ISP-100 Communications Specification

PROPRIETARY

Author:
Dana Troxel
DSP Engineer, EVI Audio

Editor:
Larry Benedict
Applications Engineer, EVI Audio

Release Date:
18 November 1997

MERLIN™ INTEGRATED SIGNAL PROCESSING

EVI AUDIO

10500 West Reno Avenue • Oklahoma City, OK 73132 • USA
Ph. 405-324-5311 • Fax 405-324-8981

TABLE OF CONTENTS

PROPRIETARY

ISP-100 Communication Protocol Overview

Two types of objects exist on the ISP-100, Audio Processing Objects and Configuration/Control Objects. All of the Objects resident on the ISP-100 are allocated and configured by Vue-It software running on an attached PC. This document describes how to control and monitor the ISP-100 after it has been successfully configured by Vue-It.

Objects on the ISP-100 are communicated with using a general messaging protocol. Message traffic is maintained at acceptable levels by using software flow control. Audio Processing objects are addressed in a message by placing their Object Identifier (OID) in the destination Id field of the message. Audio objects such as Compressors, Gates, Filters, Limiters, etc. have their own individual OIDs, which may vary from Quickmap to Quickmap. Each audio object on the ISP-100 supports the same basic set of methods (commands). These methods provide the user a way of getting or setting 1 or all of an object's properties.

The Device Manager, Slot Manager and GPI Manager (OIDs 1,3 & 4 respectively) are Configuration/Control objects. The OIDs of these objects exist for all ISP-100s. The Device Manager provides necessary housekeeping methods for the ISP-100. The Slot Manager provides the user with a way of configuring the hardware input and output cards and supports the same methods used to control the Audio Processing Objects. The GPI manager provides a set of methods/messages which allow the user to maintain the General Purpose Input port located at the back of the ISP-100.

PROPRIETARY

General Message Format

An attached user can communicate with individual objects resident on the ISP-100 through a messaging protocol derived from an early version of the AES24 specification draft. The general form of this message is listed below along with descriptions for each field.

Flag	Destination ID	Length	QID	Reply Handle	Text
Currently Always Zero	Upper 3 Bytes Reserved, Lower Byte is OID	Length = QID+Reply+ Text	Message Command	Return OID(4) Include if QID Bit #14 Set	Message Text
1 Byte	4 Bytes	1 Byte	2 Bytes	0 or 4 Bytes	0+ Bytes

General Messaging Format

Flag: Indicates the form of the message (always zero).

Destination ID: Contains the Identifier of the Object being addressed. Only the lowest byte of this ID is used for objects on the ISP-100.

Length: Contains the number of bytes remaining in the message after the Length byte. The total number of bytes in any message = Length+6.

QID: Qualified Identifier is the command issued to the object specified by the Destination ID. Obviously, only QIDs supported by the particular object addressed are valid. The most significant bit (of the 2 bytes) bit #15 (numbered 0-15) is a reply (acknowledge) bit, and bit #14 is a request for reply (acknowledge) bit.

Reply Handle: The optional reply handle is only present if the Request for Reply bit is set (bit #14). The Reply Handle of the current message becomes the destination ID of the message sent as a reply.

Text: The text contains the optional body of the message.

Hardware Communications Protocol

The ISP-100's front and back panel COMM ports communicate using the following RS232 protocol:

38400 BAUD RATE
No Parity Bit
8 Data Bits
1 Stop Bit

Requesting a Response From an Object

When the request for reply bit (bit #14) is set, then a reply message will be sent when the current message has completed execution. Some messages naturally have a reply as part of their execution. If a message does not have a natural reply, and bit #14 is set, then an execution complete message is sent. The Destination ID of this message is the ID that was previously located in the Reply Handle. There is one byte of text with this message. The byte is TRUE (1) if the message completed without errors, or FALSE (0) if there was an error in its execution.

Message Reply When There is a Natural Response

Consider the following message getting the properties of a component as an example of a message that has a natural response:

Flag	Destination ID	Length	QID	Reply Handle
0x00	0x00,00,00,08	0x06	0x45,00	0x00,00,00,09
1 Byte	4 Bytes	1 Byte	2 Bytes	4 Bytes

Get all the Properties of Audio Component (OID=8) Message Format

If the parameter set request was executed without an error then the following message is sent as a reply when the request was completed:

Flag	Destination ID	Length	QID	Text
0x00	0x00,00,00,09	0x03	0x85,00	N Property Bytes from Component with OID 8
1 Byte	4 Bytes	1 Byte	2 Bytes	1 Byte

Message Format of Get all Properties Completed True

If an error was encountered during execution then an error message will be returned with a variable Error ID indicating the type of error that occurred, no other messages are sent.

Flag	Destination ID	Length	QID	Text
0x00	0x00,00,00,01	0x03	0x3D,7F	Error ID
1 Byte	4 Bytes	1 Byte	2 Bytes	1 Byte

Message Format of Error Message for Set all the Properties request

Message Reply When No Natural Response Exists

Consider the following message setting all the properties of a component.

Flag	Destination ID	Length	QID	Reply Handle	Text
0x00	0x00,00,00,08	0x06+N	0x46,00	0x00,00,00,09	N Bytes of Text
1 Byte	4 Bytes	1 Byte	2 Bytes	4 Bytes	N Bytes

Set all the properties of Audio Component (OID=8) Message Format

PROPRIETARY

The SET_ALL_COMPONENT command has no natural response. In this example setting bit 14 forces the below reply if the set request executed without an error. If the request ACK bit is not set then no reply will be given.

Flag	Destination ID	Length	QID	Text
0x00	0x00,00,00,09	0x03	0x86,00	Non Zero Value(TRUE)
1 Byte	4 Bytes	1 Byte	2 Bytes	1 Byte

Message Format of Set all Properties Completed True

If an error was encountered during execution then an error message will be returned with an Error ID indicating the type of error that occurred, followed by an execution completed false message (both shown below).

Flag	Destination ID	Length	QID	Text
0x00	0x00,00,00,01	0x03	0x3D,7F	Error ID
1 Byte	4 Bytes	1 Byte	2 Bytes	1 Byte

Message Format of Error Message for Set all the properties request

Flag	Destination ID	Length	QID	Text
0x00	0x00,00,00,09	0x03	0x86,00	0x00 (FALSE)
1 Byte	4 Bytes	1 Byte	2 Bytes	1 Byte

Message Format of Set all Properties Completed FALSE

PROPRIETARY

General Message Comm Control Characters and Comm Flow Control

The Comm expects that each message sent and received will start with a STX (2) and end with an ETX (3) control character.

Message flow control is maintained by sending an acknowledge control character (ACK=6 or NACK =21) after a message has been received. No new messages are sent until a control character is received. If no control character is received within a 5.0 second time-out then the ISP-100 enters COMM sleep Mode (see COMM Sleep Mode section below). If the user sends a new message to the ISP-100 before it sends an ACK or NACK the user runs the risk of having the last message sent over-written.

A control character ACK (0x6) is sent if an ETX character was found in its proper place at the end of the message. A control character NACK (0x15) is sent if an ETX character was not found in its proper place at the end of the message.

Comm Control Characters and Flow Control Example

Consider the previously used example of setting the properties of a component. When the Set All Properties message is sent it is first wrapped in the control character package as shown and then sent to the ISP-100.

STX	Flag	Destination ID	Length	QID	Reply Handle	Text	ETX
0x6	0x00	0x00,00,00,08	0x06+N	0x46,00	0x00,00,00,09	N Bytes of Text	0x15
1 Byte	1 Byte	4 Bytes	1 Byte	2 Bytes	4 Bytes	N Bytes	1 Byte

After the message is sent to the ISP-100 the attached user must wait for the single byte ACK (0x6) or NACK (0x15) before another message is sent. If a NACK is sent it is up to the user to decide whether or not it wants to retry.

