our reference
Reference Chart 3.04

concerning
Network and midi commands to control the LCX

date
December 2012

LanBox LCX
Reference Chart
Version 3.04

V3.02:
Added CommonSaveData command
Minor corrections

V3.03:
Added section MIDI Program Change messages
Minor corrections

V3.04:
Expanded the Serial commands through MIDI section

LanBox Products
CDS advanced technology bv
Thames 9
3144 DG Maassluis
THE NETHERLANDS

+31 (0) 85 877 1276
info@lanbox.com
www.lanbox.com

BANK 340479876
IBAN NL31RABO0340479876
BIC RABONL2U

C. of C./K.v.K. 24264708
VAT/BTW NL804471666B01

Overview

General

This document is not a manual for the LanBox or LCedit+. This is document is aimed to be a reference for those who wish to control the LanBox through their own software and/or hardware. In order to use this document, general knowledge on how the LanBox works is required.

The LanBox programming package LCedit+ uses the same command set as described in this document, so knowing how to work with LCedit+ software can be of great help in understanding the commands. For every command we will mention the LCedit+ counterpart so you can cross reference with it. This Documentation is specifically written for the LanBox-LCX, firmware version 3.01. Although most of the commands will work with lower firmware versions as well as with the LCE and LCM and in lesser extent with the LC and LC+, this cannot be guaranteed.

Control methods

The LanBox can be controlled in different ways. Controlling all of the LanBox settings, editing, outputs and programming can be done through a TCP/IP or serial connection (these are the methods generally used by LCedit+), but these are not the only methods. If all you want to do is set Values of Mixer channels, UDP broadcasting might be the preferred way, while all commands are also available in MIDI CC messages and SysEx commands which will sometimes be the easiest way to have direct playback control.

All these communication options have a different command set which will all be covered in this Reference document. The document is divided into three main sections:

- Serial & Network Commands
- MIDI Notes
- MIDI Control Change Commands
- MIDI Show Control
- UDP Data

8bit and 16bit mode

The LanBox-LC+ only had 8bit possibilities. To keep soft- and hardware backwards compatible, every LanBox initializes data connections in 8bit operation mode. However, this means you can only use 250 Mixer channels. In order to use the LanBox-LCX, LCE or LCM to their full extend, the first thing you will have to do is to set the LanBox in 16bit operation mode. This is done with the **Common16BitMode** command.

This reference document assumes that you are working in 16bit Mode! If you are working with either a (8bit) LanBox LC/LC+, or with a LCE/LCE/LCX in 8bit Mode, be aware that all Values designating Channel Numbers will have to be 8bit instead of 16bit Values. Further, the LanBox LC and LC+ only had 8 fixed Layers, A – H.

Notation

All commands (and replies) are represented as Hexadecimal Values. These are not the actual data sent! These will always be the ASCII Values of the hexadecimal Values.

For example:

When we write:* C9 01 07 DF 7E #
The actual series of bytes sent is: 42 67 57 48 49 48 55 68 70 55 69 35
(note that we use spaces in the notation for readability, but these are obviously not sent along)

The colours have the following meaning:

RED:	Delimiter characters
BLUE:	Command numbers
MAGENTA:	Parameters and information bytes

Serial & Network Commands

Overview

In order to communicate with the LanBox, first a connection has to be established, either a serial (USB) connection or an TCP/IP connection. How this is done exactly is depends heavily on the software you are using, please refer to your software's documentation on how to establish such a connection.
The protocol will use only ASCII characters (no binary data). Each Value is represented in Hexadecimal format formed by two (8bit) or four (16bit) ASCII characters. Each message consists of a start character, a number of Values and an end character. Any deviation from this is considered a transmission error and the message will be discarded.

Network Connection

In order to control the LanBox through TCP/IP, a connection has to be made with the LanBox. By default the LanBox's TCP/IP settings are as follows:

IP Address	192.168.1.77
Subnet mask	255.255.255.0
Default gateway	192.168.1.1
Port	777

When a connection is first established, the LanBox will ask for a password. The default password is **777** and a Carriage return (ASCII 13). This is sent in ASCII values as well, so **55 55 55 13** is what is actually sent. Besides the password, a carriage return will not be necessary at the end of a command.

Serial Connection

In order to communicate with the LanBox through a serial USB connection, connect with the relevant serial port. On Mac OS and Unix systems, no extra driver is generally required. Windows does require a driver, this can be downloaded on the LanBox website and is included in the LCedit+ package.

Protocol

Each message starts with the 'start-of-message' delimiter character ***** (ASCII 42) and ends with an 'end-of-message' delimiter character **#** (ASCII 35). Reception of the start-of-message character will always result in resetting the input buffer and disposing of any received characters. Reception of the end-of-message character will start the interpretation of the message, execution of the command and reset the input buffer. The 2 characters (or is some cases 4 characters) following the start-of-message character form the command number. Each character is the hexadecimal representation of one nibble of the 8bit command number. For example command 1 would be represented as 01, command 10 as 0A, command 210 as D2, etc. The command number can be followed by any number of parameters. Each parameter can be a 8bit Value (2 ASCII characters) or a 16bit Value (4 ASCII characters). The type and amount of Values depend on the command number. It is legal for a command to have no parameters. For example, the following command is legal: *65#. The only restriction is that the total number of parameters must be smaller than 1500 bytes (250 for the LanBox LC).
After a serial message has been received, interpreted and executed the LanBox will always transmit the prompt: > (ASCII 62). If the message was only partially received, could not be interpreted or executed a question mark ? (ASCII 63) is transmitted in stead of the prompt. Just as with the commands, a reply from the LanBox will also always start with * and end with #. The data in between consists of **n** ASCII characters making up a 8bit Value (2 ASCII characters) or a 16bit Value (4 ASCII characters).

Using SysEx

It is also possible to use all of the serial commands using MIDI System Exclusive Messages (SysEx). The message format is as follows:

F0 00 20 40 ID 50 MI 2A COMMAND DATA 23 F7

Where ID is the LanBox SysEx Device ID, MI is the Master Device ID (the controller) and the COMMAND is the serial command in ASCII values of the characters. Note that the 2A and 23 are the (hexadecimal) ASCII codes for the * and #.
The reply will be:

F0 00 20 40 ID 51 MI DATA 3E F7

If the Command is not understood or wrong the 3E will be 3F (62 or 63 decimal, ASCII code for the > and ? Symbols).

For example, if we want to start cue list 16 in Layer A, the command would be * 56 01 0010 #. This would give the following SysEx command, assuming that our controller has SysEX ID 0 and the LanBox ID 1 (because a lot of software asks for decimal values, those are given as well):

Text	SysEx header				ID	type	MI	*	5	6	0	1	0	0	1	0	#	end
Hex	F0	00	20	40	01	50	00	2A	35	36	30	31	30	30	31	30	23	F7
Dec	240	0	32	64	1	80	0	42	53	54	48	49	48	48	49	48	35	247

List of Commands

General Commands

These commands are used for general control of the LanBox.

#05 CommonGetAppID

With the **CommonGetAppID** command, you can get information about the LanBox type and firmware version. The command number is 0 5 (0x0005 hex) and is followed by 2 8bit bytes, each containing 00. The reply consists of 2 8bit bytes for the device type (see Table 1), and 1 16bit byte for the firmware version.

LCedit+: Done Automatically after connecting. Information shows up in the interface status window.

LanBox Device ID	
F8 FB	LC+
F8 FD	LCX
F8 FF	LXM
F9 01	LCE

Table 1, Lanbox Device ID

General form:

* 0005 00 00 #

LanBox reply:

* ID ID VERS #>

ID ID:	Device ID
VERS:	16bit Firmware version with 2 decimal fixed point

Example:

* 0005 00 00 #

Reply:

* F8 FD 00D3 #>

F8 FD:	Device is a LCX
00D3:	firmware version 2.11(0x00D3)

#65 Common16BitMode

With the **Common16BitMode** command, you can set the LanBox (all but the LanBox-LC) in 16bit operation mode. For backward compatibility reasons, the LanBox default is set to 8bit Operation Mode. In order to make full use off the LCX, LCE or LCM possibilities, the first command you should give the LanBox is this command. The command number is 101 (0x65 hex) and has one Boolean parameter that's either 0 (0x00) for 8bit mode or > 0 for 16bit mode.

LCedit+: Done Automatically after connecting.

General form:

* 65 MO #

MO: Mode. 00 = 8bit mode, > 00 = 16bit mode

LanBox reply:

> No reply.

Example:

* 65 FF #

Set LanBox to 16bit operation Mode

Reply:

> No reply.

Example:

* 65 00 #

Set LanBox to 8bit operation Mode

Reply:

> No reply.

#B5 CommonReboot

With the **CommonReboot** command, you can reboot the LanBox. It will shut down the LanBox without saving any data and the LanBox will boot again in its last saved status. The command number is 181 (0xB5 hex) and has no parameters.

LCedit+: Tools>Reboot LanBox.

General form:	LanBox reply:
* B5 #	> No reply.

Reboots the LanBox.

#A9 CommonSaveData

The LanBox has two types of memory, its 'nonvolatile' flash ROM and its 'volatile' RAM memory. The RAM memory is very quick in storing and querying data, but requires power to work (this is what the volatile means). That means that anything stored here will be lost when you power down the LanBox. To save your work, it will have to be copied to the flash ROM. This memory works a lot slower, but will never loose its data. With the **CommonSaveData** command you copy the contents from the LanBox volatile RAM to its ROM memory. This way, the data is stored to be available after reboot. The LanBox will return to the last saved state when booting. The command number is 169 (0xA9 hex) and has no parameters.

LCedit+: Tools>Save LanBox data.

General form:	LanBox reply:
* A9 #	> No reply.

Copy the RAM contents to the ROM

Channel Control

Although DMX-512 can only broadcast a maximum of 512 Channels, the LanBox-LCX can support up to 3072 Mixer Channels (250 in 8bit operation mode). Be aware however that the maximum number of active Channels also is 3072. A Channel that is active in **n** Layers has to be counted **n** times here. So, for example, if you have the same 512 Channels active in 6 Layers, you won't be able to activate them in a 7th Layer.

Every Channel can be individually controlled (for example with the **ChannelSetData** command) or controlled through a Cue List. In order to get results, a Channel should be activated in a Layer first. By default, this is done automatically when a Channel Value is changed in a Layer, but this can be switched On and Off with the **LayerSetAutoOutput** command. When Auto Output is switched Off (default is On), a Channel will be inactive in that specific Layer until you activate it with the **ChannelSetOutputEnable** command.

#C9 ChannelSetData

With the **ChannelSetData** command, you can directly set the output Value of a number of Channels. This always has to be done in an existing Layer. The actual output of the Channel depends on your Layer settings. The command number is 201 (0xC9 hex) and has the following parameters: the Layer ID (8bit, 1 - 63 for A – BK) and n times a Channel Number (16bit, 0 - 3072) and a Channel Value (8bit, 0 - 255).

LCedit+: manipulate channels in the control panel.

General form:	LanBox reply:
* C9 LA CHA1 V1 CHA2 V2 … CHAn Vn#	> No reply.

LA	Layer ID, (8bit, 1 - 63 for A - BK) where 01 is Layer A, 02 Layer B, etc.
CHA1	1st Channel number (16bit, 1-3072)
V1	1st Value (8bit, 0 – 255)
CHA2	1st Channel number (16bit, 1-3072)
V2	1st Value (8bit, 0 – 255)
…	
...	
CHAn	nth Channel number (16bit, 1-3072)
Vn	nth Value (8bit, 0 – 255)

In Layer LA, set Channel CHA1 to a Value of V1, Channel CHA2 to V2 … Channel CHAn to Vn.

Example:	LanBox reply:
* C9 01 07 DF 7E #	> No reply.

01:	Layer A (ID 1, 0x01 hex)
07 DF:	Channel number 2015. (0x07DF hex)
7E:	Value 126 (0x7E hex)

In Layer A, set Channel 2015 to a Value of 126

Example:	LanBox reply:
* C9 1B 00 05 FF 0006 FF 0007 FF 0008 FF#	> No reply.

1B:	Layer AA (ID 26, 0x1B hex)
00 05:	Channel number 5 (0x0005 hex)
FF:	Value 255 (0xFF hex)
00 06:	Channel number 6 (0x0006 hex)
FF:	Value 255 (0xFF hex)
00 07:	Channel number 7 (0x0007 hex)
FF:	Value 255 (0xFF hex)
00 08:	Channel number 8 (0x0008 hex)
FF:	Value 255 (0xFF hex)

In Layer AA, set Channel 5 through 8 to a Value of 255.

#CD ChannelReadData

With the **ChannelReadData** command, you can ask the LanBox to return the Value of a maximum of 255 light channels within a Layer or buffer. The command number is 205 or 0xCD and has 3 parameters: the Layer ID (8bit, 1 - 63 for A - BK), the start Channel number (16bit, 1 - 3072) and the number of Channel Values to be returned (8bit, 0 - 255). To query Values in one of the LanBox Buffers, refer to Table 2 for the corresponding Layer ID's.

LCedit+: Continuous in background.

Buffer	LCX/LCE	LCM	LC+
Mixer	254 (0xFE)	254 (0xFE)	9 (0x09)
DMX out	255 (0xFF)	255 (0xFF)	10 (0x0A)
DMX in	252 (0xFC		
External Inputs	253 (0xFD)		

Table 2, LanBox Buffer ID's

General form:

* CD LA STCH NR #

LA: Layer ID, (8bit, 1 - 63 for A - BK)
STCH: Start Channel number
(16bit, 1 - 3072)
NR: number of Values to be returned,
(8bit, 1 - 255)

LanBox reply:

* V1 V2 V3 .. .. Vn #>

V1: Value 1st queried Channel
V2: Value 2nd queried Channel
V3: Value 3rd queried Channel
Vn: Value nnd queried Channel

Example:

* CD 01 00 01 10 #

01: Layer A (ID 1, 0x01 hex)
00 01: Start Channel number 1. (0x0001 hex)
10: Return 16 channels (0x10 hex)

LanBox reply:

* 00 F0 AF 56 .. .. 78 #>

00: Channel 1 has Value 0 (0x00 hex)
F0: Channel 2 has Value 240 (0xF0 hex)
AF: Channel 3 has Value 175 (0xAF hex)
56: Channel 4 has Value 86 (0x56 hex)
..
78: Channel 16 has Value 120 (0x78 hex)

Return Values of Channel 1 through 16 in Layer A

Example:

* CD FE 00 01 FF #

FE: Mixer Buffer (see Table 2)
00 01: Start Channel number 1. (0x0001 hex)
FF: Return 255 channels (0xFF hex)

Return Values of Channel 1 through 255 in the Mixer Buffer

LanBox reply:

* V1 .. Vn #>

#CE ChannelReadStatus

Every Channel has several Status Attributes in each Layer in which it's active. With the **ChannelReadStatus** command, you can query the Status Attributes of a maximum of 255 light channels within a Layer or Buffer. The command number is 206 (0xCE hex) and has 3 parameters: the Layer ID (8bit, 1 - 63 for A - BK), the start Channel number (16bit, 0 - 3072)) and the number of channels of which the Status Attributes are to be returned (8bit, 1 - 255). To query Values in one of the buffers, refer to Table 2 on page 10 for the corresponding Layer ID's. The reply consists of **n** 8bit bytes, containing flags on each Channel's status. These flags are contained in the 1st 4 bits of each byte, see Table 3 for the decoding.

LCedit+: Continuous in background.

bit	Value	attribute
0	1	Output to Mixer
1	2	Channel edit enabled
2	4	Solo Mode
3	8	Fading

Table 3, Channel status flags

General form:

* CE LA CHAN NR #

LA: Layer ID, (8bit, 1 - 63 for A - BK)
CHAN: Channel number (16bit, 1 - 3072)
NR: number of channels of which Attributes are to be returned (8bit, 1 - 255)

LanBox reply:

* A1 A2 A3 .. An #>

A1: Attributes 1st queried Channel
A2: Attributes 2nd queried Channel
A3: Attributes 3rd queried Channel
An: Attributes nnd queried Channel

Return Status Attributes of Channel CHAN through (CHAN + NR)in Layer LA

Example:

* CE 01 00 01 04 #

01: Layer A (ID 1, 0x01 hex)
00 01: Start Channel number 1. (0x0001 hex)
04: Return 4 channels (0x04 hex)

LanBox reply:

* 00 0B 04 05 #>

00: Binary 0 0 0 0 0 0 0 0
flag 0 = 0 : Output to Mixer = false
flag 1= 0 : Edit enabled = false
flag 2= 0 : Channel Solo = false
flag 3 = 0 : Channel fading = false

0B: Binary 0 0 0 0 1 0 1 1
flag 0 = 1 : Output to Mixer = true
flag 1 = 1 : Edit enabled = true
flag 2 = 0 : Channel Solo = false
flag 3 = 1 : Channel fading = true

04: Binary 0 0 0 0 0 1 0 0
flag 0 = 0 : Output to Mixer = false
flag 1 = 0 : Edit enabled = false
flag 2 = 1 : Channel Solo = true
flag 3 = 0 : Channel fading = false

05: Binary 0 0 0 0 0 1 0 1
flag 0 = 1 : Output to Mixer = true
flag 1 = 0 : Edit enabled = false
flag 2 = 1 : Channel Solo = true
flag 3 = 0 : Channel fading = false

Return Status Attributes of Channel 1 through 4 in Layer A

#CA ChannelSetOutputEnable

With the **ChannelSetOutputEnable** command, you can set the Output to Mixer Status Attribute of any number of Channels in a specific Layer. The command number is 202 (0xCA hex) and has 1 + (**n** x 2) parameters: the Layer ID (8bit, 1 - 63 for A - BK) and **n** times a Channel number (16bit, 0 - 3072) with the On/Off Boolean byte (8bit, 0 = disabled, >0 = enabled). If you want to set all Channels in a Layer to a certain Value, use Channel Number 0 (0x0000 hex).

LCedit+: Status buttons next to channels in Channel control panel

General form: LanBox reply:

```
* CA LA CHA1 B1 CHA2 B2 .. .. CHAn Bn #          > No reply.
```

LA:	Layer ID (8bit, 1 - 63 for A - BK)
CHA1:	Channel number (16bit, 0 - 3072)
B1:	On/Off Boolean byte (8bit, 0 = disabled, >0 = enabled)
CHA2:	Channel number (16bit, 0 - 3072)
B2:	On/Off Boolean byte (8bit, 0 = disabled, >0 = enabled)
.. ..	
CHAn:	Channel number (16bit, 0 - 3072)
Bn:	On/Off Boolean byte (8bit, 0 = disabled, >0 = enabled)

Example: LanBox reply:

```
* CA 05 00 09 00 00 0A 00 00 0B 00 00 0C 00 #          > No reply.
```

05:	Layer E (ID 5, 0x05 hex)
00 09:	Channel number 9. (0x0009 hex)
00:	Output enabled = false
00 0A:	Channel number 10. (0x000A hex)
00:	Output enabled = false
00 0B:	Channel number 11. (0x000B hex)
00:	Output enabled = false

Disable the output to the Mixer of channels 9 through 11 in Layer E

Example: LanBox reply:

```
* CA 02 00 00 FF #          > No reply.
```

02:	Layer B (ID 2, 0x02 hex)
00 00:	Channel number 0 : All channels.
FF:	Output enabled = true

Enable the output to the Mixer of all channels Layer B

#CC ChannelSetActive

With the **ChannelSetActive** command, you can set the Active Status Attribute of any number of channels in a specific Layer. The command number is 204 or 0xCC and has 1 + (n*2) parameters: the Layer ID (8bit, 1 - 63 for A - BK) and **n** times a Channel number (16bit, 0 - 3072) with the On/Off Boolean byte (8bit, 0 = disabled, >0 = enabled). If you want to set all Channels in a Layer to a certain Value, use Channel number 0 (0x0000 hex)

LCedit+: Status buttons next to channels in Channel control panel

General form:

* CC LA CHA1 B1 CHA2 B2 .. .. CHAn Bn #

LA:	Layer ID (8bit, 1 - 63 for A - BK)
CHA1:	Channel number (16bit, 0 - 3072)
B1:	On/Off Boolean byte (8bit, 0 = disabled, >0 = enabled)
CHA2:	Channel number (16bit, 0 - 3072)
B2:	On/Off Boolean byte (8bit, 0 = disabled, >0 = enabled)
.. ..	
CHAn:	Channel number (16bit, 0 - 3072)
Bn:	On/Off Boolean byte (8bit, 0 = disabled, >0 = enabled)

LanBox reply:

> No reply.

Example:

* CC 05 00 09 00 00 0A 00 00 0B 00 00 0C 00 #

05:	Layer E (ID 5, 0x05 hex)
00 09:	Channel number 9. (0x0009 hex)
00:	Active = false
00 0A:	Channel number 10. (0x000A hex)
00:	Active= false
00 0B:	Channel number 11. (0x000B hex)
00:	Active = false

Disactivate channels 9 through 11 in Layer E

LanBox reply:

> No reply.

Example:

* CC 02 00 00 FF #

02:	Layer B (ID 2, 0x02 hex)
00 00:	Channel number 0 : All channels.
FF:	Output enabled = true

Activate all channels Layer B

LanBox reply:

> No reply.

#CB ChannelSetSolo

With the **ChannelSetSolo** command, you can set the Solo Status Attribute of any number of channels in a specific Layer. A Channel will not actually Solo until the Layer in which it is set to Solo is set in Solo Mode (see **LayerSetSolo** command). The command number is 203 or 0xCB and has 1 + (n*2) parameters: the Layer ID (8bit, 1 - 63 for A - BK) and **n** times a Channel number (16bit, 0 - 3072) with the On/Off Boolean byte (8bit, 0 = disabled, >0 = enabled). If you want to set all Channels in a Layer to a certain Value, use Channel number 0 (0x0000 hex)

LCedit+: not implemented

General form:

* CB LA CHA1 B1 CHA2 B2 .. .. CHAn Bn #

LanBox reply:

> No reply.

LA:	Layer ID (8bit, 1 - 63 for A - BK)
CHA1:	Channel number (16bit, 0 - 3072)
B1:	On/Off Boolean byte (8bit, 0 = disabled, >0 = enabled)
CHA2:	Channel number (16bit, 0 - 3072)
B2:	On/Off Boolean byte (8bit, 0 = disabled, >0 = enabled)
.. ..	
CHAn:	Channel number (16bit, 0 - 3072)
Bn:	On/Off Boolean byte (8bit, 0 = disabled, >0 = enabled)

Example:

* CB 05 00 09 00 00 0A 00 00 0B 00 00 0C 00 #

LanBox reply:

> No reply.

05:	Layer E (ID 5, 0x05 hex)
00 09:	Channel number 9. (0x0009 hex)
00:	Solo enabled = false
00 0A:	Channel number 10. (0x000A hex)
00:	Solo enabled = false
00 0B:	Channel number 11. (0x000B hex)
00:	Solo enabled = false

Disable Solo for channels 9 through 11 in Layer E

Example:

* CB 02 00 00 FF #

LanBox reply:

> No reply.

