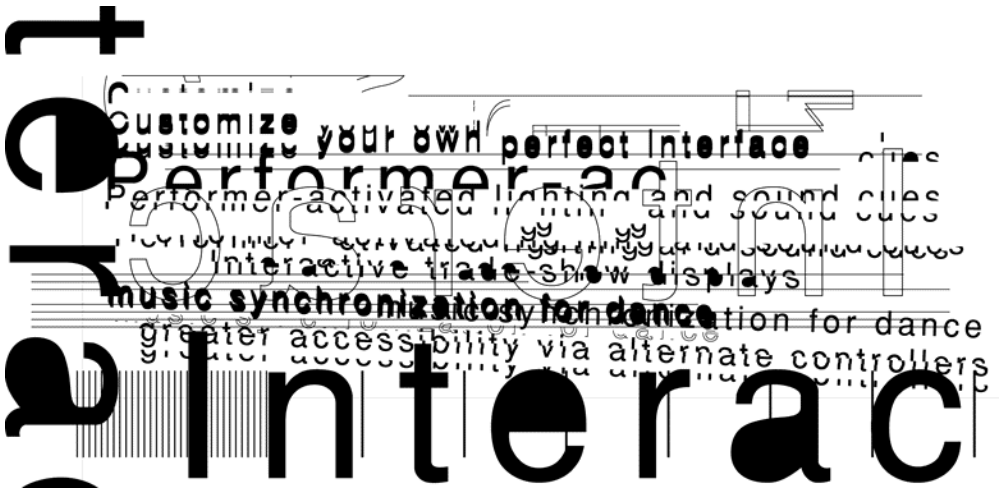
MIDI implementation v4.0

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) 648 8012
Email: info@infusionsystems.com
http://www.infusionsystems.com

10 November 2002



I-CubeX MIDI implementation v4.0

INTRODUCTION

This document describes the I-CubeX MIDI implementation v4.0. The MIDI implementation version number is the same as the firmware version number and can be obtained by sending a DUMP VERSION 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 3.5 (printed on the ROM chip as 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 4.0.

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 resetting the Digitizer, setting the mode of operation (stand-alone or host mode), setting the device ID, setting MIDI thru operation and obtaining version information. These commands have the same effect 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 reset command

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 real-time messages that are received in the Digitizer MIDI input are ignored.

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 by the Digitizer set to host mode, the RESET command sets internal parameters stored in volatile memory to initial values as specified below. When received by the Digitizer set to stand-alone mode, the RESET command retrieves the settings as stored in non-volatile memory and sets other internal parameters as specified below. 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 (when in stand-alone mode). The RESET command can also be initiated by sending a MIDI system reset (255 or FFh). The single-byte MIDI system reset message can be sent anytime during any MIDI message. The RESET command is always executed following a power-up. Upon completion of the RESET command the RESET ACK message is sent out by the Digitizer once or twice depending on the mode of operation of the Digitizer (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)

Initial values for RESET executed with Digitizer set to host mode:

Mode of operation:	0; host mode
MIDI thru activation status:	retrieved from non-volatile memory (see MIDI THRU command)
System exclusive ID:	retrieved from non-volatile memory (see SET ID command)
Sensor sampling interval:	100 ms
Sensor input sampling resolution:	7 bit
Sensor input activation status:	0; off

Sensor input muting:	0; off
Actuator output activation status:	on or off, according to initialisation values stored in non-volatile memory (see EDIT CONFIG command)

Initial values for RESET executed with Digitizer set to stand-alone mode:

Mode of operation:	1; stand-alone mode
MIDI thru activation status:	retrieved from non-volatile memory (see MIDI THRU command)
System exclusive ID:	retrieved from non-volatile memory (see SET ID command)
Sensor input muting:	0; off
Other sensor input settings:	according to settings stored in non-volatile memory (see EDIT CONFIG command)
Actuator output settings:	according to settings stored in non-volatile memory (see EDIT CONFIG command)

Command ID: DUMP VERSION (71)

This command requests the Digitizer to send out its serial number and firmware version number. Upon completion of the command the VERSION message (see Message ID: VERSION (71)) is sent out by the Digitizer.