Likewise when the ISP-100 sends the following properties in response to the previous request it will wait until a single ACK or NACK character is sent before it will send another message. If no

ACK or NACK is received within 5 seconds then the ISP-100's Comm will enter Comm Sleep Mode (See Comm Sleep Mode section).

STX	Flag	Destination ID	Length	QID	Text	ETX
0x6	0x00	0x00,00,00,09	0x03	0x86,00	N Property Bytes	0x15
1 Byte	1 Byte	4 Bytes	1 Byte	2 Bytes	1 Byte	1 Byte

Message Format of Get all Properties Completed True

Audio Processing Objects and The Slot Manager Object

The Audio Processing Objects and the Slot Manager Object are made up of one or more "primitives". When one of these two types of objects are the destination object of a message then the high byte of the message QID contains one of the four methods listed below. If a SET_PRIMITIVE or GET_PRIMITIVE method is used then the low byte of the QID contains the primitive's property identifier (PID) and the first parameter of the message contains the component's primitive number (1-N).

The following four methods can be issued to the Audio Processing Objects or the Slot Manager Object. The method is placed in the high byte portion of the QID (QID_H):

SET_ALL_COMPONENT (6) - Update the properties for all of the primitives of an object
GET_ALL_COMPONENT (5) - Retrieve the properties from all of the primitives of an object
SET_PRIMITIVE (4) - Update 1(or all) property(s) for a primitive in an object
GET_PRIMTIVE (3) - Retrieve 1(or all) property(s) for a primitive in an object

Audio Processing Components

There are 14 Audio Processing Components available on the ISP-100. Each is comprised of one or more Audio Primitives. A Primitives placement within a component can restrict the range of values for the primitive. The Composition of each Audio Component along with restricted Primitive Values within the component are listed below.

The order of the Primitives within a given component determines the ordering of a Component's properties in a SET_ALL_COMPONENT or GET_ALL_COMPONENT message. This same order determines the primitive index value passed with a SET_PRIMITIVE or GET_PRIMITIVE command.

GAIN Component

#	Primitive Names
1	Masteratten
2	Meter (Pre)
3	Meter (Post)

Primitive Subset Restrictions
Masteratten #1 <Type> = GAIN
Meter #2 <Meter Type> = PEAKM
Meter #3 <Meter Type> = PEAKM

DELAY Component

#	Primitive Names
1	Delay

Primitive Subset Restrictions
Delay #1 <Maximum Delay> Actual Value Depends on the Quickmap Configuration (Max<=2.7)

PROPRIETARY

COMBINE2 Component

#	Primitive Names
1	Masteratten
2	Masteratten

Primitive Subset Restrictions
Masteratten #1 <Type> = CUT
Masteratten #2 <Type> = CUT

LIMITER Component

#	Primitive Names
1	Limiter
2	Meter

Primitive Subset Restrictions
Meter #2 <Meter Type> = PEAKN

COMPRESSOR Component

#	Primitive Names
1	Compressor
2	Meter

Primitive Subset Restrictions
Meter #2 <Meter Type> = PEAKN

GATE Component

#	Primitive Names
1	Gate
2	Meter

Primitive Subset Restrictions
Meter #2 <Meter Type> = PEAKN

CUT Component

#	Primitive Names
1	Masteratten

Primitive Subset Restrictions
Masteratten #1 <Type> = CUT

FILTER Component

#	Primitive Names
1	Gain Trim
2-M	Filter(1-N)

The actual number of filter bands N is Quickmap dependent. Each filter band's Type parameter can be configured as 1 of the 8 Types shown below. Each Type has its own range of primitive values. If properties are sent to the filter primitive other then those listed for the given type, undetermined results will occur.

PROPRIETARY

LowPass, HighPass Filter Type Primitive Property Ranges

Property Name	Type	Range of Values
Filter Type	Uchar	LowPass, HighPass
Filter Class	Uchar	Bessel, Butterworth
Order	Uchar	1 or 2
Active Element	Uchar	N/A (0)
Gain	Float	N/A (0)
Frequency	Float	20 to 20000
Bandwidth	Float	N/A (0)
Bypass	Uchar	ON, OFF

LowShelf, HighShelf Filter Type Primitive Property Ranges

Property Name	Type	Range of Values
Filter Type	Uchar	LowShelf, HighShelf
Filter Class	Uchar	N/A (1)
Order	Uchar	1 or 2
Active Element	Uchar	N/A (0)
Gain	Float	-12.0 to +12.0
Frequency	Float	20 to 20000
Bandwidth	Float	N/A (0)
Bypass	Uchar	ON, OFF

AllPass Filter Type Primitive Property Ranges

Property Name	Type	Range of Values
Filter Type	Uchar	AllPass
Filter Class	Uchar	N/A (1)
Order	Uchar	1 or 2
Active Element	Uchar	N/A (0)
Gain	Float	N/A (0)
Frequency	Float	20 to 20000
Bandwidth	Float	N/A (0)
Bypass	Uchar	ON, OFF

PEQ Filter Type Primitive Property Ranges

Property Name	Type	Range of Values
Filter Type	Uchar	PEQ
Filter Class	Uchar	N/A (1)
Order	Uchar	1 or 2
Active Element	Uchar	N/A (0)
Gain	Float	-12.0 to +12.0
Frequency	Float	20 to 20000
Bandwidth	Float	0.083 to 3.0
Bypass	Uchar	ON, OFF

PROPRIETARY

Notch Filter Type Primitive Property Ranges

Property Name	Type	Range of Values
Filter Type	Uchar	Notch
Filter Class	Uchar	N/A (1)
Order	Uchar	1 or 2
Active Element	Uchar	N/A (0)
Gain	Float	-50.0 to +0.0
Frequency	Float	20 to 20000
Bandwidth	Float	0.083 to 3.0
Bypass	Uchar	ON, OFF

Peaked HighPass Filter Type Primitive Property Ranges

Property Name	Type	Range of Values
Filter Type	Uchar	PeakedHpf
Filter Class	Uchar	N/A (1)
Order	Uchar	1 or 2
Active Element	Uchar	N/A (0)
Gain	Float	0.0 to +20.0
Frequency	Float	20 to 20000
Bandwidth	Float	0.083 to 3.0
Bypass	Uchar	ON, OFF

Crossover Components

Three types of crossovers exist on the ISP-100: Two-Way, Three-Way and Four-Way Crossovers. The crossovers are created from individual bands of filters combined with our Masteratten primitives. Changing parameters of a crossover is a rather complex process. We strongly recommend that control of all of the crossover parameters be left to those associated with the Masteratten primitives located at the start of each band of crossover. These Masteratten primitives are restricted to type Cut for all of the crossovers.

The primitive makeup of each crossover component is shown below and on the next page.

Two-Way

#	Primitive Names
1	Masteratten (Low Cut)
2	Masteratten (High Cut)
3	Filter (Low Pass Band A)
4	Filter (Low Pass Band B)
5	Filter (High Pass Band A)
6	Filter (High Pass Band B)

PROPRIETARY

Three-Way Crossover Component

#	Primitive Names
1	Masteratten (Low Cut)
2	Masteratten (Mid Cut)
3	Masteratten (High Cut)
4	Filter (Low Pass Band A)
5	Filter (Low Pass Band B)
6	Filter (Low Pass Band C)
7	Filter (Low Pass Band D)
8	Filter (Mid Pass Band A)
9	Filter (Mid Pass Band B)
10	Filter (Mid Pass Band C)
11	Filter (Mid Pass Band D)
12	Filter (High Pass Band A)
13	Filter (High Pass Band B)
14	Filter (High Pass Band C)
15	Filter (High Pass Band D)

Four-Way Crossover Component

#	Primitive Names
1	Masteratten (Low Cut)
2	Masteratten (Mid Low Cut)
3	Masteratten (Mid High Cut)
4	Masteratten (High Cut)
5	Filter (Low Pass Band A)
6	Filter (Low Pass Band B)
7	Filter (Low Pass Band C)
8	Filter (Low Pass Band D)
9	Filter (Low Pass Band E)
10	Filter (Mid Low Pass Band A)
11	Filter (Mid Low Pass Band B)
12	Filter (Mid Low Pass Band C)
13	Filter (Mid Low Pass Band D)
14	Filter (Mid Low Pass Band E)
15	Filter (Mid High Pass Band A)
16	Filter (Mid High Pass Band B)
17	Filter (Mid High Pass Band C)
18	Filter (Mid High Pass Band D)
19	Filter (Mid High Pass Band E)
20	Filter (High Pass Band A)
21	Filter (High Pass Band B)
22	Filter (High Pass Band C)
23	Filter (High Pass Band D)
24	Filter (High Pass Band E)

DITHER Component

#	Primitive Names
1	Dither

Primitive Subset Restrictions

PROPRIETARY

None
SELECTOR Component

#	Primitive Names
1	Selector

Primitive Subset Restrictions
None

OUTPUT Meter Component

There will always be one Meter associated with every signal output from the ISP-100. This means that the actual number of Meters in an Output Meter Component is Quickmap dependent.