02:	Layer B (ID 2, 0x02 hex)
00 00:	Channel number 0 : All channels.
FF:	Solo enabled = true

Enable Solo for all channels Layer B

Layer General Control

A LanBox has multiple identical Layers, all of which run independently and can be fully controlled. (The LanBox-LC+ has 8 fixed Layers, which were called Engines. This term is still used in some of the (older) documentation and should be regarded as synonym to Layer.)
On boot, the LanBox by default has 5 active Layers, but this amount can be increased to a maximum of 31. On creation, the Layer's Attributes, status and mode are set to their default Values. Use the **LayerSet...** commands to change these settings. A Layer's Attributes and functions define the way a Layer behaves within the LanBox. The following Attributes are used:

Output Enabled: Determines if a Layer outputs its Values to the Mixer.
Default = On.
Edit/Run mode: Makes recording of Cue Lists possible.
Default = Run.
Fading: Determines if Fading is possible in a Layer. When disabled, Channels will only switch between Values.
Default = On.
Solo: If Solo is enabled, only the Channels that are in Solo Mode will respond.
Default = Off.
Paused: Determines if a Layer is Paused. This means that the Layer's sequencer is frozen (also mid-fade).
Default = Off.
Auto Activate: Determines if Channels are automatically activated when changed in the Layer.
Default = On.
Waiting: If a Layer is waiting, a next Cue Step will only be gone to if a step command (Next, Previous or Goto) is received. A Cue Step will always be completed.
Default = Off.
Locked: Only if a Layer is locked, it will be saved if the LanBox Data is saved.
Default = Off.

A LanBox Layer can operate in the following Mixing Modes:

Off: Layer does not output anything.
Copy (default): Knock Out Mode. Overwrites all Values of matching Channels in underlying Layers.
HTP: Highest takes Precedence. The Highest Value of a Channel is output to the Mixer.
LTP: Lowest takes Precedence. The Lowest Value of a Channel is output to the Mixer.
Transparent: A percentage of a Value is added to the Value of underlying Layers.
Add: Makes Values relative to underlying Layers. (Value – 128) is added to the Value of underlying Layers.

Further on, there are a Manual fade setting which determines a fade when a Channel is manually changed in a Layer, and a Chase Mode, which can manipulate Cue Lists are run.

The way Layers are ordered and how the Mix settings are applied determines what the output will eventually look like. Every Cue List and every Channel manipulation is linked to a specific Layer (except from some special cases where you can write directly to the Mixer), so it is important to keep track of where you activate which Channel.

#B1 CommonGetLayers

With the **CommonGetLayers** command, you can query the the complete index of Layers that are currently active in the LanBox. The command number is 177 (0xB1 hex) and has no parameters. The reply from the LanBox consists of **n** x 9 parameters, where **n** is the number of active Layers, in current Mixing order. These parameters are:

Internal Layer Number (8bit)
Layer ID (8bit, 1 - 63 for A - BK)
Layer Attributes (8bit flags, see Table 4)
Active Cue List (16bit, 1-999)
Active Cue Step (8bit, 0 - 99)
Cue Step Fade Time (8bit encoded, see Appendix A, Cue Time Table)
Remaining Fade Time in frames (16bit, 1 frame is 50ms)
Current Hold Time (8bit encoded, see Appendix A, Cue Time Table)
Remaining Hold Time in frames (16bit, 1 frame is 50ms)

LCedit+: Continuous in background.

bit	Val.	(default = 0 for new Layers)
0	1	Layer Output enabled
1	2	Edit/Run Mode
2	4	Fading enabled
3	8	Layer in Solo Mode
4	16	Paused
5	32	Auto Activate enabled
6	64	Sequencer Waiting
7	128	Layer Locked

Table 4, LanBox Layer attribute flags

General form:

```
* B1 #
```

Return information on active Layers

LanBox reply:

```
* n x [NR ID AT CLIS CS FT FREM HT HREM] #>
```

n x (n= number of active Layers)
in mixing order:

NR:	Internal Layer Number (8bit)
ID:	Layer ID (8bit, 1 - 63 for A - BK)
AT:	Layer Attributes (8bit Flags, Table 4)
CLIS:	Active Cue List (16bit, 1- 999)
CS	Active Cue Step (8bit, 0 - 99)
FT:	Current Fade Time (8bit encoded)
FREM:	Remaining Fade Time (16bit in frames)
HT:	Current Hold Time (8bit encoded)
HREM:	Remaining Hold Time ((16bit in frames)

(See next page for an example)

Example: LanBox reply:

```
* B1 #          * [ 5B 01 A7 0001 02 25 0023 33 014C ] [ 5C 02 A7 0002 01 25 0000 33 0071 ]
                  [ 5D 03 A6 0000 00 00 0000 00 0000 ] [ 5E 04 A7 0000 00 00 0000 00 0000 ]
                  [ 5F 05 A5 0000 00 00 0000 00 0000 ] #>
```

Top Layer:

5B: Layer has internal ID 91 (0x5B hex)
01: Layer has ID A
A7: binary : 1 0 1 0 0 1 1 1
flag 0 = 1 Output enabled = true
flag 1 = 1 Sequence Mode enabled = true
flag 2 = 1 Fading enabled = true
flag 3 = 0 Solo enabled = false
flag 4 = 0 Paused = false
flag 5 = 1 Auto Activate = true
flag 6 = 0 Sequencer Waiting = false
flag 7 = 1 Layer locked = true
0001: Layer runs Cue List 1 (0x0001 hex)
02: Current Cue Step = 2 (0x02 hex)
25: Fade time is 5.1s (time index 37 (0x25 hex), see appendix A)
0023: Remaining fade time is 1.75s: 50ms x 35 (0x0023 hex)
33: Hold time is 20s (time index 51 (0x33 hex), see appendix A)
014C: Remaining hold time is 16.6s: 50ms x 332 (0x014C hex)

Next Layer:

5C: Layer has internal ID 92 (0x5C hex)
02: Layer has ID B
A7: binary : 1 0 1 0 0 1 1 1, see top Layer
0002: Layer runs Cue List 2 (0x0002 hex)
01: Current Cue Step = 1 (0x01hex)
25: Fade time is 5.1s (time index 37 (0x25 hex), see appendix A)
0000: Remaining fade time is 0s
33: Hold time is 20s (time index 51 (0x33 hex), see appendix A)
0071: Remaining hold time is 5.65s: 50ms x 113 (0x0071 hex)

Next Layer:

5D: Layer has internal ID 93 (0x5D hex)
03: Layer has ID C
A6: binary : 1 0 1 0 0 1 1 0
flag 0 = 0 Output enabled = false
flag 1 = 1 Sequence Mode enabled = true
flag 2 = 1 Fading enabled = true
flag 3 = 0 Solo enabled = false
flag 4 = 0 Paused = false
flag 5 = 1 Auto Activate = true
flag 6 = 0 Sequencer Waiting = false
flag 7 = 1 Layer locked = true
0000: Layer runs no Cue
00: Fade time n/a
0000: Remaining fade time n/a
00: Hold time n/a
0000: Remaining hold time n/a

Next Layer:

5E: Layer has internal ID 94 (0x5E hex)
04: Layer has ID D
A7: binary : 1 0 1 0 0 1 1 1, see top Layer
0000 Layer runs no Cue
.. Other Bytes n/a

Bottom Layer:

5E: Layer has internal ID 94 (0x5E hex)
05: Layer has ID E
A5: binary : 1 0 1 0 0 1 0 1
flag 0 = 1 Output enabled = true
flag 1 = 0 Sequence Mode enabled = false
flag 2 = 1 Fading enabled = true
flag 3 = 0 Solo enabled = false
flag 4 = 0 Paused = false
flag 5 = 1 Auto Activate = true
flag 6 = Sequencer Waiting = false
flag 7 = 1 Layer locked = true
0000: Layer runs no Cue
.. Other Bytes n/a

Return Index of active Layers

#0A LayerGetStatus

With the **LayerGetStatus** command, you query the the complete state of a Layer. The command number is 10 or 0x0A and has one 8bit parameter, the ID of the Layer. The reply from the LanBox consists of all the parameters of the specific Layer. It also contains the general LanBox Information that is shown in the LCedit+ Layer Control Window.

Output Status (8bit, 0 = Off, >0 = On)
Sequence Status (8bit, 0 = Off, >0 = On)
Fade Status (8bit, 0 = Off, >0 = On)
Solo Status (8bit, 0 = Off, >0 = On)
Mix Status (8bit, see Table 5 on page 22)
Current Hold Time (8bit encoded, see Appendix A: Cue Step Time Encoding Table)
Remaining Hold Time in frames (16bit, 1 frame is 50ms)
Active Cue List (16bit, 1-999) and Active Cue Step (8bit)
Current Chase mode (8bit, see Table 6 on page 29)
Current Layer speed (8bit, 50% - 1600%)
Manual Fade type (8bit, see Table 7 on page 31)
Manual Fade time (8bit encoded, see Appendix A: Cue Step Time Encoding Table)
Cue Step Fade time (8bit encoded, see Appendix A: Cue Step Time Encoding Table)
Remaining Fade time in frames (16bit, 1 frame is 50ms)
Layer Transparency Depth (8bit, 0-100%)
LanBox Loading Indication (8bit, idle time between DMX frames, 0 – 50ms)
Pause Status (0 = Off, >0 = On)
Layer SysEx Device ID (8bit, 0-127)
Auto Activate Status (0 = Off, >0 = On)
Current Cue Step Type (8bit, see Appendix B: Cue Step Types)
Cue Step data (6 x 8bit, , see Appendix B: Cue Step Types)

LCedit+: Continuous in background, gets information on high lighted Layer.

General form:

* 0A LA #

LA	Layer ID (8bit, 1 - 63 for A - BK)

Return Status of Layer LA

LanBox reply:

* OS SS FS SS MS XX HT HREM CLIS CS CM SP MF MT FT FREM TD LI PA SY AU ST S1 S2 S3 S4 S5 S6 XX #>

OS:	Output Status (8bit, 0 = Off, >0 = On)
SS:	Sequence Status (8bit, 0 = Off, >0 = On)
FS:	Fade Status (8bit, 0 = Off, >0 = On)
SS:	Solo Status (8bit, 0 = Off, >0 = On)
MS:	Mix Status (8bit, see Table 5 on page 22)
XX:	Reserved
HT:	Current Hold Time (8bit, see Appendix A: Cue Step Time Encoding Table)
HREM:	Remaining hold time is 0.s: 50ms x 113 (0x0071 hex)
CLIS:	Active Cue List (16bit, 1-999)
CS:	Active Cue Step (8bit)
CM:	Current Chase mode (8bit, see Table 6 on page 29)
SP:	Current Layer speed (8bit, 50% - 1600%)
MF:	Manual Fade type (8bit, see Table 7 on page 31)
MT:	Manual Fade time (8bit, see Appendix A: Cue Step Time Encoding Table)
FT:	Cue Step Fade time (8bit, see Appendix A: Cue Step Time Encoding Table)
FREM:	Remaining Fade time in frames (16bit, 1 frame is 50ms)
TD:	Layer Transparency Depth (8bit, 0-100%)
LI:	LanBox Loading Indication (8bit, idle time between DMX frames, 0-50ms)
PA:	Pause Status (0 = Off, >0 = On)
SY:	Layer SysEx Device ID (8bit, 0-127)
AU:	Auto Activate Status (0 = Off, >0 = On)
ST:	Current Cue Step Type (8bit, see Appendix B: Cue Step Types)
S1 -S6	Cue Step data (6x 8bit, see Appendix B: Cue Step Types)
XX:	Reserved

(Example on next page)

Example:

```
* 0A 02#
```

02	Layer B

LanBox reply:

```
* FF FF FF 00 04 00 1B 001B 0002 01 00 7F 00 1F 04 0000 CC 2A
00 03 FF 01 03 04 1B 0A 36 A0 00 #>
```

FF:	Output Status = On
FF:	Sequence Status = On
FF:	Fade Status = On
00:	Solo Status = Off
04:	Mix Status = 4 (0x04 hex)=Transparent
00:	Reserved
1B:	Hold time is 2s (time index 27 (0x1B hex), see appendix A)
001B:	Remaining hold time is 1.35s: 50ms x 27 (0x001B hex)
0002:	Layer is running Cue List 2(0x002 hex)
01:	Current Cue Step = 1 (0x01 hex)
00:	Chase mode = Off
7F:	Layer speed = 100% (127, 0x7F hex)
00:	Manual Fade = Off
1F:	Manual Fade time = 3s (time index 31 (0x1F hex), see appendix A)
04:	Cue Step Fade time = 0.2s (time index 4 (0x04 hex), see appendix A)
0000:	Remaining Fade time 0s
CC:	Transparency Depth = 80% (204 (0cCC hex) $*^{100}/_{255}$)
2A:	LanBox has 42s (0x2A hex) idle time between frames)
00:	Layer Pause Status = Off
03:	Layer SysEx Device ID = 3 (0x03 hex)
FF:	Auto Activate Status = On
01:	Step Type = Show Scene
03	Fade type = Cross fading
04	Fade time is 0.20s (time index 4 (0x04 hex), see appendix A)
1B	Hold time is 2s (time index 27 (0x1B hex), see appendix A)
0A 36 A0	Irrelevant data, step type has 3 Data bytes
00:	Reserved

Return Status of Layer B

#45 LayerSetID

With the **LayerSetID** command, you can change the ID of a Layer. This will not effect the Layer's internal number, Mixing order or any other parameter. If a Layer ID is already used by another Layer, the LanBox will remove the other Layer's ID. The command number is 69 (0x45 hex) and has two parameters, the ID of the Layer to be changed (8bit, 1 - 63 for A - BK) and the new ID (8bit, 1 - 63 for A - BK).

LCedit+: Change the ID letter of a Layer.

General form:

```
* 45 l1 l2 #
```

l1	Layer ID (8bit, 1 - 63 for A - BK)
l2	New Layer ID (8bit, 1-63 for A - BK)

Change the ID of Layer with ID l1 to l2

LanBox reply:

> No reply.

Example:

```
* 45 01 1B #
```

01	Change ID of Layer A (ID 1, 0x01 hex)
1B	New ID is AA (ID 27, 0x1B hex)

Change the ID of Layer A to AA

LanBox reply:

> No reply.

#48 LayerSetOutput

With the **LayerSetOutput** command, you can determine if a Layer outputs its Values to the Mixer. The command number is 72 (0x48 hex) and has two parameters, the Layer ID (8bit, 1 - 63 for A - BK) and a Boolean byte (8bit), where 0 = Off and anything > 0 = On. The default Value of this setting is On.

LCedit+: The Output button in the advanced section of the Layer Control panel.

General form:

* 48 LA BO #

LA	Layer ID (1-63 for A - BK)
BO	On/Off Boolean byte (8bit, 0 = disabled, >0 = enabled)

Set output of Layer LA to On or Off

LanBox reply:

> No reply.

Example:

* 48 05 00 #

05	Layer E (ID 5, 0x05 hex)
00	Output = Off

Set output of Layer E to Off

LanBox reply:

> No reply.

#46 LayerSetFading

With the **LayerSetFading** command, you can determine if a Layer's fading should be disabled or enabled. The command number is 70 (0x46 hex) and has two parameters, the Layer ID (8bit, 1 - 63 for A - BK) and a Boolean byte (8bit), where 0 = Off and anything > 0 = On. The default Value of this setting is On.

LCedit+: The Fade button in the advanced section of the Layer Control panel.

General form:

* 46 LA BO #

LA	Layer ID (1-63 for A - BK)
BO	On/Off Boolean byte (8bit, 0 = disabled, >0 = enabled)

Set fading of Layer LA to On or Off

LanBox reply:

> No reply.

Example:

* 46 20 FF #

05	Layer AF (ID 32, 0x20 hex)
FF	Fading = Off

Set Fading of Layer AF to On

LanBox reply:

> No reply.

#4A LayerSetSolo

With the **LayerSetSolo** command, you can determine if a Layer's Solo mode should be disabled or enabled. If a Layer is in Solo Mode, only the Channels which are set to solo (with the **ChannelSetSolo** command) will be active. The command number is 74 (0x4A hex) and has two parameters, the Layer ID (8bit, 1 - 63 for A - BK) and a Boolean byte (8bit), where 0 = Off and anything > 0 = On. The default Value of this setting is Off.

LCedit+: The Solo button in the advanced section of the Layer Control panel.

General form:

```
* 4A LA BO #
```

LA	Layer ID (1-63 for A - BK)
BO	On/Off Boolean byte (8bit, 0 = disabled, >0 = enabled)

Set Solo Mode of Layer LA to On or Off

LanBox reply:

```
> No reply.
```

Example:

```
* 4A 12 00 #
```

12	Layer R (ID 18, 0x12 hex)
00	Solo Mode = Off

Set Solo Mode of Layer R to Off

LanBox reply:

```
> No reply.
```

#64 LayerSetAutoOutput

With the **LayerSetAutoOutput** command, you can determine if a Layer should automatically activate Channels if they change within a Layer. If Auto Output is disabled, a Channel's Value can be set or changed without actually activating it in the Layer. It won't become active until it is activated with the **ChannelSetOutputEnable** command. The command number is 100 (0x64 hex) and has two parameters, the Layer ID (8bit, 1 - 63 for A - BK) and a Boolean byte (8bit), where 0 = Off and anything > 0 = On. The default Value of this setting is On.

LCedit+: The Automatic Output button in the advanced section of the Layer Control panel.

General form:

```
* 64 LA BO #
```

LA	Layer ID (1-63 for A - BK)
BO	On/Off Boolean byte (8bit, 0 = disabled, >0 = enabled)

Set Auto Output of Layer LA to On or Off

LanBox reply:

```
> No reply.
```

Example:

```
* 64 03 01 #
```

03	Layer C (ID 18, 0x12 hex)
01	Solo Mode = On

Set Auto Output of Layer C to On

LanBox reply:

```
> No reply.
```

#47 LayerSetMixMode

Every Layer in the LanBox has several possible Mix Modes. The Mix Mode of a Layer can be set with the **LayerSetMixMode** command. The command number is 71 (0x47 hex) and has two parameters, the Layer ID (8bit, 1 - 63 for A - BK) and a Mix Mode number (8bit, see Table 5). The default Value of this setting is Copy (1 or 0x01 hex).

LCedit+: Setting the Mixer Mode in the advanced section of the Layer Control panel.

mode	name	description
0	Off	Layer does not output anything.
1	Copy	Knock Out Mode. Overwrites all Values of matching channels in underlying Layers.
2	HTP	Highest takes Precedence. The Highest Value of a Channel is output to the Mixer.
3	LTP	Lowest takes Precedence. The Lowest Value of a Channel is output to the Mixer.
4	Transparent	A percentage of a Value is added to the Value of underlying Layers.
5	Add	Makes Values relative to underlying Layers. (Values – 128) is added to the Value of underlying Layers.

Table 5, LanBox Mix Modes

General form:

* 47 LA MM #

LA	Layer ID (1-63 for A - BK)
MM	Mix Mode (8bit, see Table 5)

Set Mix Mode of Layer LA to MM

LanBox reply:

> No reply.

Example:

* 47 22 04 #

22	Layer AH (ID 34, 0x22 hex)
04	Mix Mode = Transparent (Mode 4, 0x04 hex)

Set Mix Mode of Layer AH to Transparent

LanBox reply:

> No reply.

#63 LayerSetTransparencyDepth

With the **LayerSetTransparencyDepth** command, you can set the factor by which the Value of a Channel is multiplied and added to the underlying Channel Value. This will only work if the relevant Layer's Mixing Mode is set to Transparent. So if a Layer is set to 50% transparent and a Channel is set to 255 in that Layer, a Value off 127 will be added to the underlying Value (to a maximum of 255). The command number is 99 (0x63 hex) and has two parameters, the Layer ID (8bit, 1 - 63 for A - BK) and a Value for the Transparency (8bit), where 0 = 0% and 255 = 100%. The default Value of the Transparency depth is 50% or 127 (0x7F hex)

LCedit+: The Transparency depth field in the advanced section of the Layer Control panel.

General form:

* 63 LA TD #

LA	Layer ID (1-63 for A - BK)
TD	Transparency depth (8bit, 0 = 0%, 255 = 100%)

Set the Transparency depth of Layer LA to TD

LanBox reply:

> No reply.

Example:

* 63 3E 33 #

3E	Layer BJ (ID 62, 0x3E hex)
33	Transparency depth = 20% (51 (0x33 hex) x $^{100}/_{255}$)

Set the Transparency depth of Layer BJ to 20%

LanBox reply:

> No reply.

#43 LayerSetLocked

The **LayerSetLocked** command sets the Locked Attribute of a Layer. If a Layer is Locked, it and its status will be saved when the LanBox Data is saved. If unlocked, the Layer won't be saved and will not be there when the LanBox is restarted. The command number is 67 (0x43 hex) and has two parameters, the Layer ID (8bit, 1 - 63 for A - BK) and a Boolean byte (8bit), where 0 = Off and anything > 0 = On. The default Value of this setting is Off.

LCedit+: The Lock button in the advanced section of the Layer Control panel.

General form:

```
* 43 LA BO #
```

LanBox reply:

```
> No reply.
```

LA	Layer ID (1-63 for A - BK)
BO	On/Off Boolean byte (8bit, 0 = disabled, >0 = enabled)

Set the Locked attribute of Layer LA to On or Off

Example:

```
* 43 19 F0 #
```

LanBox reply:

```
> No reply.
```

19	Layer Y (ID 25, 0x19 hex)
F0	Locked = On

Set the Locked attribute of Layer Y to On

#44 LayerConfigure

The **LayerConfigure** command has two ways in which it can be used, a short and a long form. In its short form it can be used to change the Mixing Order of Layers or make a new Layer, in its long form you can add Layer flags and a start up Cue Step. he command number is 68 (0x44 hex) and has either 2 parameters or 6, depending on the form.

Short form:
Places the first Layer 1 level below the second Layer. Parameters are Destination Layer ID (8bit, 1 -63 for A - BK) and the Source Layer ID (8bit, 1 -63 for A - BK). If the Destination Layer ID is 64 (0x40 hex) a new Layer will be created directly underneath the Source Layer. If The Source Layer is 0, the Destination Layer will be placed on top of the Mixing Order. If the Source Layer is 64, the Destination Layer will be deleted.

LCedit+: Swapping (dragging) Layers.

General form:

```
* 44 LD LS #
```

LD	Destination Layer ID (1-63 for A - BK)
LS	Source Layer ID (1-63 for A - BK)

Place Layer LS above Layer LD

LanBox reply:

```
> No reply.
```

Example:

```
* 44 02 04 #
```

02	Layer B (ID 2, 0x02 hex)
04	Layer D (ID 4, 0x04 hex)

Place Layer D above Layer B

LanBox reply:

```
> No reply.
```

Example:

```
* 44 03 00 #
```

03	Layer C (ID 3, 0x03 hex)
00	Place on top of Mixing Order

Place Layer C on top of Mixing Order

LanBox reply:

```
> No reply.
```

Example:

```
* 44 1B 40 #
```

1B	Layer AA (ID 27, 0x1B hex)
40	Delete Layer (Destination Layer is 64 (0x40 hex)

Delete Layer AA

LanBox reply:

```
> No reply.
```

Long form:
The Long form has the same parameters as the short form to set some Basic Settings. This is especially useful if you make a new Layer, as you can set an ID as well as start Attributes and start Cue. The extra parameters are: Layer ID (8bit, 1 -63 for A - BK), Layer Attributes (8bit flags, see Table 4 on page 16), Start Cue List (16bit, 1-999) and Start Cue Step (8bit 1 - 99).

LCedit+: Creating a new Layer, deleting Layers.

General form:

```
* 44 LD LS ID AT CLIS CS #
```

LD	Destination Layer ID (1-63 for A - BK)
LS	Source Layer ID (1-63 for A - BK)
ID	Layer ID (1-63 for A - BK)
AT	Layer Attributes (8bit, see Table 4)
CLIS	Start Cue List (16bit, 1-999)
CS	Start Cue Step (8bit 1, - 99)

Place Layer LS above Layer LD, give it ID ID, Attributes ATand start Cue List CLIS at step CS

LanBox reply:

```
> No reply.
```

(example on next page)

Example:

* 44 40 00 1B 27 0109 08 #

LanBox reply:

> No reply.

