

# I-CUBE X

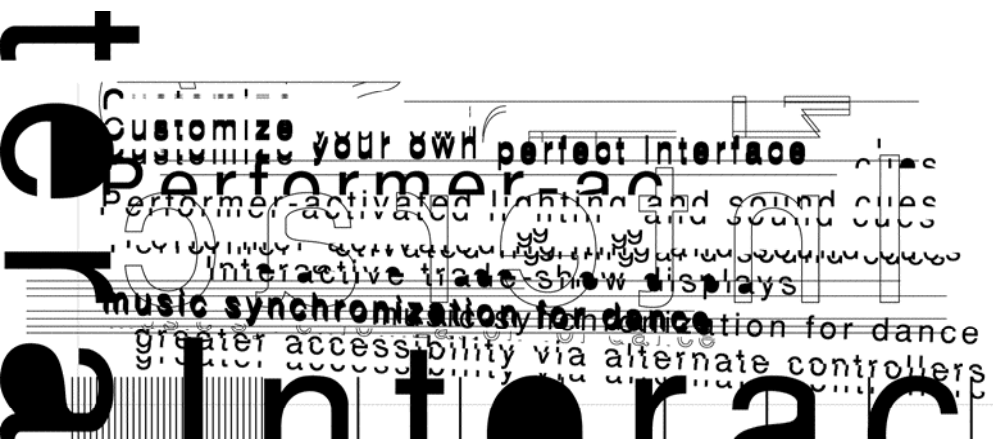


## MIDI implementation v3.5 I-CubeX Reference

*I-CubeX: The ultimate MIDI controller!*

Infusion Systems Ltd.  
P.O.Box 16178  
North Vancouver, BC  
Canada V7J 3S9  
Tel: (604) 983-3640  
Fax: (604) 983-3612  
Email: [info@infusionsystems.com](mailto:info@infusionsystems.com)  
<http://www.infusionsystems.com>

2/2/02  
© 1997-2002 Infusion Systems



# I-CubeX Digitizer firmware v3.5 MIDI implementation

**Introduction:** This document describes the I-CubeX Digitizer firmware version 3.5 MIDI implementation. The firmware version can be obtained by sending a RESET command to the Digitizer.

This specification is divided into general, stand-alone mode and host mode sections addressing programming commands as understood by the Digitizer and transmitted messages as sent by the Digitizer.

Both stand-alone and host mode are supported on Digitizer units equipped with firmware version 1.00 or greater. Units equipped with firmware versions beginning with “DCC” written on the ROM support only host mode. Note that this document does not cover other versions than firmware version 3.5.

Infusion Systems' I-CubeX Digitizer MIDI implementation uses the following data format for all system exclusive messages:

BYTE	Description
240 (F0h)	System Exclusive Status
125 (7Dh)	Manufacturer ID
{DEV}	Device ID
{CMD}	Command or Message ID
[BODY]	Main Data
247 (F7h)	End of System Exclusive

Note that all numbers in hexadecimal notation are ended with the letter h.

**Manufacturer ID:** The manufacturer identifies the manufacturer of a MIDI device. Infusion Systems uses the number 125 (7Dh). Infusion Systems does not own the manufacturer ID 125. This ID is usually reserved for general educational and/or research equipment. Please ensure that you do not have any devices with manufacturer ID 125 in your MIDI chain.

**Device ID:** This ID identifies a specific Digitizer in setups with multiple Digitizers in one MIDI chain. This setting should remain 0 for use of the Digitizer in stand-alone mode. Host mode allows for device IDs other than 0.

**Command and Message IDs and [Main Data]:** The Command and Message IDs and [Main Data] form the protocol for communication with the Digitizer. The protocol consists of commands that are sent to the Digitizer and messages that are sent by the Digitizer. The Command and Message ID description and [Main Data] formats are described starting on the next page.

**General Command and Message IDs and [Main Data]:** This section describes the messages and commands for resting the Digitizer, setting the mode of operation (stand-alone versus host mode - see below for more information), setting the device ID and setting MIDI thru operation. These commands are executed both in stand-alone and host mode. This section also describes system status messages.

The I-CubeX Digitizer uses the following general commands and messages.

