

Digitech GNX1



MIDI System Exclusive

by

Gary Barnes

Version 1.2

Copyright © 2025 Gary Barnes. All rights reserved.

Table of Contents

Introduction.....	4
Message Format.....	5
Packed Data.....	6
Number Format.....	7
Banks and Patches.....	7
Command Codes.....	8
Code 01: Device Enquiry.....	9
Code 02: Power On Announcement.....	10
Code 05: Status Request.....	10
Code 06: Response to Code 05 Status Request.....	11
Code 07: Request Amp & Cab Names.....	12
Code 08: Response to Code 07 Request Amp and Cab Names.....	12
Code 09: Request User Bank Names.....	14
Code 0A: Response to Code 09 Request User Bank Names.....	14
Code 0B: Request User Amp and Cab Patches.....	16
Code 0C: Unknown.....	16
Code 0D: Unknown.....	16
Code 0E: Unknown.....	17
Code 0F: Response to Code 0B Request Amp and Cab Patches.....	17
Code 10: Data Block Request Terminator.....	17
Code 11: Unknown.....	18
Code 12: Request Patch Names.....	18
Code 13: Response to Code 12 Request Patch Names.....	19
Code 14: Unknown.....	19
Code 15: Unknown.....	20
Code 20: Request Patch Name.....	20
Code 21: Response to Code 20 Request Patch Name.....	21
Code 22: End of Data Dump.....	22
Code 23: Unknown.....	22
Code 24: Edit Buffer Patch Data.....	22
Code 25: No Response.....	24
Code 26: LFO and Expression Pedals.....	24
Code 27: Unknown.....	26
Code 28: Sync Data.....	26
Code 27: Unknown.....	27
Code 2A: Amp or Cabinet Data Block.....	27
Code 2B: Unknown.....	28
Code 2C: Parameter Change.....	28
Code 2C Sections.....	29
Section 01 (Pickup Simulator) Parameters.....	29
Section 02 (Wah) Parameters.....	29
Section 03 (Compressor) Parameters.....	30
Section 04 (Whammy/IPS) Parameters.....	30

Type 00: Whammy.....	30
Type 01: IPS.....	31
Type 02: Detune.....	31
Type 03: Pitch.....	31
Section 05 (Warp) Parameters.....	31
Section 06 (Green Amp) Parameters.....	32
Section 07 (Green Cabinet) Parameters.....	32
Section 08 (Red Amp) Parameters.....	33
Section 09 (Red Cabinet) Parameters.....	33
Section 0A (Noise Gate) Parameters.....	34
Type 00: Silencer.....	34
Type 01: Pluck.....	34
Section 0B (Chorus/Mod) Parameters.....	35
Type 00: Chorus.....	35
Type 01: Flanger.....	35
Type 02: Phaser.....	35
Type 03: Triggered Flanger.....	36
Type 04: Triggered Phaser.....	36
Type 05: Tremelo.....	36
Type 06: Panner.....	36
Type 07: Vibrato.....	37
Type 08: Rotary Speaker.....	37
Type 09: Auto Yah.....	37
Type 0A: YaYa.....	38
Type 0B: Synthtalk.....	38
Type 0C: Envelope.....	38
Type 0D: Detune.....	38
Type 0E: Pitch.....	39
Section 0C (Delay) Parameters.....	39
Section 0D (Reverb) Parameters.....	40
Section 0E Pedal Parameters.....	40
Section 8A (LFO 1) Parameters.....	40
Section 8B (LFO 2) Parameters.....	40
Code 2D: Patch Change.....	41
Code 2E: Patch Saved With Name.....	41
Code 76: Keep Alive.....	42
Code 7E: Acknowledge.....	42
Code 7F: Error.....	43
GNX1 / GENEdit Startup Dialog.....	44

Introduction

The Digitech GNX1 is a powerful vintage guitar effects processor which shipped with the GENEdit software package for editing from a Windows machine. I wanted to create a Linux of that software, and so I contacted Digitech for documentation on the MIDI System Exclusive messages which the software uses to communicate with the device. Unfortunately, Digitech had nothing they could give me, so I set about decoding the transmitted messages.

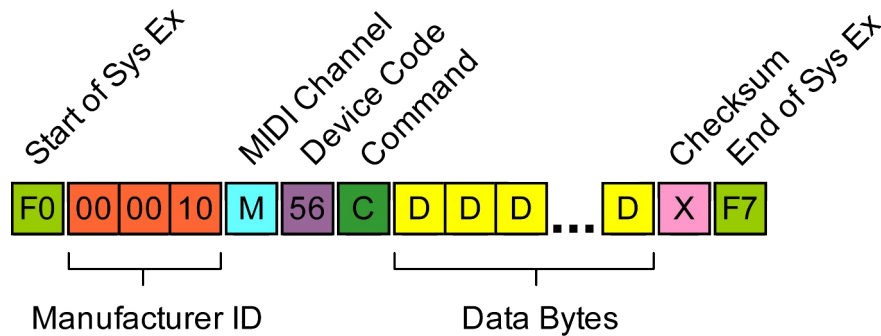
This document describes what I have discovered to date. It may not be complete but there is enough for me to build my editing software, built in the Python programming language using the Qt framework, which should mean that it is cross-platform.

My software is called GNXEdit and is available on GitHub as an open source project (although currently in development) at <https://github.com/gary-1959/GNXEdit>.

Message Format

There are many resources on the internet which detail the MIDI System Exclusive format, so I will skip that here and concentrate on the message contents applicable to the GNX1.

The general format is as follows (number is hexadecimal unless stated otherwise):



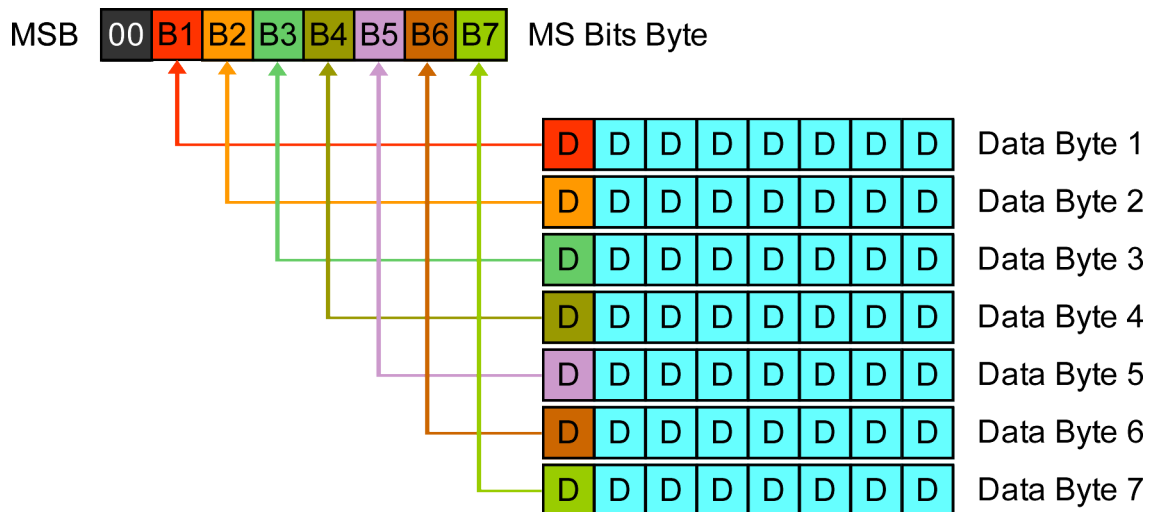
Start of Sys EX	Always 0xF0
Manufacturer ID	Three bytes 0x00, 0x00, 0x10 which is listed as the ID for DOD Electronics Corp with the Association of Musical Electronics Industry (AMIE)
MIDI Channel	0x00 to 0x0F corresponding to MIDI channels 1 to 15 (decimal), or the broadcast code 0x7E used when probing the MIDI connection for compatible devices on any channel.
Device Code	Manufacturer's unique identifier for the GNX1. Always 0x56.
Command	A code describing the contents of the message. This is detailed in this document.
Data Bytes	A variable number of data bytes, usually in a packed format which is detailed in this document
Checksum	Standard System Exclusive checksum, which is an XOR sum of all the preceding bytes excluding the 0xF0 start of message code
End of Sys Ex	Always 0xF0

Because MIDI is quite a slow format, the GNX1 encodes numbers according to their size to minimise message lengths (see below). Therefore, when encoding a complete message the number values are first encoded then the complete data package is packed.

Packed Data

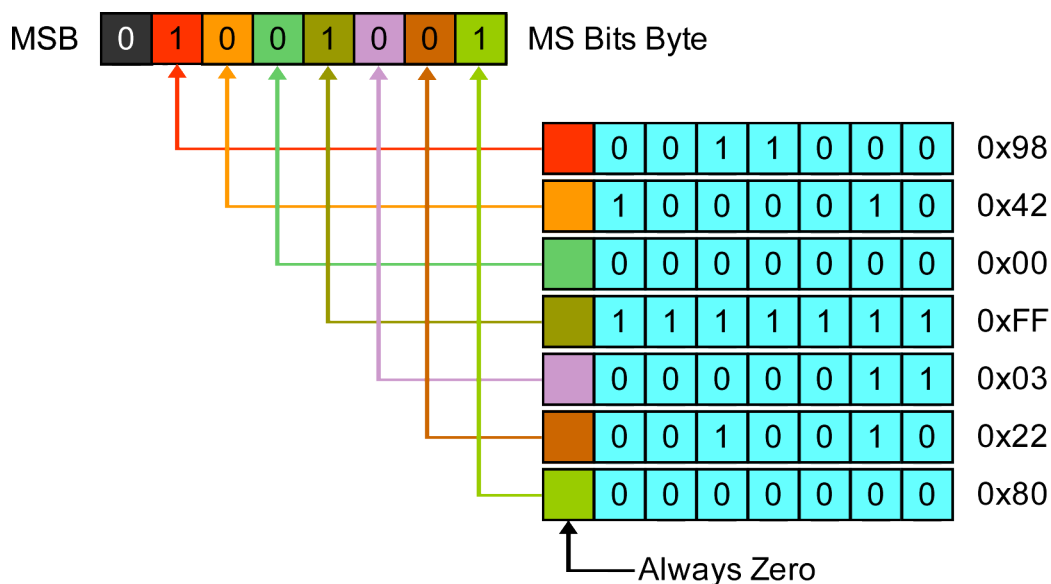
The MIDI protocol defines bytes with the most significant bit (MSB) set as commands. This means that byte values higher or equal to 0x80 (128 decimal) are a command of some description. This means that data bytes transmitted within a System Exclusive message are limited to the value 0x7F.

To overcome this the GNX1 uses a method of data packing consisting of a single byte containing the most significant bits of the following 7 bytes. Note that the last packet transmitted can be truncated to less than 7 bytes (no padding is used).