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

Command ID: SET MODE (90)

This command sets the Digitizer operation modes. The Digitizer can function in host mode, or stand-alone mode.

Stand-alone 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, so it can be directly connected to MIDI output devices such as synthesizers once it has been programmed using a host computer. This mode allows sensor inputs to be mapped to channel voice MIDI messages only (ie. no host mode type system exclusive MIDI messages are used for outputting sensor values). In stand-alone mode the actuator outputs can be set with channel voice MIDI messages as well as with the OUTPUT command. Stand-alone mode commands (including STREAM and INTERVAL), contrary to host mode commands, change the settings stored in non-volatile memory as well as volatile memory. Stand-alone mode commands are executed in both modes of operation, even though in host mode no channel voice MIDI messages are output to reflect any changed settings.

Host mode requires that the Digitizer is connected with a host computer to control and store the Digitizer settings. In host mode unprocessed sensor values are output only as system exclusive MIDI messages (ie. no stand-alone mode type channel voice MIDI messages processed from sensor values are output). Host mode allows use of multiple Digitizers in one MIDI chain. When using Max software, it allows multiple iCube / oCube Max objects for each Digitizer (to this end, some of the commands are echoed back to the host computer). In host mode the actuator outputs cannot be set with channel voice MIDI messages but only with the OUTPUT command. Host mode commands (except STREAM and INTERVAL), contrary to stand-alone mode commands, do not change the settings stored in non-volatile memory but only the settings stored

in volatile memory. Generally, host mode commands have the same effect in either mode of operation – see for details the section on host mode commands.

When switching the operation mode, all settings stored in non-volatile memory remain unchanged but all settings stored in volatile memory are re-initialized to the values stored in non-volatile memory. When switching to host mode the initial values for host mode are used (see the RESET command). When switching to stand-alone mode the initial values for stand-alone mode are used (see the RESET command). The default factory value for the mode of operation is stand-alone mode. Upon completion of the command the MODE message 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 up to 128 Digitizers in one MIDI chain. 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 default factory value is 0. Upon completion of the command the same command is sent out by the Digitizer.

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 (resulting in 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 the system exclusive messages with the manufacturer ID used by Infusion Systems as well as the Device ID used by the Digitizer. In addition, when operating in stand-alone mode, other applicable messages are the channel voice messages for the MIDI channel the Digitizer is set to using the EDIT CONFIG command. The default factory value is off (no MIDI thru). 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: 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).

There is no [BODY] for the RESET ACK message.

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 = 90;	configuration setting error
	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.

Message ID: VERSION (71)

This message contains the Digitizer firmware version number, hardware board version number as well as the serial number.

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

0xxxxxxx:	xxxxxxx = [0..127]; firmware version number * 10
0aaaaaaa:	aaaaaaa = [0..99]; hardware board version number * 10
0bbbbbbb:	bbbbbbb = [0..99]; hardware board version number decimals * 1000
0ccccccc:	ccccccc = [0..99]; 1st serial number digits
0ddddddd:	ddddddd = [0..99]; last serial number digits

Example:

240 (F0h), 125 (7Dh), 0 {DEV}, 71 {VERSION}, 40 {xxxxxxx}, 40 {aaaaaaa}, 10 {bbbbbbb}, 01 {ccccccc}, 23 {ddddddd}, 247 (F7h)
The firmware version for the Digitizer is 40 / 10 = 4.0. The hardware board version number is 40 / 10 + 10 / 1000 = 4.01. The serial number 0123. These numbers should

correspond with numbers printed on the sticker at the bottom of the Digitizer – if they don't correspond the Digitizer may have been upgraded to firmware version 4.0 after its date of purchase.