#	Primitive Names
1	Meter
…	…
N	Meter

Primitive Subset Restrictions
All Meter Types are of Type PEAKM

Audio Processing Primitives

Audio Processing Components are made up of one or more of the following primitives. See the section on data type formats for a complete description of the parameter values sent/retrieved from the ISP-100. The ISP-100 user manual will give a more detailed description of some of the primitives listed in its DSP Components section.

COMPRESSOR Primitive

Property Name	PID	Type	Range of Values	Description
Compression Ratio(CR)	1	Float	1.2,1.5,2,3,4,6,8,12,16,24	Output Level Change = Input-Level / CR ; CR Has No Units
Threshold	2	Float	-60.0 to 0	dB relative to digital clipping
Detect Time	3	Float	0.000020 to 5.0	Length of detection window in Seconds
Crest Factor	4	Float	0.0 to 1.0	Peak to Average Sensitivity No Units
Attack Time	5	Float	0.000020 to 0.050	Time in Seconds Until CR is reached
Release Time	6	Float	0.000020 to 5.0	Time in Seconds Until Release From Compression is reached
Side Chain Channel	7	Uchar	Self(0), Other(1), Max_Of_Both(2)	Source Channel For Compression Calculations
Knee	8	Uchar	Hard(1), Soft(2)	No Compression to Compression Transition Description
Bypass	9	Uchar	ON(1), OFF(0)	Bypass On Lets Signal Pass Unaltered

PROPRIETARY

LIMITER Primitive

Property Name	PID	Type	Range of Values	Description
Threshold	1	Float	-60.0 to 0	dB relative to digital clipping
Detect Time	2	Float	0.000020 to 5.0	Length of detection window in Seconds
Crest Factor	3	Float	0.0 to 1.0	Peak to Average Sensitivity No Units
Attack Time	4	Float	0.000020 to 0.050	Time in Seconds Until Limiting is reached
Release Time	5	Float	0.000020 to 5.0	Time in Seconds Until Release From Limiting is reached
Side Chain Channel	6	Uchar	Self(0), Other(1), Max_Of_Both(2)	Source Channel For Limiting Calculations
Knee	7	Uchar	Hard(1), Soft(2)	No Limiting to Limit Transition Description
Bypass	8	Uchar	ON(1), OFF(0)	Bypass On Lets Signal Pass Unaltered

GATE Primitive

Property Name	PID	Type	Range of Values	Description
Threshold	1	Float	-60.0 to 0	dB relative to digital clipping
Attenuation	2	Float	-100 to 0	Attenuation in dB applied to a Signal when threshold is passed
Detection Window	3	Float	0.000020 to 5.0	Time in seconds over which the Gate's average signal level is computed
Open Time	4	Float	0.000020 to 0.050	Time in Seconds for Release from Gate to be reached
Close Time	5	Float	0.000020 to 5.0	Time in Seconds Until Gated Level is reached
Key Channel	6	Uchar	Self(0), Other(1), Max_Of_Both(2)	Source Channel For Gating Calculations
Bypass	7	Uchar	ON(1), OFF(0)	Bypass On Lets Signal Pass Unaltered

PROPRIETARY

SELECTOR Primitive

Property Name	PID	Type	Range of Values	Description
Channel Selected	1	Uchar	1 to N (Depends on Map)	Channel Id to Select Input From

DITHER Primitive

Property Name	PID	Type	Range of Values	Description
Bit Level	1	Int	16 to 24	Bit Level Of Dither Signal
Bypass	2	Uchar	ON(1), OFF(0)	Bypass On Lets Signal Pass Unaltered

GAIN TRIM Primitive

Property Name	PID	Type	Range of Values	Description
Gain Trim	1	Float	-12.0 to +12.0	dB of Boost/Cut for Signal
Polarity	2	Uchar	Negative(0), Positive(1)	Positive Leaves Signals Phase Unaltered
Bypass	3	Uchar	ON(1), OFF(0)	Bypass On Lets Signal Pass Unaltered

DELAY Primitive

Property Name	PID	Type	Range of Values	Description
Delay Time	1	Float	0 to Maximum Delay	Seconds Of Delay to Add to the Signal
Bias	2	Uchar	Dependent On Map	Offset for Delay
Bypass	3	Uchar	ON(1), OFF(0)	Bypass On Lets Signal Pass Unaltered

FILTER Primitive

Property Name	PID	Type	Range of Values	Description
Filter Type	1	Uchar	AllPass(1), DpNotch(2), EQ(3), HighPass(4), HighShelf(5), LowPass(6), LowShelf(7), Notch(8), Peakhpf(9)	Type of Filter Used
Filter Class	2	Uchar	Bessel(0), Butterworth(1), Linkwitz-Riley(2)	Class Of Filter Used
Order	3	Uchar	1,2,3 or 4 (Situation Dependent)	Order Of Filter Used
Active Element	4	Uchar	0 or 1	Active Element of filter
Gain	5	Float	-50.0 to 12.0 (Situation Dependent)	Boost or Cut at Frequency in dB
Frequency	6	Float	20 to 20000	Critical Frequency
Bandwidth	7	Float	0.083 to 3.0	Octave Fraction
Bypass	8	Uchar	ON(1), OFF(0)	Bypass On Lets Signal Pass Unaltered

MASTERATTEN Primitive

Property Name	PID	Type	Range of Values	Description
Type	1	Uchar	Gain(0), Cut(1)	Type of Master Attenuator
Desired Gain	2	Float	-96.0 to 18.0	dB of Gain or Attenuation
Time Constant	3	Float	0.0 to 3600.0	Seconds Until Final Gain Reached
Polarity	4	Uchar	Negative(0), Positive(1)	Positive Leaves Signals Phase Unaltered
Mute	5	Uchar	ON(1), OFF(0)	Mute On Lets No Signal Pass
Bypass	6	Uchar	ON(1), OFF(0)	Bypass On Lets Signal Pass Unaltered

PROPRIETARY

METER Primitive

Property Name	PID	Type	Range of Values	Description
Meter Status	1	Uchar	ON(1), OFF(0)	On indicates Meter is Active
Meter Type	4	Uchar	PeakM(1), PeakN(2), MN_Hold(3)	Compressors, Gates and Limiter Components All use PeakN. Gain and Output Meter Components use PeakM.

Examples of Setting and Retrieving Component Properties

For the purposes of our examples consider an arbitrary Quickmap with the following Audio Components:

Component Name	OID
GAIN	8
DELAY	9
COMBINE2	10
LIMITER	11
COMPRESSOR	12
GATE	13
CUT	14
FILTER(3 Bands)	15
TWO-WAY	16
THREE-WAY	17
FOUR-WAY	18
DITHER	19
SELECTOR	20

SET_ALL_COMPONENT

The SET_ALL_COMPONENT command allows the user to set all of the properties of an Audio Component. The high byte of the QID=6 and the QID Low byte=0. The component properties are ordered in the message following its list of primitives with primitive 1 first and primitive N last. Each primitive's properties are ordered with the lowest PID first and the highest PID last.

