



# Table of Contents

<b>Introduction</b> .....	2
<b>MIDI Basics</b> .....	3
<b>General Format</b> .....	4
<b>Procedures</b> .....	5
Request Configuration Address - 00 <sup>(h)</sup> .....	5
Receive Configuration Address - 41 <sup>(h)</sup> .....	5
Request One Program - 01 <sup>(h)</sup> .....	5
Receive One Program - 42 <sup>(h)</sup> .....	6
Request RAM Area - 06 <sup>(h)</sup> .....	6
Data Response - 10 <sup>(h)</sup> .....	7
Reset Program - 20 <sup>(h)</sup> .....	7
Reset Device - 21 <sup>(h)</sup> .....	7
Reset Factory Settings - 22 <sup>(h)</sup> .....	8
Request Algorithm - 31 <sup>(h)</sup> .....	8
Receive Algorithm - 30 <sup>(h)</sup> .....	8
Request Bulk Dump - 49 <sup>(h)</sup> .....	9
Receive Large RAM Area - 48 <sup>(h)</sup> .....	9
Receive RAM Area - 47 <sup>(h)</sup> .....	10
Request Module Table - 50 <sup>(h)</sup> .....	10
Respond Module Table - 51 <sup>(h)</sup> .....	11
Request Algorithm Link Table - 52 <sup>(h)</sup> .....	12
Respond Algorithm Link Table - 53 <sup>(h)</sup> .....	12
Request Parameter Value - 17 <sup>(h)</sup> .....	13
Receive Parameter Value - 18 <sup>(h)</sup> .....	13
Receive Key Scan Code - 54 <sup>(h)</sup> .....	13
Receive Hold Key Scan Code - 55 <sup>(h)</sup> .....	14
Respond Key Accepted - 56 <sup>(h)</sup> .....	14
Request Parameter Info - 58 <sup>(h)</sup> .....	15
Respond Parameter Info - 59 <sup>(h)</sup> .....	15
Receive Parameter Value - 63 <sup>(h)</sup> .....	16
Return to Program Screen - 60 <sup>(h)</sup> .....	16
Request Error/Status Flag - 62 <sup>(h)</sup> .....	16
Respond Error/Status Flag - 61 <sup>(h)</sup> .....	17
<b>Key Scan Codes</b> .....	19
<b>Misc Information</b> .....	19

# *Introduction*

This publication has been created to document the different types of MIDI System Exclusive control available in DigiTech MIDI compatible S-DISC™ products. MIDI System Exclusive communication allows a user to control a device directly through MIDI in a very powerful and flexible manner. Much care has gone into creating and testing this documentation but DigiTech can not guarantee 100% accuracy nor can DigiTech be held responsible for presets and data lost in a product from System Exclusive Information use.

A working knowledge of the MIDI Version 1.0 Detailed Specification can be useful while implementing any manufacturer's SysEx commands. For a copy of the MIDI spec contact:

MIDI Manufacturers Association  
5316 West 57th Street  
Los Angeles, California 90056 USA  
(213) 649-6434

It is a good practice to back up any user programs and/or user algorithms in the device before you begin to work with SysEx commands since incorrect commands can produce unexpected results. Should you encounter any discrepancies between this documentation and a DigiTech S-DISC™ MIDI product please feel free to contact us at:

DOD Electronics Corp.  
Attn: Product Management Dept.  
8760 South Sandy Parkway  
Sandy, Utah 84070 USA  
(801) 566-8800

The information contained in this document only pertains completely to the following DigiTech products and their corresponding versions of software.

<b>TSR-24</b>	Software version 2.0 (or higher)
<b>GSP-2101</b>	Software version 2.0 (or higher)
<b>Valve FX</b>	Software version 1.0 (or higher)

The products listed below may have full SysEx commands implemented into them at some later date. However, they do respond to the basic Bulk Dump and Program Dump Procedures.

**TSR-12**  
**RP-10**  
**Legend II**

## MIDI Channels

All DigiTech MIDI compatible products can be set so that they 'listen' to 1 of 16 discrete MIDI channels. The devices could also be set up to receive information on all channels in an Omni mode or to none of the channels in a Disabled mode. Please refer to the product's user manual for further information on each specific product.

## Program Changes

When the device is set to receive one or all of the MIDI channels, it will recognize a Program Change command. If the number is legal (e.g. within the device's range of selectable programs) the DigiTech device will respond by changing to the new program. MIDI sends Program Changes 0 through 127. DigiTech devices begin their program numbering with 1, thus Program Change 0 will select program 1. DigiTech products also allow the user to 're-map' the Program Change commands so that Program Change 0 could call program 135, if so desired. Please refer to the product's user manual for further information.

## Continuous Controllers

A MIDI Continuous Controller may be 'Linked' to nearly any available parameter in a DigiTech S-Disc™ product. Local CC links and their ranges are treated uniquely in each program with the assignments being saved only if they are stored as part of the program. Some DigiTech S-Disc™ products also allow Global CC links. Please refer to the product's user manual for further information.

## Channel Pressure

DigiTech products treat Channel Pressure as just another type of Continuous Controller which can be linked to parameters for control.

## Pitch Bend

Pitch Bend information is a higher resolution controller that is not supported in DigiTech products due to the density of the information and the resources needed to process it properly. Many MIDI control products offer the ability to map Pitch Bend to normal Continuous Controllers if a user wanted to control a parameter with a device such as a Pitch Bend Wheel.

# General SysEx Format

The general format for the System Exclusive information is as follows (note: all SysEx values are displayed in Hex format along with this symbol <sub>(h)</sub> for clarification):

<u>Hex Value</u>	<u>Definition</u>	
F0 <sub>(h)</sub>	System Exclusive 'Begin Message' byte	
00 <sub>(h)</sub> 00 <sub>(h)</sub> 10 <sub>(h)</sub>	} Manufacturer's ID Number (DOD/DigiTech)	
0n <sub>(h)</sub>		n = channel number minus one e.g. 00 <sub>(h)</sub> is channel 1 01 <sub>(h)</sub> is channel 2 etc... 0F <sub>(h)</sub> is channel 16
nn		nn = device ID number 40 <sub>(h)</sub> identifies the TSR-24 (Extended SysEx) 41 <sub>(h)</sub> identifies the GSP-2101 (Extended SysEx) 42 <sub>(h)</sub> identifies the TSR-12 (Basic SysEx) 43 <sub>(h)</sub> identifies the RP-10 (Basic SysEx) 44 <sub>(h)</sub> identifies the Legend II (Basic SysEx) 45 <sub>(h)</sub> identifies the Valve FX (Extended SysEx)
pp	pp = Procedure number. The different procedure's names and general formats are described in the <u>Procedures</u> Section (page 5).	
dd dd .. .. .. dd dd	} dd = Data as needed by procedures.	
F7 <sub>(h)</sub>		System Exclusive 'End Message' byte

In the following section, SYS\_HEAD refers to a valid System Exclusive header. The System Exclusive header starts with the System Exclusive status byte, and includes all bytes through the DigiTech device type. F7<sup>(h)</sup> is the System Exclusive 'Message End' byte. All SysEx numbers in the format definition of each procedure are given as hexadecimal values along with this symbol <sup>(h)</sup> for clarification. Binary numbers are identified with this symbol <sup>(b)</sup>.