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

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, so it can be directly connected to MIDI output devices such as synthesizers once it has been programmed using a host computer. This mode allows sensor inputs to be mapped to channel voice MIDI messages only (ie. no host mode type system exclusive MIDI messages are used for outputting sensor values). In stand-alone mode the actuator outputs can be set with channel voice MIDI messages as well as with the OUTPUT command. Stand-alone mode commands (including STREAM and INTERVAL), contrary to host mode commands, change the settings stored in non-volatile memory as well as volatile memory. Stand-alone mode commands are executed in both modes of operation, even though in host mode no channel voice MIDI messages are output to reflect any changed settings.

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: STREAM (1)

Each sensor input of the Digitizer can be turned on or off by using the STREAM command if a signal analysis method has been selected (impulse and/or continuous signal analysis, see EDIT CONFIG command). If no signal analysis method has been selected the STREAM command has no effect. In stand-alone mode the activation status of the sensor input is stored in non-volatile memory as well as volatile memory for immediate use, while in host mode the activation status is only stored in volatile memory. Following the activation of a sensor input the MIDI message to which the sensor value is mapped as set using the EDIT CONFIG command, is sent out by the Digitizer each time the mapped sensor value changes. After a RESET command in stand-alone mode the activation status is retrieved from non-volatile memory - each sensor input is turned on or off depending on whether any of the signal analysis methods is activated (or on the activation setting as used in firmware v3.5). The factory default value of the sensor activation status is off. Upon completion of the command the same message is sent out by the Digitizer.

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 4 ms being the lowest (ie. highest rate), and 16383 ms (about 16 seconds) being the highest (ie. lowest rate). When the INTERVAL command is sent in stand-alone mode the sample interval will be stored in non-volatile memory as well as in volatile memory for immediate use, while in host mode it will be stored in volatile memory only. The factory default setting of the sampling interval is 100 ms. After a RESET in stand-alone mode, the sampling interval is retrieved from non-volatile memory while in host mode it is set to the factory default value of 100 ms. Upon completion of the command the same command is sent out by the Digitizer. If the INTERVAL command is sent with value less than 4 the sampling interval will not be set to a value less than 4 but instead a message identical to the INTERVAL command will be sent out by the Digitizer containing the current sampling interval.

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: 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, factory default = "D" (44h)
0bbbbbb:	bbbbbb = [0..127]; 2nd character, factory default = "i" (69h)
0ccccccc:	ccccccc = [0..127]; 3rd character, factory default = "g" (67h)
0ddddddd:	ddddddd = [0..127]; 4th character, factory default = "i" (69h)
0eeeeeee:	eeeeeee = [0..127]; 5th character, factory default = "t" (74h)
0fffffff:	fffffff = [0..127]; 6th character, factory default = "i" (69h)
0ggggggg:	ggggggg = [0..127]; 7th character, factory default = "z" (7Ah)
0hhhhhhh:	hhhhhhh = [0..127]; 8th character, factory default = "e" (65h)

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: CLEAR CONFIG (105)

The CLEAR CONFIG command sets all settings stored in non-volatile memory to default factory values and re-initializes all internal parameters to the initial values for a Digitizer set to stand-alone mode (see RESET – note that the RESET is not executed). See the EDIT NAME, EDIT CONFIG commands for default values of settings stored in non-volatile memory. Upon completion of the command the same command is sent by the Digitizer. Bug notice: note that the CLEAR CONFIG command activates all inputs because continuous signal analysis is turned on, ie. set to 1.