The message below sets all of the properties of a GAIN component:

MASTERATTEN Primitive Values to be set

Property Name	PID	Type	Values
Type	1	Uchar	Gain(0)
Desired Gain	2	Float	6.0 dB
Time Constant	3	Float	0.5 Seconds
Polarity	4	Uchar	Positive(1)
Mute	5	Uchar	OFF(0)
Bypass	6	Uchar	OFF(0)

Pre METER Primitive Values to be set

Property Name	PID	Type	Range of Values
Meter Status	1	Uchar	OFF(0)
Meter Type	2	Uchar	PeakM(1)

PROPRIETARY

Post METER Primitive Values to be set

Property Name	PID	Type	Range of Values
Meter Status	1	Uchar	OFF(0)
Meter Type	2	Uchar	PeakM(1)

Flag	Destination ID	Length	QID	Text
0x00	0x00,00,00,08	0x12	0x06,00	0x00,40,C0,00,00,3F,00,00,00,01,00,00,00,01,00,01
1 Byte	4 Bytes	1 Byte	2 Bytes	16 Bytes

Set all the properties of GAIN Component (OID=8) Message Format

GET_ALL_COMPONENT

The GET_ALL_COMPONENT command allows the user to get all of the properties of an Audio Component. The high byte of the QID=5 and the QID Low byte=0. The component properties are ordered in the returned message following its list of primitives with primitive 1 first and primitive N last. Each primitive's properties are ordered with the lowest PID first and the highest PID last.

Flag	Destination ID	Length	QID	Reply Handle
0x00	0x00,00,00,08	0x06	0x45,00	0x11,22,33,44
1 Byte	4 Bytes	1 Byte	2 Bytes	0 or 4 Bytes

Get all the properties of GAIN Component (OID=8) Message Format

The following message reports the properties of the GAIN component in response to the request.

Flag	Destination ID	Length	QID	Text
0x00	0x11,22,33,44	0x12	0x85,00	0x00,40,C0,00,00,3F,00,00,00,01,00,00,00,01,00,01
1 Byte	4 Bytes	1 Byte	2 Bytes	16 Bytes

Return Message Format for Get all GAIN Properties

SET_PRIMITIVE

The set primitive command allows the user to set one or all of a primitive's property(s). The high byte of the QID=4 and the QID Low byte contains the property identifier (PID) of the property(s) to be set. The PID for a primitive property is listed next to its name in the Audio Processing Primitive section. The first byte of the text portion of the SET_PRIMITIVE message is the primitive's number within its Audio Component.

In addition to the PIDs listed for each primitive there are 2 PIDs which are global to all primitives (14 & 15). PID 14 should not be used in the current software. PID 15 allows the user to set all of the properties for the primitive identified in the message. This should not be confused with the functionality of the SET_ALL_COMPONENT command which allows the user to set all of the properties of the component as opposed to all of the properties of the primitive.

The message below sets the desired gain of a GAIN component to 2.0 dB. For this example the MASTERATTEN primitive is the 1st primitive in the component so the first byte of the Text parameter is 1. The QID Low byte contains the PID for Desired Gain (2).

Flag	Destination ID	Length	QID	Text
0x00	0x00,00,00,08	0x07	0x04,02	0x01,40,00,00,00
1 Byte	4 Bytes	1 Byte	2 Bytes	5 Bytes

Set_Primitive Message to change the Desired Gain to 2.0 dB in the GAIN Component

PROPRIETARY

The next message sets all of the properties of the MASTERATTEN primitive in the GAIN component to those shown below. As with the previous example the MASTERATTEN primitive is the 1st primitive in the component so the first byte of Text is 1. The QID Low byte contains PID 15 indicating that all of the properties of the MASTERATTEN are being addressed. The properties are ordered lowest PID to Highest PID.

Property Name	PID	Type	Property Values
Type	1	Uchar	Gain(0)
Desired Gain	2	Float	6.0 dB
Time Constant	3	Float	0.5 Seconds
Polarity	4	Uchar	Positive(1)
Mute	5	Uchar	OFF(0)
Bypass	6	Uchar	OFF(0)

Flag	Destination ID	Length	QID	Text
0x00	0x00,00,00,08	0x0F	0x04,0F	0x01,00,40,C0,00,00,3F,00,00,00,01,00,00
1 Byte	4 Bytes	1 Byte	2 Bytes	13 Bytes

Set_Primitive Message to change all properties of the MASTERATTEN Primitive in the GAIN Component

GET_PRIMITIVE

The get primitive command allows the user to retrieve one or all of a primitive's property(s). The high byte of the QID=3 and the QID Low byte contains the property identifier (PID) of the property to be set. The PID for a primitive property is listed next to its name in the Audio Processing Primitive section. The first byte of the text portion of the SET_PRIMITIVE message is the primitive's number within its Audio Component.

In addition to the PIDs listed for each primitive there are 2 PIDs which are global to all primitives (14 & 15). PID 14 should not be used in the current software. PID 15 allows the user to get all of the properties for the primitive identified in the message. This should not be confused with the functionality of the GET_ALL_COMPONENT command which allows the user to get all of the properties of the component as opposed to all of the properties of the primitive.

The message below gets the desired gain of a GAIN component. For this example the MASTERATTEN primitive is the 1st primitive in the component so the first byte of the Text parameter is 1. The QID Low byte contains the PID for Desired Gain (2).

Flag	Destination ID	Length	QID	Reply Handle	Text
0x00	0x00,00,00,08	0x07	0x43,02	0x44,55,66,77	0x01
1 Byte	4 Bytes	1 Byte	2 Bytes	4 Bytes	1 Byte

Get_Primitive Message to retrieve the Desired Gain in the GAIN Component

The following message reports the Desired Gain of the GAIN component in response to the request.

Flag	Destination ID	Length	QID	Text
0x00	0x44,55,66,77	0x06	0x83,02	0x40,C0,00,00
1 Byte	4 Bytes	1 Byte	2 Bytes	4 Bytes

Returned Message for Get_Primitive Request (Desired Gain)

PROPRIETARY

The next message gets all of the properties of the MASTERATTEN primitive in the GAIN component. As with the previous example the MASTERATTEN primitive is the 1st primitive in the component so the first byte of Text is 1. The QID Low byte contains PID 15 indicating that all of the properties of the MASTERATTEN are being addressed. The returned properties are ordered lowest PID to Highest PID.

Flag	Destination ID	Length	QID	Reply Handle	Text
0x00	0x00,00,00,08	0x07	0x43,0F	0x12,34,56,78	0x01
1 Byte	4 Bytes	1 Byte	2 Bytes	4 Bytes	1 Byte

Get_Primitive Message to retrieve all of the MASTERATTEN properties in the GAIN Component

The following message reports all of the values of the MASTERATTEN primitive in the GAIN component.

Flag	Destination ID	Length	QID	Text
0x00	0x12,34,56,78	0x0F	0x83,0F	0x01,00,40,C0,00,00,3F,00,00,00,01,00,00
1 Byte	4 Bytes	1 Byte	2 Bytes	13 Bytes

Returned Message for Get_Primitive Request (All MASTERATTEN Properties)

Slot Manager Object

The Slot Manger can be viewed as a single component made up of 5 primitives one for each slot. Each slot primitive type is detected at power up. If the slot contains the same IO card that it powered down with then the cards previous state is reloaded. If the same slot contains a different card then it is configured with default settings.

SLOT MANAGER Control Examples

The following Component represents the slot manager contents if the ISP-100 has an ADC card (MIM) in slot1, Slot2 is vacant, Slots 3 and 4 contain DAC cards (MOM) and slot 5 contains a Digital card (MDM).

#	Primitive Names
1	ADC_PRIM
2	VACANT_PRIM
3	DAC_PRIM
4	DAC_PRIM
5	AES_PRIM

The following messages show how to retrieve all of the properties for all of the slots in the Slot Manager. The second message below shows all of the properties returned as a result of the first messages request.

