JLCooper Eclipse MXL2 and Eclipse MXL Developer Manual

The Eclipse MXL2, MXL and MCS-3000XL are configured to operate as a stand-alone controller.

The Eclipse MXL2, MXL and MCS-3000XL all share the same protocol, the only difference is that the Eclipse MXL2 adds a row of encoder knobs along the top edge of the unit. The commands and responses documented in this manual to refer to all of these units except when referring specifically to the additional features of the Eclipse MXL2.

Unit Communications

Typically, the unit has an interface card plugged into its slot. This may be an RS-232, RS-422, USB, or Ethernet interface card. Some units such as the MCS-3000XL have a built-in MIDI port. The Eclipse MXL2 has the USB and Ethernet interfaces integrated into the unit. The computer attached via this interface is called the Host.

> *NOTE: Though unlabeled on the unit, the LCD buttons are considered (for the following tables) to be the SELECT switches. Unless a LCD color command is received, a SELECT LED On command with cause the LCD to be Red, and an LCD LED Off command will cause it to turn Green.*

Communications Settings

The unit communicates at 31250 bits/second for MIDI and 38400 bits/second for RS-232, RS-422 and USB. The data format is 1 start bit, 8 data bits 1 stop bit and no parity.

DIP Switch Settings for Protocol Selection

The Eclipse MXL and MXL2 are preconfigured for standalone operation. The MCS-3000XL has DIP switches on the rear panel to configure the mode of operation. The MCS-3000XL must be placed in Developer Mode.

Note that the switches are mounted upside down. Switch #4 is the right most switch, when viewed from the rear. Up = off.

#4	#3	#2	#1	Mode
on	off	off	off	Developer Mode

Note: Other switch setting combinations are not used.

JLCooper Eclipse MXL2 and Eclipse MXL Developer Manual

Developer Mode

SW4 = on, SW3 = off, SW2 = off, SW1 = off

	1	2	3	4	5	6	7	8
Select	Sel 1	Sel 2	Sel 3	Sel 4	Sel 5	Sel 6	Sel 7	Sel 8
send	**B0 49 vv**	**B1 49 vv**	**B2 49 vv**	**B3 49 vv**	**B4 49 vv**	**B5 49 vv**	**B6 49 vv**	**B7 49 vv**
receive	*See LCD Buttons section*							
Aux	Aux 1	Aux 2	Aux 3	Aux 4	Aux 5	Aux 6	Aux 7	Aux 8
send	**B0 48 vv**	**B1 48 vv**	**B2 48 vv**	**B3 48 vv**	**B4 48 vv**	**B5 48 vv**	**B6 48 vv**	**B7 48 vv**
receive	same	same	same	same	same	same	same	same
Solo	Solo 1	Solo 2	Solo 3	Solo 4	Solo 5	Solo 6	Solo 7	Solo 8
send	**B0 47 vv**	**B1 47 vv**	**B2 47 vv**	**B3 47 vv**	**B4 47 vv**	**B5 47 vv**	**B6 47 vv**	**B7 47 vv**
receive	same	same	same	same	same	same	same	same
Mute	Mute 1	Mute 2	Mute 3	Mute 4	Mute 5	Mute 6	Mute 7	Mute 8
send	**B0 46 vv**	**B1 46 vv**	**B2 46 vv**	**B3 46 vv**	**B4 46 vv**	**B5 46 vv**	**B6 46 vv**	**B7 46 vv**
receive	same	same	same	same	same	same	same	same
Fader	Fader 1	Fader 2	Fader 3	Fader 4	Fader 5	Fader 6	Fader 7	Fader 8
send	**B0 07 vv 27 zz**	**B1 07 vv 27 zz**	**B2 07 vv 27 zz**	**B3 07 vv 27 zz**	**B4 07 vv 27 zz**	**B5 07 vv 27 zz**	**B6 07 vv 27 zz**	**B7 07 vv 27 zz**
receive	same	same	same	same	same	same	same	same
Touch	Touch 1	Touch 2	Touch 3	Touch 4	Touch 5	Touch 6	Touch 7	Touch 8
send	**B0 4A vv**	**B1 4A vv**	**B2 4A vv**	**B3 4A vv**	**B4 4A vv**	**B5 4A vv**	**B6 4A vv**	**B7 4A vv**
Enc Push	Encoder 1	Encoder 2	Encoder 3	Encoder 4	Encoder 5	Encoder 6	Encoder 7	Encoder 8
Send	**B0 4B vv**	**B1 4B vv**	**B2 4B vv**	**B3 4B vv**	**B4 4B vv**	**B5 4B vv**	**B6 4B vv**	**B7 4B vv**
Enc Turn	Encoder 1	Encoder 2	Encoder 3	Encoder 4	Encoder 5	Encoder 6	Encoder 7	Encoder 8
Send	**B0 4C vv**	**B1 4C vv**	**B2 4C vv**	**B3 4C vv**	**B4 4C vv**	**B5 4C vv**	**B6 4C vv**	**B7 4C vv**