## Programming commands

**MIDI System realtime commands:** The I-CubeX Digitizer can be reset as detailed under “Command ID : RESET (34)” using the MIDI system reset command (255 or FFh). A RESET ACK (35) will be sent after completion of the reset command. The MIDI system reset command can be sent anytime during any message. Any other MIDI system realtime messages that are received in the Digitizer MIDI input are ignored and immediately sent out the MIDI output.

**Command ID : RESET (34):** The RESET command resets the Digitizer, i.e. all internal circuitry and system variables stored in volatile memory are re-initialized. Non-volatile memory remains unchanged. When received in host mode, the RESET command turns off as well as un-mutes (including the muting as a result of the SOLO command) all sensor inputs, sets the resolution of all sensor inputs to lo-res (7-bit), sets the sampling interval to 100 ms and turns off all actuator outputs. When received in stand alone mode, the RESET command retrieves the settings stored in non-volatile memory for the activation of all sensor inputs and for the sampling interval and turns off all actuator outputs, sets the resolution to 7 bit as well as unmutes all sensor inputs (including the muting as a result of the SOLO command). The RESET command does not change any stand-alone mode settings stored in non-volatile memory but reloads them in volatile memory for operational use. Upon completion of the command the RESET ACK message is sent out by the Digitizer once or twice depending on the mode of operation (see Message ID : RESET ACK (35)).

There is no [BODY] associated with this command.

Example :

240 (F0h), 125 (7Dh), 0 {DEV}, 34 {RESET}, 247 (F7h)

**Command ID : SET MODE (90):** This command sets the Digitizer operation modes. The Digitizer can function in host mode (e.g for use with Max software), or stand-alone mode (e.g. for use directly with MIDI equipment and without use of a computer after the Digitizer has been configured). When switching the operation mode, all settings stored in non-volatile memory remain unchanged but all settings stored in volatile memory are re-initialized. When switching to host mode the SET MODE command turns off as well as un-mutes (including the muting as a result of the SOLO command) all sensor inputs, sets the resolution of all sensor inputs to lo-res (7-bit), sets the sampling interval to 100 ms and turns off all actuator outputs. When switching to stand-alone mode the SET MODE command retrieves the settings stored in non-volatile memory for the activation of all sensor inputs and for the sampling interval and turns off all actuator outputs, sets the resolution to 7 bit as well as unmutes all sensor inputs (including the muting as a result of the SOLO command). Upon completion of the command the MODE message (see Message ID : MODE (91)) is sent out by the Digitizer.

The [BODY] of the SET MODE command is as follows:

0000000x:      x = 0; host mode  
                  x = 1; stand-alone mode

**Command ID : DUMP MODE (91):** This command request the Digitizer to send out its operation mode. The Digitizer can function in host mode (for use with Max for Macintosh), or stand-alone mode (otherwise). Upon completion of the command the MODE message (see Message ID : MODE (91)) is sent out by the Digitizer.

There is no [BODY] for the DUMP MODE command.

**Command ID : SET ID (92):** The SET ID command sets the system exclusive device ID of the Digitizer, enabling use of upto 128 Digitizers in one MIDI chain. Upon completion of the command the same command is sent out by the Digitizer. The SET ID command can be sent with any device ID in the header of the system exclusive message, ie. the device ID will be set to the new value regardless of the current device ID of the

Digitizer receiving the SET ID command.

The [BODY] of the SET ID command:

0xxxxxxx:      xxxxxxx = [0..127]; new system  
                  exclusive device ID

**Command ID : MIDI THRU (93):** This command sets the software MIDI thru operation of the Digitizer. Note that other MIDI devices with a MIDI thru connector implement MIDI thru in hardware (involving much less delay). When MIDI thru is on, all MIDI messages that are not applicable to the Digitizer are sent out the MIDI output. Applicable messages are channel voice note-on messages for the MIDI channel the Digitizer is set to when operating in stand-alone mode and system exclusive messages with the manufacturer ID used by Infusion Systems as well as the Device ID used by the Digitizer. Upon completion of the command the same command is sent out by the Digitizer.