Flag	Destination ID	Length	QID	Reply Handle
0x00	0x00,00,00,03	0x06	0x45,00	0x11,22,33,44
1 Byte	4 Bytes	1 Byte	2 Bytes	0 or 4 Bytes

Get all the Properties of the Slot Manager (OID=3) Message Format

The following message reports the properties of all of the slots in response to the request.

Flag	Destination ID	Length	QID	Text
0x00	0x11,22,33,44	0x34	0x85,00	0x00,02,00,00,00,00,00,00,00,00,00, 01,00, 02,01,00,00,00,00,00,00,00,00,00,00 03,01,00,00,00,00,00,00,00,00,00,00 04,04,01,01,00,01,02,00,01,02
1 Byte	4 Bytes	1 Byte	2 Bytes	50 Bytes

PROPRIETARY

Return Message Format for Get all Slots Properties

The next message sets all of the properties for all of the slots to the properties listed below (same properties as shown above).

Slot1 (MIM Card)
ADC Primitive, Properties with PID numbers 1,2 have no effect since they are read only

Property Name	PID	Value
Slot Number	1	0
Slot Type	2	ADC(2)
Left Gain	3	0.0
Right Gain	4	0.0
Left Pad	5	OFF(0)
Right Pad	6	OFF(0)

Slot2 (Vacant), Properties with PID numbers 1,2 have no effect since they are read only

Property Name	PID	Value
Slot Number	1	1
Slot Type	2	Vacant(0)

Slot3 (MOM Card)
DAC Primitive, Properties with PID numbers 1,2 have no effect since they are read only

Property Name	PID	Range of Values
Slot Number	1	2
Slot Type	2	DAC(1)
Left Gain	3	0.0
Right Gain	4	0.0
Relay Mute	5	OFF(0)
Left Mute	6	OFF(0)
Right Mute	7	OFF(0)

Slot4 (MOM Card)
DAC Primitive, Properties with PID numbers 1,2 have no effect since they are read only

Property Name	PID	Range of Values
Slot Number	1	3
Slot Type	2	DAC(1)
Left Gain	3	0.0
Right Gain	4	0.0
Relay Mute	5	OFF(0)
Left Mute	6	OFF(0)
Right Mute	7	OFF(0)

Slot5 (MDM Card)
AES Primitive, Properties with PID numbers 1-2, and 7-10 have no effect since they are read only

Property Name	PID	Range of Values
Slot Number	1	4
Slot Type	2	AES_EBU(4)
Receive Mute	3	ON(1)
Transmit Mute	4	ON(1)
Pass Through	5	OFF(0)
Sample Rate Conversion	6	ON(1)
Stream Format	7	Consumer(2)
Stream Emphasis	8	NO_EMPHASIS(2)
Stream Sample Rate	9	48.0KHz(1)
Number of Bits	10	24bits(2)

PROPRIETARY

Flag	Destination ID	Length	QID	Text
0x00	0x00,00,00,03	0x34	0x06,00	0x00,02,00,00,00,00,00,00,00,00,00, 01,00, 02,01,00,00,00,00,00,00,00,00,00,00 03,01,00,00,00,00,00,00,00,00,00,00 04,04,01,01,00,01,02,02,01,02
1 Byte	4 Bytes	1 Byte	2 Bytes	50 Bytes

Set all of the Properties for all of the Slots

To turn the transmit mute off on the MDM digital card located in slot 5, the following message can be sent. For this message the PID for Transmit Mute (4) is included as the Low Byte of the QID and the 1st byte of text contains the primitive number (#5) for this component. The only property is set to OFF (0) in the next byte of text.

Flag	Destination ID	Length	QID	Text
0x00	0x00,00,00,03	0x04	0x04,04	0x05,00
1 Byte	4 Bytes	1 Byte	2 Bytes	2 Bytes

Set_Primitive Message to Turn Off the transmit mute on the digital card in slot 5

The following message will set all of the properties for the MIM card in slot 1. For this message the low byte of the QID contains the PID indicating all of the properties for the primitive (PID=15). The first byte of text contains the primitive number for the primitive representing slot1 (#1). The text reflects the properties of the primitive shown below.

Slot1 (MIM Card)
ADC Primitive, Properties with PID numbers 1,2 have no effect since they are read only

Property Name	PID	Value
Slot Number	1	0
Slot Type	2	ADC(2)
Left Gain	3	2.0
Right Gain	4	0.0
Left Pad	5	OFF(0)
Right Pad	6	ON(0)

Flag	Destination ID	Length	QID	Text
0x00	0x00,00,00,03	0x0F	0x04,0F	0x01,00,02,40,00,00,00,00,00,00,00,00,01
1 Byte	4 Bytes	1 Byte	2 Bytes	13 Bytes

Set_Primitive Message to Set all of the properties for the MIM card in slot 1

Slot Manager Primitives

The Slot Manager Component is made up of 5 of the following primitives. Each primitive will have one or more properties that can be individually addressed by a GET_PRIMITIVE or SET_PRIMITVE command using its Property Identifier (PID). See the section on data type formats for a description of each PIDs Data Type and how it is loaded into the Text of a message.

If a GET_ALL_COMPONENT/SET_ALL_COMPONENT command is issued to the slot manager then the properties of each primitive are Returned/Set in a message in the order that they are arranged in the Slot Manager component.

Some of the properties are listed as read only. These properties must be included when SET commands are issued; however the values are not changed. In the case of the Slot Number and Slot Type the values passed in are used for error checking purposes so they must be accurate, the other read only properties are simply ignored on a SET.

PROPRIETARY

ADC Primitive

Property Name	PID	Type	Range of Values	Description
Slot Number	1	Uchar	0,1,2,3,4	The Number of the Slot that the Card is in. (Read Only)
Slot Type	2	Enum	VACANT(0),DAC(1),ADC(2),AES_EBU(4)	The type of card resident in this slot. (Read Only)
Left Gain	3	Float	0.0,+2.0,+4.0,+8.0	Left Channel Gain of Card in dB
Right Gain	4	Float	0.0,+2.0,+4.0,+8.0	Right Channel Gain of Card in dB
Left Pad	5	Enum	ON(1), OFF(0)	-20.0 dB input pad is engaged(On) or not(Off)
Right Pad	6	Enum	ON(1), OFF(0)	-20.0 dB input pad is engaged(On) or not(Off)

DAC Primitive

Property Name	PID	Type	Range of Values	Description
Slot Number	1	Uchar	0,1,2,3,4	The Number of the Slot that the Card is in. (Read Only)
Slot Type	2	Enum	VACANT(0),DAC(1),ADC(2),AES_EBU(4)	The type of card resident in this slot. (Read Only)
Left Gain	3	Float	-8.0,-6.0,-4.0,0.0,+8.0	Left Channel Gain of Card in dB
Right Gain	4	Float	-8.0,-6.0,-4.0,0.0,+8.0	Right Channel Gain of Card in dB
Relay Mute	5	Enum	ON(1), OFF(0)	Both Output Channels are Muted by Relays(On) or Not(Off)
Left Mute	6	Enum	ON(1), OFF(0)	Left Output Channel is Muted in the DAC(On), or not(Off)
Right Mute	7	Enum	ON(1), OFF(0)	Right Output Channel is Muted in the DAC(On), or not(Off)

PROPRIETARY

AES Primitive

Property Name	PID	Type	Range of Values	Description
Slot Number	1	Uchar	0,1,2,3,4	The Number of the Slot that the Card is in. (Read Only)
Slot Type	2	Enum	VACANT(0),DAC(1), ADC(2),AES_EBU(4)	The type of card resident in this slot. (Read Only)
Receive Mute	3	Enum	ON(1), OFF(0)	The Receive Stream is Muted(On) or Not(Off)
Transmit Mute	4	Enum	ON(1), OFF(0)	The Transmit Stream is Muted(On) or Not(Off)
Pass Through	5	Enum	ON(1), OFF(0)	The Received Signal is Passed Through to the Transmitter Bypassing any DSP Output (On), or not(Off)
Sample Rate Conversion	6	Enum	ON(1), OFF(0)	The On Card Sample Rate Converter is Active(On) or Not(Off)
Stream Format	7	Enum	Pro(1),Consumer(2)	Format of Input Stream (Read Only)
Stream Emphasis	8	Enum	EMPHASIS(1), NO_EMPHASIS(2)	The Input Stream is Emphasized (On) or Not (Off) (Read Only)
Stream Sample Rate	9	Enum	48.0KHz(1), 44.1KHz(2),32.0KHz(3), Out of Range(4)	Sample Rate of Input Stream (Read Only)
Number of Bits	10	Enum	20bits(1),24bits(2)	Number of Valid Bits in the Input Audio Stream (Read Only)

PROPRIETARY

COMM Sleep Mode

COMM Sleep Mode provides the ISP-100 a way to re-synchronize with an external PC. It also relieves the ISP-100 from the burden of continually sending error, clip, meter and GPI messages to the PC when it is no longer attached and listening.