40	New Layer (Destination Layer ID = 64 (0x40 hex))
00	On Top of Mixing Order (Source Layer ID = 0 (0x00 hex))
1B	Layer ID = AA (ID 27 (0x1B hex))
27	Binary: 0 0 1 0 0 1 1 1 flag 0 = 1 Output enabled = true flag 1 = 1 Edit mode disabled = true flag 2 = 1 Fading enabled = true flag 3 = 0 Solo enabled = false flag 4 = 0 Paused = false flag 5 = 1 Auto Activate = true flag 6 = 0 Sequencer Waiting = false flag 7 = 0 Layer locked = false
0109	Start Cue List 265 (0x0109 hex)
08	Start Cue Step 8 (0x08 hex)

Place New Layer on top of Mixing Order, give it ID AA and stated Attributes and start Cue List 265 at step 8

Layer Playback Control

In order to control the playback of Cue Lists in a Layer, the LanBox supports a number of playback commands.

#56 LayerGo

The **LayerGo** command starts a Cue List in a Layer at an optional Cue Step. The command number is 86 (0x56 hex) and has either 2 or 3 parameters: the Layer ID (8bit, 1 - 63 for A - BK), Cue List Number (16bit, 1-999) and the optional Cue Step number (8bit, 0 - 99). A Cue List number 0 (0x0000 hex) will reset the target Layer (all Layer Attributes and settings except for Locked, MSC ID, sequencer and Layer ID).

LCedit+: Staring a Cue List by typing it in the Layer Control Window.

General form:

```
* 56 LA CLIS (CS) #
```

LA	Destination Layer ID (1-63 for A - BK)
CLIS	Start Cue List (16bit, 1-999)
(CS)	Start Cue Step (8bit) Optional

Start Cue List CLIS (at step CS) in Layer LA

LanBox reply:

```
> No reply.
```

Example:

```
* 56 1D 02A1 01 #
```

1D	Layer ID = AC (ID 29 (0x1D hex))
02A1	Cue List 673 (0x02A1 hex)
01	Cue Step 1 (0x01 hex)

Start Cue List 673, step 1 in Layer AC

LanBox reply:

```
> No reply.
```

#57 LayerClear

The **LayerClear** Command deactivates all Channels in a Layer. Note that if there is still a Cue List running, this will activate Channels again (at least, when Automatic Output is On). The command number is 87 (0x57 hex) and has 1 parameter: the Layer ID (8bit, 1 - 63 for A - BK).

LCedit+: The clear button in the Layer Control Window or the Layer\Clear Layer function.

General form:

```
* 57 LA #
```

LA	Layer ID (1-63 for A - BK)

Deactivate all Channels in Layer LA

LanBox reply:

```
> No reply.
```

Example:

```
* 57 0F #
```

0F	Layer ID = O (ID 15 (0x0F hex))

Deactivate all Channels in Layer O

LanBox reply:

```
> No reply.
```

#58 LayerPause

The **LayerPause** command halts the execution of a Cue List. The Layer will keep current Channel Values and will freeze fades. Action will be resumed again with a **LayerGo** or **LayerResume** command. The command number is 88 (0x58 hex) and has 1 parameter: the Layer ID (8bit, 1 - 63 for A - BK).

LCedit+: The pause button in the Layer Control Window.

General form:	LanBox reply:
* 58 LA #	> No reply.
LA Layer ID (1-63 for A - BK)	
Pause Layer LA	

Example:	LanBox reply:
* 58 1F #	> No reply.
1F Layer ID = AE (ID 31 (0x1F hex))	
Pause Layer AE	

#59 LayerResume

The **LayerResume** command will resume paused execution of a Cue List in a Layer. Action will be resumed again where it was originally Paused. The command number is 89 (0x59 hex) and has 1 parameter: the Layer ID (8bit, 1 - 63 for A - BK).

LCedit+: The pressed pause button in the Layer Control Window.

General form:	LanBox reply:
* 59 LA #	> No reply.
LA Layer ID (1-63 for A - BK)	
Resume Layer LA	

Example:	LanBox reply:
* 59 1F #	> No reply.
1F Layer ID = AE (ID 31 (0x1F hex))	
Resume Layer AE	

#5A LayerNextStep

The **LayerNextStep** command will make a Layer Sequencer jump to the next Cue Step. The command number is 90 (0x5A hex) and has 1 parameter: the Layer ID (8bit, 1 - 63 for A - BK).

LCedit+: The forward button in the Layer Control Window.

General form:	LanBox reply:
* 5A LA #	> No reply.
LA Layer ID (1-63 for A - BK)	
Go next Cue Step in Layer LA	

Example:	LanBox reply:
* 5A 10 #	> No reply.
10 Layer ID = P(ID 15 (0x10 hex))	
Go next Cue Step in Layer P	

#5B LayerPreviousStep

The **LayerPreviousStep** command will make a Layer Sequencer jump to the previous Cue Step. The command number is 91 (0x5B hex) and has 1 parameter: the Layer ID (8bit, 1 - 63 for A - BK).

LCedit+: The forward button in the Layer Control Window.

General form:	LanBox reply:
* 5B LA #	> No reply.
LA Layer ID (1-63 for A - BK)	
Go previous Cue Step in Layer LA	

Example:	LanBox reply:
* 5B 09 #	> No reply.
09 Layer ID = I (ID 9 (0x09 hex))	
Go previous Cue Step in Layer I	

#73 LayerNextCue

The LayerNextCue command will make a Layer Sequencer jump to the next Cue List in the LanBox' directory. The command number is 115 (0x73 hex) and has 1 parameter: the Layer ID (8bit, 1 - 63 for A - BK).

LCedit+: Not implemented.

General form:	LanBox reply:
* 73 LA #	> No reply.
LA Layer ID (1-63 for A - BK)	
Go next Cue List in Layer LA	

Example:	LanBox reply:
* 73 05 #	> No reply.
05 Layer ID = E(ID 5 (0x05 hex))	
Go next Cue List in Layer E	

#74 LayerPreviousCue

The **LayerNextCue** command will make a Layer Sequencer jump to the previous Cue List in the LanBox' directory. The command number is 116 (0x74 hex) and has 1 parameter: the Layer ID (8bit, 1 - 63 for A - BK).

LCedit+: Not implemented.

General form:	LanBox reply:
* 74 LA #	> No reply.
LA Layer ID (1-63 for A - BK)	
Go previous Cue List in Layer LA	

Example:	LanBox reply:
* 74 05 #	> No reply.
05 Layer ID = E(ID 5 (0x05 hex))	
Go previous Cue List in Layer E	

Layer Chase Control

In order to use Chases in a LanBox Layer, you have to load a Chase Cue List and set the Chase Mode of a Layer to On. A Chase Cue List is the same as a normal Cue List, but without Go or Loop Cue Steps. If a Layer is set into a Chase Mode, the Layer's sequencer will step through the Cue List Steps in the set order.

#4B LayerSetChaseMode

The **LayerSetChaseMode** sets the way a Chase in a Layer is performed. The are 9 possible modes (see Table 6). The Chase Mode will only do something in a Layer if the Chase Mode attribute of a Layer is set to On. The command number is 75 (0x4B hex) and has 2 parameters: the Layer ID (8bit, 1 - 63 for A - BK) and the desired Chase Mode. By default, the Chase mode is set to 0 (Off).

LCedit+: The chaser modes drop-down menu in the advanced section of the Layer Control window.

Type	name	description
0	Off	No chase (use Cue List)
1	Chase Up	Steps through cue steps in increasing order
2	Loop Up	Loops through cue steps in increasing order
3	Chase Down	Steps through cue steps in decreasing order
4	Loop Down	Loops through cue steps in decreasing order
5	Random	Steps through cue steps in random order
6	Loop Random	Loops through cue steps in random order
7	Bounce	Steps through cue steps in alternating order
8	Loop Bounce	Loops through cue steps in alternating order

Table 6, LanBox Chase Modes

General form:

* 4B LA CM #

LA Layer ID (1-63 for A - BK)
CM Chase Mode (see Table 6)

Set Layer LA Chase mode to Mode CM

LanBox reply:

> No reply.

Example:

* 4B 16 02#

16 Layer ID = V (ID 22 (0x16 hex))
02 Chase Mode = Up Loop

Set Layer V Chase mode to Loop Up

LanBox reply:

> No reply.

#4C LayerSetChaseSpeed

The **LayerSetChaseSpeed** you can set the speed at which a Layer's Sequencer will run. If the speed is set at 100% the Sequencer will use the time set in the running Cue List. If not, it will multiply the designated time with the Speed Factor. This will only influence Layers in Chase Mode. The Speed Factor is calculated with the following formula: 128 / (255 - **S**) x 100%, where **S** is the Speed Value. If the Speed Value is 0, the Speed Factor is set to 50%, if it is 255, the Speed Factor is set to infinite, which in practice means that Cue Steps will be performed with the LanBox frame rate (20 steps/second). The command number is 76 (0x4C hex) and has 2 parameters: the Layer ID (8bit, 1 - 63 for A - BK) and the Speed Value (8bit, 0 - 255). By default, the Chase Speed is set to 127 (100%).

LCedit+: The chase speed box in the advanced section of the Layer Control window.

General form:

* 4C LA SP #

LA	Layer ID (1-63 for A - BK)
SP	Speed Value (8bit, 0-255)

Set Chase Speed of Layer LA to SP

LanBox reply:

> No reply.

Example:

* 4C 2C AA#

2C	Layer ID = AR (ID 44 (0x2C hex))
AA	Chase speed = 150% 128/(255-170 (0XAA hex))*100%

Set Chase Speed of Layer AR to 150%

LanBox reply:

> No reply.

#4D LayerSetFadeType

Every Layer in the LanBox can be set to a Manual Fade. This means that Channel Values that are set 'Manually', which in practice means set with the **ChannelSetData** command, will fade in or out. This Fade Type is set with the **LayerSetFadeType** command. The command number is 77 (0x4D hex) and has 2 parameters: the Layer ID (8bit, 1 - 63 for A - BK) and the Fade Type (8bit, see Table 7). By default, the Fade Type is set to Off.

LCedit+: The fade types drop-down menu in the advanced section of the Layer Control window.

Type	name	description
0	Off	No fade
1	Fade In	Only fades when Channel Value increases
2	Fade Out	Only fades when Channel Value decreases
3	Cross Fade	Fades if Channel Value in- or decreases
4	Off	No fade
5	Fade In CR	Only fades when Channel Value goes up
6	Fade Out CR	Only fades when Channel Value goes down
7	Cross Fade CR	Fades if Channel Value goes up or down
CR= At Constant Rate. Will make the LSB of 16bit Values increment at a constant rate. Will make fades smoother but less accurate.		

Table 7, Manual Fade Types

General form:

* 4D LA MF #

LA Layer ID (1-63 for A - BK)
MF Manual Fade Type (8bit, see Table 7)

Set Manual Fade Type of Layer LA to MF

LanBox reply:

> No reply.

Example:

* 4D 17 03#

17 Layer ID = W (ID 23 (0x17 hex))
03 Manual Fade Type = Cross Fade

Set Manual Fade Type of Layer AR to Cross Fade

LanBox reply:

> No reply.

#4E LayerSetFadeTime

With the **LayerSetFadeTime** command, the Fade Time for the Layer's Manual Fade is set. The command number is 78(0x4E hex) and has 2 parameters: the Layer ID (8bit, 1 - 63 for A - BK) and the Fade Time (8bit encoded, see Appendix A: Cue Step Time Encoding Table). By default, the Fade Time is set to 3.0s.

LCedit+: The fade time number box in the advanced section of the Layer Control window.

General form:

* 4E LA MT #

LA Layer ID (1-63 for A - BK)
MT Manual Fade Time (
8bit encoded, see Appendix A)

Set Manual Fade TIme of Layer LA to MT

LanBox reply:

> No reply.

Example:

* 4E 0D 03#

0D Layer ID = M (ID 13 (0x0D hex))
03 Manual Fade Time = 1.5 seconds
(

Set Manual Fade Time of Layer M to 1.5 seconds

LanBox reply:

> No reply.

Layer Real Time Editing

The LanBox offers two possible ways of creating and editing Cue Lists. One is to write a whole Cue List at once (as described in the section Cue List Control), the other is way is the so called Live Editing. In order to edit a Cue List, a Layer should be set to Edit Mode. After this, you will have to tell the Layer which Cue List to edit. This can be done with the **LayerGo** command, which will clear the engine and set all Channels to the Channel Values stored in the loaded Cue List. Alternatively, the **LayerUsesCueList** command can be used. This will cause the Layer to point at a Cue List, but not load its contents.

#49 LayerSetEditRunMode

With the **LayerEditRunMode** command, a Layer can be switched between Edit and Run Mode. With this command you can set the Layer in Edit Mode which will make Live Editing possible. The command number is 73(0x4C hex) and has 2 parameters: the Layer ID (8bit, 1 - 63 for A - BK) and a Boolean byte (8bit), where 0 = Edit Mode and anything > 0 = Run Mode. The default Value of this setting is Run Mode.

LCedit+: The Record button-in the Layer Control window.

General form: LanBox reply:

* 49 LA RE # > No reply.

LA	Layer ID (1-63 for A - BK)
RE	On/Off Boolean byte (8bit, 0 = Edit Mode, >0 = Run Mode)

Set the Mode of Layer LA to RE

Example: LanBox reply:

* 49 23 00 # > No reply.

23	Layer I (ID 35, 0x23 hex)
00	Edit Mode

Set the Mode of Layer I to Editt

#0C LayerUsesCueList

With the **LayerUsesCueList** you can let a Layer point at a Cue List without loading its data. This way you can keep the data in the Layer and save it in another Cue List. The command number is 12 (0x0C hex) and has 2 parameters: the Layer ID (8bit, 1 - 63 for A - BK) and the Cue List number (16bit, 1-999).

LCedit+: Not implemented

General form: LanBox reply:

* 0C LA CLIS # > No reply.

LA	Layer ID (1-63 for A - BK)
CLIS	Cue List (16bit, 1 - 999)

Set Layer LA to point to Cue List CLIS

Example: LanBox reply:

* 0C 2B 00 A4 # > No reply.

2B	Layer AQ (ID 43, 0x2B hex)
00 A4	Cue List 164 (0x00A4 hex)

Set Layer 2B to point to Cue List 164

# 5F CueListCreate

In order to be able to manipulate Cue Lists, a Cue List has to be created first. This is done with the **CueListCreate** command. The command number is 95 (0x5F hex) and has 1 parameter: the Cue List number (16bit, 1 - 999) of the created Cue List. If the created Cue List already exists, no action is undertaken by the LanBox.

LCedit+: Starting a non existing Cue List in the Layer Control window if a Layer is in edit mode

General form:

* 5F CLIS #

CLIS	Cue List (16bit, 1 - 999)

Create Cue List CLIS

LanBox reply:

> No reply.

Example:

* 5F 01 1C #

01 1C	Cue List 284 (0x011C hex)

Create a Cue List with number 284

LanBox reply:

> No reply.

#5C LayerInsertStep

With the **LayerInsertStep** command, an empty Cue Step is created and inserted. The command number is 92 (0x5F hex) and has either 1 or 2 parameters: The Layer ID (8bit, 1 - 63 for A - BK), and an optional Cue Step number. If no Cue Step number is given, the new Cue Step will be appended, else it will be inserted at the stated index.

LCedit+: The insert or append buttons in the Single-Step Edit Section of the Layer Control window (when Layer in edit mode)

General form:

* 5C LA (CS) #

LA	Layer ID (1-63 for A - BK)
(CS)	Cue step (8bit, 1 - 99)

Append or insert a Cue Step win Layer LA

LanBox reply:

> No reply.

Example:

* 5C 1A #

1A	Layer Z (ID 26, 0x1A hex)

Append a Cue Step in Layer Z

LanBox reply:

> No reply.

Example:

* 5C 2F 02#

2F	Layer AU (ID 47, 0x2F hex)
02	Index 2 (0x02 hex)

Insert a Cue Step at index 2 in Layer AU

LanBox reply:

> No reply.

#67 LayerReplaceStep

The **LayerReplaceStep** command replaces an existing Cue Step with an empty new Cue Step. The command number is 103 (0x67 hex) and has 2 parameters: The Layer ID (8bit, 1 - 63 for A - BK), and the Cue Step number to be replaced.

> **LCedit+: The Change Cue Step button in the Single-Step Edit Section of the Layer Control window (when Layer in edit mode)**

General form: LanBox reply:

* 67 LA CS # > No reply.

LA	Layer ID (1-63 for A - BK)
CS	Cue Step (8bit, 1 - 99)

Replace Cue Step CS in Layer LA with a new Cue Step

Example: LanBox reply:

* 67 13 04 # > No reply.

13	Layer S(ID 19, 0x13 hex)
04	Cue Step 4 (0x04 hex)

Replace Cue Step 4 in Layer S with a new Cue Step

#4F LayerSetCueStepType

With the **LayerSetCueStepType** command, the Cue Step Type of the current Cue Step in a Layer can be changed. In order to do this, the Layer should be in Edit Mode and a Cue List has to be loaded. The different Cue Step Types can be found in Appendix B: Cue Step Types. The command number is 79 (0x4F hex) and has 2 parameters: The destination Layer ID (8bit, 1 - 63 for A - BK), and the Cue Step type (8bit, see Appendix B)

> **LCedit+: The Change Cue Step button in the Single-Step Edit Section of the Layer Control window (when Layer in edit mode)**

General form: LanBox reply:

* 4F LA ST # > No reply.

LA	Layer ID (1-63 for A - BK)
ST	Step Type (8bit, Appendix B: Cue Step Types)

Set the Cue Step type of the current Cue Step in Layer LA to ST

Example: LanBox reply:

* 4F 36 14 # > No reply.

36	Layer S(ID 19, 0x13 hex)
14	Step Type “Go Cue Step” (Step type 20 (0x14 hex)

Set the Cue Step type of the current Cue Step in Layer BB to “Go Cue Step”

#50 - 55 LayerSetCueStepParameters

With the **LayerSetCueStepParameters** commands, you can set the Parameter bytes of each Cue Step. Use command number 80 (0x50 hex) for the first byte, 81 (0x51 hex) for the second, etc. The maximum number of Parameter bytes for each Cue Step is 6. See Appendix B: Cue Step Types for more information on Cue Steps. The number of parameters is 2: The destination Layer ID (8bit, 1 - 63 for A - BK), and the Cue Step Data (8bit, see Appendix B)

LCedit+: Editing a Cue Step by double clicking the Single Step edit box in the advanced section of the Layer Control window. (when Layer in edit mode)

General form:	LanBox reply:
* 5X LA SX #	> No reply.

LA	Layer ID (1-63 for A - BK)
SX	Step Parameter Byte X (8bit, Appendix B: Cue Step Types)

Set the Cue Step Parameter X of the current Cue Step in Layer LA to SX

Example:	LanBox reply:
* 50 03 10 #	> No reply.

03	Layer C(ID 3, 0x03 hex)
10	Step Parameter 1 = 16 (0x10 hex) (Appendix B: Cue Step Types)

Set the Cue Step Parameter byte 1of the current Cue Step in Layer C to 16

Example:	LanBox reply:
* 52 3F 18 #	> No reply.

03	Layer BK (ID 63, 0x3F hex)
18	Step Parameter 3 = 24 (0x18 hex) (Appendix B: Cue Step Types)

Set the Cue Step Parameter byte 1of the current Cue Step in Layer BK to 24

Layer MIDI Settings

#5E LayerSetDeviceID

To control Layers with MIDI Show Control, they have to have a MSC Device ID. This is set with the **LayerSetDeviceID** command. The command number is 94 (0x5E hex) and has 2 parameters: The Layer ID (8bit, 1 - 63 for A - BK, 0 selects the LanBox), and the SysEx Device ID (8bit, 0 – 127).

LCedit+: Changing the MSC ID in the advanced section of the Layer Control Window

General form:

* 5E LA SY #

LA	Layer ID (8bit, 0-63 for A – BK, 0 selects the LanBox)
SY	Device ID (8bit, 0-127)

Set the SysEx Device ID of Layer LA to SY

LanBox reply:

> No reply.

Example:

* 5E 04 12 #

04	Layer D(ID 4, 0x04 hex)
12	Device ID 18 (0x12 hex)

Set the SysEx Device ID of Layer 4 to 18

LanBox reply:

> No reply.

#40 LayerSetSustain

The **LayerSetSustain** command influences the way a Layer behaves when receiving MIDI Notes (for more information, see the MIDI sections of the reference document). Putting a Layer in Sustain means that all Note-Off and Note-On messages with a velocity 0 will be ignored until Sustain is turned off. This way, you can set multiple Channels to a Value and reset them all at once. The command number is 64 (0x40 hex) and has 2 parameters: The Layer ID (8bit, 1 - 63 for A - BK), and a Boolean byte (8bit), where 0 = Off and anything > 0 = On. The default Value of this setting is Off.

LCedit+: Not implemented

General form:

* 40 LA SU #

LA	Layer ID (1-63 for A - BK)
SU	Boolean (8bit, 0 = Off > 0 = On)

Set the Sustain of Layer LA to SU

LanBox reply:

> No reply.

Example:

* 40 21 40 #

21	Layer AG (ID 33, 0x21 hex)
40	Sustain = On

Set the Sustain of Layer AG to On

LanBox reply:

> No reply.

#41 LayerIgnoreNoteOff

The **LayerIgnoreNoteOff** command influences the way a Layer behaves when receiving MIDI Notes (for more information, see the MIDI sections of the reference document). Putting a Layer to Ignore Note-Off mode means that all Note-Off messages will be ignored and only Note-On messages will be accepted. The command number is 65 (0x41 hex) and has 2 parameters: The Layer ID (8bit, 1 - 63 for A - BK), and a Boolean byte (8bit), where 0 = Off and anything > 0 = On. The default Value of this setting is Off.

LCedit+: Not implemented

General form:

```
* 41 LA NO #
```

LA	Layer ID (1-63 for A - BK)
NO	Boolean (8bit, 0 = Off > 0 = On)

Set Ignore Note-Off of Layer LA to NO

LanBox reply:

> No reply.

Example:

```
* 41 30 00 #
```

30	Layer AV (ID 48, 0x30 hex)
00	Ignore Note-Off = On

Set Ignore Note-Off of Layer AV to Off

LanBox reply:

> No reply.

Cue List Control

Besides editing Cue Lists in Live mode it is also possible to edit them without using a Layer. This way you can edit or create a whole Cue List at once.

#A7 CueListGetDirectory

The **CueListGetDirectory** gives you a list off the Cue Lists stored in the LanBox. Because the length of a reply of the LanBox is limited, a maximum of 80 Cue Lists are listed in one reply. Therefore, the command includes a Index Parameter. The command number is 167 (0x67 hex) and has 1 parameter: The Cue List Index (16bit, 1-999). The reply consist of **n** x 6 characters of data where **n** is the number of Cue Lists, each time giving a Cue List Number (16bit) and the number of Cue Steps of the Cue List.

LCedit+: Continuous in the background, but the result is visible in the Cue List Drop down menu in the Cue List Editor.

General form:

```
* A7 INDX #
```

INDX	Cue List Index (16bit, 1-999)

Get a maximum of 80 Cue Lists Starting at Cue List index INDX

LanBox reply:

```
* CLI1 S1 CLI2 S2 .. .. CLIN CN #>
```

CLI1	Cue List Number 1 (16bit, 1-999)
S1	Number of Cue Steps (8bit)
CLI2	Cue List Number 1 (16bit, 1-999)
S2	Number of Cue Steps (8bit)
.. ..	
CLIN	Cue List Number 1 (16bit, 1-999)
SN	Number of Cue Steps (8bit)

Example:

```
* A7 00 01 #
```

00 01	Cue List index 1 (0x0001 hex)

Get a maximum of 80 Cue Lists Starting at Cue List index 1

LanBox reply:

```
* 00 01 03 00 02 03 00 03 05 00 04 01 00 08 01 00
0B 07 00 97 07 #>
```

00 01	Cue List 1 (0x0001 hex)
03	3 Cue Steps (0x03 hex)
00 02	Cue List 2 (0x0002 hex)
03	3 Cue Steps (0x03 hex)
00 03	Cue List 3 (0x0003 hex)
05	5 Cue Steps (0x05 hex)
00 04	Cue List 4 (0x0004 hex)
01	1 Cue Steps (0x01 hex)
00 08	Cue List 8 (0x0008 hex)
01	1 Cue Steps (0x01 hex)
00 0B	Cue List 11 (0x000B hex)
07	7 Cue Steps (0x07 hex)
00 97	Cue List 151 (0x0097 hex)
07	7 Cue Steps (0x07 hex)

#60 CueListRemove

The **CueListRemove** command deletes a Cue List in the LanBox. The command number is 96 (0x60 hex) and has 1 parameter: The Cue List number (16bit, 1-999).

