

# Meade LX200 Classic Hand Controller Communications

Version 3.xx Protocol - Rob Ferrante 12/19

This document will record the results of research done to seek to understand the messaging language between the Meade LX200 Classic telescope and its hand controller. This work was originally done as part of the development of a new replacement for the original hand controller. To do that work without copying proprietary firmware, and to be able to target other MCUs, new software had to be developed that could communicate with the telescope in exactly the same way, byte-for-byte. Here we make the results of that work available in case it is of interest to others.

*This work is licensed under a Creative Commons Attribution 4.0 International License.*

*<https://creativecommons.org/licenses/by/4.0/>*

We will be describing the communications that go back and forth between the telescope's main microprocessor (CPU) and the hand controller's simpler microcontroller. For brevity, we'll term those the **cpu** and the **handset**.

For background in case the reader is not familiar with the LX200, there are several incarnations of this telescope (LX200 GPS, etc.), with later models still being built. The LX200 of concern in this document is the first model, which is now referred to as the "Classic" LX200. This model was sold around 25 years ago; it is old. The first LX200s sold behave differently than the one we describe here; they used an earlier software system (Version 2.xx). The Version 3.xx software began shipping on later LX200s and is the version on the majority of LX200 Classics that are still in operation today. The Version 3.xx LX200 Classic is the telescope we investigate.

---

## Communication

### Inverted Logic

A very important point, is that the TTL-level logic signals are inverted compared to the typical UART signal levels you would expect from microprocessors. Each side inverts the signal, so that a bit sent as a logic 1, becomes a logic 0 as it leaves the telescope, and is converted back to a 1 as it enters the handset but before the handset processor sees it. All works out unless you are an observer between the two. The data presented here will always be in its normal, non-inverted form.

### Communication Tokens

Messages from the cpu are byte sequences comprised of characters to directly display to the user on the 2-row-by-16-character handset display, and commands to operate other parts of the handset such as LEDs, sounds, and display management.

The handset processes the cpu messages with two layered rules:

- 1) If a command delimiter is received, read the next N bytes of command information and then perform the command.
- 2) Otherwise, display the byte.

The command delimiter is the ASCII Escape character, or the byte 1B (hexadecimal). We will be using hexadecimal notation for byte values unless otherwise noted.

Displayable bytes are those printable ASCII characters such as “ABCabc123!@#” etc. In general bytes between 20 (the space character) and 7E (the tilde) are printable, anything else is part of a command, coming immediately after the command name byte. There are a few exceptions to this rule however. Three special characters are programmed into the display, and “take over” the bytes 00, 01, and 02, which would normally not map to printable characters. They have been repurposed for the special *up/down arrow* character, the *star* character, and the *checkmark* character which Meade uses in certain areas of the menus. Additionally, the display will show extended characters for some of the non-ASCII byte values, and two of those are used: The byte FF will display a solid block, which is used in bar graphs displayed in certain modes, and the byte DF will display the *degree* symbol when showing degrees of declination etc.

### CPU Command Set

The set of commands is small and it will help in understanding the following dialog examples to first briefly describe all of the commands. More information about individual commands is available in Appendix A.

Some commands are modal, such that they turn “on” some mode, and then toggle it “off” later. The bytes received when the mode is “on” may be processed according to different rules. Those will be noted under the section for each command.

Again, the Escape character lets the handset know that the next byte will be the name of a command, rather than some text to display. So the “L” command, for example, will be received as the byte 1B, then the byte 4C (“L”), then the byte which contains the command data. All command name bytes are followed by a fixed number of data bytes, so that once that number of bytes has been read, the next byte encountered might be an escape character starting another command, or it might be a character to show on the handset display.

**Table of Commands**

Command Name Byte	Command Description	Data Bytes
42	“B”: Brightness of LEDs and backlight	1
4C	“L”: LEDs ON/OFF state	1
4D	“M”: Receive raw data mode (on or off)	1
43	“C”: Clear the display	0
31	“1”: Move the display cursor to the start of the first line	0
32	“2”: Move the display cursor to the start of the second line	0
44	“D”: Show the blinking cursor for data entry by user (on or off)	1
53	“S”: Sound the beeper	3
50	“P”: Set the display position by row and column	2
49	“I”: Advance or retreat the display position	1

## Handset Messages

Messages from the handset to the telescope are simply the codes for any keypad button presses the user has made. If the user clicks the Mode button, the handset will send the byte 4D (“M”) when clicked, and CD when released. There is always a down-press byte and a release byte, and the release byte is always the same byte except that the most-significant-bit is high instead of low. All down-press bytes have msb 0, and all release bytes have msb 1.

