FSR

IT-AS411 Series Serial Protocol

Rev.1.0 07/20/11

244 Bergen Blvd West Paterson NJ 07424 973-785-4347 www.fsrinc.com

Table of Contents

1 General Notes

NOTE: The IT-AS411's baud rate settings are fixed at 38,400 bps with 8 data bits, 1 stop bit and NO parity.

1.1 IT-AS411 Model Types

The IT-AS411 is available in three model types, 8 ohm stereo @ 25 watts/channel, 70 volt mono @ 25 watts, and line level output only for connection to an external amplifier. The serial protocol is identical for all three models.

1.2 Input Ranges:

Inputs range from 1 to 4.

1.3 Audio Ranges:

Input trim value range -5dB to +15dB, output level range -45dB to +5dB.

1.4 Special Characters:

Carriage returns (Enter Key, 0Dh) will be referred to in the document as <*cr*>*.* Linefeeds (0Ah) will be referred to in the document as <*lf*>*.*

1.5 Field Separators:

Fields are separated by *white* space, that is any number of spaces or tabs as long as the entire command is less than 80 characters. A <*cr*> terminates the command. Below is an example describing a command.

EX	05	<*cr*>

So the actual message would look like this:

EX 05<*cr*>

1.6 Command Request Syntax:

This document uses the following notation when describing the syntax of a command request:

BOLD – identifies the command

lower case – identifies data to be entered which is described in the text following the syntax description

" " - entry defined within double quotes is to be entered exactly as shown

< > - entry defined within these brackets is required

[ ] - entry defined within these brackets is optional

| - a vertical bar denotes a logical choice of entry

* - an asterisk following either type of bracket above denotes that data within brackets may be entered multiple times

2 IT-AS411 COMMAND LIST QUICK REFERENCE

COMMAND	DESCRIPTION	PAGE
CON	Connect Request	8
HLP	Help – Request list with format	10
MUT	Mute Request	12
RSP	Response Request	13
SBR	Set Baud Rate Request	14
STA	Status Request	15
VER	Version Number Request	16
VOL	Volume Request	17
VRB	Verbose Error messages	18
{ }	Pass-thru-Protocol Request	19

3 Request/Response Format

All requests and responses will be entirely in ASCII. This will make the IT-AS411 easy to use with the Flex, RN-8200, as well as other control systems from AMX, Crestron, etc.

All requests/responses will have a two to three character type field followed by the data required for that specific request/response. All requests may be terminated with a carriage return (0Dh), the typical terminator, which will be referred to in this document as <*cr*> or may also be terminated with a semi-colon character, ";"***.*** The semi-colon character may also be used to separate successive commands, thereby allowing multiple commands on the same line. Responses are terminated with a carriage return and line feed (0Ah) <*lf*>.

4 Acknowledging Receipt of Commands

Each request sent to the IT-AS411 will have two possible responses, one for the acknowledgement of a correct request and an error response. The normal acknowledge response will be "ACK" followed by a <*cr*><*lf*>*.* The <*cr*><*lf*> termination characters are understood in the responses shown below and are omitted.

Example:

ACK

5 Error Response

It is inevitable that errors occur in the requests sent to the IT-AS411. If an invalid *command* is sent to the IT-AS411, the IT-AS411 will respond with the message “**ERR: unknown command**“. If an invalid parameter is sent to the IT-AS411, the IT-AS411 will respond with the message “**ERR:** “ followed by the valid syntax for the errored entry.

Example:
A connect request with an incorrect type and input number:
CON 8
The error response would be:
ERR: CON <”1”|”2”|”3”|”4”> |<”?”>

6 Connect Request

The Connect request will instruct the IT-AS411 to connect the specified input to the output. The user can also inquire as to which input is currently connected. The format for the Connect request is as follows:

CON	input number	*<cr>*

Syntax: CON <"1"|"2"|"3"|"4"> | <"?"><cr>
Where:

CON	Connect Request Header
input number	Input to connect, one ASCII character 1 to 4
"?"	? for inquiring configured connection

***Example 1*:**

CON 3 <*cr*>

Instructs the IT-AS411 to connect input 3 to the output.