## REQUEST CONFIGURATION ADDRESS (00h)

SYS\_HEAD, 00<sup>(h)</sup>, F7<sup>(h)</sup>

When a **Request Configuration Address** procedure is received, the DigiTech device will respond with a **Receive Configuration Address** procedure.

## RECEIVE CONFIGURATION ADDRESS (41h)

SYS\_HEAD 41<sup>(h)</sup>, ww, xx, yy, zz, F7<sup>(h)</sup>

ww    0000000v<sup>(b)</sup> where v is bit 7 of the ram address.  
xx    0vvvvvvv<sup>(b)</sup> where v is bits 6 - 0 of the ram address.  
yy    0000000v<sup>(b)</sup> where v is bit 15 of the ram address.  
zz    0vvvvvvv<sup>(b)</sup> where v is bits 14 - 8 of the ram address.

The **Receive Configuration Address** procedure is sent in response to a **Request Configuration Address** procedure. This address can be used with the **Request RAM Area** procedure to directly access the RAM used for configuring the DigiTech device (implemented mainly for in-house programming).

## REQUEST ONE PROGRAM (01h)

SYS\_HEAD, 01<sup>(h)</sup>, yy, zz, F7<sup>(h)</sup>

When yy = 00<sup>(h)</sup>, then zz = Program numbers 1 through 128 (00-7Fh)  
When yy = 01<sup>(h)</sup>, then zz = Program numbers 129 through 256 (00-7Fh)

When a **Request One Program** procedure is received, the DigiTech device will respond with a **Receive One Program** procedure.

## RECEIVE ONE PROGRAM (42h)

SYS\_HEAD 42<sub>(h)</sub>, yy, zz, dd(1<sup>st</sup>)<sup>1</sup> dd(1<sup>st</sup>)<sup>2</sup>, ... dd(n<sup>th</sup>)<sup>1</sup>, dd(n<sup>th</sup>)<sup>2</sup>, F7<sub>(h)</sub>

When yy = 00<sub>(h)</sub>, then zz = Program numbers 1 through 128 (00-7Fh)

When yy = 01<sub>(h)</sub>, then zz = Program numbers 129 through 256 (00-7Fh)

dd(n)<sup>1</sup> 0000000v<sub>(b)</sub> where v is bit 7 of the n<sup>th</sup> byte of the received program.

dd(n)<sup>2</sup> 0vvvvvvv<sub>(b)</sub> where v is bits 6 through 0 of the n<sup>th</sup> byte of the received program.

The **Receive One Program** procedure is used to load a program into the DigiTech device. The number of program bytes to be received (n) is different for each program. If the DigiTech device is sent a **Receive One Program** procedure where the program number is not a valid RAM destination, it will be ignored. For examples of each product's actual information format, please see Misc Information (page 20).

## REQUEST RAM AREA (06h)

SYS\_HEAD, 06<sub>(h)</sub>, bb, ww, xx, yy, zz, nn, F7<sub>(h)</sub>

bb Bank number of ram being requested.

ww 0000000v<sub>(b)</sub> where v is bit 15 of the ram address.

xx 0vvvvvvv<sub>(b)</sub> where v is bits 14 - 8 of the ram address.

yy 0000000v<sub>(b)</sub> where v is bit 7 of the ram address.

zz 0vvvvvvv<sub>(b)</sub> where v is bits 6 - 0 of the ram address.

nn 1 to 7F<sub>(h)</sub>, the number of ram bytes to be sent.

The **Request RAM Area** procedure is used to have the DigiTech device send an area of its internal ram to a host midi device. When a **Request RAM Area** procedure is received, the DigiTech device will respond with a **Data Response** procedure. Use the address information from the **Receive Configuration Address** procedure to identify the RAM locations which may be accessed from the various products.

## DATA RESPONSE (10h)

SYS\_HEAD, 10<sub>(h)</sub>, bb, ww, xx, yy, z,z nn, dd(1<sup>st</sup>)<sup>1</sup>, dd(1<sup>st</sup>)<sup>2</sup>, ... dd(c<sup>th</sup>)<sup>1</sup>, dd(c<sup>th</sup>)<sup>2</sup>, F7<sub>(h)</sub>

bb Bank number of ram being sent. Currently the bank normally used in these DigiTech devices is 00h.

ww 0000000v<sub>(b)</sub> where v is bit 15 of the ram address.

xx 0vvvvvvv<sub>(b)</sub> where v is bits 14 - 8 of the ram address.

yy 0000000v<sub>(b)</sub> where v is bit 7 of the ram address.

zz 0vvvvvvv<sub>(b)</sub> where v is bits 6 - 0 of the ram address.

nn 01<sub>(h)</sub> to 7F<sub>(h)</sub>, the number of ram bytes in this response.

dd(c)<sup>1</sup> 0000000v<sub>(b)</sub> where v is bit 7 of the c<sup>th</sup> byte being sent.

dd(c)<sup>2</sup> 0vvvvvvv<sub>(b)</sub> where v is bits 6 through 0 of the c<sup>th</sup> byte being sent.

The **Data Response** procedure is sent from a DigiTech device in response to a **Request RAM Area** procedure. See the Misc Information (page 19) for the mapped descriptions of the information in this data packet. If a **Data Response** procedure is received by a DigiTech device, it is ignored. However, this data could be configured for use with the **Receive RAM Area** procedure.

## RESET PROGRAM (20h)

SYS\_HEAD, 20<sub>(h)</sub>, F7<sub>(h)</sub>

The **Reset Program** procedure causes the DigiTech product to reload the current saved program and reload the fast program cache.

## RESET DEVICE (21h)

SYS\_HEAD, 21<sub>(h)</sub>, F7<sub>(h)</sub>

The **Reset Device** procedure causes the DigiTech device to reboot the software as if the power had been turned off and then back on. If edits have been made to the current program without saving, they will be lost.



## RESET FACTORY SETTINGS (22h)

SYS\_HEAD, 22<sup>(h)</sup>, rr, xx, F7<sup>(h)</sup>

When rr = 00<sup>(h)</sup> the factory programs are reloaded.

When rr = 01<sup>(h)</sup> the factory software defaults are reloaded.

When rr = 02<sup>(h)</sup> both programs and software are reloaded.

xx     Reserved for future use.