Notes:

Faders – The fader value is an 8-bit value. vv is upper 7 bits of fader value and bit 6 of zz least significant bit of fader value.
Buttons and Encoder Push – The unit will send vv = 7Fh when a button is pressed and vv = 00h when a button is released.
LEDs - The unit will illuminate LED when vv = 7Fh and extinguish LED when vv = 00h
Touch - The unit will send vv = 7Fh when a fader is touched and vv = 00h when a fader is released.
Encoder Turn – The encoder value is a 7 bit two's compliment number. Turning the encoder clockwise yields positive values and turning the encoder counterclockwise yields negative values.

In addition, when the unit receives F0h 15h 26h 01h F7h, the unit returns F0h 15h 26h 01h revH revL F7h in reply. Where revH is the integer portion of the firmware version and revL is the fractional portion.

LCD Buttons

Command Structure

These messages are not affected by the Special Mode condition. Messages to the unit's LCD Buttons are in form:

F0h 15h 26h 04h <00uuunnn> <cc> <data> F7h

uuu is the unit number (for multiple units attached to a master.
nnn is the LCD button number (00 to 07),
cc is the command byte.
data are the parameters for the command. The length of this is variable.

Set Backlight Color (6 bit)

This sets the backlight colors for a given LCD button. The unit flashes between the On and Off backlight color at a rate of 1 Blink/Second. If no blinking is desired, set the On and Off colors identical.

This command requires two data bytes. The first is the "On" color; the second is the "Off" color. Six data bits define the backlight color as follows:

00rrggbb

The **rr** bits set the red intensity, the **gg** bits set the green intensity and, the **bb** bits set the blue intensity. A value of **00** is the minimum intensity of a given color while a value of 11 is the maximum intensity. This allows for 6 bits of color resolution or 64 unique colors to be specified for the LCD button backlight.

F0h 15h 26h 04h <00uuunnn> 00h <00rrggbb>$_{on}$ <00rrggbb>$_{off}$ F7h

*Note: Because the unit does not have a Select LED like the MCS-3000x, it has a compatibility feature to emulate the Select LED. It emulates the MCS-3000x by changing the LCD button backlight color. Sending the Select LED On message will cause the unit to select the On color. The default On color is green. Sending Select LED Off message will cause the unit to select the Off color. The default Off color is white. Either Select message will suspend the automatic flashing between the On and Off color. Sending the Set Color Command will enable automatic flashing between the On and Off color for **ALL** LCD buttons. The Set Color Command will overwrite the default Select colors.*

Set Backlight Color (21 bit)

Note: In the Eclipse MXL / MCS-3000XL, this feature is present in v1.26 and later firmware. In the Eclipse MXL2, this feature is present in v1.xx and later firmware.