The first byte is constructed as follows:

Example:



In this example the byte sequence:

0x98 0x42 0x00 0xFF 0x03 0x22 0x80

becomes:

0x49 0x18 0x42 0x00 0x7F 0x03 0x22 0x00

Number Format

A further level of encoding is employed for numbers. The GNX1 transmits numerical values up to 0x7F as a single byte. For values greater than this an encoding system is used whereby the first byte has the most significant bit set and the lower bytes contain the number of bytes following which comprise the number.

<u>Number</u>	<u>First Byte</u>	<u>Second Byte</u>	<u>Third Byte</u>
0x00 to 0x7F	0x00 to 0x7F	N/A	N/A
0x80 to 0xFF	0x81	0x80 to 0xFF	N/A
0x0100 to 0x01FF	0x82	0x01	0x00 to 0xFF
0xFFFF	0x82	0xFF	0xFF

Banks and Patches

Many commands refer to a particular bank (Factory, User, Buffer) and patch.

The User and Factory banks can have up to 48 (decimal) patches. The Buffer bank can only refer to the single edit buffer which is where patches are loaded and edited before saving to a User or Factory patch.

Banks are numbered as follows:

0x00 Factory

0x01 User

0x02 Edit Buffer

Patches are numbered 0x00 – 0x2F (0 – 47 decimal) corresponding to the User and Factory bank patches numbered 1 – 48 (decimal) on the GNX1 display. The Edit Buffer always references patch 0x00

Command Codes

These are the command codes as understood to date:

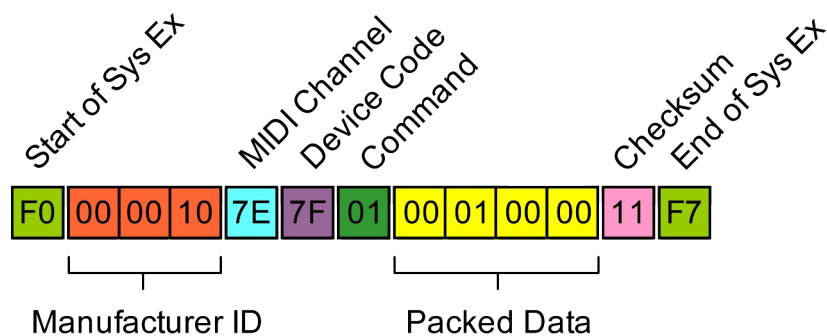
CODE (HEX)	DESCRIPTION
01	Device enquiry broadcast on all channels to any device
02	Transmitted when power applied to the device
03 - 04	Not documented
05	Status request
06	Response to Code 05 status request
07	Request Amp and Cab names
08	Response to Code 07 Amp and Cab names
09	Request User Bank Names
0A	Response to Code 09 User Bank Names
0B	Request User Amp and Cab Patches
0C	Unknown
0D	Unknown
0E	Unknown
0F	Response to Code 0B User Amp and Cab Patches
10	Response to Code 0B Terminator
11	Unknown
12	Request Patch Names
13	Response to Code 12 Patch Names
14	Unknown
15	Unknown
16 – 1F	Not documented
20	Request Patch Name
21	Response to Code 21 Patch Name
22	End of Data Dump
23	Unknown
24	Edit Buffer Patch Data
25	No Response
26	LFO and Expression Pedals
27	Unknown
28	Sync Data
29	Unknown
2A	Amp or Cabinet Data Block
2B	Unknown

CODE (HEX)	DESCRIPTION
2C	Parameter Change
2D	Patch Change
2E	Patch Saved With Name
70 - 75	Not documented
76	Keep Alive
7E	Acknowledge
7F	Error

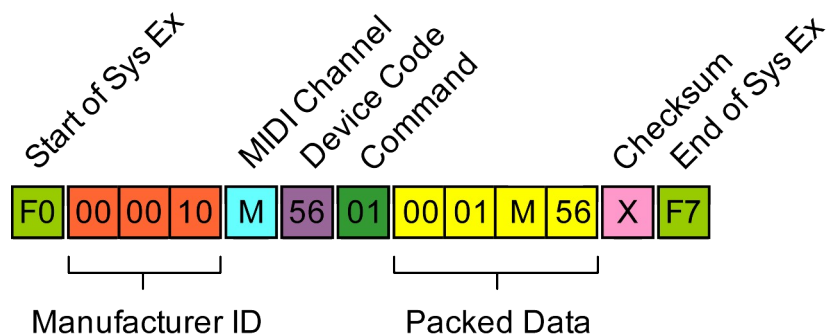
The codes marked as undocumented have not been observed or tested, based on the [Code 06](#) response.

Code 01: Device Enquiry

Broadcast on all channels to find a device on the MIDI connection.



GNX1 responds with device code and MIDI channel.

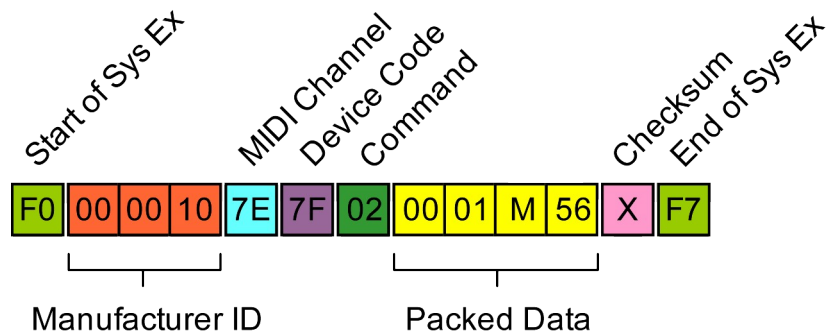


The device MIDI channel is repeated in the M data byte followed by the device code (0x56).

Note that none of the trailing 3 data bytes will be larger than 0x7F so the first data byte will always be zero. The next three data bytes are unchanged when unpacked.

Code 02: Power On Announcement

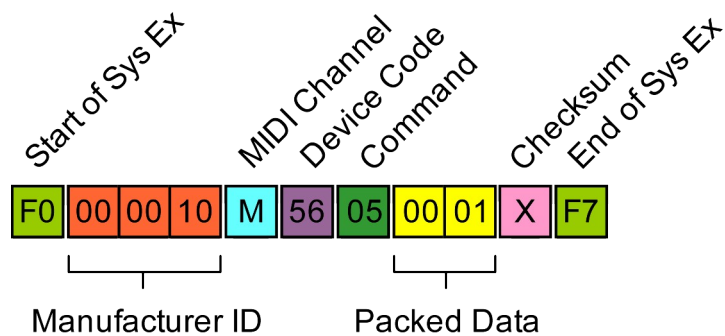
The following message is transmitted by the GNX1 when powered on, or in response to code 01 device enquiry:

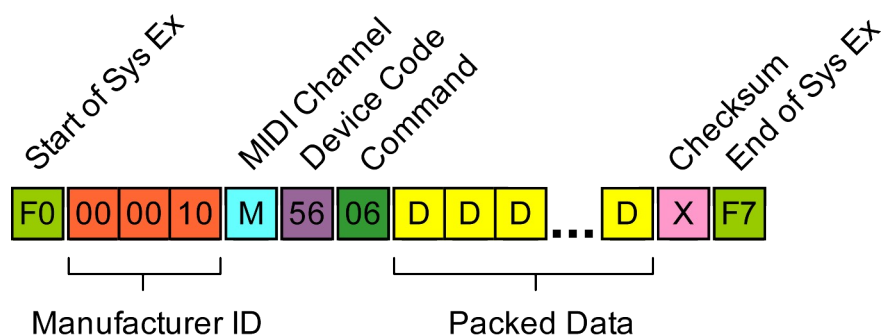


The device MIDI channel is contained in the M data byte followed by the device code (0x56).

Code 05: Status Request

Response is a code 06 message



Code 06: Response to Code 05 Status Request

Typical response (packed data):

01	00	6F	00	15	00	25	30	.	.	o	.	.	.	%	0
30	03	01	06	01	01	02	01	0
05	01	06	01	07	01	08	02
09	01	0A	02	0B	01	0C	01
0D	01	0E	01	0F	01	10	01
11	01	12	01	13	01	14	01
15	01	20	01	21	01	22	01	.	.	.	!	.	"	.	.
23	01	24	02	25	01	26	02	#	.	\$.	%	.	&	.
27	01	28	01	29	01	2A	01	'	.	(.)	.	*	.
2B	01	2C	02	2D	01	2E	01	+	.	,	.	-	.	.	.
70	01	71	01	72	01	73	01	p	.	q	.	r	.	s	.
74	01	75	01	76	01	7E	01	t	.	u	.	v	.	~	.
7F	01														

01: Current Bank (00: Factory, 01: User)

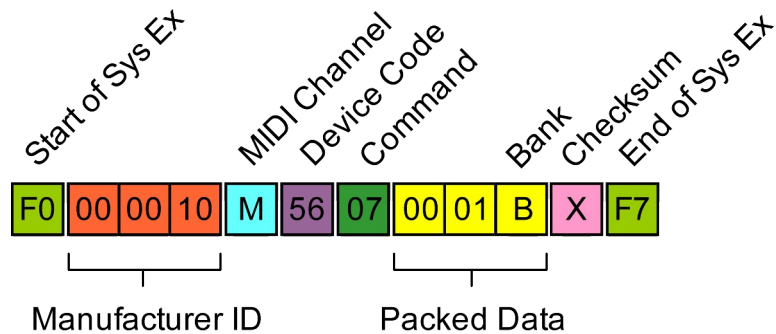
06: Current Patch Number (0x00 - 0x3F)

05: Permitted Command Code

Following bank and patch there is an index of what appear to be accepted command codes (05) followed by 0x01 or 0x02. Note absence of codes 0x03 - 0x04, 0x16 - 0x1F.

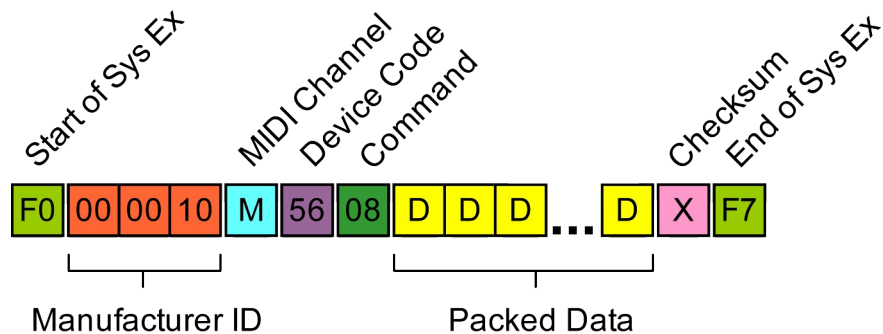
Code 07: Request Amp & Cab Names

Request amp and cab names



Bank is 0x00 for factory amp and cab names or 0x01 for user patches

Code 08: Response to Code 07 Request Amp and Cab Names



Response is an indexed list of amp and cab names. Each list is prefixed with a type code: 0x3C for amp names, 0x3D for cab names followed by a number indicating the start of the user list, then the number of names that follow. The GNX1 transmits 17 (decimal) factory amp names followed by 9 cab names. The user set depends on the number of amps saved in the user slots, and may be zero.