The **Reset Factory Settings** procedure causes the DigiTech device to reload a sections of the EPROM factory defaults.

## REQUEST ALGORITHM (31h)

SYS\_HEAD, 31<sup>(h)</sup>, aa, F7<sup>(h)</sup>

aa     Algorithms numbers 1 through 128 (00-7F<sup>(h)</sup>) where 1 through 64 are user algorithms and 65 through 128 are factory algorithms.

When a **Request Algorithm** procedure is received, then DigiTech device will respond with a **Receive Algorithm** procedure.

## RECEIVE ALGORITHM (30h)

SYS\_HEAD, 30<sup>(h)</sup>, aa, dd(1<sup>st</sup>)<sup>1</sup> dd(1<sup>st</sup>)<sup>2</sup>, ... dd(n<sup>th</sup>)<sup>1</sup>, dd(n<sup>th</sup>)<sup>2</sup>, F7<sup>(h)</sup>

aa     Algorithms numbers 1 through 128 (00-7F<sup>(h)</sup>) where 1 through 64 are user algorithms and 65 through 128 are factory algorithms.

dd(n)<sup>1</sup> 0000000v<sup>(b)</sup> where v is bit 7 of the n<sup>th</sup> byte of the received program.

dd(n)<sup>2</sup> 0vvvvvvv<sup>(b)</sup> where v is bits 6 through 0 of the n<sup>th</sup> byte of the received program.

The **Receive Algorithm** procedure is used to load an algorithm into the DigiTech device that has the ability to implement user defined Algorithms. The number of algorithm bytes to be received (n) is different for each algorithm.

**Note:** if a **Receive Algorithm** procedure is received as a result of a **Request Algorithm** procedure, byte (aa) may not match the requested algorithm number. This is due to a small bug that was left undetected. This causes no problems since the device receiving the algorithm will place the algorithm in the requested user algorithm slot **or** in the next available slot if the number is already used.

## REQUEST BULK DUMP (49h)

SYS\_HEAD, 49<sup>(h)</sup>, F7<sup>(h)</sup>

When a **Request Bulk Dump** procedure is received, the DigiTech device will respond with one or more **Receive Large RAM Area** procedures.

## RECEIVE LARGE RAM AREA (48h)

SYS\_HEAD, 48<sup>(h)</sup>, bb, ww, xx, yy, z,z nn<sup>1</sup>, nn<sup>2</sup>, nn<sup>3</sup>, dd(1<sup>st</sup>)<sup>1</sup>, dd(1<sup>st</sup>)<sup>2</sup>, ... dd(c<sup>th</sup>)<sup>1</sup>, dd(c<sup>th</sup>)<sup>2</sup>, F7<sup>(h)</sup>

- bb Bank number of RAM being received. Currently the bank normally used in these DigiTech devices is 00h.
- ww 0000000v<sup>(b)</sup> where v is bit 15 of the RAM address.
- xx 0vvvvvvv<sup>(b)</sup> where v is bits 14 - 8 of the RAM address.
- yy 0000000v<sup>(b)</sup> where v is bit 7 of the RAM address.
- zz 0vvvvvvv<sup>(b)</sup> where v is bits 6 - 0 of the RAM address.
- nn<sup>1</sup> 0vvvvvvv<sup>(b)</sup> where v is bits 6 - 0 of the transfer count.
- nn<sup>2</sup> 0vvvvvvv<sup>(b)</sup> where v is bits 13 - 7 of the transfer count.
- nn<sup>3</sup> 0vvvvvvv<sup>(b)</sup> where v is bits 20 - 14 of the transfer count.
- dd(c)<sup>1</sup> 0000000v<sup>(b)</sup> where v is bit 7 of the c<sup>th</sup> byte being received.
- dd(c)<sup>2</sup> 0vvvvvvv<sup>(b)</sup> where v is bits 6 through 0 of the c<sup>th</sup> byte being sent.

The **Receive Large RAM Area** procedures are used to load an area of a DigiTech device's internal RAM from a host MIDI device. Use the information in the **Receive Configuration** procedure to identify the RAM locations which may be accessed from the various products.

## RECEIVE RAM AREA (47h)

SYS\_HEAD, 47<sup>(h)</sup>, bb, ww, xx, yy, z,z nn, dd(1<sup>st</sup>)<sup>1</sup>, dd(1<sup>st</sup>)<sup>2</sup>, ... dd(nn<sup>th</sup>)<sup>1</sup>, dd(nn<sup>th</sup>)<sup>2</sup>, F7<sup>(h)</sup>

bb Bank number of RAM being sent. Currently the bank normally used in these DigiTech devices is 00h.

ww 0000000v<sup>(b)</sup> where v is bit 15 of the RAM address.

xx 0vvvvvvv<sup>(b)</sup> where v is bits 14 - 8 of the RAM address.

yy 0000000v<sup>(b)</sup> where v is bit 7 of the RAM address.

zz 0vvvvvvv<sup>(b)</sup> where v is bits 6 - 0 of the RAM address.

nn 01<sup>(h)</sup> to 7F<sup>(h)</sup>, the number of RAM data bytes in this response.

dd(c)<sup>1</sup> 0000000v<sup>(b)</sup> where v is bit 7 of the c<sup>th</sup> byte being sent.

dd(c)<sup>2</sup> 0vvvvvvv<sup>(b)</sup> where v is bits 6 through 0 of the c<sup>th</sup> byte being sent.

The **Receive RAM Area** procedures are used to load an area of a DigiTech device's internal RAM from a host MIDI device. Use the information in the **Receive Configuration Address** procedure to identify the RAM locations which may be accessed from the various products. The product will also perform an equivalent to a **Receive Reset Program** procedure.

## REQUEST MODULE TABLE (50h)

SYS\_HEAD, 50<sup>(h)</sup>, F7<sup>(h)</sup>

When a **Request Module Table** procedure is received, the DigiTech device will respond with a **Respond Module Table** procedure.