LCedit+: The delete button in the Cue List Editor.

General form:

```
* 60 CLIS #
```

CLIS	Cue List Number (16bit, 1-999)

Delete Cue List CLIS

LanBox reply:

```
> No reply.
```

Example:

```
* 60 00 14 #
```

00 14	Cue List 20 (0x0014 hex)

Delete Cue List 20

LanBox reply:

```
> No reply.
```

#AB CueListRead

The **CueListRead** command will return all Cue Step Data stored in a Cue List. The command number is 171 (0xAB hex) and has 3 attribute,: the Cue List Number (16bit, 1-999), the first Cue Step to return (8bit, 1 - 99) and the number of Cue Steps to return (8bit, 0 – 99). If the number of Cue Steps is 0, all Cue Steps will be returned. The reply consists of **n** x 14 characters where **n** is the number of Cue Steps, containing 7 bytes for each Cue Step. When the number of Cue Steps in a Cue List exceeds 70, it will be necessary to get the data in multiple frames.

LCedit+: Loading a Cue List in the Cue List Editor.

General form:

```
* AB CLIS S1 SN#
```

CLIS	Cue List Number (16bit, 1-999)
S1	Start Cue Step (8bit)
SN	Number of Cue Steps (8bit)

LanBox reply:

* T1 P1^1 P1^2 P1^3 P1^4 P1^5 P1^6
T2 P2^1 P2^2 P2^3 P2^4 P2^5 P2^6
.. ..
Tn Pn^1 Pn^2 Pn^3 Pn^4 Pn^5 Pn^6 #>

T1	Cue Step 1 Type
P1$^{1-6}$	Cue Step 1 Parameters 1 - 6
T2	Cue Step 2 Type
P2$^{1-6}$	Cue Step 2 Parameters 1 - 6
.. ..	
Tn	Cue Step n Type
Pn$^{1-6}$	Cue Step n Parameters 1 - 6

Read SN steps of Cue Step Data from Cue List CLIS, starting at Cue Step S1

Example:

```
* AB 00 33 01 00 #
```

00 33	Cue List 51 (0x0033 hex)
01	Start Cue Step 1 (0x01 hex)
00	All Cue Steps

LanBox reply:

```
* 01 03 1B 1B 0A 37 60
01 03 1B 1B 0A 37 70
01 03 1B 1B 0A 37 80
01 03 1B 1B 0A 37 90
17 01 05 00 00 00 00
01 03 1B 1B 0A 37 A0
01 03 1B 1B 0A 37  B0
01 03 1B 1B 0A 37 C0
01 03 1B 1B 0A 37 D0
17 06 05 00 00 00 00 #>
```

01	Show Scene
03	Fade Type Cross Fade
1B	Fading for 2 seconds (time code 27, 0x1B hex)
1B	Hold for 2 seconds (time index 27, 0x1B hex)
0A -60	n/a
01	Show Scene
03	Fade Type Cross Fade
1B	Fading for 2 seconds (time code 27, 0x1B hex)
1B	Hold for 2 seconds (time index 27, 0x1B hex)
0A -60	n/a
01 - 80	See above
01 - 90	See above
17	Loop n times to Cue Step
01	Cue Step 1 (0x01 hex)
05	5 times (0x05 hex)
00 - 00	n/a
01 - A0	See above
01 - B0	See above
01 - C0	See above
01 - D0	See above
17	Loop n times to Cue Step
05	Cue Step 5 (0x05 hex)
05	5 times (0x05 hex)

Read all steps of Cue Step Data from Cue List 51, starting at Cue Step 1

#AD CueSceneRead

The Cue Scene Data contains the Channel Data of each Cue Step. It can be retrieved with the **CueSceneRead** command. This command will return the stores Channel Values in a Cue Step. The command number is 173 (0xAD hex) and has either 2 or 3 parameters: the Cue List Number (16bit, 1-999), the Cue Step (8bit, 1-99) of which the Scene data is required and an optional start Channel (16bit, 0-3072). The start Channel is needed when the total of Channels exceeds 250. The reply of the LanBox consists of a Scene Flag (Always 0x00 on LCX, LCE and LCM), the number of Values in the Cue Scene (16bit) and a maximum of 250 times a Channel number (16bit, 1-3072) and a Channel Value (8bit, 0-255)). If the Cue Scene has more then 250 Values, the **CueSceneRead** command should be given several times, each time with a higher start Channel.

LCedit+: Loading a Cue List in the Cue List Editor.

General form:

`* AD CLIS CS (ST)#`

CLIS	Cue List Number (16bit, 1-999)
CS	Cue Step (8bit, 1-99)
(ST)	Optional: Start Channel (8bit)

LanBox reply:

`* SF CHNR CH01 V1 CH02 V2 .. .. Chnn Vn #>`

SF	Scene Flag (Always 0x00 on LCX, LCE and LCM)
CHNR	Number of Channels in scene (16bit)
CH01	1st Channel Number (16bit)
V1	1st Channel Value (8bit)
CH02	2nd Channel Number (16bit)
V2	2nd Channel Value (8bit)
.. ..	
CHnn	nth Channel Number (16bit)
Vn	nth Channel Value (8bit)

Read the Scene Data of Cue List CLIS, Cue Step CS, (starting at Channel ST)

Example:

`* AD 00 A8 02 #`

00 A8	Cue List 168 (0x0A8 hex)
02	Cue Step 2 (0x02 hex)

LanBox reply:

`* 00 00 04 0001 00 00 02 FF 00 03 00 00 04 00 #>`

00	Scene Flag 0 (0x00 hex)
00 04	4 Channels (0x004 hex)
00 01	Channel 1 (0x001 hex)
00	Value 0 (0x00 hex)
00 02	Channel 2 (0x001 hex)
FF	Value 255 (0xFF hex)
00 03	Channel 3 (0x001 hex)
00	Value 0 (0x00 hex)
00 04	Channel 4 (0x001 hex)
00	Value 0 (0x00 hex)

Read the Scene Data of Cue List 168, Cue Step 2 (No start Channel, so starting at Channel 1)

#AA CueListWrite

With the **CueListWrite** command it is possible to create a Cue List and specify the Cue Step Data. The command number is 170(0xAA hex) and has the following parameters: The Cue List number (16bit, 1-999), the number of Cue Steps (8bit, 1-99) and **n** x 7 bytes of Cue Step Data, containing the Step Type and the 6 Cue Step Parameters (all 8bit). If a Cue List has more then 70 steps, it is required to use multiple frames. In that case, only list the number of steps in the first frame, and use 0 (0x00 hex) as number of Cue Steps in the next frame.

LCedit+: Storing a Cue List in the Cue List Editor.

General form:

* AA CLIS CN
T1 $P1^1$ $P1^2$ $P1^3$ $P1^4$ $P1^5$ $P1^6$
T2 $P2^1$ $P2^2$ $P2^3$ $P2^4$ $P2^5$ $P2^6$
.. ..
Tn Pn^1 Pn^2 Pn^3 Pn^4 Pn^5 Pn^6#

LanBox reply:

> No reply.

CLIS	Cue List Number (16bit, 1-999)
CN	Number of Cue Steps (8bit, 1-99)
T1	Cue Step 1 Type
$P1^{1-6}$	Cue Step 1 Parameters 1 - 6
T2	Cue Step 2 Type
$P2^{1-6}$	Cue Step 2 Parameters 1 - 6
.. ..	
Tn	Cue Step n Type
Pn^{1-6}	Cue Step n Parameters 1 - 6

Write CN Cue Step Types of Cue List CLIS

Example:

* AA 03 BF 05
18 02 00 00 00 00 00
1B 01 00 00 01 F4 04
14 01 00 00 00 00 00
0A 02 00 75 01 00 00
14 01 00 00 00 00 00 #

LanBox reply:

> No reply.

03 BF	Cue List 959 (0x03BF hex)
05	5 Cue Steps (0x05 hex)
18	Hold (Cue Step type 24, 0x18 hex)
02	0.10 seconds (Time index 2, 0x02 hex)
00 – 00	n/a
1B	Go if Analogue Input (Cue Type 27, 0x0B hex)
01	Analogue Input 1 (0x01 hex)
00 00	Low Value 0 (0x0000 hex)
01 F4	High Value 500 (0x01F4 hex)
04	Cue Step 4 (0x04 hex)
14	Go Cue Step (Cue Step Type 20, 0x14 hex)
01	Cue step 1 (0x01 hex)
00 - 00	n/a
0A	Go Cue Step in Layer (Cue Type 10, 0x0A hex)
02	Layer B (ID 2, 0x02 hex)
00 75	Cue List 117 (0x0075 hex)
01	Cue Step 1 (0x01 hex)
00 00	n/a
14	Go Cue Step (Cue Step Type 20, 0x14 hex)
01	Cue step 1 (0x01 hex)
00 - 00	n/a

Write the following Cue List 959:

1 hold for 0.10s
2 go .4 if input 1 within 0 - 500
3 go.1
4 go 117.1 in Layer B
5 go .1

#AC CueSceneWrite

With the **CueSceneWrite** command the Channel Values associated with a Cue Step can be saved. The command number is 172 (0xAC hex) and has the following parameters: The Cue List number (16bit, 1-999), the Cue Step (8bit, 1-99), the Cue Scene Flag (8bit, always 0x00 on LCX, LCE and LCM), the number of Values in the Cue Scene (16bit) and a maximum of 250 times a Channel number (16bit, 1-3072) with a Channel Value (8bit, 0-255). If a Cue Scene has more then 250 Channel Values, it is required to use multiple frames. In that case, only list the number of Values in the first frame, and use 0 (0x00 hex) as the number of Channels Values in the next frames.

LCedit+: Storing a Cue List in the Cue List Editor.

General form:	LanBox reply:
* AC CLIS CS CF CHNR CH01 V1 CH02 V2 .. .. CHnn Vn #	> No reply.

CLIS	Cue List Number (16bit, 1-999)
CS	Cue Step (8bit, 1-99)
CF	Cue Scene Flag (0x00)
CHNR	Number of Channels in Cue Scene (16bit, 1-3072)
CH01	Channel 1 (16bit, 1-3072)
V1	Value 1 (8bit, 0-255)
CH02	Channel 2 (16bit, 1-3072)
V2	Value 2 (8bit, 0-255)
.. ..	
CHnn	Channel n (16bit, 1-3072)
Vn	Value n (8bit, 0-255)

Write Channel Values of Cue List CLIS, Cue Step CS

Example:	LanBox reply:
* AC 03 81 11 00 08 01 E0 42 01 E1 A5 01 E2 9F 01 E3 C5 01 F5 67 01 F6 00 01 F7 B5 01 F8 FF#	> No reply.

03 81	Cue List 897 (0x0381 hex)
11	Cue Step 17 (0x11 hex)
00	Cue Scene Flag (0x00)
08	Number of Channels = 8 (0x08 hex)
01 E0	Channel 480 (0x01E0 hex)
42	Value 66 (0x42 hex)
01 E1	Channel 481 (0x01E1 hex)
A5	Value 165 (0xA5 hex)
01 E2	Channel 482 (0x01E2 hex)
9F	Value 159 (0x9F hex)
01 E3	Channel 483 (0x01E3 hex)
C5	Value 197 (0xC5 hex)
01 F5	Channel 501 (0x01F5 hex)
67	Value 103 (0x67 hex)
01 F6	Channel 502 (0x01F6 hex)
00	Value 0 (0x00 hex)
01 F7	Channel 503 (0x01F7 hex)
B5	Value 181 (0xB5 hex)
01 F8	Channel 483 (0x01F8 hex)
FF	Value 255 (0xFF hex)

Write the following Channel Value for Cue List 897, Cue Step 17:

Channel	480	481	482	483	501	502	503	504
Value	66	165	159	197	103	0	181	255

#62 CueListRemoveStep

With the **CueListRemoveStep** command you can remove a single step from a stored Cue List. If the Cue Step contains Scene Data this will also be removed. The command number is 98 (0x62 hex) and has 2 parameters: The Cue List number (16bit, 1-999) and the Cue Step number (8bit, 1-99).

LCedit+: Storing a Cue List in the Cue List Editor.

General form:

* 62 CLIS CS#

CLIS	Cue List Number (16bit, 1-999)
CN	Cue Step number (8bit, 1-99)

Delete Cue Step CN of Cue List CLIS

LanBox reply:

> No reply.

Example:

* 62 02 22 60 #

02 22	Cue List 546(0x0022 hex)
60	Cue Step 96 (0x60 hex)

Delete Cue Step 96 of Cue List 546

LanBox reply:

> No reply.

LanBox Settings

With the LanBox Settings commands several Values and Tables in the LanBox can be set.

#68 CommonSetMIDIMode

The LanBox can communicate through either a serial or a MIDI connection. With the **CommonSetMIDIMode** command you can force the LanBox to switch between one of them. The command number is 104 (0x68 hex) and has one parameter: A Boolean Byte (8bit, 0= off, >0 = On)

LCedit+: Choosing a midi connection in the Interface Setup dialog

General form:	LanBox reply:
* 68 MM#	> No reply.
MM MIDI Mode (8bit, 0= off, >0 = On)	
Set MIDI Mode to MM	

Example:	LanBox reply:
* 68 FF #	> No reply.
FF Switch MIDI Mode On	
Set MIDI Mode to On	

#6B CommonMIDIBeat

The **CommonMIDIBeat** command sends a MIDI beat to the LanBox. The command number is 107 (0x6B hex) and has no parameters.

LCedit+: Not implemented

General form:	LanBox reply:
* 6B #	> No reply.
Send MIDI Beat	

# 80 CommonGetPatcher

Every Mixer Channel in the LanBox can be linked to one or more DMX channels. With the **CommonGetPatcher** command, the LanBox will return a part of the currently stored Patch Table. The command number is 128 (0x80 hex) and has 2 parameters: the first DMX Channel of which the Mixer Channel is to be returned (16bit, 1-512) and the number of DMX Channels of which the Mixer Channel is to be returned (8bit, 1-255). The reply consist of **n** Mixer Channels (16bit, 0-3072). A 0 (0x0000 hex) means the DMX Channel is not patched.

LCedit+: Loaded when opening the DMX Patch in the Patch Editor

General form:

* 80 DMX1 NR#

DMX1	First DMX Channel (16bit, 1-512)
NR	Number of Channels (8bit, 1-255)

LanBox reply:

* CH01 CH02 .. .. CHnn #>

CH01	1st Channel number (16bit, 0-3072)
CH02	2nd Channel number (16bit, 0-3072)
.. ..	
CHnn	nth Channel number (16bit, 0-3072)

Get the Patch Data of ND DMX Channels, starting with DMX Channel DMX1

Example:

* 80 01 00 04 #

01 00	Start at DMX Channel 256 (0x0100 hex)
04	Return 4 Channels (0x04 hex)

LanBox reply:

* 00 51 00 52 00 53 00 54 #>

00 51	DMX Channel 256 is patched to Mixer Channel 81 (0x0051 hex)
00 52	DMX Channel 257 is patched to Mixer Channel 82 (0x0052 hex)
00 53	DMX Channel 257 is patched to Mixer Channel 83 (0x0053 hex)
00 54	DMX Channel 256 is patched to Mixer Channel 84 (0x0054 hex)

Get the Patch Data of 4 DMX Channels, starting with DMX Channel 256

#81 CommonSetPatcher

With the **CommonSetPatcher** command, you can assign Mixer Channels to DMX Channels in the LanBox Patch Table. The command number is 129 (0x81 hex) and has up to 250 pairs of parameters, each time a DMX Channel (16bit, 1-512) and a Mixer Channel (16bit, 0-3072). If a Mixer Channel of 0 (0x0000 hex) is given, the DMX Channel will not be patched and the output will always be 0.

LCedit+: Editing the DMX patch in the Patch Editor

General form:

* 81 DMX1 CHA1 DMX2 CHA2 .. .. DMXn CHAn #

DMX1	1st DMX Channel (16bit, 1-512)
CHA1	1st Mixer Channel (16bit, 0-3072)
DMX2	2nd DMX Channel (16bit, 1-512)
CHA2	2nd Mixer Channel (16bit, 0-3072)
.. ..	
DMXn	nth DMX Channel (16bit, 1-512)
CHAn	ntth Mixer Channel (16bit, 0-3072)

LanBox reply:

No reply >

Patch DMX Channel DMX1 to DMXn to Mixer Channels CHA1-CHAn

Example:

* 81 00 01 02 01 00 02 02 02 00 03 00 00 #

00 01	DMX Channel 1(0x0001 hex)
02 01	Mixer Channel 513 (0x0201hex)
00 02	DMX Channel 2 (0x0002 hex)
02 02	Mixer Channel 514 (0x0202 hex)
00 03	DMX Channel 3 (0x0003 hex)
00 00	Mixer Channel 0 (0x0000 hex)

LanBox reply:

No reply >

Patch DMX Channels 1 and 2 to Mixer Channels 513 and 514. Leave DMX Channel 3 unpatched

#82 CommonGetGain

It is possible to give every DMX Channel a multiplication factor. In the LanBox, this is called a gain factor. The DMX output is defined by the formula: (DMX Output) = ($^{(\text{Gain Factor})}/_{128}$) x (Mixer Value). So if the Gain Factor is 255, the DMX output would be 2x the Mixer Value, and if the Gain Factor is 64, it would be ½ x the Mixer Value. The **CommonGetGain** command returns (part of) the Gain Table stored in the LanBox. The command number is 130 (0x82 hex) and has 2 parameters: the first DMX Channel of which the Gain Value is to be returned (16bit, 1-512) and the number of DMX Channels of which the Gain Value is to be returned (8bit, 1-255). The reply consist of **n** Gain Channels (8bit, 0-255).

LCedit+: Loaded when opening the Gain Table in the Patch Editor

General form:

* 82 DMX1 NR#

DMX1	First DMX Channel (16bit, 1-512)
NR	Number of Channels (8bit, 0-255)

LanBox reply:

* G1 G2 .. .. Gn #>

G1	1st Channel Gain Factor (8bit, 0-255)
G2	2nd Channel Gain Factor (8bit, 0-255)
.. ..	
Gn	nth Channel Gain Factor (8bit, 0-255)

Get the Gain Values of ND DMX Channels, starting with DMX Channel DMX1

Example:

* 82 00 48 04 #

00 48	Start at DMX Channel 72 (0x0048 hex)
04	Return 4 Channels (0x04 hex)

LanBox reply:

* 80 40 20 10 #>

80	DMX Channel 72 has Gain Factor 128 (0x80 hex)
40	DMX Channel 73 has Gain Factor 64 (0x40 hex)
20	DMX Channel 74 has Gain Factor 32 (0x20 hex)
10	DMX Channel 75 has Gain Factor 16 (0x10 hex)

Get the Gain Values of 4 DMX Channels, starting with DMX Channel 72

#83 CommonSetGain

With the **CommonSetGain** command, you can assign Gain Factors to DMX Channels in the LanBox Gain Table. The command number is 131 (0x83 hex) and has up to 250 pairs of parameters, each time a DMX Channel (16bit, 1-512) and a Gain Factor (8bit, 0-255). If a Gain Factor of 0 (0x00 hex) is given, the DMX Channel will always output 0.

LCedit+: Editing the Gain Table in the Patch Editor

General form:

* 83 DMX1 G1 DMX2 G2 .. .. DMXn Gn #

DMX1	1st DMX Channel (16bit, 1-512)
G1	1st Gain Factor (8bit, 0-255)
DMX2	2nd DMX Channel (16bit, 1-512)
G2	2nd Gain Factor (8bit, 0-255)
.. ..	
DMXn	nth DMX Channel (16bit, 1-512)
Gn	nth Gain Factor (8bit, 0-255)

LanBox reply:

No reply >

Set Gain Factor DMX Channel DMX1 through DMXn to G1-Gn

Example:

* 83 0001 80 0002 C0 0003 E0 #

00 01	DMX Channel 1(0x0001 hex)
80	Gain Factor 128 (0x80 hex)
00 02	DMX Channel 2 (0x0002 hex)
C0	Gain Factor 192 (0xC0 hex)
00 03	DMX Channel 3 (0x0003 hex)
E0	Gain Factor 224 (0xE0 hex)

LanBox reply:

No reply >

Set Gain Factor DMX Channel 1 through 3 to 128 (x1), 192 (x1,5) and 224 (x1.75)

#84 CommonGetCurveTable

In the LanBox, a curve can be assigned to every DMX Channel. The LanBox has 8 curves stored, numbered 0 – 7. Curve 0 is fixed and linear, the other 7 can be assigned (see **#8C, 8E, 90, 92, 94, 96, 98 CommonGetCurve** and **#8D, 8F, 91, 93, 95, 97, 99 CommonSetCurve** commands). The **CommonGetCurveTable** returns (part of) the Curve Table stored in the LanBox. The command number is 132 (0x84 hex) and has 2 parameters: the first DMX Channel of which the Curve is to be returned (16bit, 1-512) and the number of DMX Channels of which the Curve is to be returned (8bit, 1-255). The reply consist of **n** Curve Values (8bit, 0-7). By default, all DMX Channels have Curve 0 assigned.

LCedit+: Loaded when opening the Curve Table in the Patch Editor

General form:

* 84 DMX1 NR#

DMX1	First DMX Channel (16bit, 1-512)
NR	Number of Channels (8bit, 0-255)

LanBox reply:

* C1 C2 .. .. Cn #>

C1	1st Channel Curve (8bit, 0-7)
C2	2nd Channel Curve (8bit, 0-7)
.. ..	
Cn	nth Channel Curve (8bit, 0-7)

Get the Curves of NR DMX Channels, starting with DMX Channel DMX1

Example:

* 84 000C 04 #

00 0C	Start at DMX Channel 12 (0x000C hex)
04	Return 4 Channels (0x04 hex)

LanBox reply:

* 01 02 07 00 #>

01	DMX Channel 12 has Curve 1 (0x01 hex)
02	DMX Channel 13 has Curve 2 (0x02 hex)
07	DMX Channel 14 has Curve 7 (0x07 hex)
00	DMX Channel 15 has Curve 0 (0x00 hex)

Get the Curves of 4 DMX Channels, starting with DMX Channel 72

#85 CommonSetCurveTable

With the **CommonSetCurveTable** command, you can assign Curves to DMX Channels in the LanBox Curve Table. The command number is 133 (0x85 hex) and has up to 250 pairs of parameters, each time a DMX Channel (16bit, 1-512) and a Curve Value (8bit, 0-7).