Example Response:

ACK

This response means that the command has been received and is valid.

***Example 2*:**

CON 4<*cr*>

Instructs the IT-AS411 to connect input 4 to the output.

Example Response:

ACK

This response means that the command has been received and is valid.

Error Example:

CON 6<*cr*>

Instructs the IT-AS411 to connect input 6 the output, which is invalid since there are only 4 inputs.

Error Response:

ERR: CON <"1"|"2"|"3"|"4"> |<"?">

Connect Inquiry

If the user wishes to know what input is connected currently, the user would insert a "?" in place of the input number, an Inquiry would look like this:

Example:

CON ?<*cr*>

Example Response:

CON 2

This response informs the user that the IT-AS411 has input 2 selected.

7 Help Request

The user will have the ability to get a list of all commands from the IT-AS411. The user will issue the following request:

HLP	[cmd]	<cr>

Syntax: HLP[cmd]<cr>
Where:

HLP	Help Request header
cmd	optional command identifier

If the optional ***cmd*** is omitted, the IT-AS411 will respond with the following text message:

Response (*cmd* omitted)**:**

```
CON         Connect specified program input.
HLP         Provides help information for IT-AS411 command set.
MUT         Mute or unmute audio.
RSP         Request or decline a response.
SBR         Configure serial bit rate of downstream link.
STA         Request status of audio connections and volumes.
VER         Request current part # and version number.
VOL         Adjust audio output level or input trim levels.
VRB         Turns on or off verbose command error reporting.
{ }         Forwards a string to specified logical link (see manual).
```

Entering **HLP <cmd><cr>**, where cmd is any valid IT-AS411 command in the above list, will return specific help for the command requested.

Specific command help is listed below:

Response (*cmd* included)**:**

```
CON <"1"|"2"|"3"|"4"> |<"?">
     Connect specified program input
     (1-4) or ? for inquiry.
     "1"|"2"|"3"|"4"          Input to connect
      "?"                     ? for inquiring configured connections

HLP[ cmd]
     Provides help information for the IT-AS411 command set.
     cmd          optional command identifier

MUT <"P" | "V"> <"M"|"U">
     Mutes or unmutes program audio or voice channel.
     "P" = Program Audio channel
     "V" = Voice channel
      "M"|"U"     "M" = mute, "U" = unmute
```

RSP <”Y”|”N”><cr>

Turns on or off a response from IT-AS411.

“Y”|”N” “Y” enables a response, “N” disables.

STA<cr>

Status request returns the following:

STA <Number><MuteStatus> <OutputLevel>
<TrimInp1> <TrimInp2> <TrimInp3> <TrimInp4>
<VoiceMuteStatus> <VoiceLevel>

Number	Input (1-4)
MuteStatus	“M” = Muted, “U” = Unmuted
OutputLevel	Program Audio output level -45 to +5 in dB
TrimInp	Program Audio trim –5 to +15 in dB for each of 4 inputs
VoiceMuteStatus	“VM” = Voice Muted, “VU” = Voice Unmuted
VoiceLevel	Voice output level -45 to +5 in dB

VER<cr>

Version request returns the following:

VER <Part#> <Revision#>

Part#	“IT-AS411”	
Revision#	XX.xx,	XX=Major version number, xx=Minor version number

VOL <<”L”|”T” number |”V”> <value|”?”>><cr>

Adjust the audio output level, trim level or Voice level

“L” | “T” | “V” program audio output level, trim level or Voice level

number values of “1” to “4” applicable only when “T” designated

value volume value -5 to +15 for trims, -45 to +5 for level (in 1 dB increments) or “+” to increment or “-“ to decrement volume in 1 dB steps

“?” inquiry for the specified level, trim or voice level

VRB <”Y”|”N”><cr>

Turns on or off verbose error message reporting.

“Y”|”N” “Y” turns verbose on, “N” turns verbose off.