## RESPOND MODULE TABLE (51h)

SYS\_HEAD, 51<sub>(h)</sub>, mm(1<sup>st</sup>), nn(1<sup>st</sup>)<sup>1</sup>, nn(1<sup>st</sup>)<sup>2</sup>,... nn(1<sup>st</sup>)<sup>19</sup>, nn(1<sup>st</sup>)<sup>20</sup>, ss(1<sup>st</sup>)<sup>1</sup>, ss(1<sup>st</sup>)<sup>2</sup>,... ss(1<sup>st</sup>)<sup>5</sup>, ss(1<sup>st</sup>)<sup>6</sup>, pp(1<sup>st</sup>)<sup>1</sup>, pp(1<sup>st</sup>)<sup>2</sup>, rr(1<sup>st</sup>)<sup>1</sup>, rr(1<sup>st</sup>)<sup>2</sup>,... mm(x<sup>th</sup>), nn(x<sup>th</sup>)<sup>1</sup>, nn(x<sup>th</sup>)<sup>2</sup>,... nn(x<sup>th</sup>)<sup>5</sup>, nn(x<sup>th</sup>)<sup>6</sup>, 00<sub>(h)</sub>, ss(x<sup>th</sup>)<sup>1</sup>, ss(x<sup>th</sup>)<sup>2</sup>,... ss(x<sup>th</sup>)<sup>19</sup>, ss(x<sup>th</sup>)<sup>20</sup>, 00<sub>(h)</sub>, pp(x<sup>th</sup>)<sup>1</sup>, pp(x<sup>th</sup>)<sup>2</sup>, rr(x<sup>th</sup>)<sup>1</sup>, rr(x<sup>th</sup>)<sup>2</sup>, 00<sub>(h)</sub>, F7<sub>(h)</sub>

mm      Module I.D. number

nn(x<sup>th</sup>)<sup>1</sup>

nn(x<sup>th</sup>)<sup>2</sup>

...      Each ASCII byte for the x<sup>th</sup> Module's short name string (up to 6 characters).

nn(x<sup>th</sup>)<sup>5</sup>

nn(x<sup>th</sup>)<sup>6</sup>

00<sub>(h)</sub>

ss(x<sup>th</sup>)<sup>1</sup>

ss(x<sup>th</sup>)<sup>2</sup>

...      Each ASCII byte for the x<sup>th</sup> Module's long name string (up to 20 characters).

ss(x<sup>th</sup>)<sup>19</sup>

ss(x<sup>th</sup>)<sup>20</sup>

00<sub>(h)</sub>

pp(x<sup>th</sup>)<sup>1</sup>    0000000v<sub>(b)</sub> where v is bit 7 of the number of CPU blocks for the x<sup>th</sup> Module.

pp(x<sup>th</sup>)<sup>2</sup>    0vvvvvvv<sub>(b)</sub> where v is bits 6 through 0 of then number of CPU blocks for the x<sup>th</sup> Module.

rr(x<sup>th</sup>)<sup>1</sup>    0000000v<sub>(b)</sub> where v is bit 7 of the number of RAM blocks for the x<sup>th</sup> Module.

rr(x<sup>th</sup>)<sup>2</sup>    0vvvvvvv<sub>(b)</sub> where v is bits 6 through 0 of then number of RAM blocks for the x<sup>th</sup> Module.

The **Respond Module Table** procedure is sent in response to the **Request Module Table** procedure. This is a reference table for all the FX Modules in the DigiTech Device and is ignored if sent to a DigiTech product.

## REQUEST ALGORITHM LINK TABLE (52h)

SYS\_HEAD, 52<sup>(h)</sup>, F7<sup>(h)</sup>

When a **Request Algorithm Link Table** procedure is received, the DigiTech device will respond with a **Respond Algorithm Link Table** procedure of the current Algorithm in use.

## RESPOND ALGORITHM LINK TABLE (53h)

SYS\_HEAD, 53<sup>(h)</sup>, aa, bb, cc, dd, vv(1<sup>st</sup>), ii(1<sup>st</sup>), oo(1<sup>st</sup>), pp(1<sup>st</sup>)<sup>(1st)</sup>, nn(1<sup>st</sup>)<sup>(1st)</sup>, ...  
pp(1<sup>st</sup>)<sup>(Wth)</sup>, nn(1<sup>st</sup>)<sup>(Wth)</sup>, ... vv(x<sup>th</sup>), ii(x<sup>th</sup>), oo(x<sup>th</sup>), pp(x<sup>th</sup>)<sup>(1st)</sup>, nn(x<sup>th</sup>)<sup>(1st)</sup>, ... pp(x<sup>th</sup>)<sup>(Wth)</sup>,  
nn(x<sup>th</sup>)<sup>(Wth)</sup>, F7<sup>(h)</sup>

- aa Algorithms numbers 1 through 127 (00-7E<sup>(h)</sup>) where 1 through 64 are factory algorithms and 65 through 127 are user algorithms. 7F<sup>(h)</sup> indicates that the device was busy at the time this information was requested and that it needs to be requested again.
- bb Number of FX Modules in algorithm. The first module index is always identified as the S-DISC's inputs and the last module index is always the S-DISC's outputs.
- cc Number of FX Modules in the first S-DISC™ processor including the S-DISC's input and output module indexes.
- dd Data byte reserved for future use.
- vv(x) FX Module's I.D. number for the x<sup>th</sup> Module.
- ii(x) Number of inputs for the x<sup>th</sup> Module (this will always be 00 for the Input Module).
- oo(x) Number of outputs for the x<sup>th</sup> Module (this will always be 00 for the Output Module).
- pp(x)<sup>(w)</sup> The position of the Module to which the x<sup>th</sup> Module's w<sup>th</sup> input is connected (00 indicates that it is not connected).
- nn(x)<sup>(w)</sup> The number of the Module's output to which the x<sup>th</sup> Module's w<sup>th</sup> input is connected (00 indicates that it is not connected).

The **Respond Module Link Table** procedure is sent in response to the **Request Module Link Table** procedure. This is a reference table for an algorithm's module linking and is ignored if sent to a DigiTech product.

## REQUEST PARAMETER VALUE (17h)

**NOTE:** This procedure is only implemented in the GSP-2101 Version 3.xx software.

SYS\_HEAD, 17<sup>(h)</sup>, mm<sup>1</sup>, mm<sup>2</sup>, pp<sup>1</sup>, pp<sup>2</sup>, F7<sup>(h)</sup>

mm The effect module's position in the Algorithm

pp The Parameter's position in the effect module.

When a **Request Parameter Value** procedure is received, the device will respond with a **Receive Parameter Value** procedure.

## RECEIVE PARAMETER VALUE (18h)

**NOTE:** This procedure is only implemented in the GSP-2101 Version 3.xx software.

SYS\_HEAD, 18<sup>(h)</sup>, mm<sup>1</sup>, mm<sup>2</sup>, pp<sup>1</sup>, pp<sup>2</sup>, vv<sup>1</sup>, vv<sup>2</sup>, F7<sup>(h)</sup>

mm The effect module's position in the Algorithm

pp The Parameter's position in the effect module.

vv The Parameter's value

When a **Receive Parameter Value** procedure is received, the device's valid Parameter value is updated with vv.

## RECEIVE KEY SCAN CODE (54h)

SYS\_HEAD, 54<sup>(h)</sup>, cc, F7<sup>(h)</sup>

cc Code for the Key Scan

The **Receive Key Scan Code** procedure is used to emulate a button press on the front panel of a DigiTech Device. See the Key Scan Code Maps (page 18) for a list of Key Scan codes.