The [BODY] of the MIDI THRU command is:

0xxxxxxx:      xxxxxxx = 0; MIDI thru off  
                  xxxxxxx > 0; MIDI thru on

## Transmitted messages

**Message ID : MODE (91):** This message contains the Digitizer operation mode. The Digitizer can function in host mode (for use with Max for Macintosh), or stand-alone mode (otherwise).

The [BODY] of the MODE message is as follows:

0000000x:      x = 0; host mode  
                  x = 1; stand-alone mode

**Message ID : RESET ACK (35):** The RESET ACK message provides acknowledgement that the Digitizer has been reset. A reset is performed when the Digitizer is powered up, or when a RESET command is sent to it (see Programming Commands : RESET (34)). If the Digitizer is set to host mode and a RESET command is sent, the Digitizer will send out the RESET ACK message once (visible as one blink of the MIDI output LED). If in stand-alone mode the RESET ACK message will be sent out twice with a 200 ms interval in between the two messages (visible as two blinks of the MIDI output LED).

The [BODY] of the reset message :

0xxxxxxx:      xxxxxxx = [0..127]; firmware version  
                  number \* 10

Example :

```
240 (F0h), 125 (7Dh), 0 {DEV}, 35 {RESET ACK}, 35
{xxxxxxx}, 247 (F7h)
```

The firmware version for the Digitizer is 3.5 = 35 / 10.

**Message ID : STATUS (37):** The STATUS message indicates an error in the system. The [BODY] of the STATUS message indicates the type of problem encountered.

The [BODY] of the reset message :

```
0xxxxxxx:      xxxxxxx = 92; MIDI protocol error (a DATA
                byte of the message is > 127)
                xxxxxxx = 94; MIDI byte has been
                scrambled
                xxxxxxx = 95; receive buffer is full, data
                may be lost (too much MIDI
                data is being sent to the
                Digitizer too quickly)
```

Example :

```
240 (F0h), 125 (7Dh), 0 {DEV}, 37 {STATUS}, 95 {DATA},
247 (F7h)
```

This message indicates that too much MIDI data is being sent to the Digitizer too quickly.

## Stand-alone mode Command and Message IDs and [Main Data]

This section describes the programming protocol, consisting of commands sent to the Digitizer and messages sent by the Digitizer, for stand-alone mode. This mode allows the Digitizer to be used without the need of a host computer because all settings are stored in non-volatile memory (EEPROM). All processing and mapping happens in the Digitizer itself. This mode allows each sensor input to be mapped to a control change, note-on or pitchbend message. The 8 actuator outputs can also be programmed to respond to MIDI note-on events. See the I-CubeX manual for more information on the use of stand-alone mode.

Note that the Digitizer discards all stand-alone mode commands listed below (except for the INTERVAL command which sets the sample interval and the STREAM command which sets the acti-

vation status of a sensor input) when set to host mode. Host mode commands are executed when in stand-alone mode, but they do not change the settings stored in non-volatile memory (except INTERVAL and STREAM). For host mode commands and their effect on settings stored in volatile memory, see the section "Host mode Command and Message IDs and [Main Data]"

The I-CubeX Digitizer in stand-alone mode uses the following commands and messages to change the settings stored in non-volatile memory.

## Programming commands

**Command ID : EDIT NAME (100):** The EDIT NAME command stores the name (as ASCII encoded characters) of the configuration currently stored in the Digitizer. Upon completion of the command a NAME message is sent out by the Digitizer.

The [BODY] of the EDIT NAME command is:

```
00000001:      1; configuration number
0aaaaaaa:      aaaaaaa = [0..127]; 1st character
0bbbbbbb:      bbbbbbb = [0..127]; 2nd character
0ccccccc:      ccccccc = [0..127]; 3rd character
0ddddddd:      ddddddd = [0..127]; 4th character
0eeeeeee:      eeeeeee = [0..127]; 5th character
0ffffff:       fffffff = [0..127]; 6th character
0ggggggg:      ggggggg = [0..127]; 7th character
0hhhhhhh:      hhhhhhh = [0..127]; 8th character
```

**Command ID : DUMP NAME (101):** The DUMP NAME command is used to obtain the name of the current configuration of the Digitizer. Upon completion of the command a NAME message is sent out by the Digitizer.