LCedit+: Editing the Curve Table in the Patch Editor

General form:

* 85 DMX1 C1 DMX2 C2 .. .. DMXn Cn #

DMX1	1st DMX Channel (16bit, 1-512)
G1	1st Curve Value (8bit, 0-7)
DMX2	2nd DMX Channel (16bit, 1-512)
G2	2nd Curve Value (8bit, 0-7)
.. ..	
DMXn	nth DMX Channel (16bit, 1-512)
Gn	ntth Curve Value (8bit, 0-7)

LanBox reply:

No reply >

Set the Curve of DMX Channel DMX1 through DMXn to C1-Cn

Example:

* 85 01 37 01 01 38 06 0139 00 #

01 37	DMX Channel 311(0x0137 hex)
01	Curve 1 (0x01 hex)
01 38	DMX Channel 312 (0x0138 hex)
06	Curve 6 (0x06 hex)
01 39	DMX Channel 313 (0x0139 hex)
00	Curve 0 (0x00 hex)

LanBox reply:

No reply >

Set the Curve of DMX Channel 311 through 313 to Curve 1, 6 and 0

#8C, 8E, 90, 92, 94, 96, 98 CommonGetCurve

The LanBox has 8 Curve Tables stored that can be assigned to DMX Channels. Curve 0 is fixed and a standard linear Curve where 0 = 0 and 255 = 255. All other curves can be customized. The **CommonGetCurve** commands return these Tables. The command numbers are 140, 142, 144, 146, 148, 150 and 152 (0x8C, 0x8E, 0x90, 0x92, 0x94, 0x96, 0x98 hex) and have 2 parameters: the first Value to be returned (8bit, 0 - 255) and the number of Values to be returned (8bit, 0-255). If the number of Values is 0 (0x00 hex) all Values will be returned. The reply consists of **n** output Values (8bit, 0-255) that are linked to the requested input Values.

LCedit+: Loaded when opening the Curve editor in the Patch Editor

General form:

* 8C – 98 I1 NR#

I1	First Value to be returned (8bit, 0-255)
NR	number of Values (8bit, 0-255)

LanBox reply:

* V1 V2 .. .. Vn #>

V1	1st Curve Output Value (8bit, 0-255)
V2	2nd Curve Output Value (8bit, 0-255)
.. ..	
Vn	nth Curve Output Value (8bit, 0-255)

Get the Curve Output Values of NR Input Values, starting with Input Value I1

Example:

* 8C 00 00 #

00 0C	Start at DMX Channel 12 (0x000C hex)
04	Return 4 Channels (0x04 hex)

LanBox reply:

* FF FE FD .. .. 02 01 00 #>

FF	Output = 255 (0xFF hex)
FE	Output = 254 (0xFE hex)
FD	Output = 253 (0xFD hex)
.. ..	
02	Output = 2 (0x02 hex)
01	Output = 1 (0x01 hex)
00	Output = 0 (0x00 hex)

Get the Curve Output Values of All Input Values, starting with Input Value 00 of Curve Table 1
(Return is Default Curve 1: An inverted Linear Curve)

#8D, 8F, 91, 93, 95, 97, 99 CommonSetCurve

With the **CommonSetCurve** up to 255 output Values can be assigned to input Values in one of the 7 customizable Curve Tables in the LanBox. The command numbers are 141, 143, 145,147, 149, 151 and 153 (0x8D, 0x8F, 0x91, 0x93, 0x95, 0x97, 0x99 hex) and have **n** x 2 parameters: the Input Value (8bit, 0-255) and Output Value (8bit, 0-255).

LCedit+: Storing a Curve in the Curve editor in the Patch Editor

General form:

* 8D – 99 I1 O2 I2 O2 .. .. In On #

I1	Input Value 1 (8bit, 0-255)
O1	Output Value 1 (8bit, 0-255)
I2	Input Value 2 (8bit, 0-255)
O2	Output Value 2 (8bit, 0-255)
.. ..	
In	Input Value n (8bit, 0-255)
On	Output Value n (8bit, 0-255)

LanBox reply:

No reply >

Link output Values to Input Values in one of the 7 Curve Tables

Example:

* 97 00 00 01 01 02 04 03 09 04 F0#

00	Input 0 (0x00 hex)
00	Output 0 (0x00 hex)
01	Input 1 (0x01 hex)
01	Output 1 (0x01 hex)
02	Input 2 (0x02 hex)
04	Output 4 (0x04 hex)
03	Input 3 (0x03 hex)
09	Output 9 (0x09 hex)
04	Input 4 (0x04 hex)
F0	Output 16 (0xF0 hex)

LanBox reply:

No reply >

Link Input Values 0 through 4 to Values 0- 16 in Curve 6

#86 CommonGetSlope

Each DMX Channel has a Slope Limit assigned to it. The Slope Limit limits the amount of change per frame for a DMX Channel. The Slope Limit can be used for equipment that can not handle rapid changes in intensity and can have a Value of 0 to 255. The **CommonGetSlope** command returns (part of) the Slope Table stored in the LanBox. The command number is 134 (0x86 hex) and has 2 parameters: the first DMX Channel of which the Slope is to be returned (16bit, 1-512) and the number of DMX Channels of which the Slope is to be returned (8bit, 1-255). The reply consist of n Slope Values (8bit, 0-255). By default, all DMX Channels have Slope 255 assigned, which means any change of Value is allowed.

LCedit+: Loaded when opening the Slope Table in the Patch Editor

General form:

* 86 DMX1 NR#

DMX1	First DMX Channel (16bit, 1-512)
NR	Number of Channels (8bit, 0-255)

LanBox reply:

* S1 S2 .. .. Sn #>

S1	1st Channel Gain Factor (8bit, 0-255)
S2	2nd Channel Gain Factor (8bit, 0-255)
.. ..	
Sn	nth Channel Gain Factor (8bit, 0-255)

Get the Slope Values of ND DMX Channels, starting with DMX Channel DMX1

Example:

* 86 01 95 04 #

01 95	Start at DMX Channel 405 (0x0195 hex)
04	Return 4 Channels (0x04 hex)

LanBox reply:

* 10 7F FF 0A #>

10	DMX Channel 405 has Slope 16 (0x10 hex)
7F	DMX Channel 406 has Slope 127 (0x7F hex)
FF	DMX Channel 407 has Slope 255 (0xFF hex)
0A	DMX Channel 408 has Slope 10 (0x0A hex)

Get the Slope Values of 4 DMX Channels, starting with DMX Channel 405

#87 CommonSetSlope

With the **CommonSetSlope** command, you can assign Slopes to DMX Channels in the LanBox Slope Table. The command number is 135 (0x87 hex) and has up to 250 pairs of parameters, each time a DMX Channel (16bit, 1-512) and a Slope Value (8bit, 0-255).

LCedit+: Editing the Slope Table in the Patch Editor

General form:

* 87 DMX1 S1 DMX2 S2 .. .. DMXn Sn #

DMX1	1st DMX Channel (16bit, 1-512)
S1	1st Slope Value (8bit, 0-255)
DMX2	2nd DMX Channel (16bit, 1-512)
S2	2nd Slope Value (8bit, 0-255)
.. ..	
DMXn	nth DMX Channel (16bit, 1-512)
Sn	nth Slope Value (8bit, 0-255)

LanBox reply:

No reply >

Set the Slope of DMX Channel DMX1 through DMXn to S1-Sn

Example:

* 87 00 02 FF 00 05 7F 00 09 10 #

00 02	DMX Channel 2 (0x0002 hex)
FF	Slope 255(0xFF hex)
00 05	DMX Channel 5 (0x0005 hex)
7F	Slope 127 (0x7F hex)
00 09	DMX Channel 9 (0x0009 hex)
10	Slope 16 (0x10 hex)

LanBox reply:

No reply >

Set the Slopes of DMX channels 2, 5 and 9 to 255, 127 and 16

LanBox Global Settings

With the Global Settings commands, you can adjust all hardware settings of the LanBox. Be aware that some of these commands can disrupt the proper functioning of the LanBox.

#0B CommonGetGlobalData

The **CommonGetGlobalData** commands returns all the networking and DMX Global Settings of the LanBox. The command number is 11 (0x0B hex) and has no parameters. The reply consists of a large number of bytes and characters:

The Serial Port Baud Rate (8bit, 80-84):	See command #0006 on page 52
DMX Out Offset (16bit, 0-512)	See command #6A on page 52
Number off DMX Channels (16bit, 0-512)	See command #69 on page 53
Network Name Length (8bit, 0-13)	Number of characters in network name
Network Name, 13x 8bit	See command #AE on page 53
SysEx Device ID (8bit, 0-127)	See command #5E on page 36
LanBox IP Addres (4x 8bit)	See command #B0 on page 54
LanBox Subnet Mask (4x 8bit)	See command #B0 on page 54
LanBox Standard Gateway (4x 8bit)	See command #B0 on page 54
DMX Input Destination Layer (8bit, 1-255)	See command #B2 on page 55
DMX Input Destination Offset (16bit, 0-3072)	See command #B2 on page 55
DMX Input Source Offset (16bit, 0-3072)	See command #B2 on page 55
DMX Input Size (8bit, 0-512)	See command #B2 on page 55
UDP In Destination Layer (8bit, 1-255)	See command #B8 on page 56
UDP In Source Address (4x 8bit)	See command #B8 on page 56
UDP In Port (16bit, 0-65535)	See command #B8 on page 56
UDP In Destination Start Channel (16bit, 0-3072)	See command #B8 on page 56
UDP In Source Start Channel (16bit, 0-3072)	See command #B8 on page 56
UDP In Channel Count (16bit, 0-3072)	See command #B8 on page 56
UDP Out Port (16bit, 0-65535)	See command #B9 on page 57
UDP Out Source Start Channel (16bit, 0-3072)	See command #B9 on page 57
UDP Out Channel Count (16bit, 0-3072)	See command #B9 on page 57
UDP Out Flags (See Table 9)	See command #B9 on page 57
Clock to Channel (16bit, 0-3066)	See command #BA on page 58
MTC to Channel (16bit, 0-3066)	See command #BA on page 58
NTP time server IP address (4x8bit)	See command #BA on page 58
NTP time offset (32bit)	See command #BA on page 58
Clock Frequency Tuning (32bit)	See command #BA on page 58

LCedit+: Loaded when opening the Global Settings panel

General form: LanBox reply:

`* 0B #`

`* BR DMOF DMNR NS NETWORKNAME SY IPADDRES IPSUBNET IPGATEWA DL DIDO DISO DICC UL UDPINADD UIPO UIDS UISS UICC UOPO UOSS UOCC UF NTPS MTCS NTPIPADD NTPTIOF CLFRTU #>`

BR	Serial Port Baud Rate (8bit, 80-84)
DMOF	DMX Out Offset (16bit, 0-512)
DMNR	Number off DMX Channels (16bit, 0-512)
NS	Network Name Length (8bit, 0-13)
NETWRKNAME	Network Name, 13x 8bit
SY	SysEx Device ID (8bit, 0-127)
IPADDRES	LanBox IP Addres (4x 8bit)
IPSUBNET	LanBox Subnet Mask (4x 8bit)
IPGATEWA	LanBox Standard Gateway (4x 8bit)
DL	DMX Input Destination Layer (8bit, 1-255)
DIDO	DMX Input Destination Offset (16bit, 0-3072)
DISO	DMX Input Source Offset (16bit, 0-3072)
DICC	DMX Input Size (8bit, 0-512)
UL	UDP In Destination Layer (8bit, 1-255)
UDPINADD	UDP In Source Address (4x 8bit)
UIPO	UDP In Port (16bit, 0-65535)
UIDS	UDP In Destination Start Channel (16bit, 0-3072)
UISS	UDP In Source Start Channel (16bit, 0-3072)
UICC	UDP In Channel Count (16bit, 0-3072)
UOPO	UDP Out Port (16bit, 0-65535)
UOSS	UDP Out Source Start Channel (16bit, 0-3072)
UOCC	UDP Out Channel Count (16bit, 0-3072)
UF	UDP Out Flags (See Table 9)
NTPS	Clock to Channel (16bit, 0-3066)
MTCS	MTC to Channel (16bit, 0-3066)
NTIPADR	NTP time server IP address (4x8bit)
NTPTIOF	Optional: NTP time offset (32bit)
CLFRTU	Clock Frequency Tuning (32bit)

Get the Global Settings of the LanBox

Example: LanBox reply:

`* 0B #`

`* 83 0000 0200 08 44656D6F2D4C43584358676874 02 C0A8014C FFFFFF00 C0A80101 00 0001 0001 0000 FE 00000000 12A9 0001 0001 0200 12A9 0001 0200 02 0101 0202 00000000C7E41E800288CE70 #>`

83	Baud Rate Fixed MIDI, 31250 baud
0000	DMX Out Offset 0 (0x000 hex)
0200	512 DMX Channels (0x0200 hex)
08	Network Name Length = 8 (0x08 hex)
44656D6F2D4C43584358676874	ASCII: 68 101 109 111 45 76 67 88 (67 88 103 104 116) Text: Demo-LCX (Cxght) (Only 1st 8 characters)
02	SysEx Device ID 2 (0x02 hex)
C0A8014C	IP Address 192.168.1.77
FFFFFF00	Subnet Mask 255.255.255.0
C0A80101	Standard Gateway 192.168.1.1
FE	DMX Input Destination Layer 254 (Mixer) (0xFE hex)
0201	DMX Input Destination Offset 513 (0x0201 hex)
0001	DMX Input Source Offset 1 (0x0001 hex)
0200	DMX Input Size 512 (0x0200 hex)
FE	UDP In Destination Layer 254 (Mixer) (0xFE hex)
00000000	UDP In Source Address 0.0.0.0 (anyone)
12A9	UDP In Port 4777 (0x1249 hex)
0401	UDP In Destination Start Channel 1025 (0x0401 hex)
0001	UDP In Source Start Channel 1 (0x0001 hex)
0200	UDP In Channel Count 512 (0x0200 hex)
12A9	UDP Out Port 4777 (0x1249 hex)
0001	UDP Out Source Start Channel 1 (0x0001 hex)
0200	UDP Out Channel Count 512 (0x0200 hex)
02	UDP Out Flags: Broadcast Mixer and DMX In
0BB9	Clock Start Channel 3001 (0x0BB9)
0BBF	MTC Start Channel 3007 (0x0BBF hex)
00000000	NTP server IP 0.0.0.0 (none)
C7E41E80	NTP Time Offset 0
0288CE70	Clock Frequency Tuning 0,9000

Get the Global Settings of the LanBox

#0006 CommonSetBaudRate

The Baud Rate of the serial (MIDI) port of the LanBox can be set to several speeds. To do this the **CommonSetBaudRate** command is used. The command number is 6 (0x0006 hex). Note the extra 0's in the command number. The command has one parameter, the Baud Rate (see Table 8). The default Value is 31250 Baud. If the Baud Rate setting is 128-131 (0x80 – 0x 83 hex), the Serial Output is set to MIDI and will always have a Baud Rate of 31250. The LanBox will have to be rebooted before this setting will take effect!

LCedit+: Changing the Baud Rate in the Global Settings panel

parameter	Baud Rate	
0	38400	
1	19200	
2	9600	
3	31250 (MIDI)	Default

Table 8, LanBox Serial Port Baud Rates

General form: LanBox reply:

* 0006 BR# > no reply

BR Baud Rate, see Table 8

Set the Baud Rate of the LanBox Serial MIDI Port to BR

Example: LanBox reply:

* 0006 02 # > no reply

02 Baud Rate 9600 (See Table 8)

Set the Baud Rate of the LanBox Serial MIDI Port to 9600

#6A CommonSetDMXOffset

With the **CommonSetDMXOffset** command , the DMX Channel Offset with which the LanBox broadcasts is set. The DMX addresses up to this number will be assigned 0. The command number is 106 (0x6A hex) and has one parameter: the DMX Channel Offset (16bit, 1-512). The LanBox default Value is 0.

LCedit+: Changing the DMX Output settings in the Global Settings panel

General form: LanBox reply:

* 6A DMOF # > no reply

DMOF DMX Out Offset (16bit, 0-512)

Set the DMX Out Offset of the LanBox to DMOF

Example: LanBox reply:

* 6A 0100 # > no reply

0100 Offset 256 (0x0100 hex)

Set the DMX Out Offset of the LanBox to 256

#69 CommonSetNumDMXChannels

With the **CommonSetNumDMXChannels** command , the number of DMX Channels which the LanBox broadcasts is set. The command number is 105 (0x69 hex) and has one parameter: the Number of DMX Channels (16bit, 0-512). The LanBox default Value is 512.

LCedit+: Changing the DMX Output settings in the Global Settings panel

General form:	LanBox reply:
* 69 DMNR #	> no reply

DMNR Number of Channels (16bit, 0-512)

Set the Number of DMX Out Channels of the LanBox to DMNR

Example:	LanBox reply:
* 69 00FF #	> no reply

00FF 255 Channels (0x00FF hex)

Set the Number of DMX Out Channels of the LanBox to 255

#AE CommonSetName

With the **CommonSetName** command you can change the Network Name of the LanBox. This name has a maximum length of 13 characters (including spaces). The command number is 174 (0xAE hex) and has a maximum of 13 parameters: the ASCII codes of the characters of the new name. <u>The LanBox will have to be rebooted before this setting will take effect!</u>

LCedit+: Changing the LanBox Name in the Global Settings panel

General form:	LanBox reply:
* AE L1 L2 L3 L4 L5 L6 L7 L8 L9 LA LB LC LD #	> no reply

L1-LD Characters ACII codes (8bit, 0-255)

Set a new name for the LanBox

Example:	LanBox reply:
* AE 4C 61 6E 62 6F 78 20 4C 43 58 31 #	> no reply

4C	“L” (ASCII code 76, 0x4C hex)
61	“a” (ASCII code 97, 0x61 hex)
6E	“n” (ASCII code 110, 0x6E hex)
62	“b” (ASCII code 98, 0x62 hex)
6F	“o” (ASCII code 111, 0x6F hex)
78	“x” (ASCII code 120, 0x78 hex)
20	“ ” (ASCII code 32, 0x20 hex)
4C	“L” (ASCII code 76, 0x4C hex)
43	“C” (ASCII code 67, 0x43 hex)
58	“X” (ASCII code 88, 0x58 hex)
31	“1” (ASCII code 49, 0x31 hex)

Set a new name for the LanBox to “Lanbox LCX1”

#AF CommonSetPassword

With the **CommonSetPassword** command you can set the LanBox' password. This password consists of one 16bit number, so can be anything from 0 to 65535. The command number is 175 (0xAF hex) and has one parameter: the new password (16bit, 0-65535).

LCedit+: Changing the LanBox Password in the Global Settings panel

General form:

* AF PSWD #

PSWD	New Password (16bit, 0-65535

Set a new Password for the LanBox

LanBox reply:

> no reply

Example:

* AF 03F4 #

03F4	1012 (0x03F4 hex)

Set a new password for the LanBox to 1012

LanBox reply:

> no reply

⚠ **Use with extreme care, this function can severely disrupt the proper operation of the LanBox**

#B0 CommonSetIpConfig

The **CommonSetIpConfig** is used to change the IP settings of the LanBox. The command number is 176 (0xB0 hex) and has 3 parameters: the LanBox IP address (4x8bit), the Sub net mask (4x8bit) and the Standard Gateway IP (4x8bit). The default Values are 192.168.1.77 for the LanBox IP, 255.255.255.0 for the Sub Net Mask and 192.168.1.1 for the Standard Gateway.

LCedit+: Changing the LanBox network settings in the Global Settings panel

General form:

* B0 IPADDRES IPSUBNET IPGATEWA #

IPADDRES	(4x8bit)
IPSUBNET	(4x8bit)
IPGATEWA	(4x8bit)

Set IP settings of the LanBox

LanBox reply:

> no reply

Example:

* B0 C0A8014C FFFF0000 C0A80101 #

C0A8014C	192 (0xC0 hex)
	168 (0xA8 hex)
	1 (0x01 hex)
	76 (0x4C hex)
FFFF0000	255 (0xFF hex)
	255 (0xFF hex)
	0 (0x00 hex)
	0 (0x00 hex)
C0A80101	192 (0xC0 hex)
	168 (0xAB hex)
	1 (0x01 hex)
	1 (0x01 hex)

Set IP settings of the LanBox to 192.168.1.76 for the LanBox IP, 255.255.0.0 for the Sub Net Mask and 192.168.1.1 for the Standard Gateway.

LanBox reply:

> no reply

⚠ **Use with extreme care, this function can severely disrupt the proper operation of the LanBox**

#B2 CommonSetDmxIn

With the **CommonSetDmxIn** command the way the LanBox handles DMX in signals can be set. The command number is 178 (0xB2) and has 4 parameters: the destination Layer (8bit, 1 -63 for A - BK, see Table 2 on page 10 for Buffer ID's), The Destination Start Channel (8bit, 0-3072), the Source Start Channel (8bit, 0-512) and the number of DMX channels to be copied (8bit, 0-512).

LCedit+: Changing the DMX in settings in the Global Settings panel

General form:	LanBox reply:
* B2 DL DIDO DISO DICC #	> no reply

DL	DMX Input Destination Layer (8bit, 1-255)
DIDO	DMX Input Destination Start (16bit, 0-3072)
DISO	DMX Input Source Start (16bit, 0-512)
DICC	DMX Input Size (8bit, 0-512)

Copy DMX In Channels DISO to (DISO+DICC) to Channel DIDO to (DIDO+DICC) in Layer DL

Example:	LanBox reply:
* B2 03 0201 0001 0200 #	> no reply

03	Destination Layer C (ID 3, 0x03 hex)
0201	Dest Start Channel 513 (0x0201 hex)
0001	Input Source Start 1 (0x0001 hex)
0200	copy 512 Channels (0x0200 hex)

Copy DMX In Channels 1 to 512 to Channel 513 to 1024 in Layer C

#B8 CommonSetUdpIn

The LanBox can copy incoming UDP Values to Channels in one of its Layers or buffers (see the section UDP Data for more details). The **CommonSetUdpIn** command sets the UDP In settings of the LanBox. The command number is 184 (0xB8 hex) and has 6 parameters: the destination Layer (8bit, 1 -63 for A - BK, see Table 2 on page 10 for Buffer ID's), the Source IP address (4x8bit, default 0.0.0.0 for anyone), the UDP In Port (16bit, 0-65535, default is 4777), The Destination Start Channel (8bit, 0-3072), the Source Start Channel (8bit, 0-3072) and the number of DMX channels to be copied (8bit, 0-3072). To use Art-Net, choose UDP port 6454.

LCedit+: Changing the UDP in settings in the Global Settings panel

General form:	LanBox reply:
* B8 UL UDPINADD UIPO UIDS UISS UICC #	> no reply

UL	UDP Input Destination Layer (8bit, 1-255)
UDPINADD	Source IP address (4x 8bit)
UIPO	UDP In Port (16bit, 0-65535)
UIDS	UDP In Destination Start (16bit, 0-3072)
UISS	UDP In Source Start (16bit, 0-3072)
UICC	UDP Input Size (8bit, 0-3072)

Receive UDP from UDPINADD, port UIPO and copy UDP In Channels UISS to (UISS+UICC) to Channels UIDS to (UIDS+UICC) in Layer UL

Example:	LanBox reply:
* B8 2B C0A80164 12A9 0201 0001 0200 #	> no reply

2B	Destination Layer AQ (ID 43, 0x2B hex)
C0A80164	192 (0xC0 hex) 168 (0xA8 hex) 1 (0x01 hex) 100 (0x64 hex)
12A9	Port 4777 (0x12A9 hex)
0201	Dest Start Channel 513 (0x0201 hex)
0001	Input Source Start 1 (0x0001 hex)
0200	copy 512 Channels (0x0200 hex)

Receive UDP from 192,168,1,100, port 4777 and copy UDP In Channels 1 to 512 to Channels 513 to 1024 in Layer AQ

#B9 CommonSetUdpOut

The LanBox can Broadcast all of its Values over UDP. This is set up with the **CommonSetUdpOut** command. With the UDP Out Flags you can what is broadcasted over UDP. The command number is 184 (0xB9 hex) and has 4 parameters: The UDP Out Port (16bit, 0-65535, default is 4777), the UDP Out Source Start Channel (16bit, 0-3072), the UDP out Channel Count (8bit, 0-3072) and the UDP Out Flags (8bit, see Table 9).