## Keypad Key Press Messages

Key Name	Down Byte	Up Byte
ENTER	0D	8D
MODE	4D	CD
GO TO	47	C7
N	4E	CE
S	53	D3
E	45	C5
W	57	D7
9 - M	39	B9
8 - RET	38	B8
7 - SLEW	37	B7
6 - STAR	36	B6
5 - FOCUS	35	B5
4 - FIND	34	B4
3 - CNGC	33	B3
2 - MAP	32	B2
1 - CNTR	31	B1
GUIDE	30	B0
PREV	2E	AE
NEXT	44	C4

---

## Selected Dialogs

### Start-up:

The first thing to investigate is the start-up sequence. That's interesting because if there is no handset attached, the telescope will not respond to any commands, even those sent through one of the RS-232 ports. Below is the communication that happens automatically when power is turned on.

```
004 FF FF FF 2A
   " " " *

030 1B 31 20 20 20 20 4C 58 32 30 30 2F 32 1B 32 20 20 56 65 72 73 69 6F 6E 20 33 2E 33 34 4C
   " 1          L X 2 0 0 / 2 " 2          V e r s i o n   3 . 3 4 L

081 1B 4C 08 1B 4D 01 04 0E 15 04 04 15 0E 04 04 15 0E 1B 1B 0E 15 04 04 01 02 02 02 04 14 1C
   " L " " M " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
   08 04 04 04 04 04 15 0E 04 1B 4D 00 1B 31 20 54 45 4C 45 53 43 4F 50 45 20 20 20 20 20
   " " " " " " " " " " " M " " 1   T E L E S C O P E
   1B 32 20 4F 42 4A 45 43 54 20 4C 49 42 52 41 52 59 20 1B 31 7E
   " 2   O B J E C T   L I B R A R Y   " 1 ~
```

This is the way we will present the communication. Each message will be repeated on two lines. The first line will show three digits in gray color, then the message in a color depending on whether the bytes are coming FROM the handset TO the cpu (orange), or FROM the cpu TO the handset (blue). The three digits are the message length in bytes, as a decimal number. After that the actual message starts, with each byte shown as a hexadecimal (base 16) number. The numbers are separated by one space for clarity; they actually follow one another immediately. The second line, immediately below it, will display the same message but with any printable bytes as characters, and unprintable bytes as two small dots together. Each byte, if printable, will have its character representation immediately below it.

Using this format, you see that the first two lines represent 4 bytes sent from the handset to the cpu. Three FF bytes, followed by a 2A, which happens to be printable as the asterisk character. This small message lets the telescope know that the handset is present and awake, ready to display the cpu's messages to the user.

The next two messages are the response from the cpu. These two messages are primarily text to show to the user on the handset's two-line display. In the first message, of 30 bytes, there is initially a 1B character, which is the ASCII escape character. It does not print but is an important delimiter for the handset, telling it a new command is starting, in this case the "1" command to write something to the display on line 1. The content to write to the display is next, starting with 4 space characters (the hexadecimal number 20 corresponds to the space character.) Then everything up to the next 1B is the rest of the display ("LX200/2 2"). Not all 16 characters of the display are written, since the initial state of those final positions of the display is blank already. The next 1B is followed by a 32, printable as "2" which means get ready to receive the characters to display on line 2. Then those follow, starting with 2 space characters.

The final message is 81 characters long, too long to display nicely here, unfortunately, so it wraps around. The 1B leads off, with the "L" command. This is the command to turn on one of the LEDs, in this case the 08 tells the handset that the LED to the left of the 7 key on the keypad is to be turned on, which tells the user that the cpu starts out set to "slew" speed.

The “M” command turns on a mode for receiving raw data. The meaning of that data can only be guessed at, but it is probably stored in the handset’s RAM for some use during operation. The second “M” command turns that mode off. Finally, the top-level menu displays on lines 1 and 2.