## RECEIVE HOLD KEY SCAN CODE (55h)

SYS\_HEAD, 55<sub>(h)</sub>, cc, xx, F7<sub>(h)</sub>

cc Code for the Key Scan.

xx Time the key is pressed expressed in 0.1 second increments.

The **Receive Hold Key Scan Code** procedure is used to emulate a button press and hold on the front panel of a DigiTech Device. The device will respond with a **Receive Key Accepted** procedure. See the Key Scan Code Maps (page 18) for a list of Key Scan codes.

## RESPOND KEY ACCEPTED (56h)

SYS\_HEAD, 56<sub>(h)</sub>, cc, F7<sub>(h)</sub>

cc Code echoed for the Key Scan received. (00<sub>(h)</sub> if key is invalid)

The **Respond Key Accepted** procedure is sent from the DigiTech device in response and acceptance of a **Receive Scan Code Key** or **Receive Hold Scan Code Key** procedure but is ignored if received by a DigiTech device.

## REQUEST PARAMETER INFO (58h)

SYS\_HEAD, 58<sub>(h)</sub>, F7<sub>(h)</sub>

When a **Request Parameter Info** procedure is received, the DigiTech device will respond with a **Receive Parameter Info** procedure of the current parameter.

## RESPOND PARAMETER INFO (59h)

SYS\_HEAD, 59<sub>(h)</sub>, ss<sup>1</sup>, ss<sup>2</sup>, vv<sup>1</sup>, vv<sup>2</sup>, mm<sup>1</sup>, mm<sup>2</sup>, nn<sup>1st</sup>,... nn<sup>nth</sup>, 00<sub>(h)</sub>, dd<sup>1st</sup>,... dd<sup>nth</sup>, 00<sub>(h)</sub>, pp<sup>1st</sup>,... pp<sup>nth</sup>, 00<sub>(h)</sub>, F7<sub>(h)</sub>

ss<sup>1</sup>     0000000v<sub>(b)</sub> where v is bit 7 of the screen status (usually 00).  
 ss<sup>2</sup>     0vvvvvvv<sub>(b)</sub> where v is bits 6 through 0 of the screen status where:  
           00 = Parameter Screen        01 = Program Screen  
           02 = Algorithm Screen       03 = Algorithm Edit and other menu Screens  
 vv<sup>1</sup>     0000000v<sub>(b)</sub> where v is bit 7 of the current displayed value  
 vv<sup>2</sup>     0vvvvvvv<sub>(b)</sub> where v is bits 6 through 0 of the current displayed value  
 mm<sup>1</sup>     0000000v<sub>(b)</sub> where v is bit 7 of the maximum value for the current displayed parameter  
 mm<sup>2</sup>     0vvvvvvv<sub>(b)</sub> where v is bits 6 through 0 of the maximum value for the current displayed parameter  
 nn<sup>1</sup>  
 ...       Each ASCII byte for the parameter's long name string.  
 nn<sup>nth</sup>  
 00<sub>(h)</sub>  
 dd<sup>1</sup>  
 ...       Each ASCII byte for the parameter's short name string.  
 dd<sup>nth</sup>  
 00<sub>(h)</sub>  
 pp<sup>1</sup>  
 ...       Each ASCII byte for the parameter's current value name string.  
 pp<sup>nth</sup>  
 00<sub>(h)</sub>

The **Respond Parameter Info** procedure is sent in response to a **Request Parameter Info** procedure. This is ignored when received by a DigiTech device.



## RECEIVE PARAMETER VALUE (63h)

SYS\_HEAD, 63<sup>(h)</sup>, vv<sup>1</sup>, vv<sup>2</sup>, F7<sup>(h)</sup>

vv<sup>1</sup> 0000000v<sup>(b)</sup> where v is bit 7 of the new value to be placed

vv<sup>2</sup> 0vvvvvvv<sup>(b)</sup> where v is bits 6 through 0 of the new value to be placed

The **Receive Parameter Value** procedure allows a new parameter value to be placed at the current parameter selected. This procedure is similar to changing a parameter using a continuous controller link in that the STORE LED status does not change.

## RETURN TO PROGRAM SCREEN (60h)

SYS\_HEAD, 60<sup>(h)</sup>, F7<sup>(h)</sup>

When a **Return To Program Screen** procedure is received, the DigiTech device will return to the program title screen if it is on an algorithm or parameter screen.

## REQUEST ERROR/STATUS FLAG (62h)

SYS\_HEAD, 62<sup>(h)</sup>, F7<sup>(h)</sup>

When a **Request Error/Status Flag** procedure is received, the DigiTech device will respond with a **Receive Error/Status Flag** procedure.

## RESPOND ERROR/STATUS FLAG (61h)

SYS\_HEAD, 61<sub>(h)</sub>, ss<sup>1</sup>, ss<sup>2</sup>, ee<sup>1</sup>, ee<sup>2</sup>, ff<sup>1</sup>, ff<sup>2</sup>, mm<sup>1</sup>, m<sup>2</sup>, mm<sup>3</sup>, mm<sup>4</sup>, F7<sub>(h)</sub>

ss<sup>1</sup> 0000000v<sub>(b)</sub> where v is bit 7 of the System error ID (B1)  
 ss<sup>2</sup> 0vvvvvvv<sub>(b)</sub> where v is bits 6 through 0 of the System error ID (B1)  
 ee<sup>1</sup> 0000000v<sub>(b)</sub> where v is bit 7 of the error message (B2)  
 ee<sup>2</sup> 0vvvvvvv<sub>(b)</sub> where v is bits 6 through 0 of the error message (B2)  
 ff<sup>1</sup> 0000000v<sub>(b)</sub> where v is bit 7 of the memory error flag (internal use)  
 ff<sup>2</sup> 0vvvvvvv<sub>(b)</sub> where v is bits 6 through 0 of the memory error flag (internal use)  
 mm<sup>1</sup> 0000000v<sub>(b)</sub> where v is bit 15 of the MIDI error flag (internal use)  
 mm<sup>2</sup> 0vvvvvvv<sub>(b)</sub> where v is bits 14 through 8 of the MIDI error flag (internal use)  
 mm<sup>1</sup> 0000000v<sub>(b)</sub> where v is bit 7 of the MIDI error flag (internal use)  
 mm<sup>2</sup> 0vvvvvvv<sub>(b)</sub> where v is bits 6 through 0 of the MIDI error flag (internal use)

### Messages:

B1 = 0 - SysEx requested Error/Status (no error)

B1 = 1 - In Algorithm Edit functions:

B2 = 1 - 1 second message is being displayed, the user may abort the message

B2 = 2 - Last selected FX Module would not fit

B2 = 3 - Waiting for key release

B2 = 4 - .5 second message, may be aborted

B2 = 5 - Waiting for key press

B1 = 2 - In Algorithm Link functions

B2 = 1 - On display: "Perform AutoLink? Yes No"

B2 = 2 - On display: "Break all links before AutoLink?"