LCedit+: Changing the UDP Out settings in the Global Settings panel

bit	Value	UDP Out Setting
0	1	Broadcast DMX Out
1	2	Broadcast Mixer Channels
2	4	Broadcast External Input Values
3	8	Broadcast DMX In
4	16	Broadcast Layer List
5	32	Synchronize Layers with Same ID
6-7		Reserved

Table 9, LanBox UDP Out Flags

General form:

* B9 UOPO UOSS UOCC UF #

LanBox reply:

> no reply

UOPO	UDP Out Port (16bit, 0-65535)
UOSS	UDP Out Source Start (16bit, 0-3072)
UOCC	UDP Output Size (8bit, 0-3072)
UF	UDP Out Flags (8bit, see Table 9)

Broadcast Channels UOSS through (UOSS+UOCC) to UDP Port UOPO, using Flags UF

Example:

* B9 12A9 0201 02D0 0200 #

LanBox reply:

> no reply

12A9	Port 4777 (0x12A9 hex)	
0201	Source Start Channel 513 (0x0201 hex)	
02D0	Output Size 720 (0x02D0 hex)	
06	Binary: 0 0 0 0 1 0 1 0	
	Broadcast DMX out =	Off (0)
	Broadcast Mixer Channels =	On (1)
	Broadcast External Input =	Off (0)
	Broadcast DMX in =	On (1)
	Broadcast Layer List =	Off (0)
	Synchronize Layers =	Off(0)

Broadcast channels 513 through 1233 to UDP port 4777. Set Flags to broadcast Mixer Channels and DMX in

#BA CommonSetTime

The LanBox has an Internal Clock whose settings can be changed using the **CommonSetTime** command. The command number is 186 (0xBA hex) and has 2 to 5 parameters: the Start Channel to which the internal clock time (6 Values) is copied (16bit, 0-3066), the Start Channel to which the MIDI Time Code time (6 Values) is copied (16bit, 0-3066) and optional the NTP time server IP address (4x8bit, 0.0.0.0 to set no NTP server), the NTP time offset (32bit NTP epoch) and Clock Frequency Tuning (32bit, 2^{32} x time per internal 10ms tick).

The NTP time epoch is the number of seconds that has passed since January 1st 1900, 0:00. The LanBox default is set to Monday, April 10th 2006, 0:00 (3353616000, 0xC7E41E80 hex). The time offset is expressed as a new time epoch, the LanBox will take the difference with the default time as the Offset. Weeks, years etc. are discarded. See the example inTable 10. When setting a NTP server, be aware that the LanBox updates its time every 10 minutes. It is considered 'bad practice' to do this on a main NTP server, always select a server 'closest to home' (preferably on your own network, else your ISP's NTP server).

LCedit+: Changing the Time settings in the Global Settings panel

	hexadecimal	decimal
Default time	C7E41E80	3353616000
Set time	C7E40260	3353608800
Difference	00001C20	7200 seconds
		/3600
		2 hours

Table 10, Time Offset Example

General form:

* BA NTPS MTCS
[NTIPADDR NTPTIOF CLFRTU] #

LanBox reply:

> no reply

NTPS	Clock to Channel (16bit, 0-3066)
MTCS	MTC to Channel (16bit, 0-3066)
NTIPADR	Optional: NTP time server IP address (4x8bit)
NTPTIOF	Optional: NTP time offset (32bit NTP epoch)
CLFRTU	Optional: Clock Frequency Tuning (32bit, 2^{32} x time per internal 10ms tick)

Copy Internal Clock Values to channels NTPS through (NTPS+ 5), Copy MTC Clock Values to channels MTCS through (MTCS + 5) (And set NTP server to NTIPADR, give time offset NTPTIOF and adjust Clock Frequency to CLFRTU)

Example:

* BA 0401 0407 C2A80133 C7E472E0 02858794 #

LanBox reply:

> no reply

0401	Clock to Channel 1025 (0x0401 hex)	
0407	MTC to Channel 1031 (0x0407 hex)	
C2A80133	NTP server	192 (0xC0 hex) 168 (0xA8 hex) 1 (0x01 hex) 50 (0x33 hex)
C7E472E0	3353616000 (0xC7E41E80 hex) 3353637600 (0xC7E472E0 hex)	
	- 21600	/3600
	- 6	> GMT -6 hours
02858794	42305428 (0x02858794 hex) / 2^{32} = 0,00985s/ 10ms tick	

Copy Internal Clock Values to channels 1025 through 1030, Copy MTC Clock Values to channels 1031 through 1036, Set NTP server to 192.168.1.50, give time offset -6 hours and adjust Clock Frequency to 0,985*10ms.

#A0 CommonGet16BitTable

The LanBox can link pairs of Channels to make up one 16bit Channel. This way, when the High Channel is changed or faded, the Low Channel will follow accordingly. The Channel pairs are stored in an internal Table, which can be retrieved with the **CommonGet16BitTable** command. The command number is 160 (0xA0 hex) and has no parameters. The reply will consist of **n** pairs of Channels with first the High Channel (16bit, 0 – 3072), then the Low Channel (16bit, 0 – 3072).

LCedit+: Loaded when opening the Global Settings panel

General form:

```
* A0 #
```

LanBox reply:

```
* HCP1 LCP1 HCP2 LCP2 .. .. HCPn LCPn #>
```

HCP1	Pair 1 High Channel (16bit, 0 – 3072)
LCP1	Pair 1 Low Channel (16bit, 0 – 3072)
HCP2	Pair 2 High Channel (16bit, 0 – 3072)
LCP2	Pair 2 Low Channel (16bit, 0 – 3072)
.. ..	
HCPn	Pair **n** High Channel (16bit, 0 – 3072)
LCPn	Pair **n** Low Channel (16bit, 0 – 3072)

Return list of 16bit Channel Pairs

Example:

```
* A0 #
```

LanBox reply:

```
* 0101 0102 0103 0104 0108 0109 #>
```

0101	Channel 257 (0x0101 hex)
0102	Channel 258 (0x0102 hex)
0103	Channel 259 (0x0103 hex)
0104	Channel 260 (0x0104 hex)
0108	Channel 264 (0x0108 hex)
0109	Channel 265 (0x0109 hex)

Return list of 16bit Channel Pairs

#A1 CommonSet16BitTable

With the **CommonSet16BitTable** command you can set or delete 16bit Channel pairs. The command number is 161 (0xA1 hex) and has **n** x 3 parameters: A mode (8bit, 0 = delete, >0 = set), the High Channel (16bit, 0 – 3072) and the Low Channel (16bit, 0 – 3072). Note that the High and Low Channel can't be more then +7 or -8 Channels apart.

LCedit+: Editing the 16bit table in the Global Settings panel

```
General form:                                      LanBox reply:

   * A1 M1 HCP1 LCP1 M2 HCP2 LCP2 .. ..               > no reply
            Mn HCPn LCPn #
```

M1	Mode Pair 1 (8bit, 0 = delete, >0 = set)
HCP1	Pair 1 High Channel (16bit, 0 – 3072)
LCP1	Pair 1 Low Channel (16bit, 0 – 3072)
M2	Mode Pair 2 (8bit, 0 = delete, >0 = set)
HCP2	Pair 2 High Channel (16bit, 0 – 3072)
LCP2	Pair 2 Low Channel (16bit, 0 – 3072)
.. ..	
Mn	Mode Pair **n** (8bit, 0 = delete, >0 = set)
HCPn	Pair **n** High Channel (16bit, 0 – 3072)
LCPn	Pair **n** Low Channel (16bit, 0 – 3072)

Set **n** pairs of HCPn and LCPn, mode Mn

```
Example:                                            LanBox reply:

   * A1 FF 01E8 01E9 FF 0150 0157 00 018A 0185 #      > no reply
```

FF	Mode = set
01E8	Channel 488 (0x01E8 hex)
01E9	Channel 489 (0x01E9 hex)
FF	Mode = set
0150	Channel 336 (0x0150 hex)
0157	Channel 343 (0x0157 hex)
00	Mode = delete
018A	Channel 394 (0x018A hex)
0185	Channel 389 (0x0185 hex)

Pair up channels 488 & 489, Channels 336 & 343. Remove pair Channel 394 & 389

#A2 CommonStore16BitTable

The **CommonStore16BitTable** stores the 16bit Channel Table so it will be available when rebooting the LanBox. The command number = 162 (0xA2 hex) and has no parameters.

LCedit+: Done when saving LanBox Data

```
General form:                                      LanBox reply:

   * A2 #                                             > no reply
```

Store 16bit Table

#A3 CommonGetMIDIMapping

In the LanBox each MIDI Channel can be assigned to a Layer. This mapping is saved in the MIDI mapping Table. The **CommonGetMIDIMapping** command is used to retrieve this Table. The command number is 163 (0xA3 hex) and has no parameters. The command will return 2 Values for all 16 MIDI channels. The connected Layer ID (8bit, 1 -63 for A - BK) and the offset (16bit, 0-3072).

LCedit+: Loaded when opening the Global Settings panel

General form:

* A3 #

Return MIDI Mapping

LanBox reply:

* L1 OFF1 L2 OFF2 .. .. L0 OFFF #>

L1	Layer Channel 1 (8bit, 1 -63 for A - BK)
OFF1	Channel 1 Offset (16bit, 0 – 3072)
L2	Layer Channel 2 (8bit, 1 -63 for A - BK)
OFF2	Channel 2 Offset (16bit, 0 – 3072)
.. ..	
L0	Layer Channel 16 (8bit, 1 -63 for A - BK)
OFF0	Channel 16 Offset (16bit, 0 – 3072)

Example:

* A3 #

Return MIDI Mapping

LanBox reply:

* 04 0000 04 09F6 .. .. 35 0200 #>

04	Channel 1 to Layer D (ID 4, 0x04 hex)
0000	Channel 1 Offset 0 (0x0000 hex)
04	Channel 2 to Layer D (ID 4, 0x0004 hex)
09F6	Channel 2 Offset 127 (0x09F6 hex)
.. ..	
35	Channel 16 to Layer BA (ID 53, 0x35 hex)
0200	Channel 16 Offset 512 (0x0200 hex)

#A4 CommonSetMIDIMapping

With the **CommonSetMIDIMapping** command, the mapping of a MIDI Channel can be set. The command number has 3 parameters: The MIDI Channel (8bit, 0 – 16), the Layer ID (8bit, 1 -63 for A - BK) and the offset (16bit, 0-3072).

LCedit+:Editing the MIDI Mapping the Global Settings panel

General form:

* A4 MC LA OFFS #

MC	MIDI Channel (8bit, 0 – 16)
LA	Layer ID (8bit, 1 -63 for A - BK)
OFFS	Offset (16bit, 0-3072)

Map MIDI Channel MC to Layer LA with an offset of OFFS

LanBox reply:

> no reply

Example:

* A4 03 05 007F #

03	MIDI Channel 3 (0x03 hex)
05	Layer E (ID 5, 0x05 hex)
007F	Offset 127 (0x007F hex)

Map MIDI Channel 3 to Layer E with an offset of 127

LanBox reply:

> no reply

#A5 CommonStoreMIDIMapping

The **CommonStore16BitTable** stores MIDI Mapping Table so it will be available when rebooting the LanBox. The command number = 165 (0xA5 hex) and has no parameters.

LCedit+: Done when saving LanBox Data

General form:	LanBox reply:
* A5 #	> no reply

Store MIDI Mapping

#B3 CommonGetDigOutPatcher

In order to use the Digital Output of the LanBox, they have to be linked to unused Mixer Channels. When these channels are set to 0, the port will work like a standard RS232 port, when it is set to 255 the output will be inverted. With the **CommonGetDigOutPatcher** command the Patch Table can be retrieved. The command number is 179 (0xB3 hex) and has 2 parameters: the first Port Number of which the patched Channel is to be returned (8bit, 1 -8) and the number of ports to be returned (8bit, 0-8, 0 for all ports). The reply consists of **n** Mixer Channel Values (16bit 0 – 3072).

Lcedit+Loaded when opening the Global Settings panel

General form:

* B3 S1 NR#

S1	First Port Number (8bit, 1 - 8)
NR	Number of Ports (8bit, 0 - 8)

LanBox reply:

* SER1 SER2 .. SERn #>

SER1	1st Serial Port Channel (16bit, 0 – 3072)
SER2	2nd Serial Port Channel (16bit, 0 – 3072)
.. ..	
SER**n**	nth Serial Port Channel (16bit, 0 - 3072)

Return Digital Out Patch of NR ports, starting at Serial Port SER1

Example:

* B3 01 03 #

01	Port 1 (0x01 hex)
03	3 Ports (0x03 hex)

LanBox reply:

* 0B78 0B79 0B7A #>

0B78	Channel 2936 (0x0B78 hex)
0B79	Channel 2937 (0x0B79 hex)
0B7A	Channel 2938 (0x0B7A hex)

Return Digital Out Patch of 3 ports, starting at Serial Port 1

#B4 CommonSetDigOutPatcher

With the **CommonSetDigOutPatcher** command, the Digital outputs are patched to Mixer Channels. The command number is 180 (0xB4 hex) and has **n** x 2 parameters: the Serial Port Number (8bit, 0 - 8) and the Mixer Channel that is to be patched to this Port (16bit, 1 – 3072).

LCedit+: Editing the Digital Out in the Global Settings panel

General form:

* B4 S1 CHA1 S2 CHA2 .. .. Sn CHAn #

S1	Port Number 1 (8bit, 1 - 8)
CHA1	1st Serial Port Channel (16bit, 0 – 3072)
S2	Port Number 2 (8bit, 1 - 8)
CHA2	2nd Serial Port Channel (16bit, 0 – 3072)
.. ..	
Sn	Port Number **n** (8bit, 1 - 8)
CHAn	nth Serial Port Channel (16bit, 0 - 3072)

LanBox reply:

> no reply

Patch Serial Outputs S1 through Sn to Mixer Channels CHA1 through CHAn

Example:

* B4 01 06BF 02 06C0 #

01	Port 1 (0x01 hex)
06BF	Channel 1727 (0x06BF hex)
02	Port 1 (0x01 hex)
06C0	Channel 1728 (0x06C0 hex)

LanBox reply:

> no reply

Patch Serial Outputs 1 and 2 to Mixer Channels 1727 and 1728

Debug Commands

#A8 CommonResetNonVolatile

The **CommonResetNonVolatile** erases all Cue List, Tables and Channel Data from the LanBox and sets it to the Default Values. The command number is 168 (0xA8 hex) and has no parameters.

Lcedit+: erasing the LanBox

General form:	LanBox reply:
* A8 #	> no reply
Erase the LanBox	

⚠ **Use with extreme care, this function can severely disrupt the proper operation of the LanBox**

#DD DebugGetTotalUsage

With the **DebugGetTotalUsage** command the amount of used Memory in the LanBox can be queried. The command number is 221 and has no parameters. The reply consists of 2 32bit numbers: the used memory and the total memory.

Lcedit+:LanBox Utilities> List

General form:	LanBox reply:	
* DD #	* MEMUSE MEMTOT #>	
	MEMOUSED	Used Memory (32bit)
	MEMTOTAL	Total Memory (32bit)
Get Memory Usage of the LanBox		

Example:	LanBox reply:	
* DD #	* 00004C78 0005CA20 #>	
	00004C78	19,576 bytes used
	0005CA20	379,424 bytes total
Get Memory Usage of the LanBox		

#DE DebugGetFreeList

The **DebugGetFreeList** returns a list of unused Memory Regions in the LanBox' memory. The command number is 222 (0xDE hex) and has no parameters. The reply consists of **n** x a Memory Address (32bit) and an amount of free bytes at that Address (32bit).

Lcedit+:LanBox Utilities> List

General form:

* DE #

LanBox reply:

* ADDRESS1 FREEBYT1 … ... #>

ADDRESS1	1st address with free bytes (32bit)
FREEBYT1	Number of free bytes at 1st address (32bit)
…	
…	

List Free Memory in the LanBox

Example:

* DE #

LanBox reply:

* 000A1A1C 00000008 000A35B0 00000018 000A35D8 00000008 000A3600 00000008 000A3628 00000008 000A38B 00005C748 #>

000A1A1C	Address 662044
00000008	8 Bytes free
000A35B0	Address 669104
00000018	24 Bytes free
000A35D8	Address 669144
00000008	8 Bytes free
000A3600	Address 669184
00000008	8 Bytes free
000A3628	Address 669224
00000008	8 Bytes free
000A38B0	Address 669872
00005C748	378696 Bytes free

List Free Memory in the LanBox

#DF DebugGetCuelistUsage

The **DebugGetCueListUsage** command returns the location and size of specific Cue List Data in the LanBox' memory. The command number is 223 (0xDF hex) and has 2 parameters: the Cue List number (32 bit, 1 – 3072) and a Cue Step Number (8bit, 0 – 99). If the Cue Step Number is 0 (0x00 hex), the reply will give a 32bit memory location and and 2 16bit Values giving the reserved Memory and used Memory of all the Cue Step Type Data of the Cue List. If a Cue Step Number is not 0, the reply will be the 32bit memory location and 2 16bit Values giving the reserved Memory and used Memory of the Cue Step Data linked to that Cue Step. The Reserved and Used Memory should always be identical except if something went wrong while saving Cue List Data.

Lcedit+:LanBox Utilities> List

General form:

* DF CLIS CS #

CLIS	Cue List Number (16bit, 1-999)
CN	Cue Step number (8bit, 1-99)

LanBox reply:

* ADDRESS RESM USEM #>

ADDRESS_	Memory Address
RESM	Number of bytes reserved (16bit)
USEM	Number of bytes used (16bit)

List Memory in the LanBox used by Cue List CLIS, step CS

Example:

* DF 0001 02#

0001	Cue List (0x0001 hex)
01	Cue Step 1 (0x01 hex)

LanBox reply:

* 000A3618 0010 0010 #>

000A3618	Address 669208
0010	16 Bytes Reserved
0010	16 Bytes Used

List Memory in the LanBox used by Cue List 1, step 1

MIDI Notes

Note-On messages

All LanBox Channels can be controlled through MIDI Note-On and Note-Off messages. The Pitch of a Note-On message determines the Channel, the Velocity determines the Value. Because both Pitch and Velocity are 7bit Values and therefore can't exceed Values beyond 127, the LanBox has two mechanisms build in to handle these shortcomings.

Multiple MIDI channels can be assigned to one Layer, each with an offset. This way, it is possible to control up to 2032 (16 x 127) different Channels with the Note-On message. The MIDI Channel assignment and offset is set in the Global Settings panel in LCedit+ or with the **MIDIMapping** commands (command numbers A3 – A5).

The LanBox multiplies the Velocity of a Note-On Channel with 2, except for a Velocity of 127 which is fixed to a Value of 255. So a Velocity of 67 will set the Channel Value to 134.

Note: MIDI controllers do this automatically when making the Note-On message, but when you use software, the following is useful to know.

A MIDI message is a series of 3 numbers: the Message Type, the Pitch and the Velocity. For a Note-On message, the Message Type is 143 + the MIDI Channel. So Message Type 144 denotes a Note-On on MIDI Channel 1, Message Type 145 denotes a Note-On on MIDI Channel 2, etc.

Note-Off messages

Note-Off messages work the same as Note-On messages except for the fact that the Velocity is ignored and the relevant Channel is always set to 0.

For a Note-Off message, the Message Type is 127 + the MIDI Channel. So Message Type 128 denotes a Note-Off on MIDI Channel 1, Message Type 129 denotes a Note-Off on MIDI Channel 2, etc.

MIDI Program Change Messages

A Program Change message will start a Cue List which is determined by the Program Change value in the Layer to which the MIDI Channel is assigned. Because these values can only range from 0 to 127, the MIDI Bank Select command can be used to set the Bank. This will be the MSB of the Cue List Number. So a Program Change of 110 on Bank 3 will be Cue List number: (128 x 3) + 110 = 494.

MIDI Control Messages

A lot of functions of the LanBox can be controlled with MIDI Control Change (CC) Messages. In order to use MIDI CC Messages, MIDI Channels have to be linked to Layers in the LanBox. This is done in the Global Settings panel of LCedit+ or with the **MIDIMapping** commands (command numbers A3 – A5).

Messages

Layer General Control

Layer Playback Control

Layer Chase Control

Layer Real Time Editing

Layer MIDI Settings

Layer General Control

CC70 LayerSetFading

With the LayerSetFading command, you can determine if a Layer's fading should be disabled or enabled. The Controller Number is 70 and a Value < 64 will switch the Fading Off, a Value => 64 will switch it On.

Controller Number	Value	Action
70	0-63	Layer Fading Off
	64-127	Layer Fading On

CC71 LayerSetMixMode

Every Layer in the LanBox has several possible Mix Modes. The Mix Mode of a Layer can be set with the **LayerSetMixMode** command. The Controller Number is 71 and a the Value determines the Layer's Mix Mode.

Controller Number	Value	Action	
71	0	Off	Layer does not output anything.
	1	Copy	Knock Out Mode. Overwrites all Values of matching channels in underlying Layers.
	2	HTP	Highest takes Precedence. The Highest Value of a Channel is output to the Mixer.
	3	LTP	Lowest takes Precedence. The lowest Value of a Channel is output to the Mixer.
	4	Transparent	A percentage of a Value is added to the Value of underlying Layers.
	5	Add	Makes Values relative to underlying Layers. (Values – 128) is added to the Value of underlying Layers.
	1 - 127	No Action	

CC72 LayerSetOutput

With the **LayerSetOutput** command, you can determine if a Layer outputs its Values to the Mixer. The Controller Number is 72 and a Value < 64 will switch the Output Off, a Value => 64 will switch it On.

Controller Number	Value	Action
72	0-63	Layer Output Off
	64-127	Layer Output On

CC 74 LayerSetSolo

With the **LayerSetSolo** command, you can determine if a Layer's Solo mode should be disabled or enabled. If a Layer is in Solo Mode, only the Channels which are set to solo will be active. The Controller Number is 74 and a Value < 64 will switch the Solo Mode Off, a Value => 64 will switch it On.

Controller Number	Value	Action
74	0-63	Layer Solo Mode Off
	64-127	Layer Solo Mode On

CC99 LayerSetTransparencyDepth

With the **LayerSetTransparencyDepth** command, you can set the factor by which the Value of a Channel is multiplied and added to the underlying Channel Value. This will only work if the relevant Layer's Mixing Mode is set to Transparent. So if a Layer is set to 50% transparent and a Channel is set to 255 in that Layer, a Value off 127 will be added to the underlying Value (to a maximum of 255). The Controller Number is 99 and a the Value determines the Layer Transparency Depth, where 0 = 0% and 127 = 100%

Controller Number	Value	Action
99	0 - 127	Layer Transparency Depth 0% - 100%

CC100 LayerSetAutoOutput

With the **LayerSetAutoOutput** command, you can determine if a Layer should automatically activate Channels if they change within a Layer. If Auto Output is disabled, a Channel's Value can be set or changed without actually activating it in the Layer. The Controller Number is 100 and a Value < 64 will switch the Auto Output Off, a Value => 64 will switch it On.

Controller Number	Value	Action
100	0-63	Layer Auto Output Off
	64-127	Layer Auto Output On

Layer Playback Control

In order to control the playback of Cue Lists in a Layer, the LanBox supports a number of playback commands.

CC20&52 LayerGoCueList

The **LayerGoCueList** command starts a Cue List in a Layer at the first Cue Step. This command has 2 Controller Change Numbers, making a 14bit number possible. Control Number 20 is the MSB, Control number 52 is the LSB. The MSB should be sent first, the LSB triggers the Go action. To start at a different Cue Step then the first, use the **LayerGoCueStep** command in combination with the **LayerUsesCueList** command.