This sets the backlight colors for a given LCD button. The unit flashes between the On and Off backlight color at a rate of 1 Blink/Second. If no blinking is desired, set the On and Off colors identical.

This command requires six data bytes. The first three bytes are the "On" color; the second three bytes are the "Off" color. 21 data bits define the backlight color as follows:

0rrrrrrr 0ggggggg 0bbbbbbb

The **rrrrrrr** bits set the red intensity, the **ggggggg** bits set the green intensity and, the **bbbbbbb** bits set the blue intensity. A value of **0000000** is the minimum intensity of a given color while a value of **11111** is the maximum intensity. This allows for 21 bits of color resolution or 2^{21} (2,097,152) unique colors to be specified for the LCD button backlight.

F0h 15h 26h 04h <00uuunnn> 08h

<0rrrrrrr>$_{on}$ <0ggggggg>$_{on}$ <0bbbbbbb>$_{on}$

<0rrrrrrr>$_{off}$ <0ggggggg>$_{off}$ <0bbbbbbb>$_{off}$

F7h

Note: Sending this command places the unit in 21-bit color mode. After that, all future color commands must be 21-bit color. Six-bit color commands are no longer valid.

Set Screen Definition

This sets the basic display operation for a given LCD button. This command is as follows:

F0h 15h 26h 04h <00uuunnn> 01h <data> F7h

uuu is the unit number (for multiple units attached to a master).
nnn is the LCD button number (**0** to **7**).
data is the parameter that select the operational mode of the LCD button, defined below.

data = **00h**, **01h** or **02h**. These set the LCD button to an alphanumeric character mode. The data defining the alphanumeric text is loaded using the Set Screen Data command on the following page.

data = **00h**, a screen with four lines of 10 characters each. A 5x7 pixel character is used.

data = **01h**, a screen with three lines of 8 characters each. A 7x9 pixel character is used.

data = **02h**, a screen with two lines of 7 characters each. An 8x15 pixel character is used.

data = **20h**, sets the LCD button to display one of 32 possible RAM-based 64x32 pixel icons. Use the Set Screen Data command to select the actual icon to display.

data = **40h**, sets the LCD button to display one of 64 possible ROM-based 64x32 pixel icons. Use the Set Screen Data command to select the actual icon to display. If custom icons are desired, use RAM-based icons.

data = **41h**, sets the LCD to display a full-screen ASCII character (**20h** to **7Fh**). Use the Set Screen Data command to select the actual character to display.

Set Screen Data

This supplies data needed to write standard alphanumeric characters, full screen alphanumeric characters, RAM Icons or ROM Icons on the LCD buttons.

Note: The Set Screen Definition command then any Set Screen Data commands must be written to display standard alphanumeric characters, RAM Icons, ROM Icons or full screen alphanumeric characters.

Standard Alphanumeric Characters

For standard alphanumeric characters, each line must be written individually. In all cases, the first byte (`ln`) specifies the line number and the centering enable/disable. There are a variable number of bytes, depending on the font selected.

1. The 5x7 font allows for 4 lines of text with up to 10 characters per line.
   There are 11 bytes of data, `ln` and 10 bytes of text. Unused positions must be padded with spaces (20h).

F0 15 26 04 <00uuunnn> 03 <ln> <d_0> <d_1> <d_2> <d_3> <d_4> <d_5> <d_6> <d_7> <d_8> <d_9> F7

2. The 7x9 font allows for 3 lines of text with up to 8 characters per line.
   There are 9 bytes of data, `ln` and 8 bytes of text. Unused positions must be padded with spaces (20h).

F0 15 26 04 <00uuunnn> 03 <ln> <d_0> <d_1> <d_2> <d_3> <d_4> <d_5> <d_6> <d_7> F7

3. The 8x15 font allows for 2 lines of text with up to 7 characters per line.
   There are 8 bytes of data, `ln` and 7 bytes of text. Unused positions must be padded with spaces (20h).