### **Movement:**

Now here are some simple ones, just hitting the four direction buttons, N, S, E, W in that order, and then changing the slew speeds.

```
001 4E
    N
001 CE
    ..
001 53
    S
001 D3
    ..
001 45
    E
001 C5
    ..
001 57
    W
001 D7
    ..
001 34
    4
003 1B 4C 04
    .. L ..
001 B4
    ..
001 31
    1
003 1B 4C 02
    .. L ..
001 B1
    ..
001 30
    0
003 1B 4C 01
    .. L ..
001 B0
    ..
001 37
    7
```

```
003 1B 4C 08
    .. L ..
```

```
001 B7
    ..
```

The first 8 messages above are simple keypress messages sent to the cpu to let it know the user pressed that key. The first message is just the key that was pressed, “N”. That is actually only the down-press, when the key is released, an up-press message is sent, which is the same byte as the down-press, but with the first bit reversed. So, 4E on the press becomes CE on the release in the case of an “N” press. The same thing applies for the other three direction keys. Many key release messages are ignored by the CPU, but these four are very important: the down-press starts a motor, and the up-press stops it.

Next we get to the slew speed changes, which start with the orange 34 byte, which is the “4” character. The telescope is not expecting numeric data entry in its current state, so the secondary function of the “4” key applies, “FIND”; setting the slew speed to Find. Note what is returned from the CPU. An escape character, which tells us to pay attention for a command, then the command, 4C or “L”, then an 04 byte. This is telling the handset “activate the LED controlled by the third bit on the port that controls LEDs. (Hex 04 is binary 00000100).

Note one other thing about this message: The release of the “4” press comes after the cpu has already responded. That is typical, human fingers are slower than computers, even 1990 MCUs. Also note in this case the up-press is ignored by the cpu.

Next we just go through the other slew speeds until back at the fastest, the “7” key or 37. In each case the cpu tells the handset to set a different bit on the port that controls the LEDs. No more than one of these four is ever on at one time. Later, we’ll see that when the “ALT” LED is on, there are two on at the same time, so that there can be eight different bytes that are sent from the cpu depending on which of the four slew speed LEDs are on at the same time the ALT LED is on or off.

## Top Modes Rotation

*Starting in the first Mode, the Telescope/Object Library menu.*

```
001 4D
    M
041 1B 43 1B 31 20 52 41 20 20 3D 20 32 30 3A 32 37 3A 33 31 20 20 1B 32 20 44 45 43 20 3D 2D
    .. C .. 1 R A = 2 0 : 2 7 : 3 1 .. 2 D E C = -
    35 37 DF 33 35 3A 32 38 20 20 20
    5 7 .. 3 5 : 2 8
039 1B 31 20 52 41 20 20 3D 20 32 30 3A 32 37 3A 33 31 20 20 1B 32 20 44 45 43 20 3D 2D 35 37
    .. 1 R A = 2 0 : 2 7 : 3 1 .. 2 D E C = - 5 7
    DF 33 35 3A 32 38 20 20 20
    .. 3 5 : 2 8
001 CD
    ..
039 1B 31 20 52 41 20 20 3D 20 32 30 3A 32 37 3A 33 31 20 20 1B 32 20 44 45 43 20 3D 2D 35 37
    .. 1 R A = 2 0 : 2 7 : 3 1 .. 2 D E C = - 5 7
    DF 33 35 3A 32 38 20 20 20
    .. 3 5 : 2 8
```

```

039 1B 31 20 52 41 20 20 3D 20 32 30 3A 32 37 3A 33 31 20 20 1B 32 20 44 45 43 20 3D 2D 35 37
.. 1 R A = 2 0 : 2 7 : 3 1 " 2 D E C = - 5 7
DF 33 35 3A 32 39 20 20 20
" 3 5 : 2 9
001 4D
M
036 1B 31 7E 4C 4F 43 41 4C 3D 31 35 3A 30 34 3A 34 39 20 1B 32 20 53 49 44 45 20 3D 32 30 3A
.. 1 ~ L O C A L = 1 5 : 0 4 : 4 9 " 2 S I D E = 2 0 :
33 39 3A 31 32 20
3 9 : 1 2
001 CD
..
036 1B 31 7E 4C 4F 43 41 4C 3D 31 35 3A 30 34 3A 34 39 20 1B 32 20 53 49 44 45 20 3D 32 30 3A
.. 1 ~ L O C A L = 1 5 : 0 4 : 4 9 " 2 S I D E = 2 0 :
33 39 3A 31 32 20
3 9 : 1 2
036 1B 31 7E 4C 4F 43 41 4C 3D 31 35 3A 30 34 3A 34 39 20 1B 32 20 53 49 44 45 20 3D 32 30 3A
.. 1 ~ L O C A L = 1 5 : 0 4 : 4 9 " 2 S I D E = 2 0 :
33 39 3A 31 32 20
3 9 : 1 2
001 4D
M
038 1B 43 1B 31 7E 54 49 4D 45 52 3D 30 30 3A 30 30 3A 30 30 20 1B 32 20 46 52 45 51 20 3D 36
.. C " 1 ~ T I M E R = 0 0 : 0 0 : 0 0 " 2 F R E Q = 6
30 2E 31 20 20 20 51 20
0 . 1 Q
001 CD
..
001 4D
M
007 1B 43 1B 42 00 1B 43
.. C " B " " C
001 CD
..
001 4D
M
044 1B 42 64 1B 43 1B 31 20 54 45 4C 45 53 43 4F 50 45 20 20 20 20 20 20 1B 32 20 4F 42 4A 45
.. B d " C " 1 T E L E S C O P E " 2 O B J E
43 54 20 4C 49 42 52 41 52 59 20 1B 31 7E
C T L I B R A R Y " 1 ~
001 CD
..