The DUMP NAME command has the following [BODY]:

```
00000001:      1; configuration number
```

**Command ID : EDIT CONFIG (102):** Each of the sensor inputs, as well as all actuator outputs of the Digitizer can be edited using the EDIT CONFIG command. Upon completion of the command a CONFIG message is sent out by the Digitizer. To turn on or off a sensor input while the Digitizer is powered, send a STREAM command to the Digitizer. To change the sampling interval use the INTERVAL command. The RESET command does not change any of the settings as stored by the EDIT CONFIG command in non-volatile memory.

To edit a sensor input of the Digitizer the [BODY] of the EDIT CONFIG command consists of the following bytes:

```

00000001: 1; configuration number
000aaaaa: aaaaa = [0..31]; sensor input number
0000bbbb: bbbb = [9, 11, 14]; mapping type
           (9 = note-on, 11 = control-change, 14 = pitch-bend)
0000cccc: cccc = [0..15]; MIDI channel
000000de: d,e are sensor input polarity, on/off bits
           d = 0; inverted polarity (MIDI value increases as sensor
           value decreases)
           d = 1; normal polarity (MIDI value increases as sensor
           value increases)
           e = 0; sensor input off (effective after next power-up)
           e = 1; sensor input on (effective after next power-up)
0ffffff: fffffff = [0..127];
           note number (mapping type 9), controller number (mapping
           type 11), or 0 (mapping type 14)
0ggggggg: ggggggg = [0..127]; sensor input threshold
0hhhhhhh: hhhhhh contains the sensor input ceiling or note-
           on velocity value
hhhhhhh = [0..127]; sensor input ceiling, where the ceiling
           must be greater than the sensor input threshold (mapping
           type 11, 14)
hhhhhhh = [0..127]; note-on velocity (mapping type 9)
0: reserved

```

**Note-on mapping** means that sensor values are represented by the velocity data field of a MIDI channel voice note-on message (velocity values with normal and inverted polarity: [0, {hhhhh-hh}]).

**Control-change mapping** means that sensor values are represented by the control data field of a MIDI channel voice control-change message (control values with normal and inverted polarity: [0..127]).

**Pitch-bend mapping** means that sensor values are represented by the two pitch-bend data bytes of a MIDI channel voice pitch-bend message (pitch-bend values with normal polarity: [8191..16383], with inverted polarity: [8191..0]).

Running status is not implemented in the MIDI mapping of Digitizer firmware v3.5.

In order to program the actuator outputs the EDIT CONFIG command is used with a special value for the 2nd databyte in the [BODY]:

```

00000001: 1; configuration number
01111111: 127; actuator outputs flag
000aaaaa: aaaa = [0..15]; MIDI channel
0000bcde: b,c,d,e are response mode bits for actua
           tor outputs 8 (b) - 5 (e)
           b,c,d,e = 0; trigger mode
           b,c,d,e = 1; toggle mode
0000fghi: f,g,h,i are response mode bits for actuator
           outputs 4 (f) - 1 (i)
           f,g,h,i = 0; trigger mode
           f,g,h,i = 1; toggle mode
0jjjjjj: jjjjjj = [0..120]; base note
0: reserved
0: reserved
0: reserved

```

**Trigger mode** means a note-on message with velocity greater than zero turns actuator output on and a note-on message with velocity zero turns actuator output off.

**Toggle mode** means a note-on message with velocity greater than zero toggles actuator output both on and off (a message with velocity zero has no effect).

**Base note** is the note number which triggers the actuator output 1.

**Command ID : DUMP CONFIG (103):** The DUMP CONFIG command is used to obtain the current settings of the Digitizer. Upon completion of the command a CONFIG message is sent out by the Digitizer.

The Digitizer will send out the configuration of a Digitizer sensor input when using the DUMP CONFIG command with the following [BODY]:

```

00000001: 1; configuration number
000aaaaa: aaaaa = [0..31]; sensor input number

```

The Digitizer will send out the configuration of each Digitizer binary output as well as the sample interval value when using the DUMP CONFIG command with a special value for the 2nd databyte in the [BODY]:

```

00000001: 1; configuration number
01111111: 127; actuator outputs flag

```

**Command ID : STREAM (1):** Each sensor input of the Digitizer can be turned on or off by using the STREAM command. Upon completion of the command the activation status of the sensor input is

stored in non-volatile memory (when in stand alone mode) - in volatile memory when in host mode. The MIDI message to which the sensor value is mapped as set using the EDIT CONFIG command, is repeatedly sent out by the Digitizer at intervals specified by the sample interval. After a RESET command in host mode each sensor input is turned off. After a RESET command in stand alone mode, each sensor input is turned on or off depending on the activation setting stored in non-volatile memory.