F0 15 26 04 <00uuunnn> 03 <ln> <d_0> <d_1> <d_2> <d_3> <d_4> <d_5> <d_6> F7

Note: Only one font size can be used on an LCD button at a time.

`ln` specifies line numbering and centering in the following format:

ln = 000000nn

nn is the line to display text on.
- **nn** = **00** specifies the top line
- **nn** = **01** specifies the second line.
- **nn** = **10** specifies the third line. *Note: Not valid for 8x15 font.*
- **nn** = **11** specifies the fourth line. *Note: Not valid for 8x15 and 7x9 fonts.*

Note: Before sending another command, wait a small period (~40mS) after writing screen data to allow the unit to complete the task of rendering and writing the pixel data to the LCD. This is important, as there are a large number of pixels to process. Alternatively, the LCD Button Command Acknowledgement mechanism can be used to guarantee that the command is done processing before the next command is sent.

Writing all text at once

Note: In the Eclipse MXL / MCS-3000XL, this feature is present in v1.22 and later firmware. In the Eclipse MXL2, this feature is present in v1.00 and later firmware.

In any text mode, all the lines of text can be written with one command.

This method uses the standard Set Screen Data command except instead of writing each line with a separate Set Screen Data command, all lines are written with one Set Screen Data command. To use the Set Screen Data command to write all the lines of text on a screen, use the following command:

F0 15 26 04 <button> 03 00 <chars> F7

This command is the same as the standard Set Screen Data command except:

1. The command MUST specify the top line (**00** hex) and,
2. The number of characters MUST be a full screen of text.

Note: In 2 line x 7 character mode, ASCII data must be exactly 14 bytes. Unused bytes must be padded with spaces (20 hex). In 3 line x 8 character mode, ASCII data must be exactly 24 bytes. Unused bytes must be padded with spaces (20 hex). In 4 line x 10 character mode, ASCII data must be exactly 40 bytes. Unused bytes must be padded with spaces (20 hex).

Alphanumeric data

For ASCII characters, a value of **20h** to **7Fh** is used in the data string, with the first data byte for the left display position. Values of **00h** to **1Fh** will bring up a RAM based, user-defined sprite. Sprite information is loaded using the Set Sprite Data command. This is not currently implemented in the unit.

Note: The Set Screen Definition command then any Set Screen Data commands must be written to display standard alphanumeric characters.

Note: Before sending another command, wait a small period (~40mS) after writing screen data to allow the unit to complete the task of rendering and writing the pixel data to the LCD. This is important, as there are a large number of pixels to process. Alternatively, the LCD Button Command Acknowledgement mechanism can be used to guarantee that the command is done processing before the next command is sent.

Full Screen Alphanumeric Characters, RAM Icons and ROM Icons

Icon data

When the Set Screen Definition command has set an LCD display to a RAM Icon, ROM Icon or full screen alphanumeric character, one byte (**d_0**) of the Set Screen Data command specifies the icon or character to be displayed. `ln` is not used and should be padded with zero.

F0h 15h 26h 04h <00uuunnn> 03h <ln> <d_0> F7h

Note: The Set Screen Definition command then the Set Screen Data command must be written to display a RAM Icon, ROM Icon or full screen alphanumeric character.

Example: If it were desired to set LCD #2 to display ROM Icon #6, the following strings would be sent:

F0 15 26 04 01 01 40 F7	Makes LCD #2 display a ROM Icon
F0 15 26 04 01 03 00 05 F7	Sets Icon #6 in data

Note: ROM Icons are listed at the end of this document.

Or, if it is desired to set LCD #2 to display a large letter "J",

F0 15 26 04 01 01 41 F7	Makes LCD #2 display a large letter
F0 15 26 04 01 03 00 4A F7	Sets “J” in data

Note: Before sending another command, wait a small period (~40mS) after writing screen data to allow the unit to complete the task of rendering and writing the pixel data to the LCD. This is important, as there are a large number of pixels to process. Alternatively, the LCD Button Command Acknowledgement mechanism can be used to guarantee that the command is done processing before the next command is sent.