During COMM sleep mode the ISP-100's RS232 driver ACKs and NACKs messages sent to it however, it does not send messages from the RS232 COM port **or execute ones received**.

When the ISP-100 exits COM sleep Mode metering activity and unsolicited error messages are left in a suspended state. When the PC re-synchronizes to the ISP-100 and the ISP-100 exits COM sleep mode it is up to the PC to tell the ISP-100 to resume Metering and unsolicited error activity. This can be done 3 ways. The PC can issue a METER_RESUME command, and if any meter's Status is ON, metering activity will resume. The PC can also set any meters status to ON (whether it is already ON or not), or the PC can activate a state that has any meters whose Status set to ON.

Entering COM Sleep Mode

There are 2 basic times for the ISP-100 to enter COM sleep Mode, during power-up initialization and during normal processing. When COM sleep Mode is entered the SIGNOFF message below is sent to the PC.

At power-up the ISP-100 enters sleep mode unless default memory initialization has occurred. For this case no SIGNOFF message is sent to the PC. If default memory initialization has occurred then the ISP-100 issues the following SIGNON command to the PC. If the PC does not respond with a SYNC message within a 0.5 second time-out period then sleep mode is entered and the SIGNOFF message is sent.

Flag	Destination ID	Length	QID	Properties
0x00	0x00,00,00,01	0x0A	0x02,00	Version#
1 Byte	4 Bytes	1 Byte	2 Bytes	8 Bytes

SIGNON Message ISP-100->PC

If after sending the PC a message during normal processing the RS232 COM fails to receive an ACK/NACK before its 5.0 second time-out then sleep mode is entered.

If the ISP-100's device manager receives the following SIGNOFF command sleep mode is entered:

Flag	Destination ID	Length	QID
0x00	0x00,00,00,01	0x02	0x03,00
1 Byte	4 Bytes	1 Byte	2 Bytes

SIGNOFF Message PC->ISP-100 & ISP-100->PC

Exiting COM Sleep Mode

In Sleep mode no messages are sent to the PC. Sleep mode is exited by the following Sync up process:

The PC sends the following SIGNON command to start the COM wake-up process:

Flag	Destination ID	Length	QID
0x00	0x00,00,00,01	0x02	0x02,00
1 Byte	4 Bytes	1 Byte	2 Bytes

SIGNON Message PC->ISP-100

If default memory initialization has occurred exiting COM Sleep Mode starts with the ISP-100 sending the PC the SIGNON command indicated in Figure #1. No PC sent SIGNON command is necessary for this case. Any ACKs NACKs or AES24 messages are acknowledged (ACK or NACK) but thrown away until a SYNC message is received from the PC.

If the PC initiated this process then the ISP-100 Sends the following Sync message to the PC's Device Manager (if no ACK/NACK is received it returns to sleep mode).

Flag	Destination ID	Length	QID	Properties
0x00	0x00,00,00,01	0x0A	0x04,00	Version#
1 Byte	4 Bytes	1 Byte	2 Bytes	8 Bytes

SYNC Message ISP-100->PC

The PC then sends the following SYNC message to the ISP-100's Device Manager. This message wakes the ISP-100's COM, enabling COM messaging (this includes

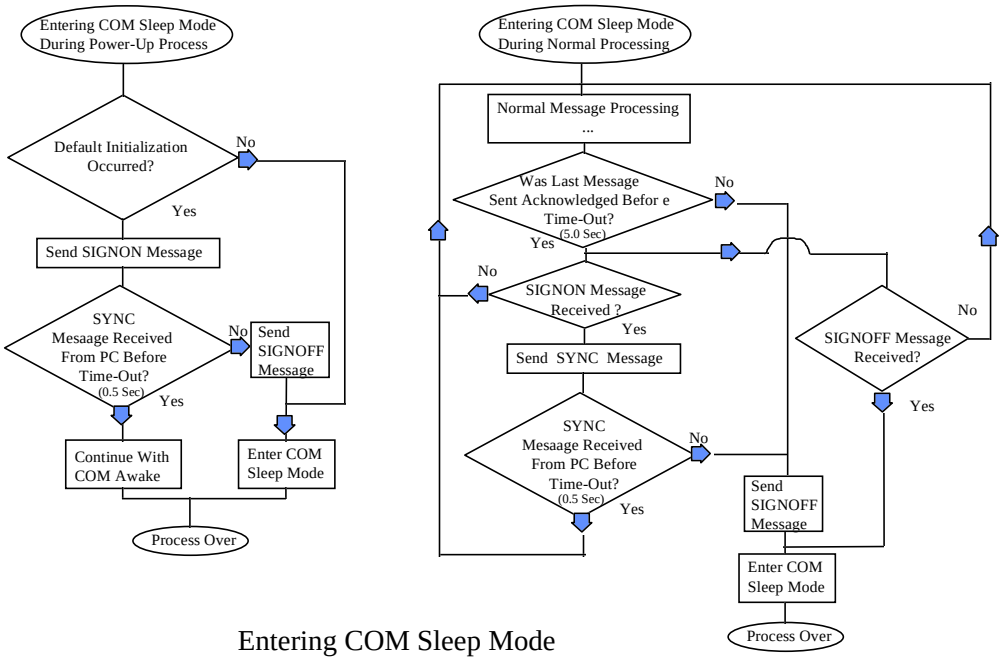
PROPRIETARY

error, clip and GPI messaging, Metering is excluded). If the message is not received before a 0.5 second time-out it returns to sleep mode.

Flag	Destination ID	Length	QID
0x00	0x00,00,00,01	0x02	0x04,00
1 Byte	4 Bytes	1 Byte	2 Bytes

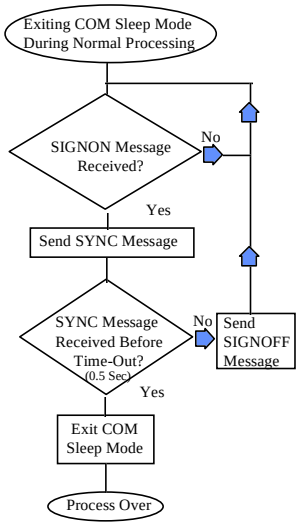
SYNC Message PC->ISP-100

After SYNC is sent from the PC to the ISP-100 it is up to the PC to query, download or whatever it needs to do to assure that the ISP-100 is in the QuickMAP/Quickset that it thinks it is in before adjusting its components.



Entering COM Sleep Mode

Figure: *Entering COM Sleep Mode Program Flow*



Exiting COM Sleep Mode

Figure: *Exiting COM Sleep Mode Program Flow*

PROPRI

Data Type Formats

When high level properties are sent to the ISP-100 it is expected that the data type sent matches the Primitive element data type being addressed. The data types used for each property of an Audio Processing Primitive are outlined in the Audio Processing Primitive section. When the data is received and/or transmitted it is done Most Significant Byte First, with the most significant bit (MSB) within each byte being transmitted first.

