Low Level Command Set

SETUP CONTROL COMMANDS

Currently, the only way to access the low level format is through the Setup Control Commands.

The following are special commands used to setup different properties of the MS-2000 and MFC-2000. The MS-2000 and the MFC-2000 recognizes these two byte commands by their prefix byte 255. These commands mimic the Ludl Interface Control Commands and expand upon them.

Almost every firmware build comes with the `LL_COMMANDS` firmware module, you can use BUILD X to check which modules you have.

The Tiger controller has it's own set of commands, see the Tiger Low Level Commands documentation.

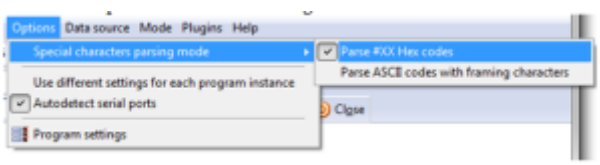
For HyperTerminal

Command	Description
255 65 Alt[255] 'A'	Switch to High Level Command Format (Note: the post byte “A” must be in caps)
255 66 Alt[255] 'B'	Switch to Low Level Command Format
255 82 Alt[255] 'R'	Reset Controller
255 72 Alt[255] 'H'	Return hundredth of a micron precision for High Level WHERE command.
255 84 Alt[255] 'T'	Return tenth of a micron precision for High Level WHERE command.

Note: Remote Switch Scanning and Transmission Delay is not supported

For Advanced Serial Port Monitor

Make Sure Special>Character Parsing mode>Parse #XX Hex Code is enabled



Command	Description
#FF#41	Switch to High Level Command Set
#FF#42	Switch to Low Level Command Set

Command	Description
#FF#52	Reset Controller

LOW LEVEL FORMAT

This serial RS-232 interface is used to hook up the MS-2000 and MFC-2000 to a PC with a protocol that imitates the Ludl Low Level command set. The purpose of the low level protocol is to provide a simple interface between a PC program and the MS-2000 and the MFC-2000, without ASCII conversion. The high level protocol is designed to allow direct human interface capability by displaying all numbers and commands in ASCII characters. The high level format is slow due to the extended transmission of ASCII characters as well as the time consumed converting back and forth from 3 byte memory stored numbers and multiple byte ASCII character numbers stored in strings. The low level format deals strictly with numbers that identify modules, commands, data_size, and data represented in 1 to 6 bytes in 2's compliment form.

NOTE: These commands apply to MS-2000 Controller firmware version 3.2 and forward.

The low level format is formed by the following 8 bit bytes:

BYTE1: Axis Identification
BYTE2: Command
BYTE3: Number of data bytes to be exchanged for this command
BYTES 4-9: Data Bytes, mostly in 2's compliment form in the order of: Least Significant Byte, Middle Byte, Most Significant Byte
LAST BYTE: The ASCII colon character (:) flags the end of the serial command

All values specified through this section of the manual use the following format:

000000	Decimal
0x0000	Hexadecimal
Ctrl<A>	ASCII character pressed with Ctrl held down
Alt[0000]	Decimal number typed with Alt held down
'A'	ASCII character typed in

RS-232 Timeout: The normal Ludl 2-second timeout is not implemented. The MS-2000 clears its buffers whenever a colon (:) is received, thereby eliminating any error prone characters received serially.

SERIAL DELAY: Due to the use of higher speed computers, there is no longer any need to delay serial communication replies; therefore, serial delays are not supported by the MS-2000.

WARNING: When using the RS-232 OUT port to daisy chain RS 232 devices, it must be taken into consideration that the MS-2000 monitors all serial traffic on the line. Although the Low level command set will not respond with an error to a foreign command like the high level command set, it is possible for the correct sequence of numbers to be entered which would match an actual command. This would result in the MS-2000 executing an unwanted command.

It is recommended that the RS-232 OUT port is not used with the low level command set.

Commands are generally broken into five groups. In three special cases the number-of-data-bytes group is omitted to speed up the communication process.

Data values are broken into 8-bit bytes for the data length times, and then each byte is sent out through serial channel to the interface, from LSB to MSB.