Set Sprite Data

Contact factory for this protocol.

Clear RAM Icon Data

This is a one data byte command that clears out all pixel data (Blank screen) from one of 32 possible RAM-based screen images (Icon). The <data> byte = **00h** to **1Fh**.

F0h 15h 26h 04h <00uuunnn> 05h <data> F7h

uuu is not used and should be padded with **00h**.
nnn is not used and should be padded with **00h**.
data is the RAM icon to clear (**00h** to **1Fh**).

Note: Before sending another command, wait a small period (~20mS) after writing Clear RAM Icon Data command to allow the unit to complete the task of clearing the pixel data. This is important, as there are a large number of pixels to process.

Set RAM Icon Data

This is a variable length data string, which defines or modifies the pixel contents of one of 32 possible RAM-based screen images. The command has the form:

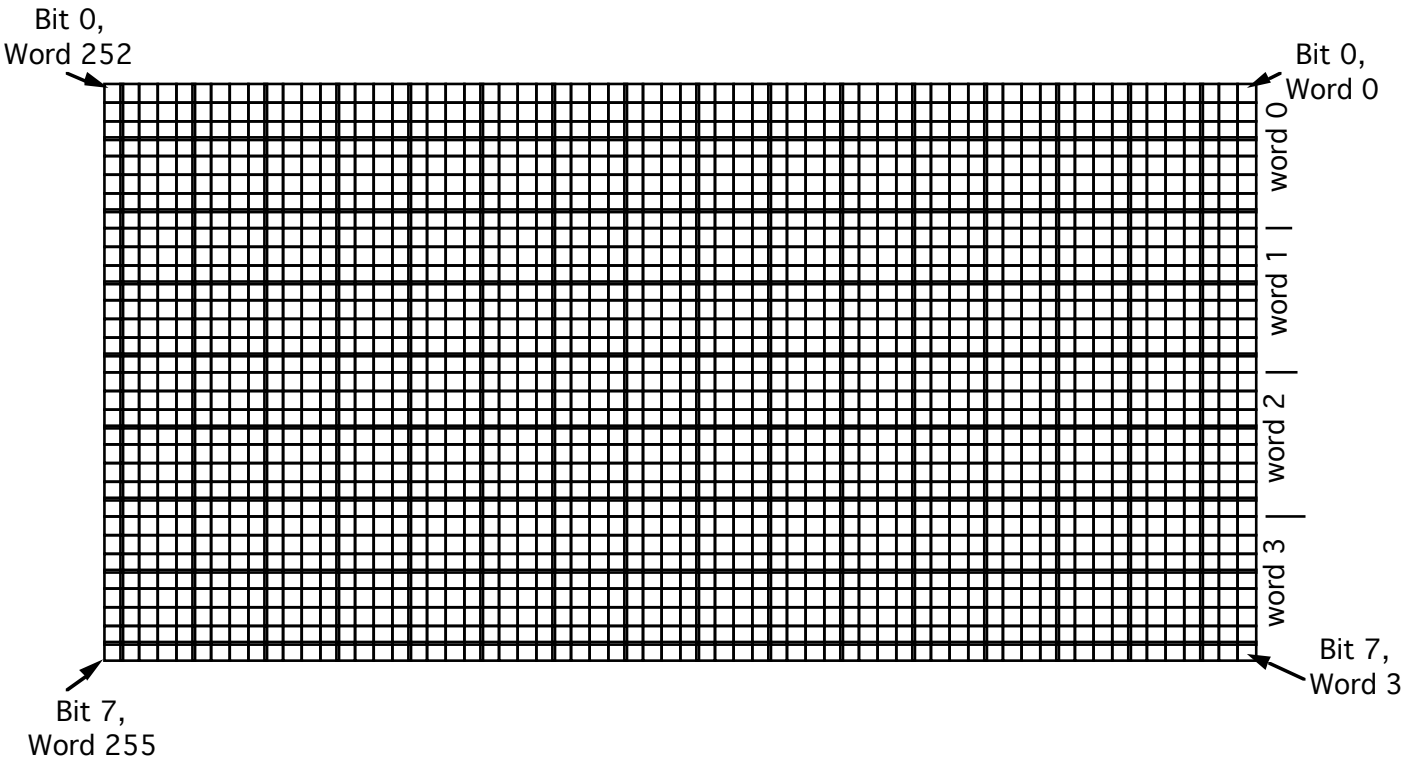
F0h 15h 26h 04h <00uuunnn> 06h <in> <op> <st_{low}> <st_{hi}> <ln_{low}> <ln_{hi}> <pixels> F7h

