



SnapAV Binary MoIP Controller Integration Protocol Document

Integration Protocol v1.1 rev20190301

Firmware 1.1.0.0

Overview

This integration protocol details how a third-party system can be used to control a SnapAV Binary MoIP Controller. With the controller online, the integration protocol will be listening for connections on **port 23 at the controllers IP address**. **NOTE: 10 simultaneous connections can be made at a time**. To get started, netcat or similar software can be used to initiate a connection and test any of the following protocol commands below.

Authentication

The protocol requires authentication before proceeding with commands. Once connected, a login prompt will be received and the third-party system must provide a valid username and password. If correct, login will be successful and other commands can be issued. If incorrect, the third-party system will be prompted for login again.

Specification

THIRD-PARTY SYSTEM <-----> SnapAV Binary MoIP Controller
i.e. MoIP IP: 192.168.0.20 Port: 23

Integration

Message Structure
Command and response messages are standard ASCII text.
? – Request message
! – Control message
- Error message
~ - Unsolicited message
\n – End of command message, ASCII hex: 0x0A dec: 11

Protocol

Protocol Command	Description/Response
?Firmware\n	Request Firmware Version. Response: ?Firmware=1.0.0.0\n
?Receivers\n	Request all Receivers current inputs. Response: ?Receivers=1:3\n Where 1 is the TX and 3 is the RX. This will be comma delimited for multiple devices.

?Devices\n	<p>Request TX and RX count.</p> <p>Response: ?Devices=1,4\n where 1 is the TX count and 4 is the RX count.</p>
?Name=T\n Where T is 0/1	<p>Request the names for either TX or RX. To request all the TX names, use 1 for the payload. To request all the RX names, use 0 for the payload. The response will be new line delimited for multiple devices where each lines format is as follows: ?Name=MODE,INDEX,NAME.</p> <p>Request for TX: ?Name=1\n Response for TX: ?Name=1,1,TX-D46A9121000B\n</p> <p>Request for RX: ?Name=0\n Response for RX: ?Name=0,1,RX-D46A91210620\n ?Name=0,2,Basement TV\n ?Name=0,3,Living Room TV\n ?Name=0,4,RX-D46A91210604\n</p>
!Switch=TX,RX\n Where TX is the index of the Transmitter you want to switch and RX is the index of the Receiver you want the switch to happen on.	<p>Switches the input on a Receiver to the desired Transmitter.</p> <p>Request to switch to Transmitter 1 on Receiver 2: !Switch=1,2\n Success Response: OK\n Error Response: #Error</p>
!Resolution=RX,R\n Where RX is the Receiver you'd like to change the resolution of and R is one of the following: 0 = Pass through resolution from the source. 1 = 1080p 60Hz 2 = 1080p 50Hz 3 = 2160p 30Hz 4 = 2160p 25Hz	<p>Changes the resolution on a given Receiver.</p> <p>Request to switch Receiver 1's resolution to Pass-Through: !Resolution=1,0\n Success Response: OK\n Error Response: #Error</p>
!OSD=RX,MSG\n Where RX is the Receiver index you'd like to display MSG on. MSG must be plain ASCII Text.	<p>Displays a plain text message on the display of the given Receiver.</p> <p>Request to display "Hello World" on Receiver 1: !OSD=1,Hello World\n Success Response: OK\n Error Response: #Error</p> <p>NOTE: To clear the text, send !OSD=1,CLEAR\n</p>
!Reboot\n	<p>Request to reboot the MoIP controller.</p> <p>Reboot Controller Request: !Reboot\n Success Response: OK\n Error Response: #Error</p>

<p>!CEC=RX,MODE\n</p> <p>Where RX is the Receiver index you'd like to control CEC on and MODE is one of the following:</p> <p>0 = CEC OFF 1 = CEC ON</p>	<p>Controls CEC for a given Receiver. MODE must either be 0 for OFF or 1 for ON.</p> <p>Request CEC Off on Receiver 1: !CEC=1,0\n</p> <p>Success Response: OK\n</p> <p>Error Response: #Error</p>
<p>!Serial=TYPE,INDEX,BAUD,DATABITS,PARITY,STOPBITS,DATA\n</p> <p>type: 0 = output (RX), 1 = input (TX)</p> <p>index: device to send baud: integer baudrate data bits: 5, 6, 7, 8 parity: n = none, e = even, o = odd stop bits: 1, 2</p> <p>data: hex data to send</p>	<p>Sends serial data to RX or TX serial port.</p> <p>Send to TX 2 at 9600-8n1 the characters "abc": !Serial=1,2,9600-8n1,61 62 63</p> <p>Success Response: OK\n</p> <p>Error Response: #Error\n</p>
<p>~Serial=TYPE,INDEX,DATA\n</p> <p>TYPE: 0 = output (RX), 1 = input (TX)</p> <p>INDEX: device to send DATA: hex data received</p>	<p>Unsolicited serial data to the connected client. This data will be sent over the protocol without a request. The third-party system should always be handling these incoming messages.</p> <p>TX #2 sent characters "abc": ~Serial=1,2,61 62 63</p>
<p>~Receivers=TX,RX\n</p> <p>Where TX is the currently selected Transmitter index and RX is the Receiver index.</p>	<p>Broadcasts all Receivers current inputs.</p> <p>Response: ?Receivers=1:3\n</p> <p>Where 1 is the TX and 3 is the RX. This will be comma delimited for multiple devices.</p>
<p>#Error\n</p>	<p>Sent whenever an invalid command was received or an internal device error has occurred.</p> <p>Consider this example with only 2 connected Transmitters and 5 connected Receivers:</p> <p>Request to switch Transmitter 2 to Receiver 6: !Switch=2,6\n</p> <p>Response: #Error</p> <p>Receiver 6 does not exist, therefore an error is returned.</p>

Example:

```
$ nc 192.168.27.51 23
Please Login to Continue
Username: binary
Password: binary
Successfully Logged In!
?Model
?Model=B-900-MOIP-4K-CTRL
```