The following is a low-level description of each data type used by the ISP-100:

ISP-100 Primitive Element Data types		
Uchar	8 bits, unsigned	MSB(7)...LSB(0)
Enum	8 bits, unsigned	MSB(7)...LSB(0)
Integer	16 bits, signed	MSB(15)...LSB(0)
Unsigned	16 bits, unsigned	MSB(15)...LSB(0)
Long	32 bits, signed	MSB(31)...LSB(0)
Float	32 bits, IEEE std 754-1985	MSB(31)...LSB(0)
Double	64 bits, IEEE std 754-1985	MSB(31)...LSB(0)

ISP-100 Data Types

The IEEE754 standard is as follows:

32-bit single precision -- [Signbit(S)][8-bit Exponent(E)][23-bit Mantissa(M)] Bias=127
64-bit extended precision -- [Signbit(S)][11-bit Exponent(E)][52-bit Mantissa(M)] Bias=1023

The data value is represented as:

$$value = (-1)^{S} x 2^{(E-Bias)} x (1.M)$$

Error Messages

When an error is detected the current process that is executing is terminated and the message below is returned to the user with a 1 byte parameter indicating the type of error that occurred. A list of potential errors can be found in the table below.

Flag	Destination ID	Length	QID	Text
0x00	0x00,00,00,01	0x03	0x3D,7F	Error
1 Byte	4 Bytes	1 Byte	2 Bytes	1 Byte

ERROR Message Format

Name	Value	Description
MSZ30_GONE	1	Out of Memory Partition MSZ30 (Miscellaneous Memory)
MSZ570_GONE	2	Out of Memory Partition MSZ570 (Scene Memory)
MSZ1540_GONE	3	Out of Memory Partition MSZ1540 (DSP Memory)
TIMERS_GONE	4	A Timer Tried to be allocated when none were left
INCORECT_SC	5	Something is wrong with the current Scene (NULL Pointer etc.)
INCORECT_ST	6	Something is wrong with the current State (NULL Pointer etc.)
INVALID_SC	7	Requested Function Cannot be performed with the scene selected.
INVALID_ST	8	Requested Function Cannot be performed with the State selected.
INVALID_MSG	9	The format of the message was not consistent with the command requested (Wrong Number of Properties?)
TIME_OUT	10	Timeout occurred while waiting for a message.
INVALID_CPI	11	Cpi Index Specified In Msg is Too High, or Points to NULL

PROPRIETARY

INVALID_CPIL	12	The CPI List specified does not exist or was in use when user tried to delete it.
INVALID_OID	13	Object Handle specified pointed to an Invalid component
REQUEST_ACK	14	An ACK should have been requested with this message
REQ_TOOLONG	15	Requested Reply is too long. It does not fit in the 254 byte message format.
LIDSZ_ERR	16	Load ID size sent did not match required size for this Load ID
INCORECT_LID	17	Load Id is invalid (NULL?)
INVLD_HSTRUCT	18	Unknown High-Level Structure Type
MAXMETERS	19	The Maximum Number of (Meters-1) was exceeded
DSPMEM_OVR	20	DSP Memory Boundary was overflowed
INVALID_POLARITY	21	Tried to set Primitive value with Invalid Polarity
INVALID_STATUSTYPE	22	Tried to set Primitive value with Invalid Status_Type
INVALID_MATTENTYPE	23	Tried to set Primitive value with Invalid Masterattenuator Type
INVALID_MATTENGAIN	24	Tried to set Primitive value with Invalid Masterattenuator Gain
INVALID_MATTENTC	25	Tried to set Primitive value with Invalid Masterattenuator Time Constant
INVALID_METERTYPE	26	Tried to set Primitive value with Invalid Meter Type
INVALID_SLOTTYPE	27	Tried to set Scene Slot Config with Invalid Slot Type
INVALID_GPIGP	28	The GPI message group requested does not exist
INCORECT_GPIGP	29	Something is wrong with the GPI message group
INVALID_EVENT	30	The GPI event list requested does not exist
INCORECT_EVENT	31	Something is wrong with the Event List
HWID_CSFAIL	32	Hardware Serial ID Checksum Fail
HWID_PGFAIL	33	Hardware Serial ID Did Not Program Correctly
IDCHIP_NOACK	34	Hardware Serial ID Did Not Respond
DSPCHIP_NOACK	35	A DSP chip Did Not Respond In Time(HARD ERROR)
GPIMSG_INV	36	The message cannot be programmed into the GPI
INVALID_GATEATTEN	37	Tried to set Gate with invalid attenuation Value
INVALID_THRESHOLD	38	Tried to set Primitive with invalid threshold value
INVALID_OTIME	39	Tried to set Primitive with invalid Opening Time value
INVALID_CTIME	40	Tried to set Primitive with invalid Closing Time value
INVALID_SCHAIN	41	Tried to set Primitive with invalid Side Chain
INVALID_CR	42	Tried to set Primitive with invalid Compression Ratio
INVALID_DETWINDOW	43	Tried to set Primitive with invalid Detection Window Time
INVALID_CREST	44	Tried to set Primitive with invalid Crest Factor
INVALID_ATIME	45	Tried to set Primitive with invalid Attack Time
INVALID_RTIME	46	Tried to set Primitive with invalid Release Time
INVALID_KNEETYPE	47	Tried to set Primitive with invalid Knee Type
INVALID_DELAYB	48	Tried to set Primitive with invalid Delay Bias
INVALID_DELAYT	49	Tried to set Primitive with invalid Delay Time
INVALID_BITLEVEL	50	Tried to set Primitive with invalid Bit Level
INVALID_FILTTYPE	51	Tried to set Primitive with invalid Filter Type
INVALID_FILTCLASS	52	Tried to set Primitive with invalid Filter Class
INVALID_FILTORDER	53	Tried to set Primitive with invalid Filter Order
INVALID_FILTELEM	54	Tried to set Primitive with invalid Filter Element Number
INVALID_FILTGAIN	55	Tried to set Primitive with invalid Filter Gain
INVALID_FILTFREQ	56	Tried to set Primitive with invalid Filter Frequency
INVALID_FILTBANDW	57	Tried to set Primitive with invalid Filter Bandwidth
INVALID_GTRIM	58	Tried to set Primitive with invalid Gain Trim
INVALID_DACGAIN	59	Tried to set Primitive with invalid DAC Gain
INVALID_ADCGAIN	60	Tried to set Primitive with invalid ADC Gain
SPIMETER_NOACK	61	Time Out Waiting For DSPs to Respond With Meter Data
INTERNAL_ERR	62	An Internal ISP-100 Error Occurred
PRIMITIVE_ERR	63	Error In a Primitive

PROPRIETARY

SCRIPT_ERR	64	There is an Error in the Quickmap Script Sent to ISP-100
INVALID_STREAMFMAT	65	Invalid Digital Audio Stream Format
INVALID_STREAMEMPH	66	Invalid Digital Audio Stream Emphasis
INVALID_STREAMRATE	67	Invalid Digital Audio Stream Sample Rate
AES_VERF	68	Digital Audio Stream Data Validity Error
AES_NO48K_NOSRC	69	Non 48KHz Sample Rate & No Sample Rate Conversion Selected
AES_NONAUDIO	70	Digital Input Data Is Not an Audio Stream
AES_OUTRANGE	71	The Input Signal Is Out Of Range
AES_MNOLOCK	72	Main Board Cannot Lock To Location Specified As Master
AES_MASTER_WSRC	73	Slot Cannot be Chosen as Master and have No SRC Selected
AES_INVALID_MASTER	74	A Slot Must be an AES Card to be Master
BATTERY_LOW	75	The Main Board Battery is Low

ISP-100 Error Messages

Device Manager Commands & Messages

The Device Manager provides a set of commands/messages that provide/report house keeping items for the ISP-100. The pertinent items are listed below. Those commands/messages associated with comm sleep mode are described in the COMM Sleep Mode section.