The ASCII colon (:) character is defined as the end of command code, and used to terminate the command loading sequence at which time the controller clears the serial buffer and attempts to process the command. If the command has errors and cannot be processed and executed, it is ignored.

Note: The MS-2000 does not support parity check.

Group 1 /Byte 1: Axis Identifier This one byte character identifies which axis or control function the command is for.

X Axis:	Dec: 24 (Hex: 0x18)	Keyboard: Ctrl<X>
Y Axis:	Dec: 25 (Hex: 0x19)	Keyboard: Ctrl<Y>
Z Axis:	Dec: 26 (Hex: 0x1A)	Keyboard: Ctrl<Z>
F Axis:	Dec: 27 (Hex: 0x1B)	Keyboard: ESC

The following are reserved for future use:

Autofocus	Dec:01(Hex: 0x01)	Keyboard: Ctrl<A>
Controller	Dec:03(Hex: 0x0)3	Keyboard: Ctrl<C>
Scan	Dec:03(Hex: 0x03)	Keyboard: Ctrl<S>

Group 2 / Byte 2: Command Identifier This is a single byte instruction code. These codes are listed in this manual. If the end command ':' is received at this point, then the command is aborted and ignored.

Group 3 / Byte 3: Data Size This is a single byte that gives the number of data bytes for this instruction. This value can also be found in command listing for different commands. Although the range of this variable is from 0 to 255, the MS-2000 only supports 0 to 6 up to firmware version 3.3.

Exceptions: There are 3 commands that do not use this data group: '?'-request status, 'G'-start motor / function, 'B'-stop motor / function.

Group 4 / Bytes 4-?: Data Bytes This group holds the data for the command whether the command is sending or receiving information. The number of bytes for this group varies with each command and is stated in Group 3. Numerical information is broken down into the 8 bit bytes. These are transmitted in the order of **L**east **S**ignificant **B**yte, **M**iddle **B**yte, then **M**ost **S**ignificant **B**yte. Positive numbers are divided down using the base of 256. Numbers that may go negative are sent in 2's compliment. For more information, see the examples for individual commands.

Group 5: Last Byte This is a one-byte end-of-command character ':'. The MS-2000 will not recognize a command until this character is received. When the ':' is received, the MS-2000 goes to a subroutine which then pulls Groups 1-4 out of the serial port buffer, and then searches the buffer until the ':' is found. Any information between Group 4 and the ':' is ignored.

COMMAND LISTING

The following are commands formatted by the MS-2000 shown in Decimal, and keyboard / ASCII form. The first command, Read Status, give examples that explain in depth the formatting which will be used for the rest of the examples.

Command: Read Status

Dec	63
Hex	0x3f
Keyboard	?
Data Size	None

The MS-2000 will respond to this command in the following manor. If the motor signal is not zero or there is a command being executed and the axis motor is enabled, the controller will return an upper case B. Otherwise, it will return a lower case b.

Example

Command: 24 63 58
Reply: 66

The above is an example of a stream of bytes that a PC would send serially to the controller and the controller's reply. In the above example the 24 represents the X axis, the 63 represents the Read Status command and the 58 is the colon which signifies the end of the command. The reply 66 is the decimal code for the ASCII character B, which means the axis is currently busy.

Example: Ctrl<X>?:b

The above example shows a way to enter this command using a terminal screen where the Ctrl<X> means that the Ctrl key is held down while the key capital X is pressed. This enters the axis identifier for the X axis. The ? stands for the command Read Status and the : signifies the end of the command.

The b is the controller's response, which means the axis is not busy. Notice that with the low level command set there are no spaces, carriage returns or line feeds. Note that, for the sake of easy recognition of the computer response, all computer responses in this manual will be either labeled so, or be printed in italics.

Command: Read Motor Position

Dec	97
Hex	0x61
Keyboard	a
Data Size	3

Requests the MS-2000 to respond with the current stage position in two's compliment form using 3 bytes. The response is in tenths of microns.

Example

Command: 24 97 03 58
Reply in Dec: 160 134 01

The above is an example of a stream of bytes that a PC would send serially to the controller and the controller’s reply.