```

Above, we have simply pressed the MODE key to rotate through the 5 top-level menus, or modes as they are called in the manual. The first mode shown is actually the second mode, since we started at the top mode and pressed the “MODE” button once at the very beginning of our conversation above. In the second and third mode, the telescope continuously sends display text every 100 msec. In the above conversation, we have truncated the repeating lines of display text which come from the telescope with the current RA and DEC in the second mode and Local and Sidereal time in the third. At the bottom, we’ve returned to the first mode that displays the Telescope and Object Library text.

You can see a few new commands from the telescope, such as the “C” command, which clears both lines of the display. Once again, the “1” and the “2” command (always coming after the “1B” escape, like all commands from the telescope) tell the handset what to print on the display. Another new command is the “B” command, which sets the display brightness. In mode 4, it is set to 0 and the display goes dark. In mode 1 it is set back to 64, which is bright. See Appendix A for protocol details on the individual commands.

## Set Time and Date

This sequence, near the end, will show how the “D” command is used to demark a time span where keypad presses are data being entered to set the time and date. When a D command has arrived with its data byte as 01, we enter a data entry mode until another D command comes with data byte 00, turning that mode off. During data entry mode, pressing a key like the “1” key tells the cpu that a “1” digit has been entered by the user. Otherwise, the cpu would interpret that key as the CNTR key, to change the slew speed.

```
001 4D
M
041 1B 43 1B 31 20 52 41 20 20 3D 20 31 36 3A 34 35 3A 32 39 20 20 1B 32 20 44 45 43 20 3D 2D
   " C " 1 R A = 1 6 : 4 5 : 2 9 " 2 D E C = -
   35 36 DF 31 30 3A 34 38 20 20 20
   5 6 " 1 0 : 4 8
001 CD
..
039 1B 31 20 52 41 20 20 3D 20 31 36 3A 34 35 3A 32 39 20 20 1B 32 20 44 45 43 20 3D 2D 35 36
   " 1 R A = 1 6 : 4 5 : 2 9 " 2 D E C = - 5 6
   DF 31 30 3A 34 38 20 20 20
   " 1 0 : 4 8
039 1B 31 20 52 41 20 20 3D 20 31 36 3A 34 35 3A 32 39 20 20 1B 32 20 44 45 43 20 3D 2D 35 36
   " 1 R A = 1 6 : 4 5 : 2 9 " 2 D E C = - 5 6
   DF 31 30 3A 34 38 20 20 20
   " 1 0 : 4 8
001 4D
M
038 1B 43 1B 31 7E 4C 4F 43 41 4C 3D 31 31 3A 31 37 3A 34 31 20 1B 32 20 53 49 44 45 20 3D 31 37
3A
   " C " 1 ~ L O C A L = 1 1 : 1 7 : 4 1 " 2 S I D E = 1
7 :
   30 37 3A 31 36 20
   0 7 : 1 6
036 1B 31 7E 4C 4F 43 41 4C 3D 31 31 3A 31 37 3A 34 31 20 1B 32 20 53 49 44 45 20 3D 31 37 3A
   " 1 ~ L O C A L = 1 1 : 1 7 : 4 1 " 2 S I D E = 1 7 :
   30 37 3A 31 36 20
   0 7 : 1 6
001 CD
..
036 1B 31 7E 4C 4F 43 41 4C 3D 31 31 3A 31 37 3A 34 31 20 1B 32 20 53 49 44 45 20 3D 31 37 3A
   " 1 ~ L O C A L = 1 1 : 1 7 : 4 1 " 2 S I D E = 1 7 :
   30 37 3A 31 36 20
   0 7 : 1 6
036 1B 31 7E 4C 4F 43 41 4C 3D 31 31 3A 31 37 3A 34 31 20 1B 32 20 53 49 44 45 20 3D 31 37 3A
   " 1 ~ L O C A L = 1 1 : 1 7 : 4 1 " 2 S I D E = 1 7 :
   30 37 3A 31 36 20
   0 7 : 1 6
001 0D
..
036 1B 31 7E 4C 4F 43 41 4C 3D 31 31 3A 31 37 3A 34 31 20 1B 32 20 53 49 44 45 20 3D 31 37 3A
   " 1 ~ L O C A L = 1 1 : 1 7 : 4 1 " 2 S I D E = 1 7 :
   30 37 3A 31 37 20
   0 7 : 1 7
027 1B 50 01 08 31 31 3A 31 37 3A 34 32 1B 50 01 08 1B 44 01 1B 4C 18 1B 53 02 02 03
   " P " " 1 1 : 1 7 : 4 2 " P " " " D " " L " " S " " "
001 8D
..
001 31
1
001 31
1
001 B1
..
```



```

001 31
    1
004 31 1B 49 01
    1 " I "
001 B1
    ..
001 31
    1
001 31
    1
001 B1
    ..
001 37
    7
004 37 1B 49 01
    7 " I "
001 B7
    ..
001 57
    W
006 1B 49 00 1B 49 00
    " I " " I "
001 D7
    ..
001 57
    W
003 1B 49 00
    " I "
001 D7
    ..
001 57
    W
006 1B 49 00 1B 49 00
    " I " " I "
001 D7
    ..
001 32
    2
004 32 1B 49 01
    2 " I "
001 B2
    ..
001 57
    W
006 1B 49 00 1B 49 00
    " I " " I "
001 D7
    ..
001 31
    1
004 31 1B 49 01
    1 " I "
001 B1
    ..
001 31
    1
001 31
    1
001 B1
    ..
001 37
    7
004 37 1B 49 01
    7 " I "