{ }<cr>

“{“[“\S”[#]]*”}” <”””string contents “””><cr>

Forwards the string specified in double-quotes to the specified link.

(See manual for a detailed description of syntax.)

8 Mute Request

The user will be able to mute the audio on the IT-AS411 by sending the mute request. The format for the mute request is as follows:

MUT	“P”|”V”	“M”|”U”	<*cr*>

Syntax MUT <”P” | “V”> <”M”|”U”><cr>

Where:

MUT	Start of mute message
“P”|”V”	“P” = Program Audio channel, “V” = Voice channel
“M”|”U”	“M” = mute, “U” = unmute

Example:

MUT P M<*cr*> – Mutes the program audio output.
MUT P U<*cr*> – Unmutes the program audio output
MUT V M<cr> - Mutes the voice output

9 Response Request

The user will be able to suppress all responses from the IT-AS411 if they like. The user should ensure that all subsequent requests to the IT-AS411 are properly formatted as it would be difficult to ascertain their correctness without a response. The format for the Response request is as follows:

RSP	“Y”|”N”	<*cr*>

Syntax: RSP <”Y”|”N”><cr>
Where:
RSP Response Request header.
“Y”|”N” Y = response enabled, N = response disabled

Example:

RSP Y<*cr*> – Response requested.
RSP N<*cr*> – Response suppressed.

10 Set Baud Rate Request

The user may alter the serial bit rate of the downstream link, SCI 2, by issuing this command. The downstream link (pass-through serial port) is used to forward serial commands received on SCI 1 to equipment connected to SCI 2. Speeds accommodated, in bits per sec, are as described below. *The default configuration is 38,400 bps.* The format for the Set Baud Rate request is as follows:

SBR	“1”|”2”|”3”|”4”|”5”|”6”|”?”	<*cr*>

Syntax: SBR “1”|”2”|”3”|”4”|”5”|”6”> | <”?”><cr>
Where:

SBR	Set Baud Rate Request header.
“1”|”2”|”3”|”4”|”5”|”6”	1=2400, 2=4800, 3=9600, 4=19200, 5=38400, 6=57600
“?”	Inquiry returns the currently configured baud rate.

Example:

SBR 1<*cr*> – Sets serial baud rate to 2400 bps.
SBR ?<*cr*> – If 2400 bps was previously configured, this command returns:
1=2400 bps

NOTE: The IT-AS411’s baud rate settings are fixed at 38,400 bps with 8 data bits, 1 stop bit and NO parity.

11 Status Request

The user may request status from the IT-AS411 at any time. The returned status will include, in order, current audio connection, mute status, output level, and trim value for each of 4 inputs. The format of the Status request is as follows:

STA	*<cr>*

Syntax: STA<cr>
Where:
STA Status Request header

Response: **STA** <*Number*><*MuteStatus*> <*OutputLevel*> <*TrimInp1*> <*TrimInp2*> <*TrimInp3*> <*TrimInp4*> <*VoiceMuteStatus*> <*VoiceLevel*>
Where:
Number Values **1**, **2**, **3**, or **4** representing the current Program Audio input.
MuteStatus **M** for muted, **U** for unmuted
OutputLevel Value representing the current Audio Output Level in range –45 to +5.
TrimInp Value representing the current Audio Trim Level in range –5 to +15 for each of 4 Audio sources.
VoiceMuteStatus **VM** for Voice muted, **VU** for Voice unmated
VoiceLevel Value representing the current Voice Level in range -45 to +5 in dB

Example:

STA<*cr*>

Requests status of the IT-AS411.

Example Response:

STA 3M –20 +5 +10 +8 -2 VM -45

This response means that program audio input three is currently connected to the output, the program audio channel is muted, and the level is -20, the input trims are +5 +10 +8 -2 respectively, Voice channel is muted and the voice level is -45.

12 Version Number Request

The user has the ability to request the IT-AS411's Part Number and the current firmware version number. The request is formatted as follows:

VER	*<cr>*

Syntax: VER<cr>
Where:
VER Version Request header

Response: **VER** *Part#* *Version#*
Where:
Part# **IT-AS411**
Version# XX.xx, where XX = Major version number and xx = Minor version number.

