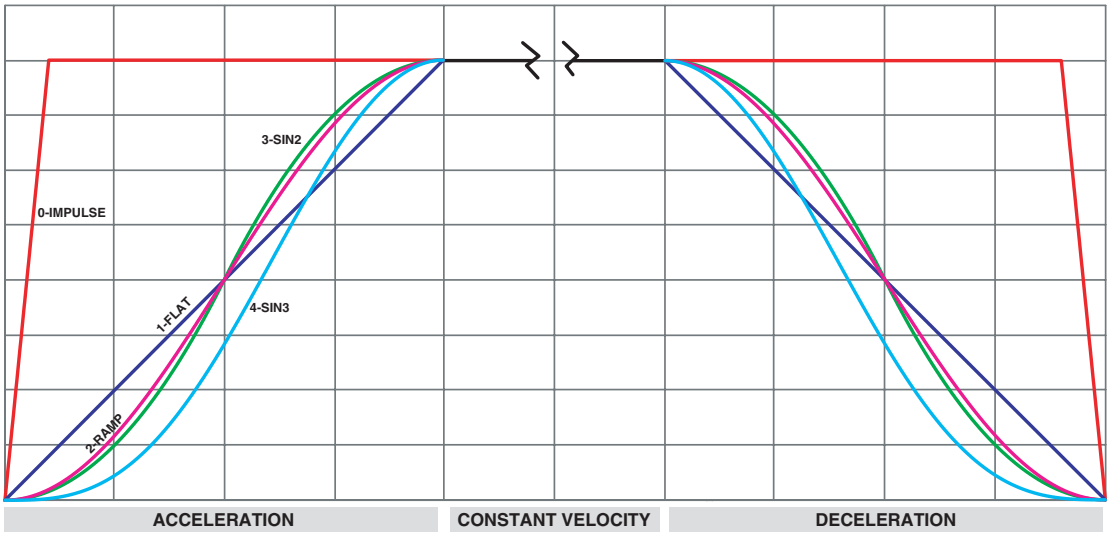
0 - Impulse
1 - Flat
2 - Ramp
3 - Sin2
4 - Sin3

For most typical applications acceleration function 1 - FLAT will provide excellent performance. Because the LT360 controls the acceleration and deceleration so precisely, very heavy loads can be rotated with ease.

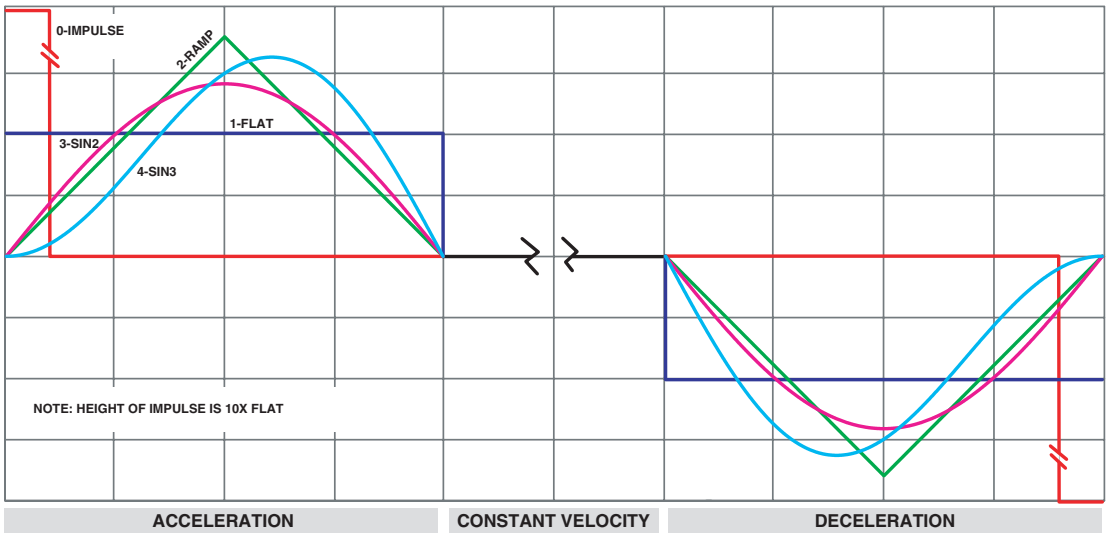
The 0 - IMPULSE function provides the fastest rise time from zero to constant velocity. However it also greatly reduces the maximum load which the LT360 can rotate, due to its higher acceleration. The acceleration of this function is about 10X that of the others.