B1 = 3 - Algorithm Store functions

B2 = 1 - On display: "Alg has been changed do you want to Link? Yes No"

B2 = 2 - On display: "Alg name is already in use." Note that this display is a 1 second message but if it is aborted then the key is not aborted, also the Algorithm naming menu is entered

B2 = 3 - On display: "Replace User Alg Yes No Cancel"

B2 = 4 - The store failed, a soft reset is needed

B2 = 5 - Waiting for a key press to abort an error message

B2 = 6 - Changed algorithm name, returning to editing the Algorithm

B2 = 7 - Algorithm has been saved successfully and the current program is being re-loaded

- B1 = 4 - Alg store failed and unit is returning to the editing functions due to an error
  - B2 = 1 - No room for algorithm
  - B2 = 2 - FX Module will not fit into the Algorithm
  - B2 = 3 - Out of CPU or RAM blocks for S-DISC #1
  - B2 = 4 - Out of CPU or RAM blocks for S-DISC #2
  - B2 = 5 - Too many parameters for a single Algorithm
  - B2 = 6 - Out of Algorithm memory
- B1 = 5 - Sending/Receiving a single Program Dump
  - B2 = 1 - Program is too large to send
  - B2 = 2 - Received Program failed, bad data
  - B2 = 3 - Receiving Program, waiting for Algorithm
  - B2 = 4 - Received Program failed, out of Algorithm memory
  - B2 = 5 - Received Algorithm failed, out of Algorithm slots
  - B2 = 6 - Received Algorithm failed, out of Program memory
  - B2 = 7 - Received Algorithm failed, waiting for unit to be powered off and then back on
- B1 = 6- Receiving a System Dump
  - B2 = 1 - Wrong Version, ignoring data
  - B2 = 2 - Bad data, resetting unit
- B1 = 7 - Receiving a single Algorithm
  - B2 = 1 - New Algorithm minus 1
- B1 = 8 - Other Status
  - B2 = 1 - Requested Algorithm doesn't exist

The **Respond Error/Status Flag** procedure is sent upon detection of a device's error status or as a response to a **Request Error/Status Flag**.

# Key Scan Codes

These are the codes for the each of the buttons or keys on the different S-DISC™ products. Some products have keys screened with two functions which are indicated by the second lable in parenthesis. Numbers are given as hexadecimal values.

<u>Key Label/ID</u>	<u>TSR-24</u>	<u>GSP-2101</u>	<u>TSR-12</u>	<u>Valve FX</u>	<u>RP-10</u>
Program Up	01 <sup>(h)</sup>	01 <sup>(h)</sup>	01 <sup>(h)</sup>		01 <sup>(h)</sup>
Program Down	02 <sup>(h)</sup>	02 <sup>(h)</sup>	02 <sup>(h)</sup>		02 <sup>(h)</sup>
Compare	03 <sup>(h)</sup>	03 <sup>(h)</sup>	03 <sup>(h)</sup>		03 <sup>(h)</sup>
Store	04 <sup>(h)</sup>	04 <sup>(h)</sup>	04 <sup>(h)</sup>	04 <sup>(h)</sup>	04 <sup>(h)</sup>
Parameter Up	05 <sup>(h)</sup>		05 <sup>(h)</sup>	05 <sup>(h)</sup>	05 <sup>(h)</sup>
Parameter Down	06 <sup>(h)</sup>		06 <sup>(h)</sup>	06 <sup>(h)</sup>	06 <sup>(h)</sup>
Parameter Left	07 <sup>(h)</sup>	07 <sup>(h)</sup>	07 <sup>(h)</sup>	07 <sup>(h)</sup>	07 <sup>(h)</sup>
Parameter Right	08 <sup>(h)</sup>	08 <sup>(h)</sup>	08 <sup>(h)</sup>	08 <sup>(h)</sup>	08 <sup>(h)</sup>
Access 1	11 <sup>(h)</sup>	11 <sup>(h)</sup>			
Access 2	12 <sup>(h)</sup>	12 <sup>(h)</sup>			
Access 3	13 <sup>(h)</sup>	13 <sup>(h)</sup>			
Access 4	14 <sup>(h)</sup>				
Prev FX Module	15 <sup>(h)</sup>	15 <sup>(h)</sup>			
Next FX Module	16 <sup>(h)</sup>	16 <sup>(h)</sup>			
Reverb	21 <sup>(h)</sup>	21 <sup>(h)</sup>	21 <sup>(h)</sup>	21 <sup>(h)</sup>	21 <sup>(h)</sup>
Delay(Smp)	22 <sup>(h)</sup>	22 <sup>(h)</sup>	22 <sup>(h)</sup>	22 <sup>(h)</sup>	22 <sup>(h)</sup>
Chorus (Fla)	23 <sup>(h)</sup>	23 <sup>(h)</sup>	23 <sup>(h)</sup>		
Flange	24 <sup>(h)</sup>				
Pitch	25 <sup>(h)</sup>	25 <sup>(h)</sup>	25 <sup>(h)</sup>	25 <sup>(h)</sup>	25 <sup>(h)</sup>
Sample	26 <sup>(h)</sup>		26 <sup>(h)</sup>		
Mod(1)	27 <sup>(h)</sup>		27 <sup>(h)</sup>	27 <sup>(h)</sup>	27 <sup>(h)</sup>
Mix(2)	28 <sup>(h)</sup>	28 <sup>(h)</sup>	28 <sup>(h)</sup>	28 <sup>(h)</sup>	28 <sup>(h)</sup>
More(3)	29 <sup>(h)</sup>	29 <sup>(h)</sup>	29 <sup>(h)</sup>	29 <sup>(h)</sup>	29 <sup>(h)</sup>
Comp(Dist)		31 <sup>(h)</sup>		31 <sup>(h)</sup>	31 <sup>(h)</sup>
Dist		32 <sup>(h)</sup>			
EQ	33 <sup>(h)</sup>	33 <sup>(h)</sup>	33 <sup>(h)</sup>	33 <sup>(h)</sup>	33 <sup>(h)</sup>
Gate		34 <sup>(h)</sup>			
Add	41 <sup>(h)</sup>	41 <sup>(h)</sup>			
Delete	42 <sup>(h)</sup>	42 <sup>(h)</sup>			
Enter	43 <sup>(h)</sup>	43 <sup>(h)</sup>			
Exit				44 <sup>(h)</sup>	
Test	45 <sup>(h)</sup>	45 <sup>(h)</sup>			
Link	46 <sup>(h)</sup>	46 <sup>(h)</sup>			
Tuner		47 <sup>(h)</sup>			
Save	48 <sup>(h)</sup>				
Name	51 <sup>(h)</sup>	51 <sup>(h)</sup>	51 <sup>(h)</sup>		51 <sup>(h)</sup>
MIDI(Utility)	52 <sup>(h)</sup>	52 <sup>(h)</sup>	52 <sup>(h)</sup>		52 <sup>(h)</sup>
Utility (MIDI)	53 <sup>(h)</sup>	53 <sup>(h)</sup>		53 <sup>(h)</sup>	53 <sup>(h)</sup>
Bypass	54 <sup>(h)</sup>	54 <sup>(h)</sup>	54 <sup>(h)</sup>	54 <sup>(h)</sup>	54 <sup>(h)</sup>