```

```

001 B7
..

001 45
E
003 1B 49 01
.. I ..
001 C5
..

001 32
2
004 32 1B 49 00
2 .. I ..
001 B2
..

001 0D
..

084 1B 44 00 1B 4C 08 1B 31 7E 4C 4F 43 41 4C 3D 31 31 3A 31 37 3A 34 32 20 1B 32 20 53 49 44
.. D .. L .. 1 ~ L O C A L = 1 1 : 1 7 : 4 2 .. 2 S I D
45 20 3D 31 37 3A 30 38 3A 31 30 20 1B 43 48 6F 75 72 73 20 66 72 6F 6D 20 47 4D 54 3A 1B
E = 1 7 : 0 8 : 1 0 .. C H o u r s f r o m G M T : ..
50 02 01 2B 30 38 1B 50 02 01 1B 49 01 1B 44 01 1B 4C 18 1B 53 02 02 03
P .. + 0 8 .. P .. .. I .. .. D .. .. L .. .. S .. ..
001 8D
..

001 30
0
001 30
0
001 B0
..

001 38
8
004 38 1B 49 00
8 .. I ..
001 B8
..

001 0D
..

042 1B 44 00 1B 4C 08 1B 31 7E 4C 4F 43 41 4C 3D 31 31 3A 31 37 3A 35 32 20 1B 32 20 53 49 44
.. D .. L .. 1 ~ L O C A L = 1 1 : 1 7 : 5 2 .. 2 S I D
45 20 3D 31 37 3A 30 38 3A 32 31 20
E = 1 7 : 0 8 : 2 1
036 1B 31 7E 4C 4F 43 41 4C 3D 31 31 3A 31 37 3A 35 32 20 1B 32 20 53 49 44 45 20 3D 31 37 3A
.. 1 ~ L O C A L = 1 1 : 1 7 : 5 2 .. 2 S I D E = 1 7 :
30 37 3A 32 32 20
0 7 : 2 2
001 8D
..

036 1B 31 7E 4C 4F 43 41 4C 3D 31 31 3A 31 37 3A 35 32 20 1B 32 20 53 49 44 45 20 3D 31 37 3A
.. 1 ~ L O C A L = 1 1 : 1 7 : 5 2 .. 2 S I D E = 1 7 :
30 37 3A 32 32 20
0 7 : 2 2
001 0D
..

036 1B 31 7E 4C 4F 43 41 4C 3D 31 31 3A 31 37 3A 35 39 20 1B 32 20 53 49 44 45 20 3D 31 37 3A
.. 1 ~ L O C A L = 1 1 : 1 7 : 5 9 .. 2 S I D E = 1 7 :
30 37 3A 32 39 20
0 7 : 2 9
001 8D
..

024 1B 53 01 01 00 1B 43 1B 31 20 44 41 54 45 20 3D 31 32 2F 31 36 2F 31 39
.. S .. .. C .. 1 D A T E = 1 2 / 1 6 / 1 9
017 1B 31 20 44 41 54 45 20 3D 31 32 2F 31 36 2F 31 39