Example: CC 20 5
CC 52 47
Will Go Cue List (5 x 128) + 47 = 687

The MSB is remembered by the LanBox for each Layer, so if only the LSB is sent, the stored MSB is used.

Controller Number	Value	Action	Controller Number	Value	Action
20	0 - 127	Cue List MSB	52	0 - 127	Cue List LSB

CC21&53 LayerUsesCueList

The **LayerUsesCueList** command loads a Cue List in a Layer without taking any further action. This command has 2 Controller Change Numbers, making a 14bit number possible. Control Number 21 is the MSB, Control number 53 is the LSB. The MSB should be sent first, the LSB triggers the Load action.

Example: CC 21 1
CC 52 116
Will Go Cue List (1 x 128) + 116 = 244

The MSB is remembered by the LanBox for each Layer, so if only the LSB is sent, the stored MSB is used.

Controller Number	Value	Action	Controller Number	Value	Action
21	0 - 127	Cue List MSB	52	0 - 127	Cue List LSB

CC59 LayerResume

The **LayerResume** command will resume paused execution of a Cue List in a Layer. Action will be resumed again where it was originally Paused. The Control Number is 59, the Value is not used.

Controller Number	Value	Action
59	Not Used	Resume execution of a Cue List

CC86 LayerGoCueStep

The **LayerGoCueStep** will trigger a Go of a loaded Cue List at a designated Cue Step. To load a Cue List, use the **LayerUsesCueList** command. The Control Number is 86 and the Value designates the Cue Step.

Controller Number	Value	Action
86	1 - 99	Go Cue Step

CC87 LayerClear

The **LayerClear** Command deactivates all Channels in a Layer. Note that if there is still a Cue List running, this will activate Channels again (at least, when Automatic Output is On). The Control Number is 87, the Value is not used.

Controller Number	Value	Action
87	Not Used	Clear the Layer Channel Values

CC88 LayerPause

The **LayerPause** command halts the execution of a Cue List. The Layer will keep current Channel Values and will freeze fades. Action will be resumed again with a **LayerGo** or **LayerResume** command. The Control Number is 88, the Value is not used.

Controller Number	Value	Action
88	Not Used	Pause Layer execution

CC90 LayerNextStep

The **LayerNextStep** command will make a Layer Sequencer jump to the next Cue Step. The Control Number is 90, the Value is not used.

Controller Number	Value	Action
90	Not Used	Go next Cue Step

CC91 LayerPreviousStep

The **LayerPreviousStep** command will make a Layer Sequencer jump to the previous Cue Step. The Control Number is 91, the Value is not used.

Controller Number	Value	Action
91	Not Used	Go previous Cue Step

Layer Chase Control

In order to use Chases in a LanBox Layer, you have to load a Chase Cue Lists and set the Chase Mode of a Layer to On. A Chase Cue List is the same as a normal Cue List, but without Go or Loop Cue Steps. If a Layer is set into a Chase Mode, the Layer's sequencer will step through the Cue List Steps in the set order.

CC75 LayerSetChaseMode

The **LayerSetChaseMode** sets the way a Chase in a Layer is performed. The are 9 possible modes. The Chase Mode will only do something in a Layer if the Chase Mode attribute of a Layer is set to On. The Control number is 75 and the Value determines the Chase Mode

Controller Number	Value	Action	
75	0	Off	No chase (use Cue List)
	1	Chase Up	Steps through cue steps in increasing order
	2	Loop Up	Loops through cue steps in increasing order
	3	Chase Down	Steps through cue steps in decreasing order
	4	Loop Down	Loops through cue steps in decreasing order
	5	Random	Steps through cue steps in random order
	6	Loop Random	Loops through cue steps in random order
	7	Bounce	Steps through cue steps in alternating order
	8	Loop Bounce	Loops through cue steps in alternating order
	9 - 127	No Action	

CC76 LayerSetChaseSpeed

The **LayerSetChaseSpeed** you can set the speed at which a Layer's sequencer will run. If the speed is set at 100% the Sequencer will use the time set in the running Cue List. If not, it will multiply the designated time with the Speed Factor. This will only Layers in Chase Mode. The Speed Factor is calculated with the following formula: 128 / (255 - 2**S**) x 100%, where **S** is the Speed Value. If the Speed Value is 0, the Speed Factor is set to 50%, if it is 127, the Speed Factor is set to infinite, which in practice means that Cue Steps will be performed with the LanBox frame rate (20 steps/second). The Controller Number is 76 and a the Value determines the Layer Chase Speed , where 0 = 50% and 127 = infinite.

Controller Number	Value	Action
76	0 - 127	Layer Chase Speed 50% - infinite

CC77 LayerSetFadeType

Every Layer in the LanBox can be set to a Manual Fade. This means that Channel Values that are set 'Manually', which in practice means set with the **MIDI Note-On** command, will fade in or out. This Fade Type is set with the **LayerSetFadeType** command. The Control Number is 77, the Value determines the Fade Type.

Controller Number	Value	Action	
77	0	Off	No fade
	1	Fade In	Only fades when Channel Value increases
	2	Fade Out	Only fades when Channel Value decreases
	3	Cross Fade	Fades if Channel Value in- or decreases
	4	Off	No fade
	5	Fade In CR	Only fades when Channel Value goes up
	6	Fade Out CR	Only fades when Channel Value goes down
	7	Cross Fade CR	Fades if Channel Value goes up or down
	8 - 127	Loop Bounce	Loops through cue steps in alternating order

CR= At Constant Rate. Will make the LSB of 16bit Values increment at a constant rate. Will make fades smoother but less accurate.

CC78 LayerSetFadeTime

With the **LayerSetFadeTime** command, the Fade Time for the Layer's Manual Fade is set. The Control Number is 78 and the Value determines the Fade Time (8bit encoded, see Appendix A: Cue Step Time Encoding Table).

Controller Number	Value	Action
78	1 - 92	Fade Time code

Layer Real Time Editing

It is possible to edit Cue Lists using Control Change commands using so called Live Editing commands. In order to edit a Cue List, a Layer should be set to Edit Mode. After this, you will have to tell the Layer which Cue List to edit. This can be done with the **LayerGo** command, which will clear the engine and set all channels to the Channel Values as stored in the loaded Cue List. Alternatively, the **LayerUsesCueList** command can be used. This will cause the Layer to point at a Cue List, but not load its contents.

CC73 LayerSetEditRunMode

With the **LayerSetEditRunMode** command, a Layer's mode can be switched Edit Mode and will make Live Editing possible. The Control Number is 73 and a Value < 64 will put the Layer in Edit Mode, a Value => 64 will put it in Run Mode.

Controller Number	Value	Action
73	0-63	Layer in Edit Mode
	64-127	Layer in Run Mode

CC79 LayerSetCueStepType

With the **LayerSetCueStepType** command, the Cue Step Type of the current Cue Step in a Layer can be changed. In order to do this, the Layer should be in Edit Mode and a Cue List has to be loaded. The different Cue Step Types can be found in Appendix B: Cue Step Types. The Control Number is 79 and the Value determines the Cue Step Type.

Controller Number	Value	Action
79	1 - 56	Cue Step Type (see Appendix B)
	57 – 127	Not Used

CC80-85 LayerSetCueStepParameters

With the **LayerSetCueStepParameters** commands, you can set the Parameter bytes of each Cue Step. Use Control Number 80 for the first byte, 81 for the second, etc. The maximum number of Parameter bytes for each Cue Step is 6. See Appendix B: Cue Step Types for more information on Cue Steps. Note that only Data Values of 0 – 127 can be set using MIDI!

Controller Number	Value	Action
80	0-127	Cue Step Data 1
81	0-127	Cue Step Data 2
82	0-127	Cue Step Data 3
83	0-127	Cue Step Data 4
84	0-127	Cue Step Data 5
85	0-127	Cue Step Data 6

CC92 LayerInsertStep

With the **LayerInsertStep** command, an empty Cue Step is created and inserted. The Control Number is 92, the Value is ignored. The new Cue Step will be inserted on the current index. The Cue List Number is set using the **LayerUsesCueList** Command before using this command.

Controller Number	Value	Action
92	Not used	Insert Cue Step

CC95 CueListCreate

In order to be able to manipulate Cue Lists, a Cue List has to be created first. This is done with the **CueListCreate** command. The Control Number is 95, the Value is not used. The Cue List Number is set using the **LayerUsesCueList** Command before using this command. If the created Cue List already exists, no action is undertaken by the LanBox.

Controller Number	Value	Action
95	Not used	Create Cue List

CC96 CueListRemove

The **CueListRemove** command deletes a Cue List in the LanBox. The Control Number is 96, the Value is not used. The Cue List Number is set using the **LayerUsesCueList** Command before using this command. If the removed Cue List doesn't exists, no action is undertaken by the LanBox.

Controller Number	Value	Action
96	Not used	Remove Cue List

CC98 CueListRemoveStep

With the **CueListRemoveStep** command you can remove a single step of the loaded Cue List. If the Cue Step contains Scene Data these will also be removed. The Control Number is 98, the Value determines the Cue Step. The Cue List Number is set using the **LayerUsesCueList** Command before using this command.

Controller Number	Value	Action
98	1 - 99	Remove Cue Step Cue Step

CC103 LayerReplaceStep

he **LayerReplaceStep** command replaces an the current Cue Step with an empty new Cue Step. The Control Number is 103, the Value is not used.

Controller Number	Value	Action
103	Not used	Replace Cue Step

Layer MIDI Settings

CC64 LayerSetSustain

The **LayerSetSustain** command influences the way a Layer behaves when receiving MIDI Notes (for more information, see the MIDI Note section of the reference document). Putting a Layer in Sustain means that all Note-Off and Note-On messages with a velocity 0 will be ignored until Sustain is turned Off. This way, you can set multiple Channels to a Value and reset them all at once. The Controller Number is 64 and a Value < 64 will switch the Sustain Mode Off, a Value => 64 will switch it On.

Controller Number	Value	Action
64	0-63	Sustain Off
	64-127	Sustain On

CC65 LayerIgnoreNoteOff

The **LayerIgnoreNoteOff** command influences the way a Layer behaves when receiving MIDI Notes (for more information, see the MIDI sections of the reference document). Putting a Layer to Ignore Note-Off mode means that all Note-Off messages will be ignored and only Note-On messages will be accepted. The Controller Number is 65 and a Value < 64 will switch the Ignore Note-Off Mode Off, a Value => 64 will switch it On.

Controller Number	Value	Action
65	0-63	Ignore Note-Off Off
	64-127	Ignore Note-Off On

CC94 LayerSetDeviceID

To control Layers with MIDI Show Control, they have to have a MSC Device ID. This is set with the **LayerSetDeviceID** command. The Control Number is 94, the Value determines the ID.

Controller Number	Value	Action
94	1 - 127	Layer Device ID

MIDI Show Control

It is possible to control the LanBox with MIDI Show Control (MSC). MSC is a subset of the MIDI SysEx standard. The purpose of MIDI Show Control is to allow MIDI systems to communicate with and to control dedicated intelligent control equipment in theatrical, live performance, multimedia, audio-visual and similar environments.

The LanBox implementation of MSC is consistent with the recommended minimum set 2 (No time code, full data

capability). In order to receive MSC commands each Layer should have its own MSC device ID. When using more than one LanBox in a MSC system, each Layer in each LanBox should have a unique device ID. This ID is stored in the non-volatile memory and can be changed with in LCedit+ or with the **LayerSetDeviceID** command.

MIDI Show Control messages always have the following format:

F0 7F ID 02 CF CO <data> F7

The device ID points at the Layer, the Command Format CF is always 7F (Lighting). The Data depends on the command CO and is given in ASCII Values. If more then one Data set is required, 00 is used as a delimiter.

MSC Commands

MSC 01 GO

The **GO** command starts a Cue List at a Cue Step in the targeted Layer. The MSC command number is 1 (0x01 hex), the Cue List number and Cue Step number are send together, separated by a decimal point , ASCII 42, 0x2E hex. If no Cue List and Step are given, the Layer will perform a Go on the current Cue List.

General form:

F0 7F ID 02 7F 01 <Cue List> E9 <Cue Step> F7

ID	Layer Device ID
<Cue List>	Cue List in ASCII Values
<Cue Step>	Cue Step in ASCII Values

Go Cue List <Cue List>, Step <Cue Step> in Layer with Device ID ID

Example:

F0 7F 04 02 7F 01 31 36 36 E9 38 37 F7

04	Layer with Device ID 4 (0x04 hex)
31 36 36	Cue List 166 (ASCII 49 54 54, 0x31 0x36 0x36 hex)
38 37	Cue Step 87 (ASCII 56 55, 0x38 0x37 hex)

Go Cue List 166, Step 87 in Layer with Device ID 4

MSC 02 STOP

The **STOP** command pauses the execution of a Cue List in the targeted Layer. The MSC command number is 2 (0x02 hex) and requires no further data.

General form:

F0 7F ID 02 7F 02 F7

ID	Layer Device ID

Pause Layer with Device ID ID

Example:

F0 7F 2F 02 7F 02 F7

2F	Layer with Device ID 47 (0x2F hex)

Pause Layer with Device ID 47

MSC 03 RESUME

The **RESUME** command resumes the execution of a paused Cue List in the targeted Layer. The MSC command number is 3 (0x03 hex) and requires no further data.

General form:

F0 7F ID 02 7F 03 F7

ID	Layer Device ID

Resume Layer with Device ID ID

Example:

F0 7F 71 02 7F 03 F7

71	Layer with Device ID 113 (0x71 hex)

Pause Layer with Device ID 71

MSC 04 TIMED_GO

The **TIMED_GO** command starts a Cue List at a Cue Step in the targeted Layer at a designated Time. The MSC command number is 4 (0x04 hex) and requires a Time and a Cue List number plus Cue Step number. The Cue List number and Cue Step number are send together, separated by a decimal point , ASCII 42, 0x2E hex. If the LanBox has not internal timer set, the time data will be ignored.

General form:

F0 7F ID 02 7F 04 HR MN SC FR FF <Cue List> E9 <Cue Step> F7

ID	Layer Device ID
HR	Hours, 0 – 23 (0 – 1D hex)
MN	Minutes, 0 – 59 (0 – 3B hex)
SC	Seconds (0 – 59 (0 – 3B hex
FR	Frames 0 – 23 (0 – 1D hex)
FF	Sub Frames 0 – 99 (1 - 63 hex)
<Cue List>	Cue List in ASCII Values
<Cue Step>	Cue Step in ASCII Values

Go Cue List <Cue List>, Step <Cue Step> in Layer with Device ID ID at time HR:MN:SC.FR.FF

Example:

F0 7F 1B 02 7F 04 13 1E 0A 00 00 38 37 33 E9 38 F7

1B	Layer with Device ID 27 (0x1B hex)
13	19 Hours (0x13 hex)
1E	30 Minutes (0x1E hex)
0A	10 Seconds (0x0A hex)
00	0 Frames (0x00 hex)
00	0 Sub Frames (0x00 hex)
38 37 33	Cue List 873 (ASCII 56 55 51, 0x38 0x37 0x33 hex)
38	Cue Step 8 (ASCII 56, 0x38 hex)

Go Cue List 873, Step 8 in Layer with Device ID 1B at time 19:30:10.00.00

MSC 05 LOAD

The **LOAD** command loads a Cue List into the designated Layer without executing it. It will wait for a **GO** command. The MSC command number is 5 (0x05 hex), the Cue List number and Cue Step number are send together, separated by a decimal point , ASCII 42, 0x2E hex.

General form:

F0 7F ID 02 7F 05 <Cue List> E9 <Cue Step> F7

ID	Layer Device ID
<Cue List>	Cue List in ASCII Values
<Cue Step>	Cue Step in ASCII Values

Load Cue List <Cue List>, Step <Cue Step> in Layer with Device ID ID

Example:

F0 7F 17 02 7F 05 37 39 34 E9 32 F7

17	Layer with Device ID 23 (0x17 hex)
37 39 34	Cue List 734 (ASCII 55 57 52, 0x37 0x39 0x34 hex)
32	Cue Step 2 (ASCII 50, 0x32 hex)

Load Cue List 734, Step 2 in Layer with Device ID 23

MSC 06 SET

The **SET** command sets the Value of a Control Change command. In general, all functions that can be called on with the MIDI CC messages can be called on this way. The MSC command number is 6 (0x06 hex). The data consists of the Control Change Number and the Value, both in 2 bytes, LSB first. An optional time can be set as well. See the section **MIDI Control Messages** for the Control Change Numbers.

General form:

F0 7F ID 02 7F 06 CCNR VALU [HR MN SC FR FF] F7

ID	Layer Device ID
CCNR	Control Change Number, LSB first
VALU	Value, LSB first
Optional:	
HR	Hours, 0 – 23 (0 – 1D hex)
MN	Minutes, 0 – 59 (0 – 3B hex)
SC	Seconds (0 – 59 (0 – 3B hex
FR	Frames 0 – 23 (0 – 1D hex)
FF	Sub Frames 0 – 99 (1 - 63 hex)

Set CC number CCNR setting to Value VALU in Layer with Device ID ID (at time HR:MN:SC:FR:FF)

Example:

F0 7F 61 02 7F 06 47 00 04 00 F7

61	Layer with Device ID 97 (0x61 hex)
47 00	Control Change Number 71 (0x47 hex), **SetLayerMixMode**
04 00	Value 4 (0x04 hex), Transparent

Set Layer Mix Mode to Transparent in Layer with Device ID 97

MSC 07 FIRE

The **FIRE** command triggers a Macro Cue in the targeted Layer. In the LanBox Macro Cues are the same as normal Cue Lists, but can only have a Cue List Number of 1 – 127. The MSC command number is 7 (0x07 hex). The data consists of the Macro Cue List, this time as a number instead of ASCII values.

General form:

F0 7F ID 02 7F 07 <Cue List> F7

ID	Layer Device ID
<Cue List>	Cue List number (as number)

Load Cue List <Cue List>, Step <Cue Step> in Layer with Device ID ID

Example:

F0 7F 12 02 7F 07 77 F7

12	Layer with Device ID 18 (0x12 hex)
77	Cue List 119 (0x77 hex)

Load Cue List 734, Step 2 in Layer with Device ID 23

MSC 08 ALL_OFF

The **ALL_OFF** command sets all Attributes (Output, EditRun, Fading, Solo) of the targeted Layer to Off. The LanBox remembers the original settings and will restore these with the **RESTORE** command. The MSC command number is 8 (0x08 hex) and has no further data.

General form:

F0 7F ID 02 7F 08 F7

ID	Layer Device ID

Set all Attributes of Layer with Device ID ID to Off

Example:

F0 7F 4D 02 7F 08 F7

4D	Layer with Device ID 77 (0x4D hex)

Set all Attributes of Layer with Device ID 77 to Off

MSC 09 RESTORE

The **RESTORE** command restores all Attributes (Output, EditRun, Fading, Solo) of the targeted Layer to their original Values. The MSC command number is 9 (0x09 hex) and has no further data.

General form:

F0 7F ID 02 7F 09 F7

ID	Layer Device ID

Restore all Attributes of Layer with Device ID ID

Example:

F0 7F 2D 02 7F 09 F7

2D	Layer with Device ID 45 (0x2D hex)

Restore all Attributes of Layer with Device ID 45

MSC 0A RESET

The **RESET** command terminates all running Cue Lists in the targeted Layer and resets the Layer to its original (saved) state. The MSC command number is 10 (0x0A hex) and had no further data.

General form:

F0 7F ID 02 7F 0A F7

ID	Layer Device ID

Restore all Attributes of Layer with Device ID ID

Example:

F0 7F 2D 02 7F 0A F7

2D	Layer with Device ID 45 (0x2D hex)

Restore all Attributes of Layer with Device ID 45

MSC 0B GO_OFF

The GO_OFF command works exactly the same way as the GO command. It is meant for devices that can't replace a running Cue List. The LanBox can do this, so its function has no use for the LanBox.

MSC 10 GO_JAM

The **GO_JAM** command does exactly the same as the **GO** command, but also forces the Clock Time to the Go Time of that Cue List. The MSC command number is 16 (0x10 hex). The format is the same as the **GO** command.

MSC 11 STANDBY_+

The **STANDBY_+** command places the next Cue Step into Standby position. The Current Cue Step will finish running (if it was), but when the duration time is over, the Cue Step in the Standby position will be executed. The MSC command number is 17 (0x11 hex) and has as optional data the Cue Step to set to Standby.

General form:

F0 7F ID 02 7F 11 [<cue step>] F7

ID	Layer Device ID
optional:	
<cue step>	Cue Step in ASCII Values

Set next Cue Step (or Cue Step <cue step>) to Standby in Layer with Device ID ID

Example:

F0 7F 3E 02 7F 11 31 31 F7

3E	Layer with Device ID 62 (0x3E hex)
31 31	Cue Step 11 (ASCII 49 49, 0x31 0x31 hex)

Set Cue Step11 to Standby in Layer with Device ID 62

MSC 12 STANDBY_-

The **STANDBY_-** command places the previous Cue Step into Standby position. The Current Cue Step will finish running (if it was), but when the duration time is over, the Cue Step in the Standby position will be executed. The MSC command number is 18 (0x12 hex) and has as optional data the Cue Step to set to Standby.

General form:

F0 7F ID 02 7F 12 [<cue step>] F7

ID	Layer Device ID
optional:	
<cue step>	Cue Step in ASCII Values

Set previous Cue Step (or Cue Step <cue step>) to Standby in Layer with Device ID ID

Example:

F0 7F 13 02 7F 12 F7

13	Layer with Device ID 19 (0x13 hex)

Set previous Cue Step11 to Standby in Layer with Device ID 19

MSC 13 SEQUENCE_+

The **SEQUENCE_+** command places the next Cue List into Standby position. The Current Cue Step will finish running (if it was), but when the duration time is over, the Cue List in the Standby position will be executed. The MSC command number is 19 (0x13 hex) and has as optional data the Cue List + Step to set to Standby. The Cue List number and Cue Step number are send together, separated by a decimal point , ASCII 42, 0x2E hex.

General form:

F0 7F ID 02 7F 13 [<cue list> E9 <cue step>] F7

ID	Layer Device ID
optional:	
<Cue List>	Cue List in ASCII Values
<Cue Step>	Cue Step in ASCII Values

Set next Cue List (or Cue List <cue list>) to Standby in Layer with Device ID ID

Example:

F0 7F 07 02 7F 13 F7

07	Layer with Device ID 7 (0x07 hex)

Set the next Cue List to Standby in Layer with Device ID 7

MSC 14 SEQUENCE_-

The **SEQUENCE_-** command places the previous Cue List into Standby position. The Current Cue Step will finish running (if it was), but when the duration time is over, the Cue List in the Standby position will be executed. The MSC command number is 20 (0x14 hex) and has as optional data the Cue List + Step to set to Standby. The Cue List number and Cue Step number are send together, separated by a decimal point , ASCII 42, 0x2E hex.

General form:

F0 7F ID 02 7F 14 [<cue list> E9 <cue step>] F7

ID	Layer Device ID
optional:	
<Cue List>	Cue List in ASCII Values
<Cue Step>	Cue Step in ASCII Values

Set previous Cue List (or Cue List <cue list>) to Standby in Layer with Device ID ID

Example:

F0 7F 44 02 7F 14 F7

44	Layer with Device ID 68 (0x44 hex)

Set the previous Cue List to Standby in Layer with Device ID 68

UDP Data

The LanBox is capable of both broadcasting its buffer Values using UDP and receiving Values to write them directly into a buffer. To do this, network and UDP have to be correctly set up using LCedit+ or the proper serial or TCP/IP commands.

