Controlling ENCO DAD using Axia GPIO IP interface

Version:1.01, 29 November, 2010
Adapted from a document prepared by ENCO Systems

Using The Axia Routing Protocol

You can access an Axia IP driver or Axia node by opening a Telnet session to the appropriate IP address port 93. Using the local loopback to the driver this would be:

```
telnet 127.0.0.1 93
```

When connected, you must send commands in all UPPER CASE. A few examples:

DST (this will list all current destination mappings)
SRC (this will list all current source mappings)

```
DST 1 ADDR:"239.192.0.1"        (sets destination 1 to Axia channel 1) *
DST 2 ADDR:"239.192.1.45"       (sets destination 2 to Axia channel 45) *
SRC 8 ADDR:"239.192.0.105"      (sets source 1 to Axia channel 105) *
```

**NOTE: Axia system channel numbers do not necessarily correspond to the last octet of the output device's IP address as the list above might imply. These channel numbers are generally specified in the configuration Web pages served by Axia devices, and may range from 1 – 30,000. See Axia product documentation for instructions on how to set channel numbers.*

You can also set DAD 5.1a or above to issue these commands via Send Text DCLs. You must first configure the [SEND TEXT] IO lines in the CFI file to the port, IP address and termination (delimiter) character:

```
[SEND_TEXT] A_IO=-93 127.0.0.1 ^M
```

Notice that DAD requires the minus (-) sign for ports under 1024.

You can then build commands to control the routing as above. Notice that you do not need to include the double quotes (") around the addresses when the commands are issued by DCL:

```
SEND TEXT A 'DST 1 ADDR:239.192.0.1' SEND TEXT A 'DST 2
ADDR:239.192.1.45' SEND TEXT A 'SRC 1 ADDR:239.192.0.2'
```

You can set different Send Text levels (A-Z) to talk to different addresses, so this structure will allow DAD to control various Axia nodes, Axia consoles, and other workstation IP audio drivers (routing) as well as its own local IP audio driver settings.

NOTE: When communicating with any Axia device through an IP address other than the local loopback (127.0.0.1), you must authenticate to the unit before you can change any settings. This is can be done by sending the following command:

```
SEND TEXT A 'LOGIN <password>'
```

The default password is blank, but the space and blank must be sent, so the command is:

```
SEND TEXT A 'LOGIN '
```

This login can precede each command or it can be a separate command cut that is setup as a startup command cut so that DAD authenticates when it first starts.

Communication With Axia GPIO Nodes

It is also possible to use these communication settings to talk to Axia GPIO nodes. Since these will have an IP external to the PC, you must LOGIN as described above. You may then use Send Text commands to control the outputs of the node. The IO on the nodes is arranged into channels, each of which has 5 GPIs and 5 GPOs. So an 8 channel node actually has 40 ins and 40 outs. You control each output by specifying the characters L for low, H for high or X for don't change. The outputs are normally high, so you "close" an output by commanding it to go low. The command to close output 1 is therefore:

```
SEND TEXT A 'GPO 1 LXXXX'
```

and to open it:

```
SEND TEXT A 'GPO 1 HXXXX'
```

To operate GPOs higher than 5, you must use appropriate channel numbers. For instance, to pulse GPO 13 for 1 second, use this command sequence:

```
SEND TEXT A 'GPO 3 XXLXX' DELAY NEXT CMD 1000 SEND TEXT A 'GPO 3
XXHXX'
```

Currently the inputs from the Axia GPIO node are not supported, but ENCO says they intend to add this soon. Please contact ENCO for more information.

Communication Between Dad And Axia Consoles

Console to DAD Control (GPIOVK FILES)

The Axia IP audio driver contains "GPIO" functionality that can be used to control DAD. This is quoted because no hardware contact closures are used to provide this connectivity. The console uses an Axia protocol to send messages to the GPIO component of the IP audio driver. This in turn uses a user definable profile to convert these Axia GPIO messages into DAD commands that are then sent to UDP port 2002 which DAD listens to and executes the received commands.

Each playback device in the IP Audio Driver (Source 1 -16) has 5 GPIs and 5 GPOs associated with it (see Axia documentation for more details). These are "hard coded" in Axia to represent ON, OFF, PREVIEW, START and STOP. The only ones of any concern at this point are START and STOP.