# Misc Information

## RAM ADDRESS MAPS

The System Configuration data pack is defined as follows:

<u>data type</u>	<u>data byte size</u>	<u>Description</u>
conf_addr	2	Current RAM address echo
reserved_1	2	
reserved_2	2	
reserved_3	2	
reserved_4	2	
reserved_5	2	
reserved_6	2	
fact_prg_end	2	byte count of current factory algorithms
dummy_1	2	
user_prg_end	2	byte count of current user algorithms
dummy_2	2	
version_number	2	Current software version (displayed in Hex)
fact_prg_tbl	4	physical address of factory program table
fact_prg_indexs	4	physical address of factory program index
fact_alg_tbl	4	physical address of factory algorithm table
fact_alg_indexs	4	physical address of factory algorithm index
user_prg_tbl	4	physical address of user program table
user_prg_indexs	4	physical address of user program index
user_alg_tbl	4	physical address of user algorithm table
user_alg_indexs	4	physical address of user algorithm index

56 total data bytes

note: each data byte is represented with two bytes through MIDI

## TSR24 SysEx Dump of Factory Program Number 1

```
<<SysEx Header>>
F0 00 00 10 00 40

<<Procedure and Program#>>
42 00 00

<<Algorithm#>>
00 40

<<16 Character Program Name - dynamic>>
00 42 00 69 00 67 00 20 00 26 00 20 00 42 00 72 00 69 00 74 00 65 00 20 00 52 00 65 00 76

<<Carriage Return>>
00 0D

<<16 Character Alg Name for display - dynamic>>
00 47 00 69 00 67 00 61 00 56 00 65 00 72 00 62 00 20 00 31 00 7E 00 34 00 20 00 20 00 20 01 0A

<<NULL to indicate end of character strings>>
00 00

<<Number of Linked Local CCs>>
00 00

<<CC#, Parameter#, MinValue (4 bytes), MaxValue (4 bytes) for each CC Link (CC#129=no link)>>

<<Parameter# Asssignments for Access buttons 1,2,3, and 4 (255=no assignment)>>
00 09    00 11    00 12    00 13

<<Total count for Parameters>>
00 1C

<<Values for Parameters 00, 01, 02, 03....>>
00 01    00 00    00 28    00 24    00 01    00 23    00 00    00 00    00 05    00 28
00 28    00 32    00 09    00 1A    00 07    00 1D    00 1D    00 04    00 17    00 17
00 32    00 3C    00 55    00 1E    00 32    00 00    00 50    00 01

<<Z-Reg init count for S-Disc 1>>
00 04

<<Z-Reg init strings for S-Disc 1>>
01 45    00 00    00 00    01 70
01 46    00 00    00 00    00 00
01 47    00 00    00 00    00 00
01 48    00 00    00 00    00 10

<<Z-Reg init count for S-Disc 2>>
00 00

<<Z-Reg init strings for S-Disc 2>>

<<Software Version number (ex: 2.0)>>
00 00    00 02

<<Seamless Program Change Hold and Ramp Times>>
00 0A    00 14

<SysEx End>
F7
```

# Misc Information

## GSP-2101 SysEx Dump of Factory Program Number 1

```
<<SysEx Header>>
F0 00 00 10 00 41

<<Procedure and Program#>>
42 00 00

<<Algorithm#>>
00 5C

<<20 Character Program Name for line 1 of display - dynamic>>
00 44 00 72 00 79 00 20 00 53 00 61 00 74 00 75 00 72 00 61
00 74 00 65 00 64 00 20 00 54 00 75 00 62 00 65

<<Carriage Return>>
00 0D

<<20 Character Alg Name for line 2 of display - dynamic>>
00 4E 00 6F 00 20 00 44 00 69 00 67 00 69 00 74

<<Carriage Returns>>
00 0D 00 0D

<<20 Character Quick Key Parameter Names for line 4 of display - dynamic>>
00 47 00 61 00 69 00 6E 00 31 00 20 00 15 00 47 00 61 00 69
00 6E 00 32 00 20 00 15 00 4D 00 56 00 6F 00 6C 00 20 00 20

<<NULL to indicate end of character strings>>
00 00

<<Number of Linked Local CCs>>
00 02

<<CC#, Parameter#, Max Value (4 Bytes), Min Value (4 Bytes) for each CC Link (CC#129=Blank
Link)>>
00 04 00 07 00 1F 00 00 00 00 00 00
00 04 00 08 00 1F 00 00 00 00 00 00

<<Parameter# Asssignment for Access buttons 1,2,3, and 4 (4 is ignored and 255=no assignment)>>
00 07 00 08 00 10 01 7F

<<Total count for Parameters>>
00 1C

<<Values for Parameters 00, 01, 02, 03....>>
00 00 00 06 00 03 00 02 00 01 00 00 00 00 00 16 00 16 00 09
00 06 00 06 00 02 00 07 00 06 00 08 00 14 00 0D 00 02 00 01
00 00 00 06 00 0A 00 00 00 1B 00 23 00 32 00 00

<<Z-Reg init count for S-Disc 1>>
00 07

<<Z-Reg init strings for S-Disc 1>>
01 40 00 00 00 00 01 70 01 41 00 00 00 00 00 00
01 42 00 00 00 00 00 00 01 43 00 00 00 00 00 10
01 46 00 00 00 00 00 01 01 47 00 00 00 00 00 01
01 48 00 00 00 00 00 00

<<Z-Reg init count for S-Disc 2>>
00 00

<<Z-Reg init strings for S-Disc 2>>

<<Seamless Program Change Hold and Ramp Times>>
01 14 00 01

<SysEx End>
F7
```