The graphs on the following page show the velocity and acceleration profile curves for the five different LT360 functions.

LT360 VELOCITY PROFILES



LT360 ACCELERATION PROFILES



3.17 Get Moving

■ Class	Get (in)
■ RS-232 String	Get Moving [CR]
■ Returns	*CCW[0]* or *CW[0]* or *NO[0]*

Description
This command returns the rotational status of the LT360. Three possible string values can be returned:

CCW - rotating counter clockwise
CW - rotating clockwise
NO - not rotating

This command is very useful when you want to know if the LT360 has completed a movement. It can be used in a polling loop to see when the LT360 step or goto command has finished. Another alternative method would be to call the *Get Position* command until the desired location is returned.

When using either method in a polling loop, the frequency of calls should be reasonable. Typically, testing for movement completion once every second is more than adequate.

3.18 Get Pulse Input

■ Class	Get (in)
■ RS-232 String	Get PulseInput [CR]
■ Returns	*OFF[0]* or *ON[0]*

Description
This command returns the status for the Pulse Input BNC function of the LT360. The returned string value is either OFF or ON.

This parameter enables the TTL Pulse Input feature. When *PulseInput* is set ON, the LT360 will respond to TTL pulses at the BNC connector. When it is set OFF, it will ignore any pulses.

Each pulse will trigger a step. The width of the pulse should be at least 10uSec, and the triggering can be set on the rising or falling edge. Any pulses which occur while the LT360 is moving will be ignored.

If you are not using the TTL pulse input, it is probably best to keep this feature OFF to prevent false triggering due to any noise pickup at the open connector.

3.19 Get Analog Input

■ Class	Get (in)
■ RS-232 String	Get AnalogInput [CR]
■ Returns	*OFF[0]* or *ON[0]*

Description
This command returns the status for the Analog Input BNC function of the LT360. The returned string value is either OFF or ON.

This parameter enables the Analog Input feature. When *AnalogInput* is set ON, the LT360 will respond to the DC voltage at the analog input BNC connector. When it is set OFF, it will ignore that input.

The LT360 has a scale factor of 10mV/deg for the analog input. The range can be either unipolar (0 ... +3.600V) or bipolar (0 ... ±1.800V). The input impedance of the analog input is 1M Ohm. Although the input is an unbalanced BNC connector, the actually circuitry is differential. This helps to reject ground noise.

The input source must be very stable and of low noise. The LT360 has a resolution of 0.1 degrees and therefore responds to changes of 1mV at the analog input. When this input is enabled, the LT360 will sample the input and move to the location as indicated by the voltage. After that move is completed, it then samples the analog input again and moves to the new position. Monitoring of the analog input is real time.

If you are not using the analog input, this input *must* be set OFF. Otherwise the LT360 will be controlled by any noise received at the input.

3.20 Get Smart Torque

■ Class	Get (in)
■ RS-232 String	Get SmartTorque [CR]
■ Returns	*OFF[0]* or *ON[0]*

Description
This command returns the status for the Smart Torque feature of the LT360. The returned string value is either OFF or ON.

This parameter controls the behavior of the stepper motor. When *SmartTorque* is set ON, the motor is powered down to half the driving power 2 seconds after each movement stops. Motor power is returned to full (as controlled by the Maximum Torque setting) when the motor starts again.

This feature takes advantage of the fact that the worm gear drive system employed by the LT360 is irreversible. Meaning, the platter is self-locking at its current position. The load cannot rotate the platter itself. Therefore, it is not necessary to maintain full power on the motor merely to hold its current position when the platter is not in movement.

The use of *SmartTorque* greatly reduces the thermal heating in the motor and driver circuitry, and produces a very efficient drive system. Under most conditions and use, this feature should always be kept ON.

If a particular application demands that the motor power be kept constant at all times, even when stationary (SmartTorque=OFF), then the maximum torque should be to 70% or less to prevent excessive heating. It is difficult to envision what kinds of applications would require this behavior, but the option is available if needed.