An example response follows:

Example Response to code 07 request for Amp and Cab Names (factory, unpacked):

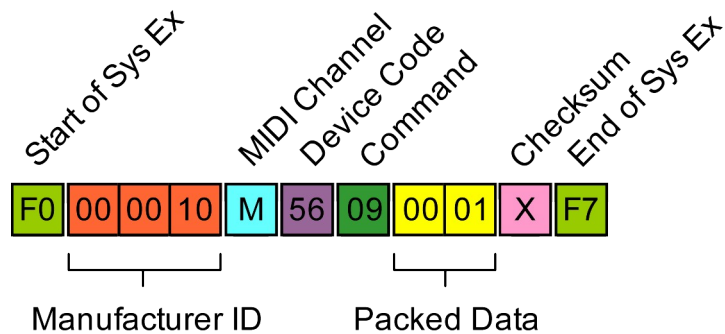
	02	00	02	3C	11	11	00	44	. . . < . . . D
	49	52	45	43	54	00	01	42	I R E C T . . B
	4C	4B	46	41	43	00	02	42	L K F A C . . B
Amp Names	4F	55	54	49	51	00	03	52	O U T I Q . . R
	45	43	54	49	46	00	04	48	E C T I F . . H
	4F	54	52	4F	44	00	05	54	O T R O D . . T
	57	45	45	44	20	00	06	42	W E E D . . B
	52	54	43	4D	42	00	07	43	R T C M B . . C
	4C	4E	54	55	42	00	08	42	L N T U B . . B
	52	54	53	54	4B	00	09	43	R T S T K . . C
	52	55	4E	43	48	00	0A	48	R U N C H . . H
	49	47	41	49	4E	00	0B	42	I G A I N . . B
	4C	55	45	53	20	00	0C	4D	L U E S . . M
	4F	44	47	41	4E	00	0D	46	O D G A N . . F
	55	5A	5A	20	20	00	0E	42	U Z Z . . B
	41	53	53	4D	4E	00	0F	48	A S S M N . . H
	49	57	41	54	47	00	10	41	I W A T G . . A
	43	4F	55	53	54	00	3D	09	C O U S T . = .
	09	00	44	49	52	45	43	54	. . D I R E C T
Cab Names	00	01	41	4D	32	58	31	32	. . A M 2 X 1 2
	00	02	42	52	34	58	31	32	. . B R 4 X 1 2
	00	03	56	20	34	58	31	32	. . V 4 X 1 2
	00	04	42	52	32	58	31	32	. . B R 2 X 1 2
	00	05	41	4D	31	58	31	32	. . A M 1 X 1 2
	00	06	42	4C	32	58	31	32	. . B L 2 X 1 2
	00	07	46	4E	34	58	31	32	. . F N 4 X 1 2
	00	08	47	52	34	58	31	32	. . G R 4 X 1 2
	00								.

KEY

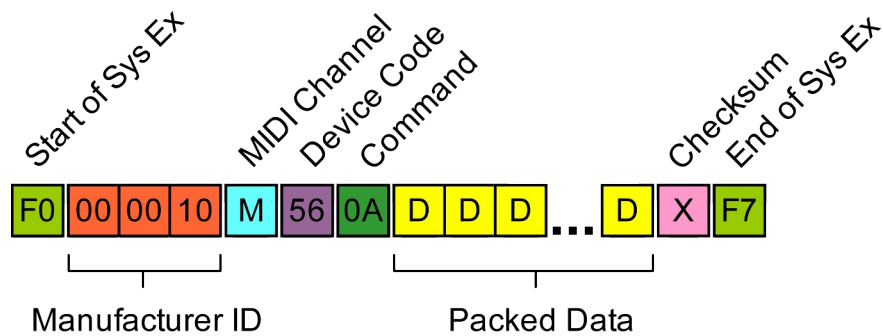
- : 00 = Factory, 01 = User
- : 3C = Amps, 3D = Cabs
- : Offset to User Names
- : Number of items
- : Index No.
- : ASCII Name
- : String terminator (null)

Code 09: Request User Bank Names

The GNX1 divides groups the user patches into 16 groups of 3 and an editable name is assigned to each of these banks. Command code 09 is used to request these names.



Code 0A: Response to Code 09 Request User Bank Names



Names are returned as a set of null terminated ASCII strings followed by some spurious data.

Example response follows:

Example response:

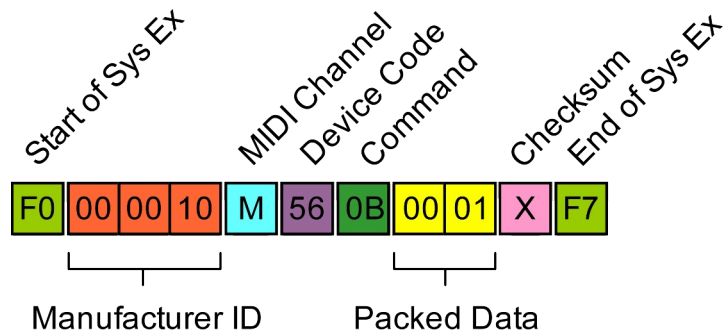
		02	03	00	04	01	10	00	F0		
User Bank Names		53	48	4F	43	41	53	00	47	S	H O C A S . G
		45	4E	45	54	58	00	53	54	E	N E T X . S T
		41	52	53	20	00	41	4D	50	A	R S . A M P
		53	20	20	00	42	4C	55	45	S	. B L U E
		53	20	00	43	4E	54	52	59	S	. C N T R Y
		20	00	52	4F	43	4B	20	20		. R O C K
		00	43	4C	45	41	4E	20	00		. C L E A N .
		4D	45	54	41	4C	20	00	53	M	E T A L . S
		54	55	44	49	4F	00	4A	41	T	U D I O . J A
		5A	5A	20	20	00	56	49	4E	Z	Z . V I N
		54	41	47	00	46	55	4E	4B	T	A G . F U N K
		20	20	00	48	59	42	52	49		. H Y B R I
		44	00	45	58	50	52	53	4E	D	. E X P R S N
		00	53	50	45	43	49	41	00	.	S P E C I A .
Character Set 00-1F		01	02	01	00	02	03	00	00	.	.
		80	00	80	00	01	02	03	04	.	.
		05	06	07	08	09	0A	0B	0C	.	.
		0D	0E	0F	10	11	12	13	14	.	.
		15	16	17	18	19	1A	1B	1C	.	.
		1D	1E	1F	20	21	22	23	24	.	! " # \$
		25	26	27	28	29	2A	2B	2C	%	& ' () * + ,
		2D	2E	2F	30	31	32	33	34	-	. / 0 1 2 3 4
		35	36	37	38	39	3A	3B	3C	5	6 7 8 9 : ; <
		3D	3E	3F	40	41	42	43	44	=	> ? @ A B C D
		45	46	47	48	49	4A	4B	4C	E	F G H I J K L
		4D	4E	4F	50	51	52	53	54	M	N O P Q R S T
		55	56	57	58	59	5A	5B	5C	U	V W X Y Z [\
		5D	5E	5F	00	01	02	03	04]	^ _
		05	06	07	08	09	0A	0B	0C	.	.
		0D	0E	0F	10	11	12	13	14	.	.
		15	16	17	18	19	1A	1B	1C	.	.
		1D	1E	1F						.	.

Changes to VSWITCH level cause this message to be transmitted, but the value does not appear in the data blocks

Code 0B: Request User Amp and Cab Patches

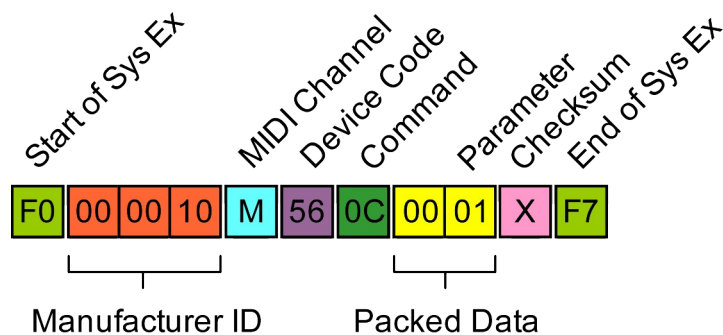
The GNX1 manual says that there are 9 memory slots for storing Hypermodel amps. Mine seems to allow saving beyond the 9. Either there is extra memory available or other RAM is being overwritten.

The 0B command requests data about these locations:



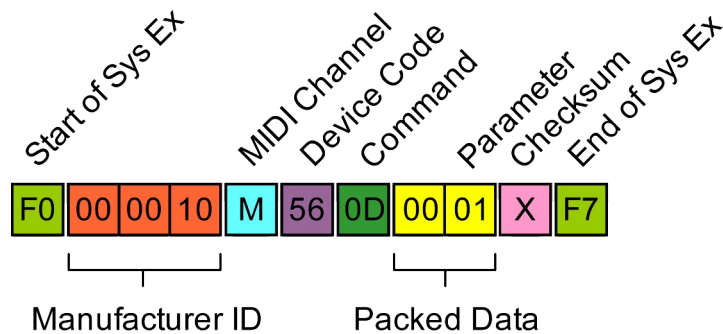
Code 0C: Unknown

Message below with parameter 0x01 returns 0x7E. Other parameters in range 0x00 to 0x0F return code 0x7F.



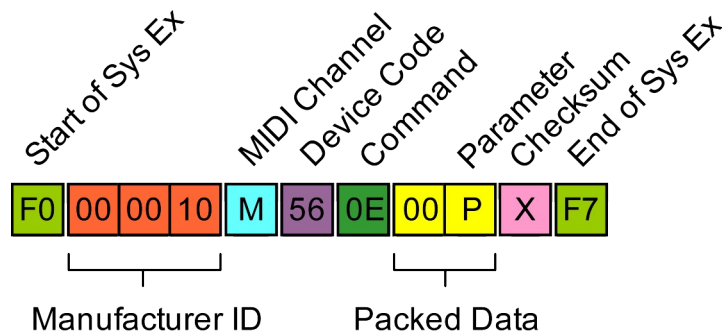
Code 0D: Unknown

Message below with parameter 0x01 returns 0x7E. Other parameters in range 0x00 to 0x0F return code 0x7F.



Code 0E: Unknown

Message below with parameter 0x01 returns a 0xF0 message. Other parameters in range 0x00 to 0x0F return code 0x7F.