In the above example the 24 represents the X axis, the 97 represents the Read Motor Position command. The 3 means that the controller should return 3 bytes of data, and the 58 is the colon, which signifies the end of the command. In the reply are three bytes: lsb:160, the mb:134, and the msb:01.

Conversion: `160+(134*256)+(1*256*256)=100000` tenths of a micron or 10 millimeters from the origin.

The example below shows the same example above as it would appear on a computer serial port terminal program such as Hyperterminal (see command 63 for this manuals formatting information). As can be seen the numbers 160 134 01 correspond to non-legible ASCII characters. For this reason it is next to impossible to use a terminal program with the low level command set.

<X>a<C>: áå_

Note: As can be seen in the Read Motor Position Command, many low level commands are incompatible with terminal screens, so no terminal screen example will be given throughout the rest of the manual for those commands.

Command: Read Increment Value

Dec	100
Hex	0x64
Keyboard	d
Data Size	3

Requests the MS-2000 to respond with current setting for the distance of increment moves. The number is a three byte two's compliment number representing a position offset in tenths of a micron.

Example:

Command: 24 100 03 58
Reply in Dec: 160 134 01

The above is an example of a stream of bytes that a PC would send serially to the controller and the controller’s reply.

In the above example the 24 represents the X axis, the 100 represents the Read Increment Value command. The 3 means that the controller should return 3 bytes of data, and the 58 is the colon, which signifies the end of the command. In the reply are three bytes: lsb:160, the mb:134, and the msb:01.

Conversion: 160+(134*256)+(1*256*256)=100000 tenths of a micron or 10 millimeters from the origin.

Command: Read Identification

Dec	105
Hex	0x69
Keyboard	i
Data Size	6

Requests the MS-2000 to respond with the identification code for the Axis Id. The response for X, Y, and Z axis' is EMOT :. The fifth and sixth bytes are spaces (ASCII code 32).

Note: The MS-2000 does not support consecutive 105 commands to read the version information.

Example:

Command: 24 105 58
Reply in Dec: 69 77 79 84 32 58

Reply Converted to ASCII: EMOT :

The above is an example of a stream of bytes that a PC would send serially to the controller and the controller’s reply. The example below shows the same example above as it would appear on a computer serial port terminal program such as Hyperterminal (see command 63 for this manuals formatting information).

<X>i: EMOT : , there is a space after T and before.

Command: Read Motor Position and Status

Dec	108
Hex	0x6C
Keyboard	l
Data Size	4

Requests the MS-2000 to respond with the current stage position in two's compliment form using 3 bytes followed by the status byte. The response is in tenths of microns. See command 126 (Read Status Byte) for more information on the status byte.

Example

Command: 24 108 03 58
Reply in Dec: 160 134 01 20

The above is an example of a stream of bytes that a PC would send serially to the controller and the controller’s reply.

In the above example the 24 represents the X axis, the 97 represents the Read Motor Position command. The 3 means that the controller should return 3 bytes of data, and the 58 is the colon, which signifies the end of the command. In the reply are three bytes, the lsb:10, the mb:1, and the msb:2, plus the status byte. This can be translated as follows:

160+(134*256)+(1*256*256)=100000 tenths of a micron or 10 millimeters from the origin.

See command 126 (Read Status Byte) for more information on the status byte.

Command: Read Current Speed

Dec	111
Hex	0x6F
Keyboard	o
Data Size	2

Requests the MS-2000 to respond with current instantaneous value of the servo speed trajectory. The number returned is a signed two-byte number representing the velocity in □m/s.

Example:

Command: 24 111 02 58
Reply in Dec: 78 02

The above is an example of a stream of bytes that a PC would send serially to the controller and the controller's reply.

In the above example the 24 represents the X axis, the 111 represents the Read Current Speed command. The 02 means that the controller should return 2 bytes of data, and the 58 is the colon, which signifies the end of the command. The reply is made up of an lsb and msb, which would convert as follows:

78+(2*256) =590 µm/s or 0.59 mm/second

Command: Read Ramp Time

Dec	113
Hex	0x71
Keyboard	q
Data Size	1

Requests the MS-2000 to respond with current setting for the time to ramp up and down. This is a one-byte number between 1 and 255. It represents the number of milliseconds the ramp from start speed to maximum speed at the beginning of a move and from maximum speed to start speed at the end of a move will take.