The [BODY] of the CLEAR CONFIG command:

00000001: 1; configuration number

Factory default values for each sensor input and all actuator outputs are re-instated for:

MIDI thru:	0; off
System exclusive ID:	0
Mode of operation:	1; stand-alone
Configuration name:	“Digitize”
Sensor sampling rate:	100 ms
Sensor input MIDI mapping type:	3; control-change
Sensor input MIDI channel mapping:	0
Sensor input MIDI note number mapping:	sensor input number + 1
Sensor input impulse end notification:	0; off
Sensor input impulse maximum set to constant:	0; off
Sensor input cont. and/or impulse signal differentiation:	0; off
Sensor input continuous signal averaging:	0; off
Sensor input impulse signal analysis:	0; off
Sensor input continuous signal analysis:	1; on
Sensor input threshold:	0
Sensor input ceiling:	127
Sensor input noise gate:	0; off
Sensor input time window:	0; off
Actuator output MIDI mapping type:	1; note-on
Actuator output MIDI channel mapping:	0
Actuator output MIDI base mapping:	64
Actuator output response mode:	1; toggle mode
Actuator output initialisation:	1; on
Actuator output MIDI value threshold:	1

Command ID: EDIT CONFIG (106)

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 the Digitizer is re-initialized with the new settings and a CONFIG message is sent out by the Digitizer. The EDIT CONFIG command can be used to start sending out MIDI messages calculated from sensor values immediately while the Digitizer is powered, but it is also possible to send a STREAM command to the Digitizer. Activating either or both signal analysis methods will enable the sending of MIDI messages. 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.

Editing sensor inputs

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
0bbbcccc:	bbb = [0..6]; MIDI mapping type (0 = note-off, 1 = note-on, 2 = key-pressure, 3 = control-change, 4 = program-change, 5 = after-touch, 6 = pitch-bend) cccc = [0..15]; MIDI channel
0ddddddd:	ddddddd = [0..127]; note number (mapping type 0..2), controller number (mapping type 3), irrelevant for mapping type 4..6
00efghij:	e,f,g,h,i,j determine signal processing settings, see notes for combining e = [0,1]; impulse end notification (0 = off, 1 = on) f = [0,1]; impulse maximum/minimum constant (0 = off, 1 = on) g = [0,1]; continuous and/or impulse signal differentiation (0 = off, 1 = on) h = [0,1]; continuous signal averaging (0 = off, 1 = on) i = [0,1]; impulse signal analysis (0 = off, 1 = on) j = [0,1]; continuous signal analysis (0 = off, 1 = on)
0kkkkkkk:	kkkkkkk = [0..127]; sensor input threshold
0mmmmmmm:	mmmmmmm contains sensor input ceiling or constant value activated with {f} mmmmmmm = [0..127]; sensor input ceiling mmmmmmm = [0..127]; constant value activated with {f}
0nnnnnnn:	nnnnnnn = [0..127]; noise gate
0000pppp:	pppp = [0..15]; time window

Factory default settings for each sensor input are:

Sensor input MIDI mapping type {bbb}:	3; control-change
Sensor input MIDI channel mapping {cccc}:	0
Sensor input MIDI note number mapping {ddddddd}:	sensor input nr {aaaaa} + 1
Sensor input impulse end notification {e}:	0; off
Sensor input impulse maximum set to constant {f}:	0; off
Sensor input cont. and/or impulse signal diff. {g}:	0; off
Sensor input continuous signal averaging {h}:	0; off
Sensor input impulse signal analysis {i}:	0; off
Sensor input continuous signal analysis {j}:	1; on
Sensor input threshold {kkkkkkk}:	0; minimum
Sensor input ceiling {mmmmmmm}:	127; maximum
Sensor input noise gate {nnnnnnn}:	1
Sensor input time window {pppp}:	0; off

Mapping sensor inputs

Impulse signal analysis is useful when the sensor signal can be characterized as an impulse, ie. as either rising from below a minimum value to a maximum value after which it eventually drops below a certain minimum value, or as dropping below a minimum value after which it eventually rises above a certain maximum value. Subsequent impulses may appear asynchronously. Continuous signal analysis is useful when the sensor signal can be characterized as a continuously varying signal without any specific start or end.