.. 1 D A T E = 1 2 / 1 6 / 1 9

```

```

001 0D
..

017 1B 31 20 44 41 54 45 20 3D 31 32 2F 31 36 2F 31 39
.. 1 D A T E = 1 2 / 1 6 / 1 9
017 1B 31 20 44 41 54 45 20 3D 31 32 2F 31 36 2F 31 39
.. 1 D A T E = 1 2 / 1 6 / 1 9
027 1B 50 01 08 31 32 2F 31 36 2F 31 39 1B 50 01 08 1B 44 01 1B 4C 18 1B 53 02 02 03
.. P " " 1 2 / 1 6 / 1 9 " P " " " D " " L " " S " " "
001 8D
..

001 31
1
001 31
1
001 B1
..

001 32
2
004 32 1B 49 01
2 " I "
001 B2
..

001 45
E
003 1B 49 01
" I "
001 C5
..

001 45
E
006 1B 49 01 1B 49 01
" I " " I "
001 C5
..

001 31
1

001 31
1
001 B1
..

001 39
9
004 39 1B 49 00
9 " I "
001 B9
..

001 0D
..

030 1B 44 00 1B 43 55 70 64 61 74 69 6E 67 1B 32 70 6C 61 6E 65 74 61 72 79 20 64 61 74 61 2E
.. D " " C U p d a t i n g " 2 p l a n e t a r y d a t a .
001 8D
..

025 1B 43 1B 44 00 1B 4C 08 1B 31 20 44 41 54 45 20 3D 31 32 2F 31 36 2F 31 39
.. C " D " " L " " 1 D A T E = 1 2 / 1 6 / 1 9
017 1B 31 20 44 41 54 45 20 3D 31 32 2F 31 36 2F 31 39
.. 1 D A T E = 1 2 / 1 6 / 1 9
017 1B 31 20 44 41 54 45 20 3D 31 32 2F 31 36 2F 31 39
.. 1 D A T E = 1 2 / 1 6 / 1 9
001 4D
M
038 1B 43 1B 31 7E 54 49 4D 45 52 3D 30 30 3A 30 30 3A 30 30 20 1B 32 20 46 52 45 51 20 3D 36
.. C " 1 ~ T I M E R = 0 0 : 0 0 : 0 0 " 2 F R E Q = 6
30 2E 31 20 20 20 51 20

```

```

0 . 1 Q
001 CD
..
001 4D
M
007 1B 43 1B 42 00 1B 43
.. C " B " " C
001 CD
..
001 4D
M
044 1B 42 64 1B 43 1B 31 20 54 45 4C 45 53 43 4F 50 45 20 20 20 20 20 20 1B 32 20 4F 42 4A 45
.. B d " C " 1 T E L E S C O P E " 2 O B J E
43 54 20 4C 49 42 52 41 52 59 20 1B 31 7E
C T L I B R A R Y " 1 ~
001 CD
..