Example:

Command: 24 113 01 58
Reply in Dec: 78

The above is an example of a stream of bytes that a PC would send serially to the controller and the controller's reply.

In the above example the 24 represents the X axis, the 113 represents the Read Ram Time command. The 1 means that the controller should return 1 byte of data, and the 58 is the colon, which signifies the end of the command. The reply 78 means that the controller will allow 78 milliseconds for ramping up and down.

Command: Read Start Speed

Dec	114
Hex	0x72
Keyboard	r
Data Size	2

Dummy function. Do not use.

Command: Read Maximum Speed

Dec	115
Hex	: 0x73
Keyboard	s
Data Size	2

Requests the MS -2000 to respond with current setting for the maximum speed the stage is allowed to move. The number returned is a straight two-byte number representing a speed □m/s.

Example:

Command: 24 114 02 58
Reply in Dec: 78 02

The above is an example of a stream of bytes that a PC would send serially to the controller and the controller's reply. In the above example the 24 represents the X axis, the 115 represents the Read Maximum Speed command. The 02 means that the controller should return 2 bytes of data, and the 58 is the colon, which signifies the end of the command. The reply is made up of an lsb and msb, which would convert as follows:

78+(2*256) =590 µm/s or 0.59 mm/second

Command: Read Target Position

Dec	116
Hex	0x74
Keyboard	t
Data Size	3

Requests the MS-2000 to respond with current target position. The number is a three byte, two's compliment, number representing a position offset in tenths of a micron.

Example:

Command: 24 116 03 58
Reply in Dec: 160 134 01

The above is an example of a stream of bytes that a PC would send serially to the controller and the controller's reply.

In the above example the 24 represents the X axis, the 116 represents the Read Target Position command. The 3 means that the controller should return 3 bytes of data, and the 58 is the colon, which signifies the end of the command. In the reply are three bytes: lsb:160, the mb:134, and the msb:01.

Conversion: `160+(134*256)+(1*256*256)=100000` tenths of a micron or 10 millimeters from the origin.

Command: Read Status Byte

Dec	126
Hex	0x7E
Keyboard	~
Data Size	1

Requests the MS-2000 to respond with the Status Byte. The number is one byte, which can be broken down into 8 bits that represent the following internal flags:

Bit 0	0 = No Motor Signal, 1 = Motor Signal (i.e., axis is moving)
Bit 1	Always 1, as servos cannot be turned off
Bit 2	0 = Pulses Off, 1 = Pulses On
Bit 3	0 = Joystick/Knob disabled, 1 = Joystick/Knob enabled
Bit 4	0 = motor not ramping, 1 = motor ramping
Bit 5	0 = ramping up, 1= ramping down

Bit 6	Upper limit switch: 0 = open, 1 = closed
Bit 7	Lower limit switch: 0 = open, 1 = closed

Example:

Command: 24 126 58
Reply: 138

The above is an example of a stream of bytes that a PC would send serially to the controller and the controller's reply.

In the above example the 24 represents the X axis, the 126 represents the Read Status Byte command. The 58 is the colon, which signifies the end of the command. The reply can be broken into its individual bits as follows:

B7: 1 -Axis is at upper limit
B6: 0 -Lower limit switch open
B5: 0 -Ramping down if ramping
B4: 0 -Not ramping
B3: 1 -Joystick is enabled
B2: 0 -Pulses are not being used
B1: 1 -Servo Encoders are in use
B0: 0 -Motors are not turned on

Command: Start / Enable Motor

Dec	71
Hex	47
Keyboard	G
Data Size	0

Enables the function. Mainly used to turn on / start / enable the motor for an axis specified. Does not give or receive data so the data field is omitted and the end character ':' follows directly.

Example:

Command: 24 71 58
Response: There is no response

Command: Stop / Disable Motor

Dec	66
Hex	42
Keyboard	B

Data Size	0

Disables the function. Mainly used to turn off / stop / disable the motor for an axis specified. Does not give or receive data so the data field is omitted and the end character ':' follows directly. Starting with firmware version 3.3, a disabled axis / function will reply to the Status command with a not busy 'b' even if the current position and target position do not match.