The impulse signal analysis settings are the activation status {i} which will also start sampling of the sensor input, the time window setting {pppp} within which {pppp} + 1 sensor values are compared so as to find the highest value (peak search, where the threshold {kkkkkkk} is smaller than {mmmmmmm}) or lowest value (dip search, where the threshold {kkkkkkk} is greater than {mmmmmmm}), the end notification setting {e} which provides a way to determine the duration of the impulse, the maximum/minimum constant setting {f} which provides a way to output each impulse maximum/minimum (peak/dip) as a constant value (ie. the time window {pppp} is not used). If the sensor value rises above {mmmmmmm} ({mmmmmmm} greater than {kkkkkkk}) or drops below {mmmmmmm} ({mmmmmmm} smaller than {kkkkkkk}) before {pppp} + 1 samples have been obtained, the peak/dip search is stopped and is immediately output as peak/dip value. Bug notice: note that activating the maximum/minimum constant setting {f} conflicts with setting the sensor input ceiling value such that when looking for a dip, the ceiling value cannot be used for detecting whether the sensor value drops below it – in other words, a constant value smaller than the threshold and applying a sensor value close to the threshold results in the MIDI output value alternating between the constant value and zero at a rate determined by the sampling interval. Other bug notice: when looking for a dip, due to a rounding-off error, the Digitizer will output the maximum MIDI value (127, or 16383 when using pitch-bend) when the sensor value is exactly equal to the Top value.

The continuous signal analysis settings are the activation status {j} which will also start sampling of the sensor input, the averaging setting {g} which enables calculation of the average of a number of sensor values, the time window setting {pppp} within which {pppp} + 1 sensor values are taken to calculate the average value, the differentiation setting {g} which enables calculation of the difference between the current and last output value.

Continuous signal analysis can be combined with impulse signal analysis and will be useful for sensor signals that have an onset with a local maximum or minimum after which the sensor value continues for some time to stay above the minimum or below the maximum before it eventually drops below the minimum or rises above the maximum.

Impulse signal analysis can also be applied to continuously varying signals such as AC (alternating current) signals and recurring impulses of which only the peak (or dip) value is relevant (deactivate impulse end notification ({e} = 0) to avoid sending out MIDI value zero).

In both impulse and signal analysis the 12-bit sensor value is scaled between the threshold {kkkkkkk} and ceiling {mmmmmmm}, and the resulting value has to increase beyond the noise gate {nnnnnnn} divided by 2 (mapping types 1-5) or beyond the noise gate {nnnnnnn} times 4 (mapping type 6) before being output. After passing the noise gate the value itself or the absolute difference between the current and last calculated value can be output by activating the differentiation setting {g}.

If either impulse or continuous signal analysis is active, the calculated values are represented by the 2nd data field of a MIDI channel voice note-off (header 80h), note-on (header 90h), key-pressure (header A0h) message with note number {ddddddd}, or by the 2nd data field of a control-change (header B0h) message with control number {ddddddd}, or by the 1st data field of a program-change (header C0h) or after-touch (header D0h) message or by the two data fields of a pitch-bend (header E0h) message.

If both impulse and continuous signal analysis are active the sensor value's start and end are characterized using impulse signal analysis and represented as one of the channel voice MIDI messages listed above, while the signal between start and end is characterized using continuous signal analysis and always represented as the key-pressure value of a MIDI channel voice key-pressure (header A0h) message. Note that if averaging is activated, {pppp} will also be used for determining the averaging window, ie. {pppp} + 1 samples will be taken for peak/dip detection and subsequently {pppp} + 1 samples will be taken for averaging.

For example, if impulse signal analysis is activated ({i} = 1), the threshold {kkkkkkk} is smaller or equal than the ceiling {mmmmmmm} and note-on mapping has been selected, then, once the sensor value rises above the threshold {kkkkkkk}, a note-on message with velocity value greater than zero will be sent and no new note-on message with velocity greater than zero will be sent until the sensor value has dropped below {kkkkkkk}. If continuous signal analysis is activated as well ({j} = 1), then, after sending a note-on message, the sensor value will be represented by the key-pressure data field of a MIDI channel voice key-pressure message, until the sensor drops below {mmmmmmm} upon which a note-on message with velocity zero will be sent out.