```

That large sequence of messages includes the keypad presses to enter the hour-minute-seconds of the time, the timezone offset, and the month-day-year of the date, and ends with pressing the MODE button twice to get back to the top menu. It also includes a few clicks on the “W” and “E” button, which lets you move the cursor for the digit you are entering. Note that, like the number keys, the “E” and “W” keys are not interpreted as motor movement commands because they are contained inside the two “D” commands and so are in data entry mode, where they act to move the cursor right and left.

The repeating lines of RA/DEC and LOCAL/SIDE time have been pared down as usual.

You’ll see several new commands here, which are detailed in Appendix A, but something to be aware of is that the numeric entry is modal, (the “D” command) and that when it is “on”, the ALT LED on the handset is lit. The ENTER key is 0D on press, 8D on release, and it is important because that tells the telescope that the user wants to enter data. Then the cpu responds with the “D” command to start data entry. A long press on the ENTER key is often expected for this, and knowing when the key release happens is necessary for the cpu to distinguish between a long and a short press of the key.

## Appendix A - LX200 Commands

The following commands are those sent from the CPU to the handset in software version 3 and later. The handset does not send any “commands” as such to the CPU, only keypresses the user has made. That doesn’t mean the handset is not involved though. It must decode all of the commands from the CPU, control the display, control the backlight and LEDs, keypad, etc. Where names have been given in parenthesis in the command heading, it is only a guess. Only Meade knows why the command characters were chosen as they were.

All items in angle brackets are placeholders for required or optional bytes, which can be numbers from 0 to 255 (0x00 to 0xFF), with that number representing either numerical quantities, on-off flags (0x00 is off), or text characters.

---

### **C** (*Clear*)

`0x1B 0x43 <optional1> <optional2> ... <optional16>`

Clears both lines of the display. Up to 16 characters of text can optionally follow, and will start at row 1 column 1.

---

### **1**

`0x1B 0x31 <c1> <c2> ... <c16>`

Write characters on row 1, beginning at column 1. There can be up to 16 characters, and any less will leave the display as it was at the columns not given.

---

### **2**

`0x1B 0x32 <c1> <c2> ... <c16>`

Write characters on row 2, beginning at column 1. There can be up to 16 characters, and any less will leave the display as it was at the columns not given.

---

### **B** (*Backlight*)

`0x1B 0x42 <n1>`

Brightens or dims the display backlight, the 4 backlight LEDs, and the 4 slew speed LEDs (only one of which is on). Larger numbers drive the LEDs brighter, with 0x00 turning them off completely.

---

## L (LEDs)

0x1B 0x4C <n1>

The number following is a set of bit flags, with the 5 LSbs controlling the 5 LEDs as follows:

0x01 GUIDE LED  
0x02 CNTR LED  
0x04 FIND LED  
0x08 SLEW LED  
0x10 ALT LED

---

## P (Prompt)

0x1B 0x50 <n1> <n2>

Followed by two hex numbers..e. 0x01 and 0x08, for row and column. Shows the blinking cursor at the designated row and column, and sets that row/col as the current position.

---

## D (Data)

0x1B 0x44 <flag>

The flag can be 0x00 (off), or 0x01 (on). When turned on, keypad input is accepted, rather than the function inputs for those keys. The cpu will echo back the numbers for display. The CPU will also make the cursor skip over any punctuation delimiters that are not supposed to be edited, by sending the "I" (increment) command. For example, when editing dates the forward slash character is skipped. It will also keep the cursor from advancing after the last input digit. These are always fixed-position inputs, so there is always a last digit to be expected.

---

## S (Sound)

0x1B 0x53 <n1> <n2> <n3>

Followed by three numbers. n1=number of beeps, n2=duration of beep, n3=duration of spacing between beeps. Volume is constant.

---

## I *(Increment)*

`0x1b 0x49 <n1>`

The number n1 can be 0x00 (meaning reverse), or 0x01 (meaning forward). This command will advance the cursor and position by one when followed by "forward", and reverse the cursor and current position by one when with "reverse". Advancing the cursor is done to skip over a display position. Note that entering a digit key automatically moves the position forward to the next character, so sending an 0x01 with this command advances again, skipping whatever was next after the number just entered.