Example:

Command: 24 71 58
Response: There is no response

Command: Write Motor Position

Dec	65
Hex	0x41
Keyboard	A
Data Size	3

Requests the MS-2000 to write the given position to the current position count buffer. The position is given in two's compliment form using 3 bytes. The number represents the position in tenths of microns.

Example:

Command: 24 65 03 160 134 01 58
Reply: There is no reply

The above is an example of a stream of bytes that a PC would send serially to the controller. The 24 represents the X axis, the 65 represents the Write Motor Position command. The 3 means that the controller should read three bytes of data. The three bytes are: lsb:160, the mb:134, and the msb:01. The 58 is the colon which signifies the end of the command.

Conversion: `160+(134*256)+(1*256*256)=100000` tenths of a micron or 10 millimeters from the origin.

Reverse Conversion:

10 millimeters *10,000 to get tenths of microns =100,000

lsb = remainder of 100,000 / 256 = 160 mb = remainder of 100,000 / 256 / 256 = 134 msb = remainder of 100,000 / 256 / 256 / 256 =1

Command: Write Target Position (move)

Dec	84
Hex	0x54

Keyboard	T
Data Size	3

Requests the MS-2000 to write the given position to the target position buffer. The position is given in two's compliment form using 3 bytes. The number represents the position in tenths of microns.

Example:

Command: 24 84 03 160 134 01 58
Reply: There is no reply

The above is an example of a stream of bytes that a PC would send serially to the controller. The 24 represents the X axis, the 84 represents the Write Target Position command. The 3 means that the controller should read three bytes of data. The three bytes are: lsb:160, the mb:134, and the msb:01. The 58 is the colon which signifies the end of the command.

Conversion: `160+(134*256)+(1*256*256)=100000` tenths of a micron or 10 millimeters from the origin.

Reverse Conversion:

10 millimeters *10,000 to get tenths of microns =100,000

lsb = remainder of 100,000 / 256 = 160 mb = remainder of 100,000 / 256 / 256 = 134 msb = remainder of 100,000 / 256 / 256 / 256 =1

Command: Increment Move Up

Dec	43
Hex	0x2B
Keyboard	+
Data Size	0

Requests the MS-2000 to add the Increment Value to the Current Position Value and place the result in the Target Position Buffer. There is no data or response.

Example:

Command: 24 43 0 58
Reply in Dec: No reply

The above is an example of a stream of bytes that a PC would send serially to the controller. In the above example the 24 represents the X axis, the 43 represents the Increment Move Up command. The 0 means that there is no data. The 58 is the end of command character.

Command: Increment Move Down

Dec	45

Hex	0x2D
Keyboard	-
Data Size	0

Requests the MS-2000 to subtract the Increment Value to the Current Position Value and place the result in the Target Position Buffer. There is no data or response.

Example:

Command: 24 45 0 58
Reply in Dec: No reply

The above is an example of a stream of bytes that a PC would send serially to the controller. In the above example the 24 represents the X axis, the 45 represents the Increment Move Down command. The 0 means that there is no data. The 58 is the end of command character.

Command: Write Increment Value

Dec	68
Hex	0x44
Keyboard	D
Data Size	3

Requests the MS-2000 to write the given position to the Increment Value buffer. The position is given in two's compliment form using 3 bytes. The number represents the position in tenths of microns. The Increment Value is used for making successive Relative Moves.

Example:

Command: 24 68 03 160 134 01 58
Reply: There is no reply

The above is an example of a stream of bytes that a PC would send serially to the controller. The 24 represents the X axis, the 68 represents the Write Increment Value command. The 3 means that the controller should read three bytes of data. The three bytes are: lsb:160, the mb:134, and the msb:01. The 58 is the colon which signifies the end of the command.

Conversion: `160+(134*256)+(1*256*256)=100000` tenths of a micron or 10 millimeters from the origin.