If either impulse or continuous signal analysis is activated, MIDI messages calculated as defined by the signal analysis method will be sent out. If both are deactivated no MIDI output is generated. MIDI running status is implemented in the MIDI output mapping when using stand-alone mode.

Editing actuator outputs

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
01111110:	127; actuator outputs flag
0aaabbbb:	aaa = [0..3]; MIDI mapping type (0 = note-off, 1 = note-on, 2 = key-pressure, 3 = control-change) bbbb = [0..15]; MIDI channel
0ccccccc:	ccccccc = [0..120]; base note number or control number
0000defg:	d,e,f,g are response mode bits for actuator outputs 8 (d) .. 5 (g) d,e,f,g = 0; trigger mode d,e,f,g = 1; toggle mode
0000hijk:	h,i,j,k are response mode bits for actuator outputs 4 (h) .. 1 (k) h,i,j,k = 0; trigger mode h,i,j,k = 1; toggle mode
0000mnpq:	m,n,p,q are initialisation bits for actuator outputs 8 (m) .. 5 (q) m,n,p,q = 0; actuator output is off after power-up or reset m,n,p,q = 1; actuator output is on after power-up or reset
0000rstu:	r,s,t,u are initialisation bits for actuator outputs 4 (r) .. 5 (u) r,s,t,u = 0; actuator output is off after power-up or reset r,s,t,u = 1; actuator output is on after power-up or reset
0vvvvvvv:	vvvvvvv = [0..127]; MIDI value threshold

Factory default settings for all actuator outputs are:

Actuator output MIDI mapping type {aaa}:	1; note-on
Actuator output MIDI channel mapping {bbbb}:	0
Actuator output MIDI base mapping {ccccccc}:	64
Actuator output response mode {d,e,f,g,h,i,j,k}:	1; toggle mode
Actuator output initialisation {m,n,p,q,r,s,t,u}:	1; on

Actuator output MIDI value threshold {vvvvvvv}: 1

Mapping actuator outputs

Trigger mode means a note-off (header 80h), note-on (header 90h), key-pressure (header A0h) or control-change (header B0h) message with velocity or data field greater than {vvvvvvv} turns actuator output on and a message with velocity or data field zero turns actuator output off.

Toggle mode means a note-off (header 80h), note-on (header 90h), key-pressure (header A0h) or control-change (header B0h) message with velocity or data field greater than {vvvvvvv} toggles actuator output both on and off (a message with velocity or data field zero has no effect).

Base note is the note number or control number which triggers or toggles actuator output 1.

Running status is implemented in the MIDI input mapping when using stand-alone mode.

Command ID: DUMP CONFIG (107)

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 data byte in the [BODY]:

00000001: 1; configuration number
0aaaaaaa: aaaaaaa = [127]; actuator outputs flag

Command ID: EDIT CONFIG 3.5 (102)

For backward compatibility reasons, each of the sensor inputs, as well as all actuator outputs of the Digitizer can be edited using the EDIT CONFIG command, but only wrt. settings available in firmware version 3.5. Upon completion of the command a CONFIG message is sent out by the Digitizer. The EDIT CONFIG command can be used to turn on or off a sensor input while the Digitizer is powered, but it is also possible to 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.

Editing sensor inputs

To edit a sensor input of the Digitizer wrt. settings as defined in firmware 3.5, 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 and 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 immediately) e = 1; sensor input on (effective immediately)
0fffffff:	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:	hhhhhhh 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

Factory default settings are:

Sensor input MIDI mapping type {bbbb}:	11; control-change
Sensor input MIDI channel mapping {cccc}:	0
Sensor input polarity {d}:	1; normal
Sensor activation status {e}:	0; off
Sensor input MIDI note number mapping {f}:	sensor input number {aaaaa} + 1
Sensor input threshold {ggggggg}:	0
Sensor input ceiling {hhhhhhh}:	127

Mapping sensor inputs

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, {hhhhhhh}]).

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 implemented in the MIDI output mapping of the Digitizer firmware v4.0 when using stand-alone mode.