uuu	is not used and should be padded with 0.
nnn	is not used and should be padded with 0.
in	is the icon number, **00h** to **1Fh**.
op	is the operation (00 = "OR", 01 = "AND", 02 = "XOR" with existing data).
st_{low}, st_{hi}	are the starting address (as two nibbles) into the icon's memory block.
ln_{low}, ln_{hi}	are the number of bytes of pixel data being sent (as two nibbles).
pixels	are the actual pixel bitmap. The upper right-most pixel is the low bit of the 0 byte as shown below. All pixel data are nibblized in sending, with low nibble first. The length specified by **ln_{low}** and **ln_{hi}** are the number of bytes, not nibbles, being sent.

Note: If there are a full screen worth of pixels (256 bytes) to send, set ln_{low} = 00h *and* ln_{hi} = 00h.

Note: Pixel data is not retained when the power is off. Pixel data will be random after power on and therefore RAM Icons should be cleared and set after power on.

Note: Before sending another command, wait a small period (~20mS) after writing Set RAM Icon Data command to allow the unit to complete the task of clearing the pixel data. This is important, as there are a large number of pixels to process.



ROM Based Icons

Up to 64 Icons are stored within the unit. The factory-supplied Icons are as follows:

#0 Scissors
#1 Undo
#2 Eye Dropper
#3 Eraser
#4 Zoom In
#5 Zoom Out
#6 Paint Bucket
#7 Paint Brush
#8 Pencil
#9 Hand
#10 Piece of Film
#11 2 Pieces of Film
#12 Insert Film
#13 Delete Film
#14 +1 Frame
#15 -1 Frame
#16 Slate
#17 Phone On Hook
#18 Phone Off Hook
#19 Phone Handset
#20 Speaker
#21 Notes on staff
#22 More Notes
#23 Tempo Indication
#24 Sine Wave
#25 Happy Face
#26 Sad Face
#27 Gun
#28 Open Folder
#29 FM Station Logo
#30 Up Arrow
#31 Down Arrow
#32 Jog Wheel
#33 Key
#34 Return to Zero
#35 Rewind
#36 Fast Forward
#37 Stop
#38 Play
#39 Record

ID request and Power Up Status

To request id and power-up status, send the following command:

F0h 15h 26h 01h F7h

The unit will reply with the following response:
(The ID message is also sent automatically a few seconds after power up.)

F0h 15h 26h 01h revH revL F7h

revH is the integer portion of the firmware rev number,
revL is the fractional portion of the firmware rev number,

For example, if the unit replies with **F0h 15h 26h 01h 01h 0Ah F7h**, then the firmware version is 1.10.

Status Request

Note: In the Eclipse MXL / MCS-3000XL, this feature is present in v1.20 and later firmware. In the Eclipse MXL2, this feature is present in v1.00 and later firmware.