The [BODY] of the STREAM command consists of a single 7-bit byte with the following format :

```
0xyyyyyy :    x = 1; on
              x = 0; off
```

yyyyyy = [0..31]; sensor input number, where the first sensor input number = 0, and the last (32nd) sensor input number = 31

Example :

In order to turn the 31st sensor input on, the following message is sent.

```
240 (F0h), 125 (7Dh) , 0 {DEV}, 1 {STREAM}, 94 {x = 1,
yyyyyy = 30}, 247 (F7h)
```

In order to turn the same sensor input off, the following message is sent

```
240 (F0h), 125 (7Dh) , 0 {DEV}, 1 {STREAM}, 30 {x = 0,
yyyyyy = 30}, 247 (F7h)
```

Command ID : INTERVAL (3): The Digitizer's global sampling interval can be set to speed up or slow down data acquisition. The interval is set in milliseconds with 4ms being the fastest, and 16383ms (about 16 seconds) being the slowest. When the INTERVAL command is sent in stand-alone mode the sample interval will be stored in non-volatile memory - in host mode it will be stored in volatile memory. After a RESET in host mode, the sampling interval is 100ms by default. After a RESET in stand-alone mode, the sampling interval is retrieved from non-volatile memory. Upon completion of the command the same command is sent out by the Digitizer.

The INTERVAL command [BODY] is composed of 2 bytes each containing a 7-bit number. The first byte is the most significant byte, while the second byte is the least significant byte.

The [BODY] of the INTERVAL command :

```
0xxxxxxx:    xxxxxx = [0..127]; sample interval MSB
0yyyyyyy:    yyyyyy = [0..127]; sample interval LSB
```

where sample interval = xxxxxx \* 128 + yyyyyy

Example : In order to set the sampling interval to 1 second (1000 ms), the following command is sent :

```
240 (F0h), 125 (7Dh) , 0 {DEV}, 3 {INTERVAL}, 7
{xxxxxxx}, 104 {yyyyyy}, 247 (F7h)
```

The sample interval is  $7 * 128 + 104 = 1000$

## Transmitted messages

**Message ID : CONFIG (103):** The CONFIG message contains either the configuration of a sensor input or the configuration of an actuator output as well as the sample interval value.

When receiving the configuration of a sensor input, the [BODY] of the CONFIG message is identical to the [BODY] of the EDIT CONFIG command.

When receiving the configuration of the actuator outputs, the [BODY] of the CONFIG message is identical to the [BODY] of the EDIT CONFIG command, except for databytes 7 and 8 which hold the 14 bit sample interval:

```
00000001:    1; configuration number
01111111:    127; actuator outputs flag
0000aaaa:    aaaa = [0..15]; MIDI channel
0000bcde:    b,c,d,e are response mode bits for
              actuator outputs 5 (e) - 8 (b)
              b,c,d,e = 0; trigger mode
              b,c,d,e = 1; toggle mode
0000fghi:    f,g,h,i are response mode bits for
              actuator outputs 1 (i) - 4 (f)
              f,g,h,i = 0; trigger mode
              f,g,h,i = 1; toggle mode
0jjjjjjj:    jjjjjj = [0..120]; base note
0kkkkkkk:    kkkkkk = [0..127]; sample interval MSB
0mmmmmmm:    mmmmmm = [0..127]; sample interval
              LSB
0:           reserved
```

**Trigger mode** means a note-on message with velocity of at least 64 turns actuator output on and a note-on message with velocity 0 turns actuator output off

**Toggle mode** means a note-on message with velocity of at least 64 toggles actuator output both on and off

**Base note** is the note number which triggers the actuator output 1

**Sample interval** = kkkkkkk \* 128 + mmmmmmm

**Message ID : NAME (101):** The NAME message contains the name (as ASCII encoded characters) of the configuration currently stored in the Digitizer.