Editing actuator outputs

To program the actuator outputs use the EDIT CONFIG command with a special value for the 2nd data byte in the [BODY], although it does not allow for editing of all settings defined in firmware version 4.0 but only the settings as defined in firmware version 3.5:

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 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
0jjjjjjj:	jjjjjj = [0..120]; base note
0:	reserved
0:	reserved
0:	reserved

Factory default settings are:

Actuator output MIDI channel mapping {aaaa}:	0
Actuator output response mode {b,c,d,e,f,g,h,i}:	1; toggle mode
Actuator output MIDI base note mapping {jjjjjjj}:	64

Mapping actuator outputs

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 or toggles actuator output 1.

Running status is implemented in the MIDI input mapping of the Digitizer firmware v4.0 when using stand-alone mode.

Command ID: DUMP CONFIG 3.5 (103)

The DUMP CONFIG 3.5 command is used to obtain the current settings of the Digitizer limited to settings available in firmware 3.5. Upon completion of the command a CONFIG 3.5 message is sent out by the Digitizer.

The Digitizer will send out the configuration of a Digitizer sensor input when using the DUMP CONFIG 3.5 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 data byte in the [BODY]:

00000001: 1; configuration number
0aaaaaaa: aaaaaaa = [127]; actuator outputs flag

Transmitted messages

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.

Message ID: CONFIG (106)

The CONFIG message contains either the configuration of a sensor input or the configuration of an actuator output.

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

Message ID: CONFIG 3.5 (103)

The CONFIG 3.5 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 3.5 message is identical to the [BODY] of the EDIT CONFIG 3.5 command.

When receiving the configuration of the actuator outputs, the [BODY] of the CONFIG 3.5 message is identical to the [BODY] of the EDIT CONFIG 3.5 command, except when the 2nd data byte is 127 in which case data bytes 7 and 8 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 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
0jjjjjjj:	jjjjjj = [0..120]; base note
0kkkkkkk:	kkkkkkk = [0..127]; sample interval MSB
0mmmmmmm:	mmmmmmm = [0..127]; sample interval LSB
0:	reserved

Where:

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 or toggles actuator output 1

sample interval = kkkkkkk * 128 + mmmmmmm

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. In host mode unprocessed sensor values are output only as system exclusive MIDI messages (ie. no stand-alone mode type channel voice MIDI messages processed from sensor values are output). Host mode allows use of multiple Digitizers in one MIDI chain. When using Max software, it allows multiple iCube / oCube Max objects for each Digitizer (to this end, some of the commands are echoed back to the host computer). In host mode the actuator outputs cannot be set with channel voice MIDI messages but only with the OUTPUT command. Host mode commands (except STREAM and INTERVAL), contrary to stand-alone mode commands, do not change the settings stored in non-volatile memory but only the settings stored in volatile memory. Generally, host mode commands have the same effect in either mode of operation – see for details below.

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

Programming commands

Command ID: STREAM (1, 01h)

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 volatile memory only, when in host mode, and in both volatile as well as non-volatile memory when in stand-alone memory. In host mode STREAM messages are output, while in stand-alone mode channel voice messages are output. After a RESET command in host mode each sensor input is turned off, while in stand-alone mode the activation status is retrieved from 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, 125, 0 {DEV}, 1 {STREAM}, 94 {x = 1, yyyyyy = 30}, 247 (F0h, 7Dh, 00h, 01h, 5Eh, F7h)

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

240, 125, 0 {DEV}, 1 {STREAM}, 30 {x = 0, yyyyyy = 30}, 247 (F0h, 7Dh, 00h, 01h, 1Eh, F7h)

Command ID: RES (2, 02h)

In host mode, each sensor input can either be sampled with 12-bit (hi-res) or 7-bit (lo-res) resolution. After a RESET, the resolution for each input is set to lo-res by default (in host mode). Upon completion of the command the same command is sent out by the Digitizer. In stand-alone mode this command is only useful when sending the SAMPLE command because in stand-alone mode sensor inputs are always sampled with 12-bit resolution.