To access the GPIO features of the driver, click the GPIO button in the IP Audio Driver screen. Here you enter the file name of the GPIO profile which can be any name with the extension .GPIOVK. Here is an example of a GPIOVK profile that enables an Axia console to start and stop DAD's PBK1 (using device 1) and PBK2 (using device 2):

```
################################################################################
# Device identifiers:
# DEVA -Device #01 (A)
# DEVB -Device #02 (A)
# ...
# DEVP -Device #16 (A)
#
# Note: for compatibility reasons, device IDs DEV0-DEV7 are also supported.

# Event constants:
# 0 ON
# 1 OFF
# 2 PREV
# 3 START
# 4 STOP

UDP_DSTPORT=2002

DEVA.3COMMAND="<DADCMD><ID>99999</ID><COMMAND>play pbk1</COMMAND></DADCMD>"
DEVA.4COMMAND="<DADCMD><ID>99999</ID><COMMAND>stop pbk1</COMMAND></DADCMD>"

DEVB.3COMMAND="<DADCMD><ID>99999</ID><COMMAND>play pbk2</COMMAND></DADCMD>"
DEVB.4COMMAND="<DADCMD><ID>99999</ID><COMMAND>stop pbk2</COMMAND></DADCMD>"
################################################################################
```

The # sign signifies a comment. The Device identifier comments explain that the first IP driver device (device #01) is referenced as DEVA in this file. Device #16 is referenced as DEVP. The Event constants comments explain that a START command maps to event 3 and a STOP command maps to event 4.

The first active line of the file is UDP_DSTPORT which configures the GPIO module to send output commands to UDP port 2002 which is the port DAD listens on. The rest of the file contains a series of command interpreter lines which identify an incoming command then send a corresponding command to the UDP_DSTPORT.

Commands coming in from the console contain both device and event information expressed as <device>.<event>. For instance a START command for device 1 is identified by DEVA.3. The rest of the formatting in these actions lines is critical for proper communication with DAD so just imitate the above examples. You can replace the DCL in the middle of the string (e.g. play pbk1) with any valid DCL. You can also send multiple commands on a single line by repeating all the formatting around each DCL command including the quote marks ("). For example, the following will STOP and NEXT PBK1.

```
"<DADCMD><ID>99999</ID><COMMAND>stop
pbk1</COMMAND></DADCMD>""<DADCMD><ID>99999</ID><COMMAND>next
pbk1</COMMAND></DADCMD>"
```

When a console surface channel is assigned an Axia network channel # that corresponds to one of the devices originating from an IP driver, pressing the console's channel ON button will send a Start command to that device. Pressing the channel OFF button will send a Stop command to that device.

DAD to Console Control (SEND TEXT)

You can also have DAD send control commands to an Axia console to turn a channel on and off for instance. This works similar to talking to a GPIO Node as described above. You must build Send Text DAD Control of Axia Devices

DCLs in DAD and send them to the local machine's IP audio driver. This is done using the local loopback IP 127.0.0.1 on port 93. In the DAD CFI file, make the following entry for whatever Send Text level (A-Z) you choose to use. This example is using level A:

```
[SEND_TEXT] A_IO=-93 127.0.0.1 ^M
```

You do not need to authenticate to the local address, so no LOGIN or password is needed.
The Send Text command strings follow the GPIO protocol described earlier. Each IP audio device (1-16) has a GPI channel (1-16) associated with it. This GPI channel has 5 inputs that correspond to ON, OFF, PREVIEW, START and STOP. If you send an ON command to GPI 1, the IP audio driver communicates with the Axia console and turns on any surface channel that has IP audio device #1 routed to it.

The GPIs are triggered when they go low, but for some reason, they all start out as low. So you either need to send two commands, one to make them all high, then another to send the desired GPI low, or you can send a single command that toggles the states of both the ON and OFF GPIs together. I have used the second method and have found the following to work:

```
SEND TEXT A 'GPI 1 LHXXX' # turns on console channel that has IP audio device 1 assigned to it
SEND TEXT A 'GPI 1 HLXXX' # turns off console channel that has IP audio device 1 assigned to it
```

It is possible to put these commands into the DAD.GPO file so that DAD will automatically put itself "UP" on the console whenever something is played. Here is an example of a GPO file that will turn the console on any time any virtual machine in DAD plays audio to the channel assignment mapped to IP Audio Device 1:

```
BOARD1_PGM_START C "SEND TEXT A 'GPI 1 LHXXX'" BOARD1_PGM_STOP C
"SEND TEXT A 'GPI 1 HLXXX'"
```

The second line will turn the module off when the audio stops playing. See DADpro32 System Reference Manual, Section 10 for more information on DAD Command Language and using GPO files.

CAUTION: If you setup both Console-to-DAD control AND DAD-to-Console control, you can create a control feedback loop that can cause serious trouble. Currently there is no way around this so we recommend you use only one direction of control. If you want to start DAD primarily from the console channel buttons, use Console-to-DAD control. If you wish to operate DAD primarily from the touchscreen or other DAD interfaces, use the DAD-to-Console control.