Example:

VER<*cr*>
To which the IT-AS411 and firmware revision 2.05 will respond with the following:
VER IT-AS411 02.05

13 Volume Request

The Volume request is used to adjust input trims, program audio level and voice level. It can be used to inquire about the current value of a program audio input trim, program audio output level or voice level. The levels can be set as a value, or adjusted up or down with "+" or "-".

VOL	"L" | "T" number | "V"	value|"?"	*<cr>*

Syntax: VOL <<"L"|"T" number | "V"> <value|"?">><cr>

Where:

VOL Volume request Header

"L" | "T" output Level, Trim

number values of "1" to "4", applicable only when "T" designated

"V" Voice Level

value volume value -5 to +15 for trims, -45 to +5 for level (in 1 dB increments), -45 to +5 for Voice level or "+" to increment volume in 1 dB increments, or "-" to decrement volume in 1 dB decrements. Multiple "+" or "-" may be specified consecutively, each representing 1 dB step.

"?" inquiry for the specified level, trim or voice

Example:

VOL L -23<*cr*>

Instructs the IT-AS411 to set the program audio output level to -23.

VOL V -30<cr>

Instructs the IT-AS411 to set the voice level to -30.

VOL T 1 10

Instructs the IT-AS411 to set input 1's Trim level to 10.

VOL L +++

Instructs the IT-AS411 to increment program output level +3dB.

Example Response:

ACK

This response means that the command has been received and is valid.

Error Example:

VOL T 2 40<*cr*>

Instructs the IT-AS411 to set input 2's trim to "40" which is invalid.

Error Response:

ERR: VOL <<"L"|"T" number> <value|"?">

The IT-AS411 would send the error message above indicating that the "40" in the message was invalid (trim values are from -5dB to +15dB).

VOLUME INQUIRY

If the user wishes to query a trim, program audio level or voice level, the user would insert a "?" in place of the Value, an Inquiry would look like this:

Example:

VOL L ?<*cr*>

This requests the value of the program audio output level, to which the IT-AS411 would respond:

Example Response:

VOL L –10

This response informs the user that the current program audio output level setting is –10dB.

14 Verbose Request

The user can, during debugging, turn on verbose mode. This mode will expand the standard error messages to include the expected command syntax along with a description of the commands and possible parameters as formatted above in the specific help for that errored command. If an input command is unidentified, verbose mode will output the contents of the general HLP command. If verbose mode is off, then any errored input will result in just the command syntax being output as described in **Syntax** for each command. The verbose request is defined as follows:

VRB	“Y”|”N”	<*cr*>

Syntax: VRB <”Y”|”N”><cr>

Where:

VRB	Verbose request header
“Y”|”N”	“Y” turns verbose on, “N” turns verbose off.

Example:

If the user tries to connect program audio input 6 (which is not in valid range of 1-4) using the CON command, the error message would appear as follows:

```
ERR:  CON <”1”|”2”|”3”|”4”> |<”?”>
      Connect  audio of the specified input
      (1-4) or ? for inquiry.
      ”1”|”2”|”3”|”4”         Input to connect
       “?”                    ? for inquiring configured connections
```

15 Pass-thru-Protocol Request

The Pass-thru-Protocol facilitates reaching end devices not able to be directly connected with the originator of a command.

The IT-AS411 is capable of forwarding a serially received request to a device further downstream using the Pass-thru-Protocol. The request to be forwarded is essentially composed of two parts. The first part, an ASCII string referred to as the Drop Preamble, describes the logical link to which the second part, it too an ASCII string, will be forwarded to. For IT-AS411, the logical link may only be a serial link but devices further downstream may have the capability to designate other types of logical links that can be incorporated in the contents of the drop preamble. The Pass-thru-Protocol request is defined as follows:

“{“Drop Preamble”}”	“”””Drop String””””	<*cr*>

Note: the syntax description below follows our convention of characters within quotes as being required, contents enclosed within [ ] brackets as being optional and the * character meaning that it may be repeated 1 or more times. Also note that the syntax is presented as a BNF grammar to fully specify the syntax. This form of grammar decomposes the initial description of the Drop String into progressively smaller units that identify the full extent of the composition of characters allowed. Examples are provided below for clarification.