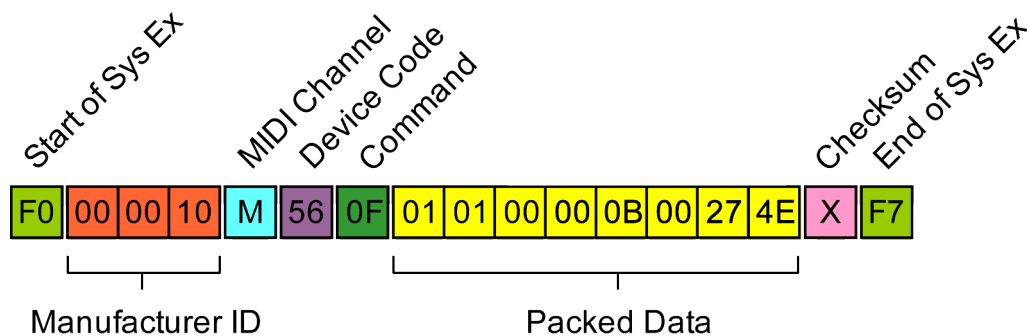


Returns unpacked 00 00 03 00 and GNX1 shows “SNDBLK” followed by “BADACK” error indicating that a bulk transfer is expected.

Code 0F: Response to Code 0B Request Amp and Cab Patches

There is an initial response, followed by blocks of patch data with a final terminating response.

Initial response:

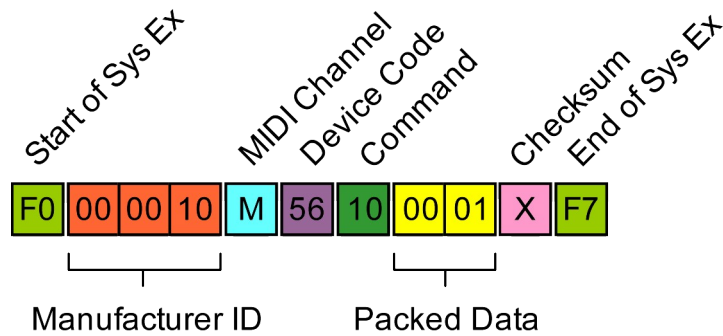


Unpacked data format is as follows:

Offset (dec)	Description
0	Always 0x01
1	Always 0x00
2	Always 0x00
3	Number of data messages that follow including terminator (minimum 0x01)
4	Always 0x00
5	Examples: 0x07F6 for 1 user patch, 0x0FEC for 2 patches (2x value), 0x47A6 for 9 patches. Probably an indicator of number of bytes following although exact correlation is unclear
6	

Data blocks in amp/cab pairs follow, 1 pair for each occupied slot. Each data block has a 0x2A command code.

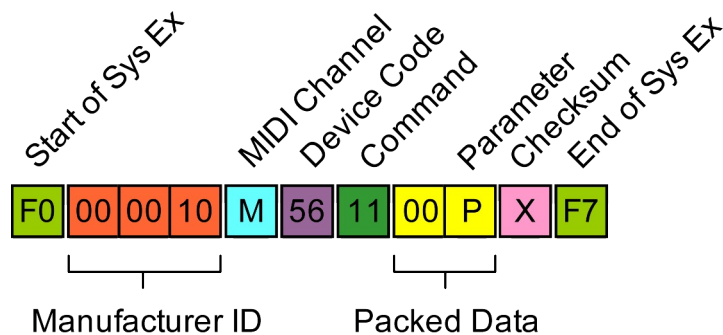
Code 10: Data Block Request Terminator



The final terminating message following data requested by a 0x0B command has a 0x10 code.

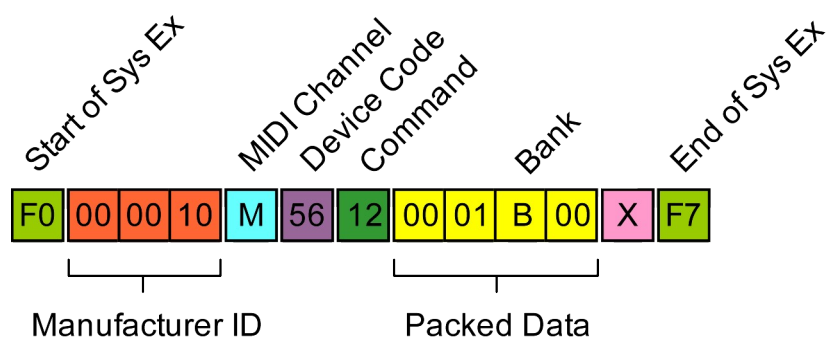
Code 11: Unknown

Message below with parameters in range 0x00 to 0x0F return code 0x7F.

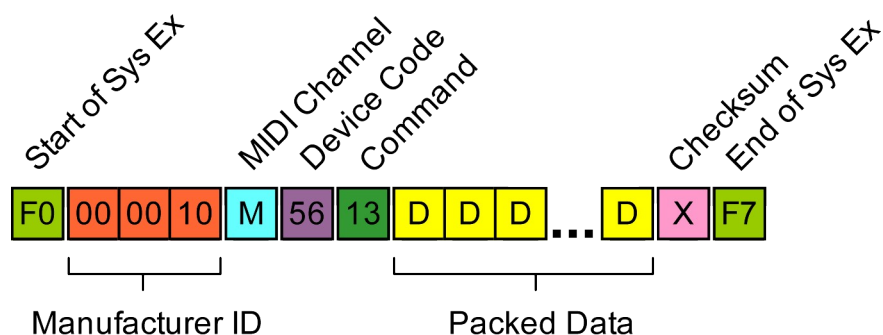


Code 12: Request Patch Names

Message below causes patch names to be transmitted. Parameter B is 0x00 for factory and 0x01 for user names.



Code 13: Response to Code 12 Request Patch Names



Names are returned as a set of 48 (decimal) null terminated ASCII strings. Note that the returned data does not indicate which bank it is.

Example response follows:

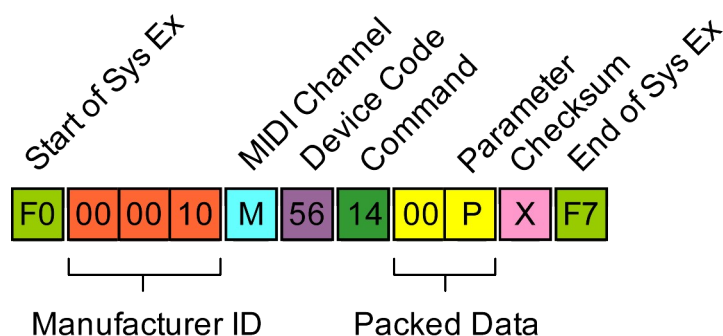
01	30	48 59 42 52 49 44	. O H Y B R I D
00	43 4C 4E 43 48 4F	00	. C L N C H O .
32	43 48 55 4E 4B 00 57		2 C H U N K . W
41	52 50 4D 45 00 42 4C		A R P M E . B L
4B	42 41 53 00 4D 45 41		K B A S . M E A
54	58 32 00 45 52 49 43		T X 2 . E R I C
20	4A 00 43 41 52 4C 4F		J . C A R L O
...			
59	41 20 20 00 53 54 55		Y A . S T U
54	54 52 00 54 52 49 50		T T R . T R I P
4C	54 00 44 49 56 42 4F		L T . D I V B O
4D	00		M .

KEY

- : Number of names
- : ASCII Name
- : String terminator (null)

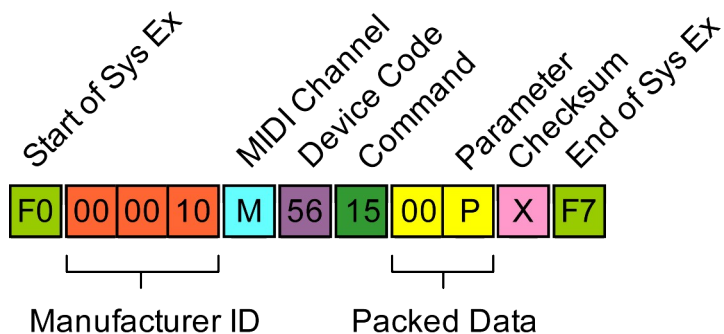
Code 14: Unknown

Message below with parameters in range 0x00 to 0x0F return code 0x7F.



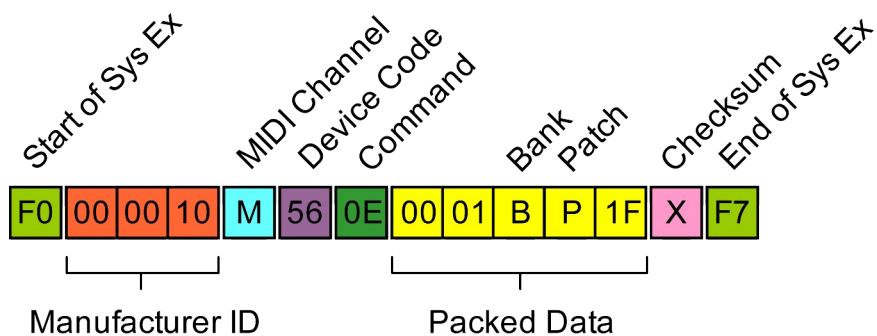
Code 15: Unknown

Message below with parameters in range 0x00 to 0x0F return code 0x7F.



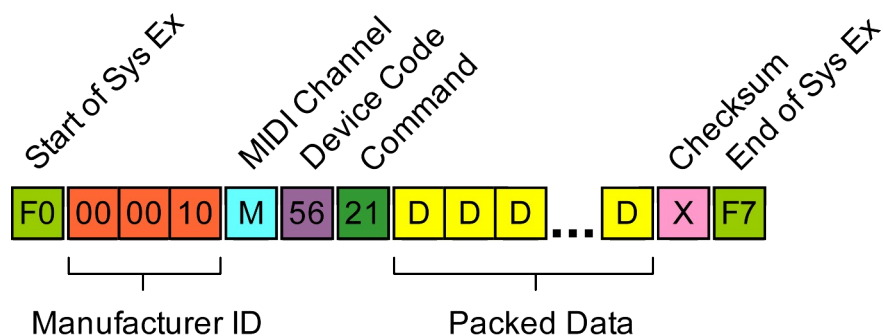
Code 20: Request Patch Name

The following requests the patch name and additional data from the specified buffer and patch. Parameter B is 0 for factory, 2 for user and 3 for edit buffer banks. Patch number is 0 – 47 (decimal), or 0 when querying the edit buffer.