## VALVE FX SysEx Dump of Factory Program Number 1

```
<<SysEx Header>>
F0 00 00 10 00 45

<<Procedure and Program#>>
42 00 00

<<Algorithm#>>
00 40

<<15 Character Program Name for line 1 of display - dynamic>>
00 20 00 53 00 6F 00 6C 00 6F 00 20 00 4D 00 69 00 6F

<<Carriage Return>>
00 0D

<<16 Character Alg Name for line 2 of display - dynamic>>
00 44 00 43 00 68 00 6F 00 7E 00 32 00 54 00 44 00 6C 00 79 00 7E 00 52 00 65 00 76 00 72 00 62

<<NULL to indicate end of character strings>>
00 00

<<Number of Linked Local CCs>>
00 06

<<Link#, CC#, Parameter#, Max Value (4 Bytes), Min Value (4 Bytes) for each CC Link>>
00 00 00 04 00 0F 00 46 00 00 00 00 00 00
00 01 00 15 00 00 00 01 00 00 00 00 00 00
00 02 00 16 00 03 00 01 00 00 00 00 00 00
00 03 00 19 00 3D 00 01 00 00 00 00 00 00
00 04 00 18 00 2D 00 01 00 00 00 00 00 00
00 05 00 17 00 20 00 01 00 00 00 00 00 00

<<Total count for Parameters>>
00 4E

<<Values for Parameters 00, 01, 02, 03...>>
00 00 00 14 00 04 00 01 00 02 00 16 00 32 00 3E 00 01 00 08
00 19 00 00 00 29 00 33 00 0A 00 45 00 07 00 03 00 18 00 0B
00 03 00 11 00 16 00 02 00 17 00 19 00 03 00 12 00 00 00 03
00 00 00 64 00 00 00 32 00 1E 00 1E 00 2D 00 19 00 01 00 50
00 00 00 05 00 1A 00 64 00 0D 00 01 00 11 00 06 00 00 00 06
00 32 00 10 00 00 00 3C 00 00 00 3C 00 1A 00 64 00 00 00 64
00 1A 00 00 00 21 00 09 00 09 00 1C 00 1A 00 04 00 0E 00 0C
00 64 00 00 00 64 00 1A 00 64 00 00 00 64 00 1A

<<Z-Reg init count for S-Disc 1>>
00 1B

<<Z-Reg init strings for S-Disc 1>>
01 40 00 00 00 00 01 70 01 41 00 00 00 00 00 00
01 43 00 00 00 00 00 10 01 45 00 00 00 00 00 70
01 46 00 00 00 00 00 00 01 47 00 00 00 00 00 00
01 4A 01 7F 00 10 00 00 01 4B 01 4C 01 7C 01 7F
01 4C 01 4C 01 7C 01 7F 01 4D 01 78 01 43 00 09
01 4E 00 01 00 00 00 00 01 4F 00 14 01 2E 00 0F
01 50 00 14 01 2E 00 0F 01 51 00 00 00 00 00 00
01 52 00 0A 01 57 00 07 01 53 01 76 00 28 01 78
01 54 00 00 00 00 00 10 01 55 00 00 00 00 00 70
01 56 00 00 00 00 00 00 01 57 01 7F 00 0F 00 00
01 58 00 16 00 00 00 00 01 59 00 33 00 33 01 73
01 5A 00 33 00 33 01 73 01 58 00 11 00 00 00 00
01 4D 00 00 01 40 00 12 01 4B 00 3E 01 7E 01 7F
01 4C 00 06 01 7F 01 7F

<SysEx End>
F7
```



# Misc Information

## TSR-12 SysEx Dump of Factory Program Number 1

```
<<SysEx Header>>
F0 00 00 10 00 42

<<Procedure and Program#>>
42 00 00

<<Algorithm#>>
00 49

<<16 Character Program Name for line 1 of display - dynamic>>
00 42 00 69 00 67 00 20 00 26 00 20 00 42 00 72 00 69 00 67 00 68 00 74 00 20 00 52 00 65 00 76

<<Carriage Return>>
00 0D

<<16 Character Alg Name for line 2 of display - dynamic>>
00 50 00 45 00 51 00 35 00 7E 00 42 00 69 00 67 00 20 00 52 00 65 00 76 00 65 00 72 00 62

<<NULL to indicate end of character strings>>
00 00

<<Number of Linked Local CCs>>
00 00

<<Link#, CC#, Parameter#, Max Value (4 Bytes), Min Value (4 Bytes) for each CC Link>>

<<Total count for Parameters>>
00 24

<<Values for Parameters 00, 01, 02, 03....>>
00 32 00 01 00 09 00 32 00 00 00 1A 00 33 00 00 00 05 00 0F
00 0A 00 03 00 0F 00 0F 00 03 00 0F 00 14 00 03 00 0F 00 16 00 12
00 46 00 01 00 00 00 03 00 00 00 09 00 3C 00 00 00 09 00 09 00 1D
00 1D 00 09 00 10 00 2D

<<Z-Reg init count for S-Disc 1>>
00 07

<<Z-Reg init strings for S-Disc 1>>
01 40 00 00 00 00 01 70 01 41 00 00 00 00 00 00
00 43 00 00 00 00 00 10 01 45 00 00 00 00 01 70
01 46 00 00 00 00 00 00 01 47 00 00 00 00 00 00
01 48 00 00 00 00 00 10
```

<SysEx End>  
F7

## Legend II SysEx Dump of Factory Program Number 1

```
<<SysEx Header>>
F0 00 00 10 00 44

<<Procedure and Program#>>
42 00 00

<<Algorithm#>>
00 60

<<15 Character Program Name for line 1 of display - dynamic>>
00 47 00 72 00 75 00 6E 00 63 00 68 00 79

<<Carriage Return>>
00 0D

<<16 Character Alg Name for line 2 of display - dynamic>>
00 4E 00 6F 00 20 00 44 00 69 00 67 00 69 00 79 00 61 00 6C

<<NULL to indicate end of character strings>>
00 00

<<Number of Linked Local CCs>>
00 03

<<Link #, CC#, Parameter#, Max Value (4 Bytes), Min Value (4 Bytes) for each CC Link>>
00 00 00 04 00 0F 00 64 00 00 00 00 00 00
00 01 00 15 00 00 00 01 00 00 00 00 00 00
00 02 00 16 00 03 00 01 00 00 00 00 00 00

<<These Bytes only appear in Program 1 (In house testing)>>
00 57

<<Total count for Parameters>>
00 20

<<Values for Parameters 00, 01, 02, 03...>>
00 00 00 14 00 01 00 01 00 00 00 16 00 33 00 2D 00 01 00 08
00 19 00 00 00 29 00 33 00 0A 00 64 00 07 00 03 00 1E 00 0E
00 05 00 0F 00 13 00 04 00 0F 00 1A 00 03 00 1E 00 00 00 03
00 00 00 00

<<Z-Reg init count for S-Disc 1>>
00 06

<<Z-Reg init strings for S-Disc 1>>
01 40 00 00 00 00 01 70
01 41 00 00 00 00 00 00
01 43 00 00 00 00 00 10
01 45 00 00 00 00 01 70
01 46 00 00 00 00 00 00
01 47 00 00 00 00 00 00

<SysEx End>
F7
```