QUICKSET_SAVE

Changes to Audio processing components resident on the ISP-100 are made to Active Memory and will be lost when power is cycled or a new Quickset is activated. The Quickset Save command can be issued to the Device Manager if the information needs to be saved for later recall. The Quickset save command format is as follows:

Flag	Destination ID	Length	QID	Text
0x00	0x00,00,00,01	0x03	0x0D,00	Quickset Number to Save Active Memory to (1toN)
1 Byte	4 Bytes	1 Byte	2 Bytes	1 Byte

Quickset Save Message Format

QUICKSET_ACTIVATE

The Quickset activate command is issued to the Device Manager to activate one of the Quicksets saved on the ISP-100. The ISP-100 is muted while the new Quickset is activated. The Quickset activate command has the following format:

Flag	Destination ID	Length	QID	Text
0x00	0x00,00,00,01	0x03	0x12,00	Quickset Number to Load into Active Memory (1toN)
1 Byte	4 Bytes	1 Byte	2 Bytes	1 Byte

Quickset Activate Message Format

PROPRIETARY

GLOBAL_MUTE

The Global Mute command is issued to the Device Manager to Mute all of the channels of Audio on the ISP-100. After the Global Mute is issued the Box will remain muted until the Global Unmute is received, regardless of changes made to components. The Global Mute command has the following format:

Flag	Destination ID	Length	QID
0x00	0x00,00,00,01	0x02	0x28,00
1 Byte	4 Bytes	1 Byte	2 Bytes

Global Mute Message Format

GLOBAL_UNMUTE

The Global Unmute command is a complement to the Global Mute command. The unmute command will unmute those components on the ISP-100 whose properties are not in a muted state. If IO cards or components were muted prior to the global unmute then they will remain muted after the unmute is issued. Issuing an Unmute Command without a Global Mute being executed previously has no effect. The Global UnMute command has the following format:

Flag	Destination ID	Length	QID
0x00	0x00,00,00,01	0x02	0x29,00
1 Byte	4 Bytes	1 Byte	2 Bytes

Global Unmute Message Format

BATTERY LOW

When the battery on the ISP-100 is Low a Battery Low message (shown below) is returned to the PC indicating that it is time to change the battery. The message is repeated once every 3 seconds until the low battery is replaced with a charged battery.

Flag	Destination ID	Length	QID	Text
0x00	0x00,00,00,01	0x03	0x06,00	0x4B
1 Byte	4 Bytes	1 Byte	2 Bytes	1 Byte

Battery Low Message Format

AUDIO CLIPS

When Audio clips occur in the ISP-100 the locations of the clips are reported back to the PC in a Clip Message. The clip locations are reported by clip Ids included in the message. If a clip location is specified as PRE it indicates that the audio signal was clipped as it arrived from the input card. The text of the message contains one identifier for each location reporting a clipped signal. The Clip Id locations with their associated values are given below.

IN1A(1), IN1B(2), IN2A(3), IN2B(4), OUT2A(5), OUT2B(6), OUT3A(7), OUT3B(8), OUT4A(9), OUT4B(10), OUT5A(11), OUT5B(12), PRE(13)

Flag	Destination ID	Length	QID	Text
0x00	0x00,00,00,01	0x02+N	0x01,00	Clip Id1,…,Clip IdN
1 Byte	4 Bytes	1 Byte	2 Bytes	N Byte

Clip Message Format

DIGITAL CARD ERRORS

Errors associated with the digital card are reported to the PC in a Digital Card Error Message. The text of the message contains the Number of Errors(N) followed by N pairs which indicate the Error

PROPRIETARY

Location and Error ID. The Error Ids used are the same as the errors listed in the table, shown in the Error Messages Section.

There are 3 potential locations from which digital real-time operational errors may occur, Input Slot1, IO Slot 2 or the External reference. The following errors may occur from slot 1 or 2:

AES_VERF<68> Data Validity Error
AES_NO48K_NOSRC<69> A non 48KHz Sample Rate & No Sample rate Conversion Selected
AES_NONAUDIO<70> Non Audio Input Stream
AES_OUTRANGE<71> Input Signal is Out of Range
AES_MNOLOCK<72> Main Board Cannot Lock to location specified as the Master
AES_MASTER_WSRC <73> Slot Cannot be chosen as master & have sample rate conversion present
AES_INVALID_MASTER <74> A slot must be an AES Card to be master

Only the following error can occur from the external reference location:

AES_MNOLOCK<72> Main Board Cannot Lock to location specified as the Master

If an error occurs from any of the three locations corresponding LEDs turn red on the front panel(Slot1, Slot2 or Host<if external reference>).

Errors are sent back to the PC to the device Manager with a digital error QID<5,0>. The message report the number of errors N followed by the error Location(1of3), and Enum(1of5) for each error. If present they will be sent once per second.

Flag	Destination ID	Length	QID	Property1		Property N-1	Property N
0x00	0x00,00,00,01	0x05	0x05,00	Number of Errors(N)	……	Location(N) Enum	Error(N) Enum
1 Byte	4 Bytes	1 Byte	2 Bytes	1 Byte		1 Byte	1 Byte

UNSOLICITED ERRORS SUSPEND/RESUME

Digital errors, Audio Clips and Battery Low messages form a group of messages referred to as unsolicited errors. When COMM Sleep Mode is entered unsolicited error messages are suspended. Unsolicited errors may also be suspended by issuing a SUSPEND_UNSOLICITED message to the device manager. To enable the sending of these messages a RESUME_UNSOLICITED message must be sent to the Device Manager.

There is no text associated with these two messages.

Flag	Destination ID	Length	QID
0x00	0x00,00,00,01	0x03	0x35,00
1 Byte	4 Bytes	1 Byte	2 Bytes

SUSPEND_UNSOLICITED Message Format

Flag	Destination ID	Length	QID
0x00	0x00,00,00,01	0x03	0x36,00
1 Byte	4 Bytes	1 Byte	2 Bytes

RESUME_UNSOLICITED Message Format

General Purpose Interface (GPI) Manager Messages

PROPRIETARY

The GPI manager maintains the GPI port located at the rear of the ISP-100. A GPI event occurs when a one of 8 switches close on the GPI port. Each of the 8 switch positions can have one or more events associated with it. When a switch has more then one event programmed on it, repeated assertions of the switch will march through the list executing an event at each assertion. When the end of the list is reached the first message in the list will be executed on the next assertion.

Each time the execution of an event is completed the below message is sent to the user indicating the switch ID number and the event in the switch list which was last executed. The switch ID numbers are hexadecimal representations of the bit number for that switch. Switch numbers 1,2,3,4,5,6,7 & 8 have switch Id numbers 1,2,4,8,16,32,64 & 128.

Flag	Destination ID	Length	QID	Property1	Property2	Property3
0x00	0x00,00,00,04	0x05	0x03,00	Scene #	Switch ID#	List # (1-N)
1 Byte	4 Bytes	1 Byte	2 Bytes	1 Byte	1 Byte	1 Byte

GPI Event Executed Message Format

In addition to the 8 switch closures on the back panel, GPI events can be triggered by sending the following message to the GPI manager on the ISP-100:

Flag	Destination ID	Length	QID	Property1
0x00	0x00,00,00,04	0x03	0x03,00	Switch ID#
1 Byte	4 Bytes	1 Byte	2 Bytes	1 Byte

Execute GPI Event command Message Format

The user needs to be aware that if multiple events are programmed into a switch's event list, the pointer to that list will be reset to its beginning position at power up or any time a QuickMAP is activated. Additionally all of the pointers can be reset by issuing the following GPI Pointer Reset command.

Flag	Destination ID	Length	QID
0x00	0x00,00,00,04	0x02	0x04,00
1 Byte	4 Bytes	1 Byte	2 Bytes

GPI Pointer Reset command Message Format

PROPRIETARY