Code 21: Response to Code 20 Request Patch Name

The following is received in response to a code 20 patch name request:



The unpacked data block is arranged as follows (edit buffer request):

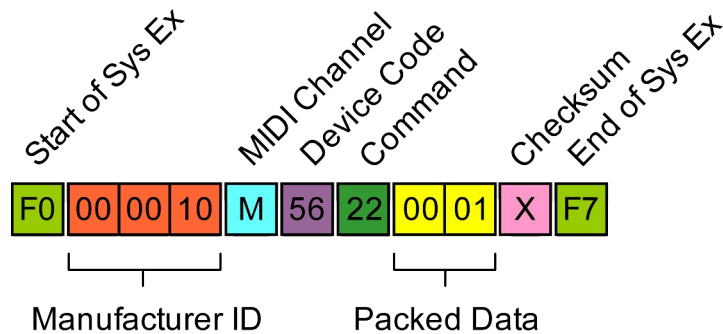
01	02	00	48	59	42	52	49	.	.	.	H	Y	B	R	I
44	00	00	08	09	A9	00		D

: Bank
 : Patch
 : ASCII Name (6 bytes)
 : String terminator (null)

If unacknowledged, the GNX1 will display "BADACK" error.

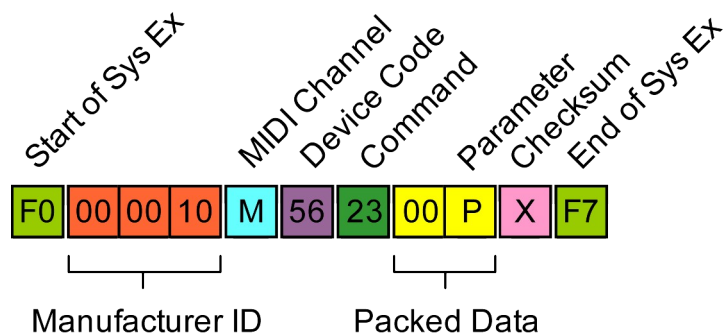
Code 22: End of Data Dump

Code is transmitted by the GNX1 at the end of a multi-message data dump.



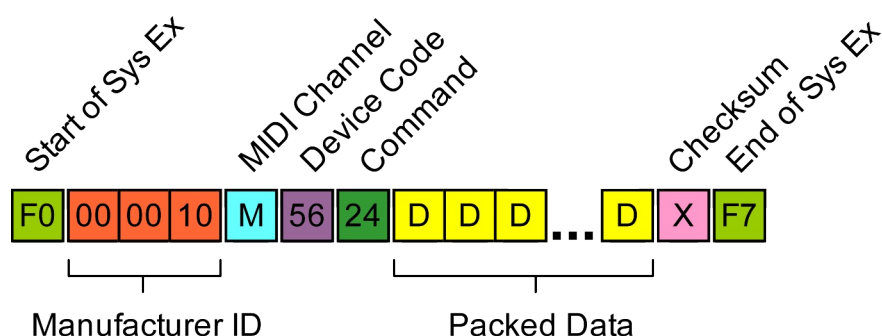
Code 23: Unknown

Message below with parameters in range 0x00 to 0x0F return code 0x7F.



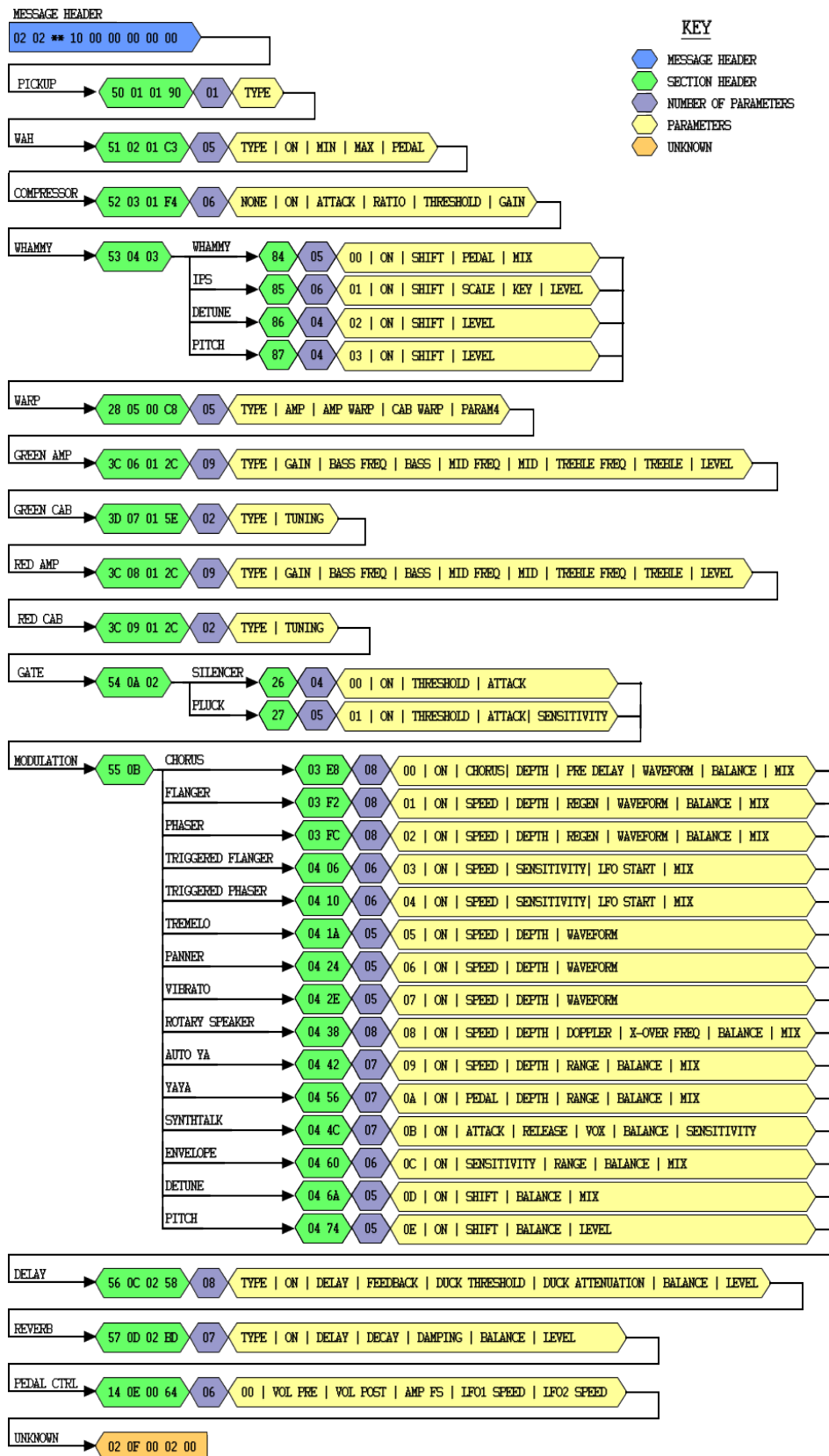
Code 24: Edit Buffer Patch Data

Patch effects data is returned in a code 24 message:



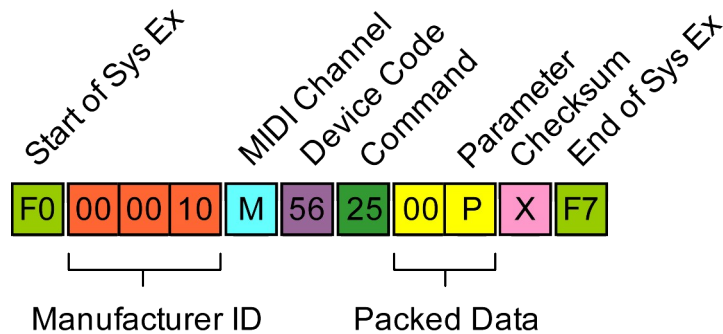
The data block contains data for effects settings where only the required number of values are included for each effect type which makes the data length for each effect section variable. Furthermore, the multi-byte number formatting described at the beginning of this document is used for effects parameter values which introduces further variability. The consequence of this is that the data block has to be analysed sequentially.

Code 24 data map:



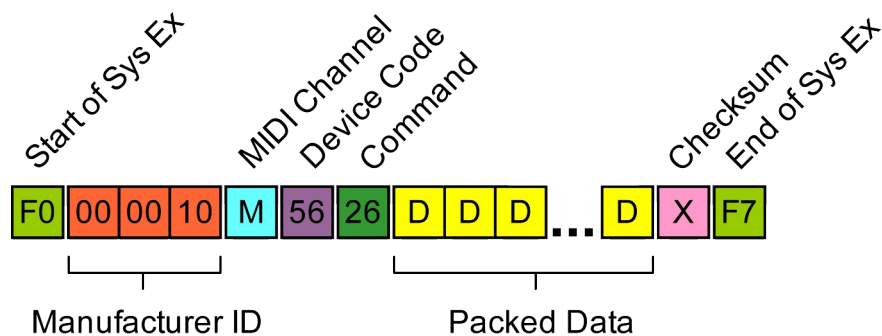
Code 25: No Response

Message below with parameters in range 0x00 to 0x0F return no response at all.



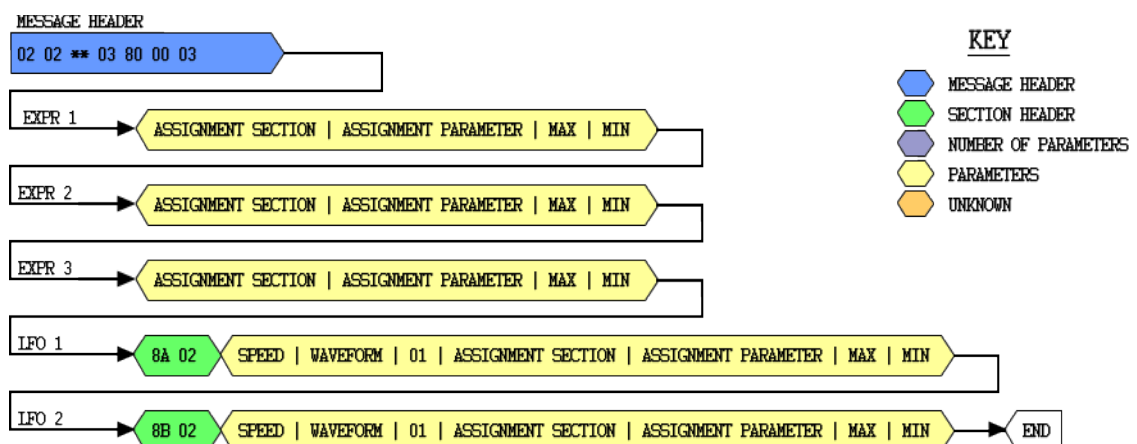
Code 26: LFO and Expression Pedals

Patch LFO and expression assignment data is returned in a code 26 message:



The data block contains data for effects settings where only the required number of values are included for each effect type which makes the data length for each effect section variable. Furthermore, the multi-byte number formatting described at the beginning of this document is used for effects parameter values which introduces further variability. The consequence of this is that the data block has to be analysed sequentially.