The [BODY] of the NAME message is identical to the [BODY] of the EDIT NAME command.

## Host mode Command and Message IDs and [Main Data]

This section describes the programming protocol, consisting of commands sent to the Digitizer and messages sent by the Digitizer, for host mode. This mode requires that the Digitizer is connected with a host computer to control and store the Digitizer settings. Host mode allows full access to the raw sensor input values and uses a more compact encoding method of sensor input values compared to stand-alone mode when sampling many inputs simultaneously. All sensor input values are encoded in system exclusive messages. Host mode also allows use of multiple Digitizers in one MIDI chain and multiple iCube / oCube Max objects for each Digitizer (to this end, some of the commands are echoed back to the host computer). See the I-CubeX manual for more information on the use of host mode.

Note that the Digitizer discards all stand-alone mode commands listed below (except for the INTERVAL command which sets the sample interval and the STREAM command which sets the activation status of a sensor input) when set to host mode. Host mode commands are executed when in stand-alone mode, but they do not change the settings stored in non-volatile memory (except INTERVAL and STREAM).

The I-CubeX Digitizer in host mode uses the following commands and messages.

## Programming commands

**Command ID : STREAM (1):** Each sensor input of the Digitizer can be turned on or off by using the STREAM command. Upon completion of the command the activation status of the sensor input is stored in non-volatile memory (when in stand alone mode) - in volatile memory when in host mode. The MIDI message to which the sensor value is mapped as set using the EDIT CONFIG command, is repeatedly sent out by the Digitizer at intervals specified by the sample interval. After a RESET command in host mode each sensor input is turned off. After a RESET command in stand alone mode, each sensor input is turned on or off depending on the activation setting stored in non-volatile memory.

The [BODY] of the STREAM command consists of a single 7-bit byte with the following format :

```
0xyyyyyy :      x = 1; on
                x = 0; off
```

yyyyyy = [0..31]; sensor input number, where the first sensor input number = 0, and the last (32nd) sensor input number = 31

Example :

In order to turn the 31st sensor input on, the following message is sent.

```
240 (F0h), 125 (7Dh) , 0 {DEV}, 1 {STREAM}, 94 {x = 1,
yyyyyy = 30}, 247 (F7h)
```

In order to turn the same sensor input off, the following message is sent

```
240 (F0h), 125 (7Dh) , 0 {DEV}, 1 {STREAM}, 30 {x = 0,
yyyyyy = 30}, 247 (F7h)
```

**Command ID : RES (2):** Each sensor input can either sample sensor values at 12 bit (hi-res) or 7-bit (lo-res) resolution. After a RESET, the resolution for each input is lo-res by default. Upon completion of the command the same command is sent out by the Digitizer.

The RES command's [BODY] is formatted the same as the STREAM command (the Command ID is the only difference):

0xyyyyyy:        x = 1; hi-res (12-bit mode)  
                   x = 0; lo-res (7-bit mode)

yyyyyy = [0..31]; sensor input number, where the first sensor input number = 0, and the last (32nd) sensor input number = 31

Example : In order to turn the 2nd sensor input to hi-res mode, the following message is sent.

240 (F0h), 125 (7Dh) , 0 {DEV}, 2 {RES}, 65 {x = 1, yyyyyy = 1}, 247 (F7h)

In order to turn the same sensor input to lo-res mode, the following message is sent

240 (F0h), 125 (7Dh) , 0 {DEV}, 2 {RES}, 1 {x = 0, yyyyyy = 1}, 247 (F7h)

**Command ID : INTERVAL (3):** The Digitizer's global sampling interval can be set to speed up or slow down data acquisition. The interval is set in milliseconds with 4ms being the fastest, and 16383ms (about 16 seconds) being the slowest. When the INTERVAL command is sent in stand-alone mode the sample interval will be stored in non-volatile memory - in host mode it will be stored in volatile memory. After a RESET in host mode, the sampling interval is 100ms by default. After a reset in stand alone mode the sampling interval is retrieved from non-volatile memory. Upon completion of the command the same command is sent out by the Digitizer.