To request the current fader positions and LED states, send the following command:

F0h 15h 26h 02h F7h

The unit will reply with the following response:

F0h 15h 26h 02h

$fader_1$ $fader_2$ $fader_3$ $fader_4$ $fader_5$ $fader_6$ $fader_7$ $fader_8$

$mute_L$ $mute_H$ $solo_L$ $solo_H$ aux_L aux_H

on_1 on_2 on_3 on_4 on_5 on_6 on_7 on_8

off_1 off_2 off_3 off_4 off_5 off_6 off_7 off_8

F7h

$fader_x$	is the current 7 bit position of the fader
$mute_x$	is a bitmap of the current state of the Mute LEDs
$solo_x$	is a bitmap of the current state of the Solo LEDs
aux_x	is a bitmap of the current state of the Aux LEDs
on_x	is the current 'On' color value of the LCD backlight
off_x	is the current 'Off' color value of the LCD backlight

0	0	0	0	Mute 4	Mute 3	Mute 2	Mute 1

muteL Bitmap

0	0	0	0	Mute 8	Mute 7	Mute 6	Mute 5

muteH Bitmap

0	0	0	0	Solo 4	Solo 3	Solo 2	Solo 1

soloL Bitmap

0	0	0	0	Solo 8	Solo 7	Solo 6	Solo 5

soloH Bitmap

0	0	0	0	Aux 4	Aux 3	Aux 2	Aux 1

auxL Bitmap

0	0	0	0	Aux 8	Aux 7	Aux 6	Aux 5

auxH Bitmap

Note: The LCD Switch 'On' and 'Off' colors are not valid in 21-bit color mode.

LCD Button Command Acknowledgement

Note: In the Eclipse MXL / MCS-3000XL, this feature is present in v1.22 and later firmware. In the Eclipse MXL2, this feature is present in v1.00 and later firmware.

Because text and graphic commands to the LCD buttons can require a significant amount of time to complete, a simple feedback mechanism has been added to notify the host application that the unit is ready to receive another text or graphic command for the LCD buttons.

To use this feedback mechanism, the feature must first be enabled. The command to enable or disable this mechanism is as follows:

F0h 15h 26h 04h 00h 07h <data> F7h

data is the enable or disable value.
A value of 0 will disable this feedback mechanism.
A value of 1 will enable this feedback mechanism.

To enable the feedback mechanism, send the following command:

F0h 15h 26h 04h 00h 07h 01h F7h

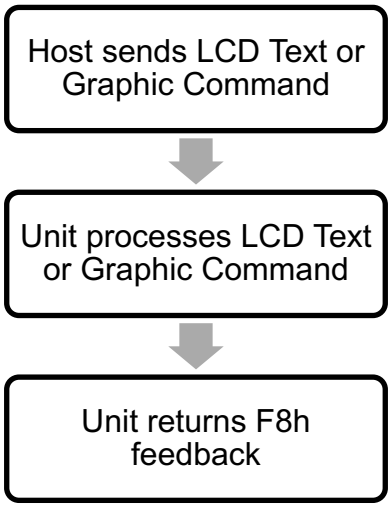
To disable the feedback mechanism, send the following command:

F0h 15h 26h 04h 00h 07h 00h F7h

When the feedback mechanism is enabled, any text or graphic command to the LCD buttons triggers the unit to return the following message when the unit has completed receiving, rendering and displaying the requested data to the LCD buttons.

F8h

The following flowchart illustrates the use of this feedback mechanism.



Fader Message Behavior

Note: In the Eclipse MXL / MCS-3000XL, this feature is present in v1.24 and later firmware. In the Eclipse MXL2, this feature is present in v1.00 and later firmware.

By default, the unit sends fader position messages regardless of the state of the Fader Touch status.

To change the behavior of the fader, use the following command:

F0h 15h 26h 05h <data> F7h

data is the enable or disable value.
A value of 0 will allow the unit to send fader position messages regardless of the touch sensor.
A value of 1 will require that the fader knob be touched before the fader values are sent.

To configure the unit to send fader position messages regardless if the fader knob is touched, send the following command:

F0h 15h 26h 05h 00h F7h

To configure the unit to send fader position messages only if the fader knob is touched, send the following command:

F0h 15h 26h 05h 01h F7h