3.21 Get Display Polarity

■ Class	Get (in)
■ RS-232 String	Get DisplayPolarity [CR]
■ Returns	*UNIPOLAR[0]* or *BIPOLAR[0]*

Description
This command returns the mode for the Display Polarity of the LT360. The returned string value is either UNIPOLAR or BIPOLAR.

This parameter changes how the LT360 represents the degree position on the front display of the chassis, and in the Win32 software. When *DisplayPolarity* is set to UNIPOLAR, the LT360 will display the position in the range of 0 to 360 degrees. All position values are positive. When it is set to BIPOLAR, the range will be 0 to ±180 degrees. Both positive and negative values will be shown.

3.22 Get Input Polarity

■ Class	Get (in)
■ RS-232 String	Get InputPolarity [CR]
■ Returns	*UNIPOLAR[0]* or *BIPOLAR[0]*

Description
This command returns the mode for the Analog Input Polarity of the LT360. The returned string value is either UNIPOLAR or BIPOLAR.

This parameter changes how the LT360 interprets the DC voltage received at the Analog Input connector. When *InputPolarity* is set to UNIPOLAR, the LT360 will accept voltages in the range of 0.000 to +3.600 volts. All position values are positive. When it is set to BIPOLAR, the range will be 0.000 to ±1.800 volts. Both positive and negative values are used.

Voltages higher than the allowable range for the selected mode are invalid.

The LT360 has a scale factor of 10mV/deg for the analog input. The range can be either unipolar (0 ... +3.600V) or bipolar (0 ... ±1.800V). The input impedance of the analog input is 1M Ohm. Although the input is an unbalanced BNC connector, the actually circuitry is differential. This helps to reject ground noise.

The input source must be very stable and of low noise. The LT360 has a resolution of 0.1 degrees and therefore responds to changes of 1mV at the analog input. When this input is enabled, the LT360 will sample the input and move to the location as indicated by the voltage. After that move is completed, it then samples the analog input again and moves to the new position. Monitoring of the analog input is real time.

If you are not using the analog input, this input *must* be set OFF. Otherwise the LT360 will be controlled by any noise received at the input.

3.23 Get Output Polarity

■ Class	Get (in)
■ RS-232 String	Get OutputPolarity [CR]
■ Returns	*UNIPOLAR[0]* or *BIPOLAR[0]*

Description
This command returns the mode for the Analog Output Polarity of the LT360. The returned string value is either UNIPOLAR or BIPOLAR.

This parameter changes how the LT360 generates the DC voltage output at the Analog Output connector. When *OutputPolarity* is set to UNIPOLAR, the LT360 will produce voltages in the range of 0.000 to +3.600 volts. All position values are positive. When it is set to BIPOLAR, the range will be 0.000 to ±1.800 volts. Both positive and negative values are produced.

The LT360 has a scale factor of 10mV/deg for the analog output. The range can be either unipolar (0 ... +3.600V) or bipolar (0 ... ±1.800V). The output impedance of the analog output is <200 Ohms. Although the output is an unbalanced BNC connector, it is highly recommended that the input on the other device be differential to reject ground noise. Since the smallest resolution of the LT360 is 0.1 degree, the equivalent resolution in the output is 1mV. The input impedance of the other device should be at least 100K Ohm to prevent loading errors.

3.24 Get Motor Home Check

- Class — Get (in)
- RS-232 String — Get MotorHomeChk [CR]
- Returns — *OFF[0]* or *ON[0]*

Description
This command returns the status for checking stepper motor synchronization. The returned string value is either OFF or ON.

When *MotorHomeChk* is set ON, the firmware checks that the motor returns to its home position after each movement. If disabled, no checking is performed. If checking is enabled and the motor does not end its operation in the Home position, an error message is shown on the front panel display.

The stepper motor produces a specific number steps for a given amount of platter rotation. The LT360 utilizes microstepping for additional precision control. One particular phase of the motor is designated as the *Home* position. For any given platter rotation, the motor should always end in its Home position.

In normal operation this error should never occur. If it does, the unit must be powered off to clear the error, or this feature disabled. It is possible for this error to be produced by noise in the stepper motor controller, or by the load exceeding the torque capability thereby causing the motor to skip steps.