UDP broadcasting

The LanBox can broadcast all its buffer Values All Values are in network order (MSB first). The default port is 4777 but can be changed. Each packet consists of a header and one or more messages. If a message has an odd length, a padding byte is added to make sure that the next message will have a 16bit alignment. Each UDP package has the following format:

Header:

C0 B7 SEQU

where: SEQU is the 16bit sequence number

One or more messages:

MT BU LENG INDX <data>

where:		
	MT	Message Type: C9 for Buffer Broadcast, D1 for Layer Information Broadcast
	BU	Buffer ID
	LENG	16bit message length in 8bit bytes (complete message, including MT, BU LENG and INDX, excluding padding)
	INDX	16bit start index of the broadcast channels (only for Message type C9)
	<data>	consists of **n** x 8bit Values

The following Buffer IDs are used:

252 (0xFC hex)	DMX Input Buffer
253 (0xFD hex)	Analog Input Buffer
254 (0xFE hex)	Mixer Buffer
255 (0xFF hex)	DMX Output Buffer

Layer Information is Broadcast as:

n x (**n**= number of active Layers) in mixing order:

NR:	Internal Layer Number (8bit)
ID:	Layer ID (8bit, 1 - 63 for A - BK)
AT:	Layer Attributes (8bit Flags, Table 4 on page 16)
CLIS:	Active Cue List (16bit, 1-999)
FT:	Current Fade Time (8bit encoded, see Appendix A)
HT:	Current Hold Time (8bit encoded, see Appendix A)
FREM:	Remaining Fade Time (16bit in frames)
HREM:	Remaining Hold Time ((16bit in frames)

Example:

The LanBox is set to broadcast DMX Input, Channel 1 through 512, and the Mixer Channels 256 through 512. The UDP packet would look like this:

C0 B7 55 87 C9 FC 02 06 00 01 <512 x Data> C9 FE 01 07 01 00 <257x Data> 00

where:		
	C0 B7	UDP packet Header
	55 87	Sequence Number
	C9	Message Header
	FC	Buffer ID 252 (0xFC hex), DMX Input Buffer
	02 06	Package length 518 (0x0206 hex), 512 channels+ 6x 8bit message header
	00 01	Start Channel 1 (0x0001 hex)
	<512x Data>	
	C9	Message Header
	FE	Buffer ID 254 (0xFE hex),Mixer Buffer
	01 07	Message length 263 (0x0107 hex), 257 channels + 6x 8bit message header
	01 00	Start Channel 256 (0x0100hex)
	<257x Data>	
	00	Padding Byte

UDP Receiving

The LanBox can be configured to copy incoming UDP channels from another LanBox or 3rd part software into one of its Layers or Mixer Buffers. To do this, network and UDP have to be correctly set up using LCedit+ or the proper serial or TCP/IP commands.

Note: In order to copy DMX Values to a Layer, these Channels will have to be activated in this Channel.

Each UDP packet should have the following format:

Header:

C0 B7 SEQU

where: SEQU is the 16bit sequence number

One or more messages:

MT BU LENG INDX <data>

where:		
MT	Message Type.	CA for Buffer Write: ignores LanBox Settings and copies Values directly to Buffer and Offset stated in the Message. C9 for Buffer Broadcast: the LanBox copies Mixer buffer Data to Layer specified in UDP receive Settings D1 for Layer Information Broadcast.
BU	Buffer ID (8bit, 1 - 63 for Layer A - BK, 254 for Mixer)	
LENG	16bit message length in 8bit bytes (complete message, including MT, BU LENG and INDX, excluding padding)	
INDX	16bit start index of the broadcast channels (only Message Type CA and C9)	
<data>	consists of **n** x 8bit Values	

Example:

To send DMX Data to Channel 1 through 20 of Layer with ID A and Channel 21 through 40 of Layer with ID B.

C0 B7 55 87 CA 01 00 1A 00 01 <20x Data> CA 02 00 1A 00 15 <20x Data>

where:	
C0 B7	UDP packet Header
55 87	Sequence Number
C9	Message Header
01	Layer A (ID 1, 0x01 hex)
02 06	Package length 26 (0x001A hex), 20 channels+ 6x 8bit message header
00 01	Start Channel 1 (0x0001 hex)
<20x Data>	
C9	Message Header
02	Layer B (ID 2 0x02 hex)
01 07	Message length 26 (0x001A hex), 20 channels + 6x 8bit message header
00 15	Start Channel 21 (0x0015 hex)
<20x Data>	

Art-Net

If the UDP port for Send or Receive is set to **6454**, the LanBox automatically switches to Art-Net packet format. Please refer to the the Art-Net documentation for more information on format.

Appendix A: Cue Step Time Encoding Table

In order to be able to use one 8bit byte for time, the LanBox uses a time encoding table. This means that instead of a time Value in (milli)seconds, one of the numbers from the table below should be used.

NR.	HEX	Time (mm:ss.hh)	NR.	HEX	Time (m:ss.hh)	NR.	HEX	Time (m:ss.hh)
1	01	00:00.05	32	20	00:03.30	63	3F	01:00.00
2	02	00:00.10	33	21	00:03.60	64	40	01:06.00
3	03	00:00.15	34	22	00:03.90	65	41	01:12.00
4	04	00:00.20	35	23	00:04.30	66	42	01:18.00
5	05	00:00.25	36	24	00:04.70	67	43	01:30.00
6	06	00:00.30	37	25	00:05.10	68	44	01:36.00
7	07	00:00.35	38	26	00:05.60	69	45	01:48.00
8	08	00:00.40	39	27	00:06.20	70	46	02:00.00
9	09	00:00.45	40	28	00:06.80	71	47	02:12.00
10	0A	00:00.50	41	29	00:07.50	72	48	02:24.00
11	0B	00:00.55	42	2A	00:08.20	73	49	02:42.00
12	0C	00:00.60	43	2B	00:09.10	74	4A	03:00.00
13	0D	00:00.65	44	2C	00:10.00	75	4B	03:18.00
14	0E	00:00.70	45	2D	00:11.00	76	4C	03:42.00
15	0F	00:00.75	46	2E	00:12.00	77	4D	03:54.00
16	10	00:00.80	47	2F	00:13.00	78	4E	04:18.00
17	11	00:00.85	48	30	00:15.00	79	4F	04:48.00
18	12	00:00.90	49	31	00:16.00	80	50	05:06.00
19	13	00:00.95	50	32	00:18.00	81	51	05:42.00
20	14	00:01.00	51	33	00:20.00	82	52	06:18.00
21	15	00:01.10	52	34	00:22.00	83	53	06:48.00
22	16	00:01.20	53	35	00:24.00	84	54	07:30.00
23	17	00:01.30	54	36	00:27.00	85	55	08:12.00
24	18	00:01.50	55	37	00:30.00	86	56	09:06.00
25	19	00:01.60	56	38	00:33.00	87	57	10:00.00
26	1A	00:01.80	57	39	00:36.00	88	58	11:00.00
27	1B	00:02.00	58	3A	00:39.00	89	59	12:00.00
28	1C	00:02.20	59	3B	00:43.00	90	5A	13:00.00
29	1D	00:02.40	60	3C	00:47.00	91	5B	15:00.00
30	1E	00:02.70	61	3D	00:51.00	92	5C	For ever
31	1F	00:03.00	62	3E	00:56.00			

Table A1: Cue Step Time Encoding

Appendix B: Cue Step Types

Nr	HEX	Description	Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6
1	01	Show Scene	Fade type	Fade Time	Hold Time			
2	02	Show Scene of Cue List	Fade type	Fade Time	Hold Time	Cue List high	Cue List low	Cue Step
10	0A	Go Cue Step in Layer	Layer ID	Cue List high	Cue List low	Cue Step		
11	0B	Clear Layer	Layer ID					
12	0C	Pause Layer	Layer ID					
13	0D	Resume Layer	Layer ID					
14	0E	Start Layer	Layer ID					
15	0F	Stop Layer	Layer ID					
16	10	Configure Layer	Source Layer ID	Dest. Layer ID	New Layer ID	Cue List high	Cue List low	Cue Step
17	11	Reset Layer	Layer ID					
18	12	Go Trigger	Layer ID	Cue List high	Cue List low	Trigger ID	Channel high	Channel low
20	14	Go Cue Step	Cue Step					
21	15	Go Next in Layer	Layer ID					
22	16	Go Previous Step in Layer	Layer ID					
23	17	Loop to Cue Step	Cue Step	Nr of Loops				
24	18	Hold	Hold Time					
25	19	Hold Until		Day	Hours	minutes	seconds	frames
27	1B	Go if Analogue Input	Analogue Port	Value 1 high	Value 1 low	Value 2 high	Value 2 low	Cue Step
28	1C	Go if Channel	Layer ID	Channel High	Channel Low	Value 1	Value 2	Cue Step
30	1E	Set Layer attributes	Layer ID	Fade Enabled	Output enabled	Solo	Locked	
31	1F	Set Layer Mix Mode	Layer ID	Mix Mode	Transp. Start	Transp. End	Fade Time	
32	20	Set Layer Chaser	Layer ID	Chase Mode	Speed Start	Speed End	Fade Time	
40	28	Write Midi Stream	Number of Bytes	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
49	31	Write Stream to Serial Port 1	Number of Bytes	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
50	32	Write Stream to Serial Port 2	Number of Bytes	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
51	33	Write Stream to Serial Port 3	Number of Bytes	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
52	34	Write Stream to Serial Port 4	Number of Bytes	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
53	35	Write Stream to Serial Port 5	Number of Bytes	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
54	36	Write Stream to Serial Port 6	Number of Bytes	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
55	37	Write Stream to Serial Port 7	Number of Bytes	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
56	38	Write Stream to Serial Port 8	Number of Bytes	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
112	70	Comment						

Table B1: Cue Step Types

Wait Flag

Each Cue Step can be given a Wait Flag. This Flag will result in the Sequencer waiting for a Step Command (Go, Next, Previous) before proceeding to the next Cue Step. The Flag is the MSB of the Cue Step Type Byte, what means that 128 (0x80 hex) is added to the Cue Step Type number. So Cue Step Type 1 (0x01 hex) becomes 129 (0x81 hex), Type 20 (0x14 hex) becomes 148 (0x94 hex) etc.

Cue Step Type 1 Show Scene

The **Show Scene** Cue Step type shows the Cue Step Data assigned to the Cue Step for a given time (Hold Time) and has the possibility to fade.

Cue Type Nr:	1 (0x01 hex)	
Parameter:	1	Fade Type: see Table 7 on page 31
	2	Fade time, encoded, see Appendix A
	3	Hold time, encoded, see Appendix A
	4 – 6	Not used

Cue Step Type 2 Show Scene of Cue List

The **Show Scene** Cue Step type shows the Cue Step Data assigned to a Cue Step of another Cue List for a given time (Hold Time) and has the possibility to fade.

Cue Type Nr:	2 (0x02 hex)	
Parameter:	1	Fade Type: see Table 7 on page 31
	2	Fade time, encoded, see Appendix A
	3	Hold time, encoded, see Appendix A
	4 – 5	Cue List Number 16bit, MSB first
	6	Cue Step Number, 8bit

Cue Step Type 10 Go Cue Step in Layer

The Go Cue Step in Layer starts a Cue List at the given Cue Step in the given Layer.

Cue Type Nr:	10 (0x0A hex)	
Parameter:	1	Layer ID: 1 – 63 for A - BK
	2 - 3	Cue List Number 16bit, MSB first
	4	Cue Step Number, 8bit
	5 - 6	Not used

Cue Step Type 11 Clear Layer

The **Clear Layer** Cue Step type clears a LanBox Layer of all its Channel Data

Cue Type Nr:	11 (0x0B hex)	
Parameter:	1	Layer ID: 1 – 63 for A - BK
	2 - 6	Not used

Cue Step Type 12 Pause Layer

The **Pause Layer** Cue Step type pauses the execution of a Cue List in a LanBox Layer.

Cue Type Nr:	12 (0x0C hex)	
Parameter:	1	Layer ID: 1 – 63 for A - BK
	2 - 6	Not used

Cue Step Type 13 Resume Layer

The **Resume Layer** Cue Step type resumes the execution of a Paused Cue List in a LanBox Layer.

Cue Type Nr:	13 (0x0D hex)	
Parameter:	1	Layer ID: 1 – 63 for A - BK
	2 - 6	Not used

Cue Step Type 14 Start Layer

The **Start Layer** Cue Step type starts the execution of a loaded Cue List in a LanBox Layer.

Cue Type Nr:	14 (0x0E hex)	
Parameter:	1	Layer ID: 1 – 63 for A - BK
	2 - 6	Not used

Cue Step Type 15 Stop Layer

The **Stop Layer** Cue Step type stops the execution of the running Cue List in a LanBox Layer.

Cue Type Nr:	15 (0x0F hex)	
Parameter:	1	Layer ID: 1 – 63 for A - BK
	2 - 6	Not used

Cue Step Type 16 Configure Layer

The **Configure Layer** Cue Step type Layers can be placed in different order or a new Layer can be created.

Cue Type Nr:	16 (0x10 hex)	
Parameter:	1	Source Layer ID: 1 – 63 for A – BK 64 (0x40 hex) for New Layer
	2	Destination Layer ID: 1 – 63 for A – BK Layer will be placed directly under this Layer 0 (0x00 hex) to place Layer on top of Layer Stack
	3	New Layer ID: 1 – 63 for A – BK 0 (0x00 hex) for No ID
	4 – 5	Start Cue List Number 16bit, MSB first
	6	Start Cue Step Number, 8bit

Cue Step Type 17 Reset Layer

The **Reset Layer** Cue Step Type Clear a Layer and sets all its flags to default.

Cue Type Nr:	17 (0x11 hex)	
Parameter:	1	Layer ID: 1 – 63 for A - BK
	2 - 6	Not used

Cue Step Type 18 Trigger

The **Trigger** Cue Step Type takes a Channel Value, and maps its Value to start 1 of 64 Cue Lists.

Cue Type Nr:	18 (0x12 hex)	
Parameter:	1	Layer ID: 1 – 63 for A – BK to start Triggered Cue List
	2 - 3	Triggered Cue Lists start Cue List Number 16bit, MSB first
	4	Trigger ID Number 8bit
	5 – 6	Mixer Channel to Map to Cue Lists, 16bit, MSB first 3061 – 3068 for Analog Inputs

Cue Step Type 20 Go Cue Step

The **Go Cue Step** Cue Step Type starts a Cue Step in the current Cue List.

Cue Type Nr:	20 (0x14 hex)	
Parameter:	1	Cue Step Number, 8bit
	2 - 6	Not Used

Cue Step Type 21 Go Next in Layer

The **Go Next in Layer** Cue Step Type gives a **LayerNextStep** command to specified Layer.

Cue Type Nr:	21 (0x15 hex)	
Parameter:	1	Layer ID: 1 – 63 for A - BK
	2 - 6	Not Used

Cue Step Type 22 Go Previous in Layer

The **Go Previous in Layer** Cue Step Type gives a **LayerPreviousStep** command to specified Layer.

Cue Type Nr:	22 (0x16 hex)	
Parameter:	1	Layer ID: 1 – 63 for A - BK
	2 - 6	Not Used

Cue Step Type 23 Loop Cue Step

The **Loop Cue Step** Cue Step Type starts the defined Cue Step **n** times before continuing to the next Cue Step.

Cue Type Nr:	23 (0x17 hex)	
Parameter:	1	Start Cue Step Number, 8bit
	2	Number of Loops 8bit
	3 – 6	Not used

Cue Step Type 24 Hold

The **Hold** Cue Step Type sets the Cue List to wait for a specific time before continuing execution.

Cue Type Nr:	24 (0x18hex)	
Parameter:	1	Hold Time, encoded, see Appendix A
	2 – 6	Not used

Cue Step Type 25 Hold Until

The **Hold Until** Cue Step Type waits for a specific time (internal LanBox clock or MTC) before executing the next Cue Step. The Time Source is encoded in the Day Value, the Time Comparison is encoded in the Hour Value,

Cue Type Nr:	25 (0x19 hex)			
Parameter:	1	Not used		
	2	Day, 0 – 6 for Monday – Sunday, 128 for no (every) Day		
		Time Source	MTC	+ 0 (0x00 hex)
			Internal Clock	+ 32 (0x20 hex)
			Locked MTC	+ 64 (0x40 hex)
			Locked Clock	+ 96 (0x60 hex)
		note: Locked MTC responds only to Locked MTC Values, Locked Clock only to NTP verified Internal Clock Values.		
	3	Hour, 0 – 23, 128 for no (every) hour		
		Comparison:	Time is after	+ 0 (0x00 hex)
			Time is before	+ 32 (0x20 hex)
			Transition to after	+ 64 (0x40 hex)
			Transition to before	+ 96 (0x60 hex)
	4	Minutes, 0 – 59		
	5	Seconds, 0 – 59		
	6	frames, 0 – 30		

Cue Step Type 27 Go If Analog Input

The **Go If Analog Input** Cue Step Type checks the Value of analog Input and performs the specified comparison. If Value 1 < Value 2, a Value between Value 1 and 2 will trigger a Go, if Value 1 > Value 2, a Value outside the specified range will trigger a Go.

Cue Type Nr:	27 (0x1B hex)	
Parameter:	1	Analog Input Number
	2 – 3	Value 1 (16bit)
	4 – 5	Value 2 (16bit)
	6	Cue Step Number (8bit)

Cue Step Type 28 Go If Channel

The **Go If Channel** Cue Step Type checks the Value of Channel in a specified Layer and performs a comparison. If Value 1 < Value 2, a Value between Value 1 and 2 will trigger a Go, if Value 1 > Value 2, a Value outside the specified range will trigger a Go.

Cue Type Nr:	28 (0x1C hex)	
Parameter:	1	Layer ID: 1 – 63 for A – BK, 254 (0xFE hex) for the Mixer Buffer
	2 – 3	Channel Number (16bit)
	4	Value 1 (8bit)
	5	Value 2 (8bit)
	6	Cue Step Number (8bit)

Cue Step Type 30 Set Layer Attributes

The **Set Layer Attributes** Cue Step Type sets the Attributes of a specified Layer. For all Attributes, a Value of 0 sets the attribute to Off, a Value > 0 to On

Cue Type Nr:	30 (0x1E hex)	
Parameter:	1	Layer ID: 1 – 63 for A – BK
	2	Fade enabled, 0 for Off, >0 for On
	3	Output enabled, 0 for Off, >0 for On
	4	Solo enabled, 0 for Off, >0 for On
	5	Locked, 0 for Off, >0 for On
	6	Not used

Cue Step Type 31 Set Layer Mix Mode

The **Set Layer Mix Mode** Cue Step Type sets the Mix Mode and Transparency Depth of a specified Layer. The Transparency Depth can be set to fade from Value 1 to Value 2 in a given time.

Cue Type Nr:	31 (0x1F hex)		
Parameter:	1	Layer ID: 1 – 63 for A – BK	
	2	Mix Mode:	
		0: Off	Layer does not output anything.
		1: Copy	Knock Out Mode. Overwrites all Values of matching channels in underlying Layers.
		2: HTP	Highest takes Precedence. The Highest Value of a Channel is output to the Mixer.
		3: LTP	Lowest takes Precedence. The Lowest Value of a Channel is output to the Mixer.
		4: Transparent	A percentage of a Value is added to the Value of underlying Layers.
		5: Add	Makes Values relative to underlying Layers. (Value– 128) is added to the Value of underlying Layers.
	3	Transparency Depth Value 1	
	4	Transparency Depth Value 2	
	5	Fade time, encoded, see Appendix A	
	6	Not used	

Cue Step Type 32 Set Layer Chaser

The **Set Layer Chaser** Cue Step Type sets the Chase Mode and Chase Speed of a specified Layer. The Chase Speed can be set to fade from Value 1 to Value 2 in a given time.

Cue Type Nr:	32(0x20 hex)		
Parameter:	1	Layer ID: 1 – 63 for A – BK	
	2	Mix Mode:	
		0: Off	No chase (use Cue List)
		1: Chase Up	Steps through cue steps in increasing order
		2: Loop Up	Loops through cue steps in increasing order
		3: Chase Down	Steps through cue steps in decreasing order
		4: Loop Down	Loops through cue steps in decreasing order
		5: Random	Steps through cue steps in random order
		6: Loop Random	Loops through cue steps in random order
		7: Bounce	Steps through cue steps in alternating order
		8: Loop Bounce	Loops through cue steps in alternating order
	3	Chase Speed Value 1	
	4	Chase Speed Value 2	
	5	Fade time, encoded, see Appendix A	
	6	Not used	

Cue Step Type 40 Write MIDI Stream

The **Write MIDI Stream** Cue Step Type outputs up to 5 8bit Values to the MIDI out Port of the LanBox.

Cue Type Nr:	40 (0x28 hex)	
Parameter:	1	Number of MIDI Bytes (0 - 5)
	2	Byte 1
	3	Byte 2
	4	Byte 3
	5	Byte 4
	6	Byte 5

Cue Step Type 49 – 56 Write Serial Stream

The **Write Serial Stream** Cue Step Type outputs up to 5 8bit Values to one of the Serial out Ports of the LanBox. The Step Type number defines the port, 49 for Port 1, 56 for Port 8.

Cue Type Nr:	49 – 56 (0x31 - 0x38 hex)	
Parameter:	1	Number of Bytes (0 - 5)
	2	Byte 1
	3	Byte 2
	4	Byte 3
	5	Byte 4
	6	Byte 5

Cue Step Type 70 Comment

The **Comment** Cue Step Type does nothing but writing a comment in a Cue List. The 6 8bit Parameters contain 8 encoded base64 (6bit) characters (see Table B2).

Cue Type Nr:	49 – 56 (0x31 - 0x38 hex)		
Parameter:	1	x1	1st character code
		x2	
		x3	
		x4	
		x5	
		x6___	
		x7	2nd character code
		x8	
	2	x1	
		x2	
		x3	
		x4___	
		x5	3rd character code
		x6	
		x7	
		x8	
	3 – 6	etc.	

0	space	8	H	16	P	24	X	32	f	40	n	48	v	56	3
1	A	9	I	17	Q	25	Y	33	g	41	o	49	w	57	4
2	B	10	J	18	R	26	Z	34	h	42	p	50	x	58	5
3	C	11	K	19	S	27	a	35	i	43	q	51	y	59	6
4	D	12	L	20	T	28	b	36	j	44	r	52	z	60	7
5	E	13	M	21	U	29	c	37	k	45	s	53	0	61	8
6	F	14	N	22	V	30	d	38	l	46	t	54	1	62	9
7	G	15	O	23	W	31	e	39	m	47	u	55	2	63	-

Table B2, LanBox base64 encoding

Appendix C: SysexWriteToMixer command

The **SysexWriteToMixer** command is a special case. It works like a serial command sent in a SysEx format, but with some subtle differences. The command enables you to write binary data (7bit, 0-127) directly into the Mixer buffer of a LanBox. **SysexWriteToMixer** is like standard LanBox SysEx calls, but as it's a 7 bit binary call instead of ASCII, the command starts with an $ (ASCII 36, 0x24 hex) instead of *, which is followed by 2 0's. The command has 2 + **n** parameters: the Start Channel (14bit, 1 – 3072), the number of Channels (14bit, 1 – 3072) and **n** Values (7bit, 0-127).

General form:

F0 00 20 40 ID 50 MI 24 00 00 OFFS NUMR V1 V2 V3 .. .. Vn F7

ID	LanBox Device ID
MI	Master Device ID
OFFS	Start Channel (14bit, 1 – 3072)
NUMR	Number of Channels (14bit, 1 – 3072)
V1	Value 1 (7bit, 0 – 127)
V2	Value 2 (7bit, 0 – 127)
V3	Value 3 (7bit, 0 – 127)
..	
..	
Vn	Value n (7bit, 0 - 127)

Set all Attributes of Layer with Device ID ID to Off

As the speed is mainly determent by the MIDI speed, its a good idea to keep the data length as short as possible.