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, 125, 0 {DEV}, 2 {RES}, 65 {x = 1, yyyyyy = 1}, 247 (F0h, 7Dh, 00h, 02h, 41h, F7h)

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

240, 125, 0 {DEV}, 2 {RES}, 1 {x = 0, yyyyyy = 1}, 247 (F0h, 7Dh, 00h, 02h, 01h, F7h)

Command ID: INTERVAL (3, 03h)

The Digitizer's sampling interval can be set to speed up or slow down data acquisition. The sample interval applies to all sensor inputs, ie. it is not possible to set a different sample interval for each sensor input. The interval is set in milliseconds with 4ms being the lowest (ie. highest rate), and 16383ms (about 16 seconds) being the highest (ie. lowest rate). When the INTERVAL command is sent in host mode the sampling interval is stored in volatile memory, while in stand-alone mode it is stored in both volatile as well as non-volatile memory. After a RESET in host mode, the sampling interval is set to 100ms, while in stand-alone mode it is retrieved from non-volatile memory. Upon completion of the command the same command is sent out by the Digitizer. If the INTERVAL command is sent with value 0 the sampling interval will not be set to 0 but instead an INTERVAL command will be sent out by the Digitizer with the current sampling interval.

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, 125, 0 {DEV}, 3 {INTERVAL}, 7 {xxxxxxx}, 104 {yyyyyyy}, 247 (F0h, 7Dh, 00h, 03h, 07h, 68h, F7h)

The sample interval is 7 * 128 + 104 = 1000

Command ID: SAMPLE (4, 04h)

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 is sent out by the Digitizer, whether in host or stand-alone mode.

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, 125, 0 {DEV}, 4 {SAMPLE}, 11 {yyyyy}, 247 (F0h, 7Dh, 00h, 04h, 0Bh, F7h)

Command ID: MUTE (32, 20h)

The MUTE command stops the sampling of all sensor inputs. It can be used in both modes of operation. 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, 125, 0 {DEV}, 32 {MUTE}, 247 (F0h, 7Dh, 00h, 20h, F7h)

Command ID: SET MUTE (50, 32h)

The SET MUTE command stops the sampling of all sensor inputs. It can be used in both modes of operation. 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, 125, 0 {DEV}, 50 {SET MUTE}, 1 {xxxxxxx}, 247 (F0h, 7Dh, 00h, 32h, F7h)

Command ID: OUTPUT (48, 30h)

The OUTPUT command can be used to turn any of the 8 available actuator outputs on or off. It can be used in both modes of operation. After a RESET with the Digitizer in host mode, all outputs are turned off. 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:

0xyyyyyy:	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, 125, 0 {DEV}, 48 {RES}, 65 {x = 1, yyyyyy = 1}, 247 (F0h, 7Dh, 00h, 30h, 41h, F7h)

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

240, 125, 0 {DEV}, 48 {RES}, 1 {x = 0, yyyyyy = 1}, 247 (F0h, 7Dh, 00h, 30h, 01h, 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, 00h)

The Digitizer, when in host mode, sends it's sensor input values, after being activated through a STREAM command, 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)

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 :

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, 125, 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 (F0h, 7Dh, 00h, 00h, 64h, 5Dh, 18h, 15h, 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, 04h)

The SAMPLE DATA message contains a single snapshot of a sensor connected to a sensor input, whether the Digitizer is in host or in stand-alone mode.

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, 125, 0 {DEV}, 4 {SAMPLE DATA}, 11 {xxxxx}, 64 {yyyyyyy}, 247 (F0h, 7Dh, 00h, 04h, 0Bh, 40h, 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, 125, 0 {DEV}, 4 {SAMPLE DATA}, 11 {xxxxx}, 10 {yyyyyyy}, 10 {zzzzz}, 247 (F0h, 7Dh, 00h, 04h, 0Bh, 0Ah, 0Ah, F7h)

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