If this error is shown repeatedly, either there is a problem internal in the LT360 or the torque load is too great for the drive system.

3.25 Get Revision Code

■ Class	Get (in)
■ RS-232 String	Get RevCode [CR]
■ Returns	*nn[0]*

Description
This command returns the revision code of the controller PC board, as an ASCII code for the revision letter. For example, '65' = A.

3.26 Get Output Mode

■ Class	Get (in)
■ RS-232 String	Get OutputMode [CR]
■ Returns	*CONT[0]* or *START[0]* or *STOP[0]*

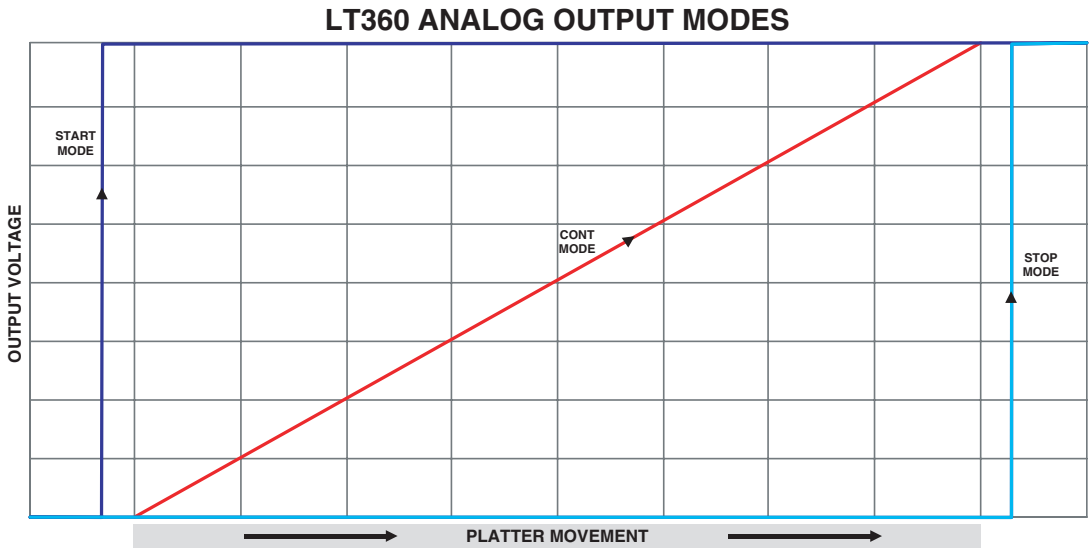
Description
This command returns the operating mode for the Analog Output. The returned string value is either CONT or START or STOP.

When *CONT* is the mode, the Analog Output will continuously follow the platter position as it is moving. This is a real time output mode where the analog output voltage always represents the current platter position.

When *START* is the mode, the Analog Output will only produce a single step change output voltage representing the final destination position at the start of the movement. For example, if the current position is 0.0 degrees and the destination is 100.0 degrees, the Analog Output voltage will change from 0.000V to 1.000V at the start of the movement.

When *STOP* is the mode, the Analog Output will only produce a single step change output voltage representing the final destination position after the movement stops. For example, if the current position is 0.0 degrees and the destination is 100.0 degrees, the Analog Output voltage will remain at 0.000V during the movement and then change to 1.000V when the movement stops. The final output voltage changes 2 seconds after the platter movemet stops.

If the movement is aborted before it reaches the destination, the output voltage will be updated to the correct final value in all modes.


LT360 ANALOG OUTPUT MODES
START MODE
CONT MODE
STOP MODE
OUTPUT VOLTAGE
PLATTER MOVEMENT

3.27 Get Revolution

■ Class	Get (in)
■ RS-232 String	Get Revolution [CR]
■ Returns	±*nnn[0]*

Description
This command returns the internal Revolution Counter value. The integer value string provides the full rotation count.

The Revolution Counter is internally incremented by +1 each time the LT360 rotates CCW across zero degrees, and decremented by -1 each time it rotates CW across zero degrees.

The Revolution Counter is useful for keeping track of the total number of circular rotations when issues such as cable wrapping are involved.

The count value will be stored and maintained during power Off, and restored to the previous value during power On.