Code 26 data map:



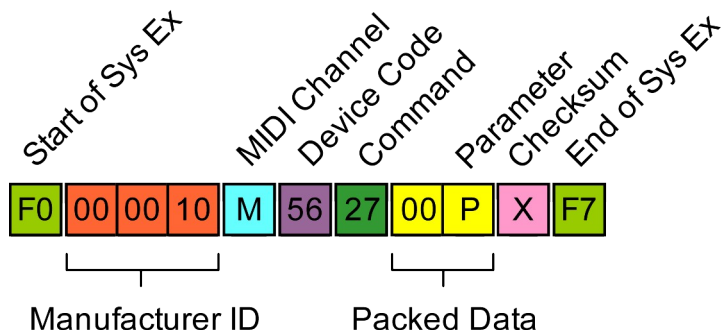
Code 26 Assignable Parameters

SECTION - PARAM	DESCRIPTION
FF FF	No Link
03 02	Compressor Attack
03 03	Compressor Ratio
03 04	Compressor Threshold
03 05	Compressor Gain
04 02	IPS/Whammy Shift Amount
04 03	IPS/Whammy Pedal
04 04	IPS/Whammy Mix
05 01	Amp Channel
05 02	Amp Warp
05 03	Cab Warp
05 04	Warp
06 01	Green Amp Gain
06 08	Green Amp Level
08 01	Red Amp Gain
08 08	Red Amp Level
0A 02	Gate Threshold
0A 03	Gate Attack
0B 02	Modulation Speed
0B 03	Modulation Depth
0B 04	Modulation Regeneration
0B 06	Modulation Balance
0B 07	Modulation Mix
0C 03	Delay feedback
0C 04	Delay Duck Threshold
0C 05	Delay Duck Attenuation
0C 06	Delay Balance
0C 07	Delay Level
0D 02	Reverb Pre Delay
0D 03	Reverb Decay
0D 05	Reverb Balance
0D 06	Reverb Level
0E 01	Vol Pre
0E 02	Vol Post
0E 04	LFO1 Speed

SECTION - PARAM	DESCRIPTION
0E 05	LFO2 Speed

Code 27: Unknown

Message below with parameters in range 0x00 to 0x0F return code 0x7F.



Code 28: Sync Data

A code 28 message is sent by the GNX1 when syncing with GENEdit. Values seem consistent for all patches so probably contains capability data:

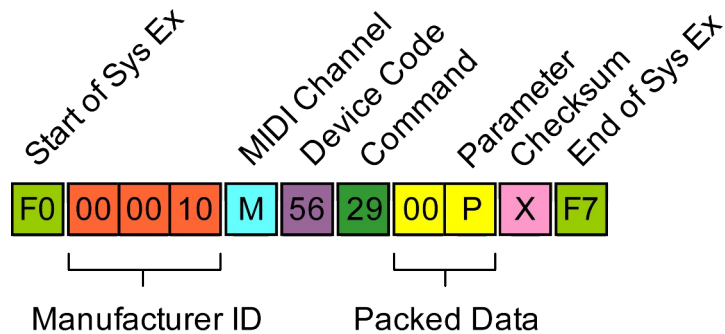
```

01 02 00 10 00 00 00 01 . . . . .
50 01 01 01 00 00 51 02 P . . . . Q .
01 01 01 00 52 03 01 01 . . . . R . .
02 00 53 04 01 01 03 00 . . S . . . .
28 05 00 00 3C 06 01 01 ( . . . < . . .
04 00 3D 07 01 01 06 00 . . = . . . .
3C 08 01 01 04 00 3D 09 < . . . . = .
01 01 08 00 54 0A 01 01 . . . . T . .
07 00 55 0B 01 02 0A 00 . . U . . . .
56 0C 02 02 0B 00 0B 01 V . . . . .
57 0D 02 02 0C 00 0C 01 W . . . . .
14 0E 00 00 02 0F 02 00 . . . . .
0D 00 0D 01 . . . . .
: Resemble codes seen elsewhere

```

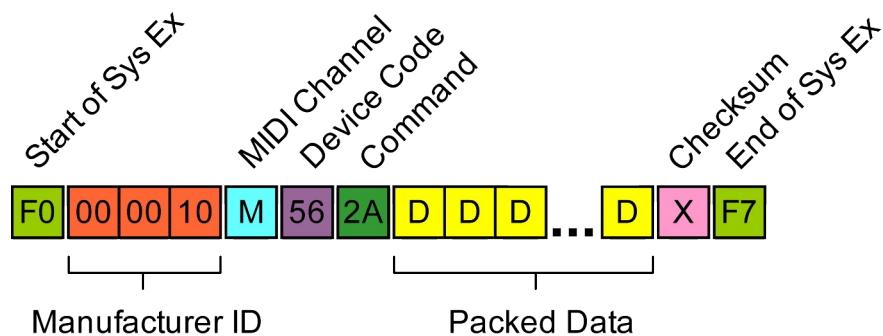
Code 27: Unknown

Message below with parameters in range 0x00 to 0x0F return code 0x7F.



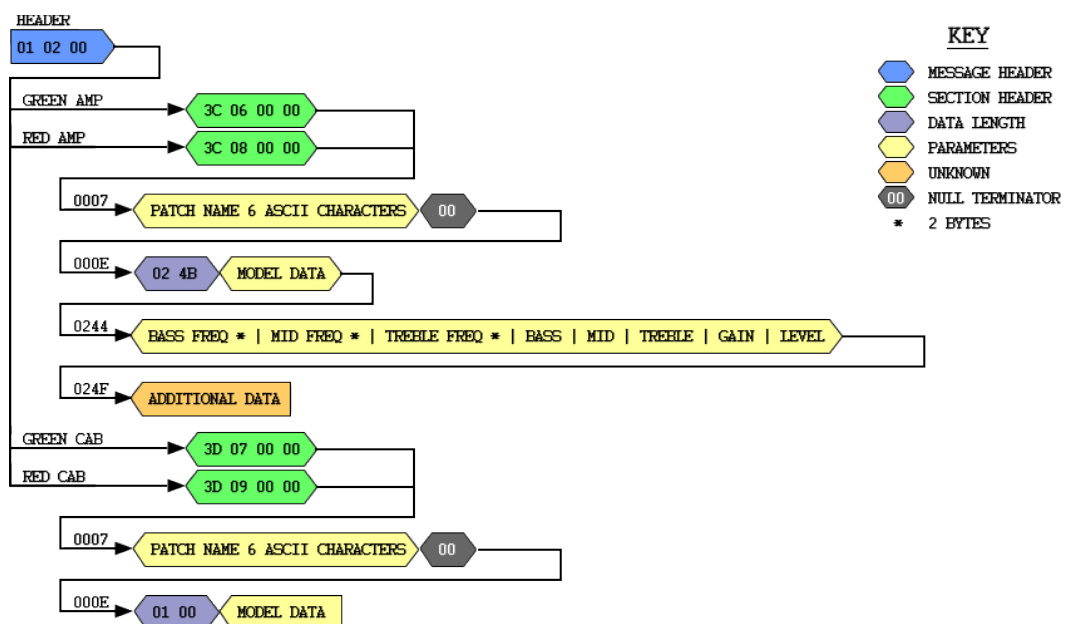
Code 2A: Amp or Cabinet Data Block

The code 2A is used to signify a bulk data message containing amp and cabinet data:



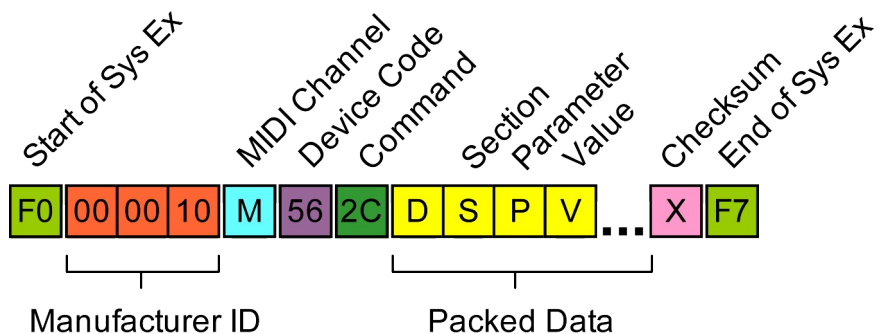
The numerical values in these data blocks are fixed 1 or 2 byte values and do not use the Number Format described at the beginning of this document. In particular, the values for equalizer frequencies consist of two bytes which can be combined to form a 16 bit integer (ms – ls order).

Code 2A data map:



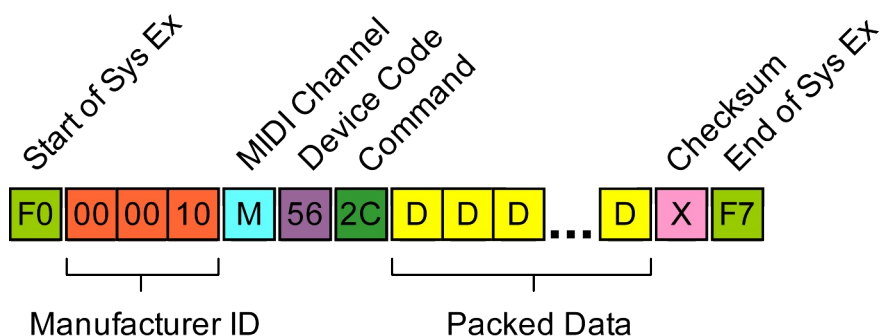
Code 2B: Unknown

Message below with parameters in range 0x00 to 0x0F return code 0x7F.



Code 2C: Parameter Change

Parameter changes are transmitted during editing with the following code:



Each parameter is described by a section/parameter byte pair which are detailed below. The parameter values use the Number Format described at the beginning of this document, which will vary the overall length of the message.