Reverse Conversion:

10 millimeters *10,000 to get tenths of microns =100,000

lsb = remainder of 100,000 / 256 = 160 mb = remainder of 100,000 / 256 / 256 = 134 msb = remainder of 100,000 / 256 / 256 / 256 =1

Command: Write Ramping Time

Dec	81
Hex	0x51
Keyboard	Q
Data Size	1

Requests the MS-2000 to write the given byte to the Ramping Time buffer. Value Range is from 0 to 256 in the unit of milliseconds. The ramp time sets the stage acceleration at Max_Speed / Ramp_Time. For short moves the acceleration will be at the same rate as for long moves, but the duration of the ramp will be less than the full ramp time. To minimize damage to the servo motors it is recommended that the ramp time always be greater than 50ms when ramping to full motor speed.

Example:

Command: 24 81 01 45 58
Reply: There is no reply

The above is an example of a stream of bytes that a PC would send serially to the controller. The 24 represents the X axis, the 81 represents the Write Ramping Time command. The 01 means that the controller should read one byte of data. The 45 is the byte of data which means the ramp time will be set to 45 milliseconds. The 58 is the colon which signifies the end of the command.

Command: Write Start Speed

Dec	82
Hex	0x52
Keyboard	R
Data Size	2

Dummy function – do not use.

Command: Write Top Speed

Dec	83
Hex	0x53
Keyboard	S
Data Size	2

Requests the MS-2000 to write the given speed to the Top Speed buffer. The speed is divided down into two 8-bit bytes by dividing the number down by 256. The number represents the speed in µm/s.

Example:

Command: 24 83 2 112 23 58
Reply: There is no reply

The above is an example of a stream of bytes that a PC would send serially to the controller. The 24 represents the X axis, the 83 represents the Write Top Speed command. The 2 means that the controller should read two bytes of data. The two bytes are: lsb:112 and the msb:23. This means the top speed the axis will travel is at 6000 □m/s. The 58 is the colon which signifies the end of the command.

Conversion: 112+(23*256)=6000 µm/s or 6 mm/s.

Reverse Conversion:

6 millimeters/second *1000 to get microns =6000 microns/second

lsb = remainder of 6000 / 256 = 112 msb = remainder of (6000 / 256)† / 256 = 23

†drop remainder of first division and take remainder of second division

Command: Write Vector Speed

Dec	94
Hex	0x5E
Keyboard	
Data Size	2

Instructs the MS-2000 to immediately ramp motors to given velocity value and continue at that speed until instructed otherwise. The velocity is a two-byte value. The binary number represents the velocity in □m/s. The acceleration rate is set by the Write-Ramping-Time and Write-Top-Speed settings. (see Write-Ramping-Time command).

Example:

Command: 24 94 2 112 23 58
Reply: There is no reply

The above is an example of a stream of bytes that a PC would send serially to the controller. The 24 represents the X axis, the 94 represents the Write Vector Speed command. The 2 means that the controller should read two bytes of data. The two bytes are: lsb:112 and the msb:23. The stage ramps to the speed 6000 □m/s. The 58 is the colon which signifies the end of the command.

Conversion: 112+(23*256)=6000 µm/s or 6 mm/s.

Command: Joystick / Control Device Enable

Dec	74
Hex	0x4A
Keyboard	J
Data Size	0

Enables the control device function. Allows enabling a control device such as a Joystick or Command Knob to be re-enabled.

Example:

Command: 24 74 58 OR 24 74 0 58

The above is an example of a stream of bytes that a PC would send serially to the controller. The 24 represents the X axis, the 74 represents the Enable Joystick command. The data size is 0 and can either be included or left off on the MS-2000. The 58 is the colon which signifies the end of the command.

Command: Joystick / Controller Disable

Dec	75
Hex	0x4B
Keyboard	K
Data Size	0

Disables the control device function. Allows disabling a control device such as a Joystick or Command Knob so that no external signals are allowed to affect move functions during PC control.

Example:

Command: 24 75 58 OR 24 75 0 58

The above is an example of a stream of bytes that a PC would send serially to the controller. The 24 represents the X axis, the 74 represents the Disable Joystick command. The data size is 0 and can either be included or left off on the MS-2000. The 58 is the colon which signifies the end of the command.

ms2000, lowlevel, manual