The INTERVAL command [BODY] is composed of 2 bytes each containing a 7-bit number. The first byte is the most significant byte, while the second byte is the least significant byte.

The [BODY] of the INTERVAL command :

0xxxxxxx:        xxxxxxx = [0..127]; sample interval MSB  
 0yyyyyyy:        yyyyyyy = [0..127]; sample interval LSB

where sample interval = xxxxxxx \* 128 + yyyyyyy

Example : In order to set the sampling interval to 1 second (1000 ms), the following command is sent :

240 (F0h), 125 (7Dh) , 0 {DEV}, 3 {INTERVAL},  
 7 {xxxxxxx}, 104 {yyyyyyy}, 247 (F7h)

The sample interval is  $7 * 128 + 104 = 1000$

**Command ID : SAMPLE (4):** The SAMPLE command provides a snapshot of the data coming out of a single sensor input. In order to use SAMPLE for a sensor input, that sensor input must be turned off (it is redundant to sample a sensor input that is turned on since it is already being sampled continuously). Upon completion of the command the SAMPLE DATA message (see Message ID : SAMPLE DATA (4)) is sent out by the Digitizer.

The [BODY] of the SAMPLE command is composed of one byte, the sensor input number:

000yyyyy:        yyyyy = [0..31]; sensor input number,

Where the first sensor input number = 0, and the last (32nd) sensor input number = 31

Example : In order to sample sensor input 12 (provided it is turned off), the following message is sent :

240 (F0h), 125 (7Dh) , 0 {DEV}, 4 {SAMPLE}, 11 {yyyyy},  
 247 (F7h)

**Command ID : MUTE (32):** The MUTE command stops the sampling of all sensor inputs. After completion of the MUTE command, the Digitizer does not send out any MIDI messages representing sensor values. After a RESET, the Digitizer is un-muted by default. The MUTE command is a toggle. Sending it the first time after power-up mutes all active sensor inputs, sending it a second time un-mutes them, and so forth. The Digitizer does not send out any messages upon completion of this command.

There is no [BODY] associated with this command.

Example :

240 (F0h), 125 (7Dh) , 0 {DEV}, 32 {MUTE}, 247 (F7h)

**Command ID : SET MUTE (50):** The SET MUTE command stops the sampling of all sensor inputs. After completion of the SET MUTE command, the Digitizer does not send out any MIDI messages representing sensor values. After a RESET, the Digitizer is un-muted by default. The Digitizer does not send out any messages upon completion of this command.

There [BODY] of the MUTE command:

0xxxxxxx:        xxxxxxx = 0; all sensor inputs un-muted  
                   xxxxxxx = [1..127]; all sensor inputs muted



Example :

```
240 (F0h), 125 (7Dh) , 0 {DEV}, 50 {SET MUTE}, 1
{xxxxxxx}, 247 (F7h)
```

**Command ID : SOLO (52):** The SOLO command mutes all sensor inputs except one, ie. the Digitizer still samples all active (turned on) sensor inputs but only sends out the MIDI message representing the sensor values coming from the chosen input. After a RESET, all sensor inputs are un-muted by default. Upon completion of the command no message is sent out by the Digitizer.

The [BODY] of the SOLO command:

```
0xxxxxxx:      xxxxxxx = [0..31]; sensor input number
                 that is not muted
                 xxxxxxx = 127; un-mute all sensor inputs
                 (undo effect of SOLO command)
```

**Command ID : OUTPUT (48):** The OUTPUT command can be used to turn any of the 8 available actuator outputs on or off. After a RESET, all outputs are turned off by default. Upon completion of the command the same command is sent out by the Digitizer.

The [BODY] of the OUTPUT command consists of a single 7-bit byte with the following format:

```
0yyyyyyy:      x = 1; on
                 x = 0; off
```

yyyyyy = [0..7]; actuator output number, where the first actuator output number = 0, and the last (8th) actuator output number = 7

Example : In order to turn the 2nd actuator output on, the following message is sent.

```
240 (F0h), 125 (7Dh) , 0 {DEV}, 48 {RES}, 65 {x = 1,
yyyyyy = 1}, 247 (F7h)
```

In order to turn the same actuator output off, the following message is sent

```
240 (F0h), 125 (7Dh) , 0 {DEV}, 48 {RES}, 1 {x = 0,
yyyyyy = 1}, 247 (F7h)
```

## Transmitted messages

Note that in the following listing the commands as echoed back by the Digitizer (to allow multiple Max objects to work with a Digitizer) are not included.