Syntax: “{“Drop Preamble”}” “””Drop String”””<cr>
Where:

Drop Preamble [“\S[#]”]*, That is: a \ character, followed by a S character identifying a *serial* logical link, followed by an optional number designating which serial logical link. That sequence may be repeated as many as 5 times, a practical limit.
For example, \S\S1\S2\S\S2.
This designates a drop preamble that will traverse five logical links (each which must have Pass-thru-Protocol capability) before reaching the end device to which the Drop String is intended to target (further examples later).

Drop String Drop String is enclosed within double quote characters.
= [SpecialASCIISeq] [Non-Special ASCII Char] [ASCIIHexSeq] DropString

SpecialASCIISeq = “\<” | “\>” | “\\” | “\”” (ie a backslash char, \, precedes the special characters <, >, \, and “, which have special uses within the Drop String (see below).

Non-Special ASCII Char = ASCII characters

ASCIIHexSeq = “<” [ASCIIHexBytePair]*”>”, ie zero or more pairs of ASCII hex bytes separated by a space and enclosed within <> characters.
For example, <AA BB CC 0F 0D>.

ASCIIHexBytePair = XX, where X = 0-9, A-F.

Example:

The target device to which the Drop String contents are intended may be a EIKI projector and the user may wish to turn it on, which is the sequence C00 followed by a carriage return. If the projector is directly connected to the downstream serial port of the IT-AS411then the Pass-thru-Protocol request will be the following:

{\S} “C00 <0D>”<cr>

The above request has the drop preamble directing the Drop String enclosed within double quotes to the IT-AS411downstream serial link. The IT-AS411 will interpret the drop preamble as this device being the final forwarding device and will send the string C00 <cr> (the 0D ASCII hex pair represents a carriage return) to its downstream serial link.

If the projector were connected via two successive IT-AS411devices, or perhaps a single IT-AS411and a second Pass-thru-Protocol capable device then the request would be the following:

{\S\S} “C00 <0D>”<cr>

If the end target device, that is the one receiving the Drop String, is attached to a PtP capable device via 4 intervening PtP capable devices and contains a combination of both ASCII characters and ASCII hex sequences then the (fictitious) command might look as follows:

{\S\S3\S2\S1\S} “A more complex example <AA BB 55 AB> with multiple embedded ASCII hex seqs <01 02 FF 0D>”

The above request addresses 5 PtP capable devices. The first device interprets the Drop Preamble directing the Drop String to its serial logical link, S. This device strips off its logical link, the first \S, and sends the remainder out serial logical link S, ie {\S3\S2\S1\S} “A more complex example <AA BB 55 AB> with multiple embedded ASCII hex seqs <01 02 FF 0D>”.

Each succeeding device interprets the Drop Preamble destination, strips off its address and forwards the remaining request. This occurs until the last device in the Drop Preamble in which the remaining request is: {\S} “A more complex example <AA BB 55 AB> with multiple embedded ASCII hex seqs <01 02 FF 0D>”.

The last device will strip off the Drop Preamble entirely (as nothing remains of it after stripping off \S and parse the Drop String. The parsing will output any non-special sequences as is and will interpret any ASCII hex sequences and output the corresponding hex values. Thus for our example above, the final output is:

A more complex example 0xAA 0xBB 0x55 0xAB with multiple embedded ASCII hex seqs 0x01 0x02 0xFF 0x0D

Where the hex sequences 0xAA, 0xBB, etc are raw hex values and NOT ASCII hex.

The response to any PtP request by the end device will be returned upstream via any and all intervening PtP capable devices to the originating device. Thus, if there were a response by the EIKI projector to the Power On command, it will be forwarded to the originator.