Code 2C data map:



KEY

- ▶ MESSAGE HEADER
- ▶ SECTION HEADER
- ▶ PARAMETERS
- 00 NULL TERMINATOR
- * FORMATTED

Code 2C Sections

SECTION	NAME
01	Pickup Simulator
02	Wah
03	Compressor
04	Whammy/IPS
05	Warp
06	Green Amp
07	Green Cabinet
08	Red Amp
09	Red Cabinet
0A	Noise Gate
0B	Chorus/Mod
0C	Delay
0D	Reverb
0E	Pedal

Section 01 (Pickup Simulator) Parameters

PARAMETER	NAME	VALUES
00	Type	00: Off 01: Single Coil>Humb 02: Humb>Single Coil

Section 02 (Wah) Parameters

PARAMETER	NAME	VALUES
00	Type	00: Cry, 01: Boutique, 02: Full
01	On/Off	00: Off, 01: On
02	Min	00 – 63 (0 – 99)
03	Max	00 – 63 (0 – 99)
04	Pedal	00 – 63 (0 – 99)

Section 03 (Compressor) Parameters

PARAMETER	NAME	VALUES
00		
01	On/Off	00: Off, 01: On
02	Attack	00: Fast, 01: Medium, 02: Slow
03	Ratio	00: 1.2:1 01: 1.5:1 02: 1.8:1 03: 2.0:1 04: 2.5:1 05: 3.0:1 06: 4.0:1 07: 5.0:1 08: 8.0:1 09: 10:1 0A: 20:1 0B: Inf
04	Threshold	00 – 63 (0 – 99)
05	Gain	00 – 14 (0 – 20)

Section 04 (Whammy/IPS) Parameters*Type 00: Whammy*

PARAMETER	NAME	VALUES
00	Type *	00: Whammy
01	On/Off	00: Off, 01: On
02	Shift	00: 1 Oct Up 01: 2 Oct Up 02: 2 nd Down 03: 2 nd Dn Rev 04: 4 th Down 05: 1 Oct Dn 06: 2 Oct Dn 07: Dive Bomb 08: m3>Maj3 Up 09: 2nd>Maj3 Up 0A: 3 rd >4th Up 0B: 4 th >5th Up 0C: 5 th >Oct Up 0D: Harm Oct Up 0E: Harm Oct Dn 0F: Oct Up>Dn
03	Pedal	0 – 63 (0 – 99)
04	Mix	00 – 63 (0 – 99)

Type 01: IPS

PARAMETER	NAME	VALUES
00	Type *	01: IPS
01	On/Off	00: Off, 01: On
02	Shift	00: Oct Down 01: 7 th Down 02: 6 th Down 03: 5 th Down 04: 4 th Down 05: 3 rd Down 06: 2 nd Down 07: 2 nd Up 08: 3 rd Up 09: 4 th Up 0A: 5 th Up 0B: 6 th Up 0C: 7 th Up 0D: Oct Up
03	Scale	00: Major 01: Minor 02: Dorian 03: Mixolydian 04: Lydian 05: Harm Minor
04	Key	00: E 01: F 02: Gb 03: G 04: Ab 05: A 06: Bb 07: B 08: C 09: Db 0A: D 0B: Eb
05	Level	00 – 63 (0 – 99)

Type 02: Detune

PARAMETER	NAME	VALUES
00	Type *	02: Detune
01	On/Off	00: Off, 01: On
02	Shift	00 / 0C / 18 (-24 / 0 / 24) (Step 2)
03	Level	00 – 63 (0 – 99)

Type 03: Pitch

PARAMETER	NAME	VALUES
00	Type *	03: Pitch
01	On/Off	00: Off, 01: On
02	Shift	00 / 18 / 30 (-24 / 0 / 24) (Step 1)
03	Level	00 – 63 (0 – 99)

* Type change causes GENX1 to transmit associated parameters

Section 05 (Warp) Parameters

PARAMETER	NAME	VALUES
00		
01	Amp Select	00: Green, 01: Red, 02: Yellow
02	Amp	00 – 62 (1 – 99) Green → Red
03	Cabinet	00 – 62 (1 – 99) Green → Red
04		

Section 06 (Green Amp) Parameters

PARAMETER	NAME	VALUES
00	Amp Type *	00: Direct 01: Blkfac 02: Boutiq 03: Rectif 04: Hotrod 05: Tweed 06: Brtcmb 07: Clntub 08: Brstk 09: Crunch 0A: Hlgain 0B: Blues 0C: Modgan 0D: Fuzz 0E: Bassmn 0F: Hiwatg 10: Acoust 11: User 1 12: User 2 13: User 3 14: User 4 15: User 5 16: User 6 17: User 7 18: User 8 19: User 9 1A: Custom
01	Gain	00 – 63 (0 – 99)
02	Bass Freq	00 – FA (50Hz – 300Hz) Step 1Hz **
03	Bass Level	00/0C/18 (-12/0/+12)
04	Mid Frequency	00 – 125C (300Hz – 5KHz) Step 1 Hz
05	Mid Level	00/0C/18 (-12/0/+12)
06	Treble Freq	00 – 1D4C (500Hz – 8KHz) Step 1Hz
07	Treble Level	00/0C/18 (-12/0/+12)
08	Level	00 – 63 (0 – 99)

* selecting amp type causes additional parameters to be transmitted

** 1 or 2 bytes (see FLAGS byte in message construct)

Section 07 (Green Cabinet) Parameters

PARAMETER	NAME	VALUES
00	Cabinet Type *	00: Direct 01: Am2x12 02: Br4x12 03: V 4x12 04: Br2x12 05: Am1x12 06: Bl2x12 07: Fn4x12 08: Gr4x12 09: User 1 0A: User 2 0B: User 3 0C: User 4 0D: User 5 0E: User 6 0F: User 7 10: User 8 11: User 9 12: Custom
01	Tuning	00/18/30 (-12/0/+12 in 0.5 steps)

Section 08 (Red Amp) Parameters

PARAMETER	NAME	VALUES
00	Amp Type *	00: Direct 01: Blkfac 02: Boutiq 03: Rectif 04: Hotrod 05: Tweed 06: Brtcmb 07: Clntub 08: Brstk 09: Crunch 0A: Hlgain 0B: Blues 0C: Modgan 0D: Fuzz 0E: Bassmn 0F: Hiwatg 10: Acoust 11: User 1 12: User 2 13: User 3 14: User 4 15: User 5 16: User 6 17: User 7 18: User 8 19: User 9 1A: Custom
01	Gain	00 – 63 (0 – 99)
02	Bass Freq	00 – FA (50Hz – 300Hz) Step 1Hz **
03	Bass Level	00/0C/18 (-12/0/+12)
04	Mid Frequency	00 – 125C (300Hz – 5KHz) Step 1 Hz
05	Mid Level	00/0C/18 (-12/0/+12)
06	Treble Freq	00 – 1D4C (500Hz – 8KHz) Step 1Hz
07	Treble Level	00/0C/18 (-12/0/+12)
08	Level	00 – 63 (0 – 99)

* selecting amp type causes additional parameters to be transmitted

** 1 or 2 bytes (see FLAGS byte in message construct)

Section 09 (Red Cabinet) Parameters

PARAMETER	NAME	VALUES
00	Cabinet Type *	00: Direct 01: Am2x12 02: Br4x12 03: V 4x12 04: Br2x12 05: Am1x12 06: Bl2x12 07: Fn4x12 08: Gr4x12 09: User 1 0A: User 2 0B: User 3 0C: User 4 0D: User 5 0E: User 6 0F: User 7 10: User 8 11: User 9 12: Custom
01	Tuning	00/18/30 (-12/0/+12 in 0.5 steps)

Section 0A (Noise Gate) Parameters

Type 00: Silencer

PARAMETER	NAME	VALUES
00	Type *	00: Silencer
01	On/Off	00: Off, 01: On
02	Threshold	00 – 28 (0 – 40)
03	Attack	00 – 09 (0 – 9)

Type 01: Pluck

PARAMETER	NAME	VALUES
00	Type *	01: Pluck
01	On/Off	00: Off, 01: On
02	Threshold	00 – 28 (0 – 40)
03	Attack	00 – 09 (0 – 9)
04	Sensitivity	00 – 63 (0 – 99)

* Type change causes GENX1 to transmit associated parameters

Section 0B (Chorus/Mod) Parameters**Type 00: Chorus**

PARAMETER	NAME	VALUES
00	Type *	00: Chorus
01	On/Off	00: Off, 01: On
02	Chorus	00 – 62 (1 – 99)
03	Depth	00 – 62 (1 – 99)
04	Pre Delay	00 – 13 (1 – 20)
05	Waveform	00: Triangle, 01: Sine, 02: Square
06	Balance	00 – 63 – C6 (-99 – 0 – 99) **
07	Mix	00 – 63 (0 – 99)

Type 01: Flanger

PARAMETER	NAME	VALUES
00	Type *	01: Flanger
01	On/Off	00: Off, 01: On
02	Speed	00 – 62 (1 – 99)
03	Depth	00 – 62 (1 – 99)
04	Regen	00 – 63 (0 – 99)
05	Waveform	00: Triangle, 01: Sine, 02: Square
06	Balance	00 – 63 – C6 (-99 – 0 – 99) **
07	Mix	00 – 63 (0 – 99)

Type 02: Phaser

PARAMETER	NAME	VALUES
00	Type *	02: Phaser
01	On/Off	00: Off, 01: On
02	Speed	00 – 62 (1 – 99)
03	Depth	00 – 62 (1 – 99)
04	Regen	00 – 63 (0 – 99)
05	Waveform	00: Triangle, 01: Sine, 02: Square
06	Balance	00 – 63 – C6 (-99 – 0 – 99) **
07	Mix	00 – 63 (0 – 99)

Type 03: Triggered Flanger

PARAMETER	NAME	VALUES
00	Type *	03: Triggered Flanger
01	On/Off	00: Off, 01: On
02	Speed	00 – 62 (1 – 99)
03	Sensitivity	00 – 62 (1 – 99)
04	LFO Start	00 – 63 (0 – 99)
05	Mix	00 – 63 (0 – 99)

Type 04: Triggered Phaser

PARAMETER	NAME	VALUES
00	Type *	04: Triggered Phaser
01	On/Off	00: Off, 01: On
02	Speed	00 – 62 (1 – 99)
03	Sensitivity	00 – 62 (1 – 99)
04	LFO Start	00 – 63 (0 – 99)
05	Mix	00 – 63 (0 – 99)

Type 05: Tremelo

PARAMETER	NAME	VALUES
00	Type *	05: Tremelo
01	On/Off	00: Off, 01: On
02	Speed	00 – 62 (1 – 99)
03	Depth	00 – 63 (0 – 99)
04	Waveform	00: Triangle, 01: Sine, 02: Square

Type 06: Panner

PARAMETER	NAME	VALUES
00	Type *	06: Panner
01	On/Off	00: Off, 01: On
02	Speed	00 – 62 (1 – 99)
03	Depth	00 – 63 (0 – 99)
04	Waveform	00: Triangle, 01: Sine, 02: Square

Type 07: Vibrato

PARAMETER	NAME	VALUES
00	Type *	07: Vibrato
01	On/Off	00: Off, 01: On
02	Speed	00 – 62 (1 – 99)
03	Depth	00 – 62 (1 – 99)
04	Waveform	00: Triangle, 01: Sine, 02: Square

Type 08: Rotary Speaker

PARAMETER	NAME	VALUES
00	Type *	08: Rotary Speaker
01	On/Off	00: Off, 01: On
02	Speed	00 – 63 (0 – 99)
03	Depth	00 – 63 (0 – 99)
04	Doppler	00 – 63 (0 – 99)
05	X-Over Freq	00 – 82 (200Hz – 1.50kHz) ** Step 10Hz
06	Balance	00 – 63 – C6 (-99 – 0 – 99) **
07	Mix	00 – 63 (0 – 99)