**Message ID : STREAM DATA (0):** The Digitizer sends it's sensor values in the STREAM DATA message.

The [BODY] of the STREAM DATA message contains a list of sensor values from all active (turned on) sensors, in ascending order. Inactive sensors are not included in the list. Both 7-bit and 12-bit sensor values are packed together - 12-bit values are sent using two bytes, where the first byte contains the first 7 bits of the sensor value and the second byte contains the last 5 bits of the sensor value:

```
0yyyyyyy:      (7-bit lo-res mode)
or
0yyyyyyy, 000zzzzz:  (12-bit hi-res mode)
```

yyyyyy = [0..127]; the most significant bits (or a full data byte in lo-res mode)

zzzzz = [0..31]; the 5 least significant bits (used in 12-bit hi-res mode only)

Example :

The Digitizer has sensor input 1 turned on in lo-res mode (7-bit), sensor input 10 turned on in hi-res mode (12-bit), and sensor input 15 turned on in lo-res mode (7-bit). All other sensor inputs are turned off. The Digitizer acquires the value 100 (sensor input 1), 3000 (sensor input 10), and 21 on (sensor input 15). The following message is sent by the Digitizer :

```
240 (F0h), 125 (7Dh) , 0 {DEV}, 0 {STREAM DATA},
100 {yyyyyyy of sensor input 1}, 93 {yyyyyyy of sensor input
10}, 24 {zzzzz of sensor input 10}, 21 {yyyyyyy of sensor
input 15}, 247 (F7h)
```

The sampled value from sensor input 10 is  $93 * 32$  (i.e. 5 bit shift) + 24 = 2976 + 24 = 3000

**Message ID : SAMPLE DATA (4):** The SAMPLE DATA message contains a single snapshot of a sensor connected to a sensor input.

The [BODY] of the SAMPLE DATA message is composed of two or three bytes - the sensor input number, the first data byte, and, if the sensor input is set to 12-bit hi-res mode, the second data byte.

The [BODY] of the SAMPLE DATA message :

```
000xxxxx, 0yyyyyyy:      (7-bit lo-res mode)
or
000xxxxx, 0yyyyyyy, 000zzzzz:  (12-bit hi-res mode)
```

xxxxx = [0..31]; sensor input number, where the first sensor input number = 0, and the last (32nd) sensor input number = 31

yyyyyyy = [0.127]; the most significant bits (or a full data byte in lo-res mode)

zzzzz = [0..31]; the 5 least significant bits (used in 12-bit hi-res mode only)

Example :

When sampling sensor input 12 (provided it is turned off) and the sensor input is set to lo-res mode, the following message is received:

```
240 (F0h), 125 (7Dh) , 0 {DEV}, 4 {SAMPLE DATA}, 11
{xxxxx}, 64 {yyyyyyy}, 247 (F7h)
```

The sampled value from sensor input 12 is 64.

When sampling sensor input 12 (provided it is turned off) and the sensor input is set to hi-res mode, the following message is received:

```
240 (F0h), 125 (7Dh) , 0 {DEV}, 4 {SAMPLE DATA},
11 {xxxxx}, 10 {yyyyyyy}, 10 {zzzzz}, 247 (F7h)
```

The sampled value from sensor input 12 is  $10 * 32$  (i.e. 5 bit shift) + 10 = 330.

## Disclaimer:

### *Waiver of Liability*

Infusion Systems Ltd. offers no warranty, limited or otherwise, on any of its products. Infusion Systems Ltd. advises any user to immediately notify us on the discovery of any major failure in the functionality of any piece of equipment or software purchased from us.

UNDER NO CIRCUMSTANCES WILL INFUSION SYSTEMS LTD. BE LIABLE FOR ANY LOST PROFITS, LOST SAVINGS, ANY INCIDENTAL DAMAGES, OR ANY CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PRODUCT, EVEN IF INFUSION SYSTEMS LTD. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.