Type 09: Auto Yah

PARAMETER	NAME	VALUES
00	Type *	09: Auto Yah
01	On/Off	00: Off, 01: On
02	Speed	00 – 62 (1 – 99)
03	Depth	00 – 62 (1 – 99)
04	Range	00 – 31 (0 – 50)
05	Balance	00 – 63 – C6 (-99 – 0 – 99) **
06	Mix	00 – 63 (0 – 99)

Type 0A: YaYa

PARAMETER	NAME	VALUES
00	Type *	0A: YaYa
01	On/Off	00: Off, 01: On
02	Pedal	00 – 63 (0 – 99)
03	Depth	00 – 62 (1 – 99)
04	Range	00 – 31 (0 – 50)
05	Balance	00 – 63 – C6 (-99 – 0 – 99) **
06	Mix	00 – 63 (0 – 99)

Type 0B: Synthtalk

PARAMETER	NAME	VALUES
00	Type *	0B: Synthtalk
01	On/Off	00: Off, 01: On
02	Attack	00 – 63 (0 – 99)
03	Release	00 – 63 (1 – 100)
04	Vox	00 – 63 (0 – 99)
05	Balance	00 – 63 – C6 (-99 – 0 – 99) **
06	Sensitivity	00 – 62 (1 – 99)

Type 0C: Envelope

PARAMETER	NAME	VALUES
00	Type *	0C: Envelope
01	On/Off	00: Off, 01: On
02	Sensitivity	00 – 62 (1 – 99)
03	Range	00 – 62 (1 – 99)
04	Balance	00 – 63 – C6 (-99 – 0 – 99) **
05	Mix	00 – 63 (0 – 99)

Type 0D: Detune

PARAMETER	NAME	VALUES
00	Type *	0D: Detune
01	On/Off	00: Off, 01: On
02	Shift Amount	00/18/30 (-24/0/+24) (Step 1)
03	Balance	00 – 63 – C6 (-99 – 0 – 99) **
04	Mix	00 – 63 (0 – 99)

Type 0E: Pitch

PARAMETER	NAME	VALUES
00	Type *	0E: Pitch
01	On/Off	00: Off, 01: On
02	Shift Amount	00/0C/24 (-12/0/+24) (Step 1)
03	Balance	00 – 63 – C6 (-99 – 0 – 99) **
04	Level	00 – 63 (0 – 99)

* Type change causes GENX1 to transmit associated parameters

** 1 or 2 bytes (see FLAGS byte in message construct)

Section 0C (Delay) Parameters

PARAMETER	NAME	VALUES
00	Type *	00: Mono 01: Ping Pong 02: Analog 03: Alg Pong
01	On/Off	00: Off, 01: On
02	Delay Time	00 – 7D0 (0ms – 2s) Step 1ms ***
03	Feedback	00 – 64 (0 – 99 + R-Hold)
04	Duck Thresh	00 – 63 (1 – 99 + Off)
05	Duck Atten	00 – 63 (0 – 99)
06	Balance	00 – 63 – C6 (-99 – 0 – 99) **
07	Level	00 – 63 (0 – 99)

* Type change causes GENX1 to transmit associated parameters

** 1 or 2 bytes (see FLAGS byte in message construct)

*** 1, 2 or 3 bytes (see FLAGS byte in message construct)

Section 0D (Reverb) Parameters

PARAMETER	NAME	VALUES
00	Type *	00: Studio 01: Room 02: Club 03: Plate 04: Hall 05: Amphitheatre 06: Church 07: Garage 08: Arena 09: Spring
01	On/Off	00: Off, 01: On
02	Pre Delay	00 – 0F (0 – 15)
03	Decay	00 – 62 (1 – 99)
04	Damping	00 – 64 (0 – 99)
05	Balance	00 – 63 – C6 (-99 – 0 – 99) **
06	Level	00 – 63 (0 – 99)

* Type change causes GENX1 to transmit associated parameters

** 1 or 2 bytes (see FLAGS byte in message construct)

Section 0E Pedal Parameters

PARAMETER	NAME	VALUES
00	Unknown	01: Volume Pre 02: Volume Post
01	Value	Min – Max Expression Setting

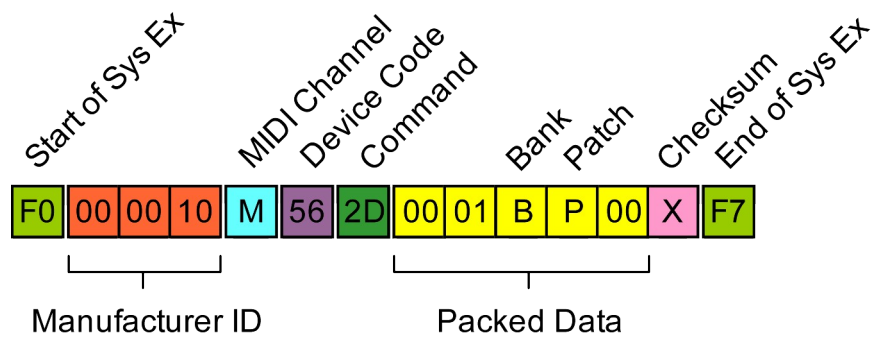
Section 8A (LFO 1) Parameters

PARAMETER	NAME	VALUES
TBC	Speed	00 – B9 (0.05Hz – 0.99Hz) Step 0.01Hz (1Hz – 10Hz) Step 0.1Hz
TBC	Waveform	00: Triangle, 01: Sine, 02: Square

Section 8B (LFO 2) Parameters

PARAMETER	NAME	VALUES
TBC	Speed	00 – B9 (0.05Hz – 0.99Hz) Step 0.01Hz (1Hz – 10Hz) Step 0.1Hz
TBC	Waveform	00: Triangle, 01: Sine, 02: Square

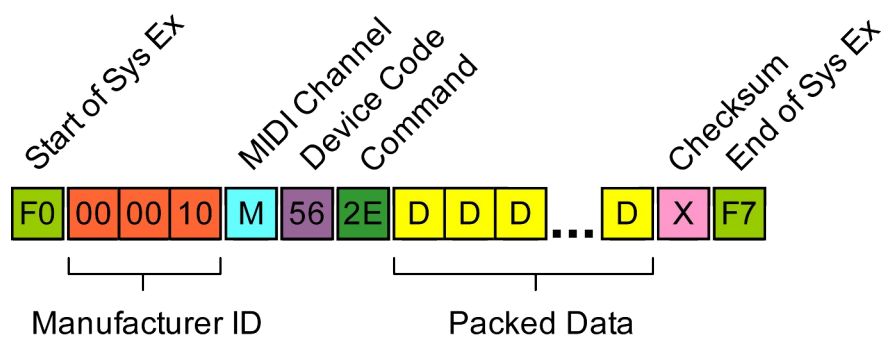
Code 2D: Patch Change



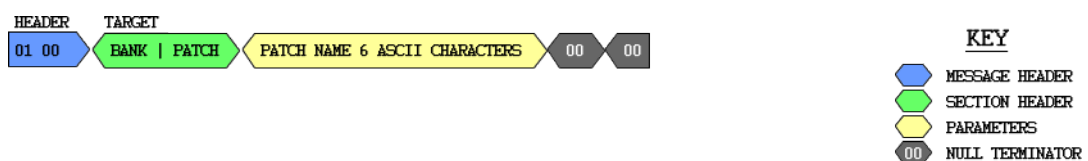
The GNX1 emits the following when a new patch is selected for editing. Parameter B is 0 for factory, 2 for user. Patch number is 0 – 47 (decimal).

Code 2E: Patch Saved With Name

The GNX1 emits the following message when the current edit buffer is saved to a patch location with a specified name:

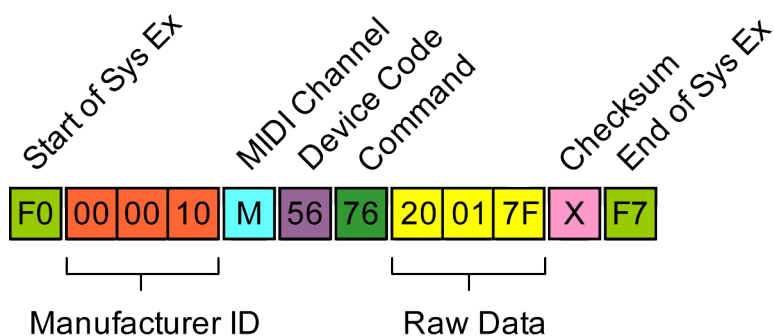


Code 2E data map:



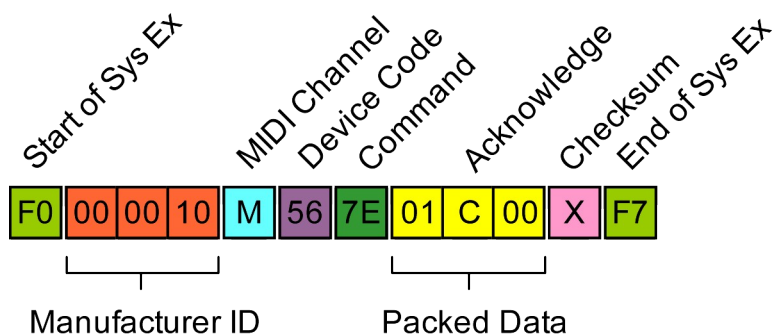
Code 76: Keep Alive

The Digitech GENEdit programs persistently sends a 'keep alive' command to the GNX1, which the GNX1 acknowledges commands with a code 7E command:



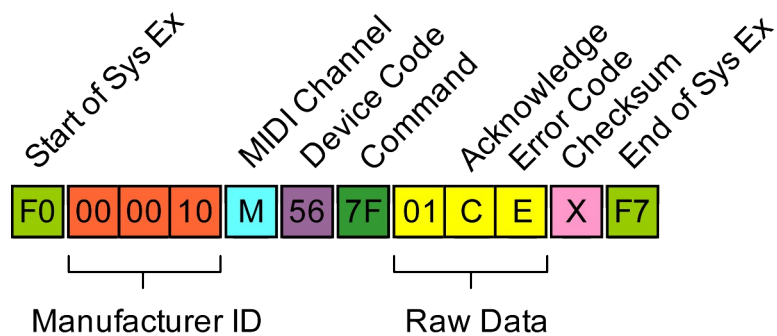
Code 7E: Acknowledge

The GNX1 acknowledges commands with a code 7E command, or notifies



Code 7F: Error

The GNX1 reports errors (e.g.. unrecognised code, system exclusive checksum error, with a code 7F command:



ERROR CODE	DESCRIPTION
00	No error
01	Unrecognised command
08	System exclusive checksum error

GNX1 / GENEdit Startup Dialog

When the GENEdit Software establishes a link to a GNX1 device it enters a dialog to synchronise the two:

