

# iPAudioDriver

Sound Card Replacement for Windows®



Document Version 2.4, Jan 2009  
IP-Audio Driver version  
2.4.8.12 or higher



# Customer Service

## We support you...

### **By Phone/Fax in the USA.**

- Customer service is available from 9:30 AM to 6:00 PM USA Eastern Time, Monday through Friday at +1 216.241.7225. Fax: +1 216.241.4103. Please call 216-622-0247 for Axia 24/7 phone support.

### **By Phone/Fax in Europe.**

- Service is available from Axia Europe in Germany at +49 81 61 42 467. Fax: +49 81 61 42 402.

### **By E-Mail.**

- The address is [Support@AxiaAudio.com](mailto:Support@AxiaAudio.com).

### **Via World Wide Web.**

- The Axia Web site has a variety of information which may be useful for product selection and support. The URL is <http://www.AxiaAudio.com>. Our Axia Discussion Boards are at <http://forums.AxiaAudio.com>

## Feedback

We welcome feedback on any aspect of the Livewire products or this manual. In the past, many good ideas from users have made their way into software revisions or new products. Please contact us with your comments.

## Updates

All of our products are undergoing constant improvement. Periodic updates may become available - to determine if this is the case, visit our web site periodically, or contact us for advice concerning whether a newer release is more suitable to your needs.

Our electronic newsletter has announcements of major software updates for existing products, as well as keeping you up to date on the latest Axia, Telos, and Omnia product releases.

To subscribe go to <http://www.axiaaudio.com/news/eNews.htm>.

## Trademarks

Telos Systems, Axia Audio, Livewire, the Livewire Logo, the Axia logo, Element, SmartQ, Status Symbols, Zephyr, Zephyr Xstream, Zephyr Xport, Omnia, the Omnia logo, and the Telos logo, are trademarks of TLS Corporation. All other trademarks are the property of their respective holders.



### **Axia Audio**

2101 Superior Ave. Cleveland, OH 44114 USA  
+1 (216) 241-7225  
[Inquiry@AxiaAudio.com](mailto:Inquiry@AxiaAudio.com)

### **Axia Europe**

Johannisstraße 6, 85354 Freising, Germany  
+49 81 61 42 467  
[Inquiry@AxiaAudio.com](mailto:Inquiry@AxiaAudio.com)

Copyright © 2006-2009 by TLS Corporation. Published by Axia Audio. We reserve the right to make improvements or changes in the products described in this manual, which may affect the product specifications, or to revise the manual without notice. All rights reserved.

## Notice

All versions, claims of compatibility, trademarks, etc. of hardware and software products not made by Axia mentioned in this manual or accompanying material are informational only. Axia makes no endorsement of any particular product for any purpose, nor claims any responsibility for operation or accuracy.

## Warranty

This product is covered by a 90-day limited warranty, the full text of which is included in Appendix D of this manual.

## Service & Support

You must contact Axia before returning any software for replacement. Axia will issue a Return Authorization number, which must be written on the exterior of your shipping container. Be sure to adequately insure your shipment for its replacement value. Packages without proper authorization may be refused. US customers please contact Axia technical support at +1 (216) 241-7225. All other customers should contact their local representative to arrange for service.

If you call Axia Support with a software question, we strongly recommend being near your computer when you call, so our Support Engineers can verify information about your configuration and the conditions under which the problem occurs. Please have your software license key handy.

## Feedback is welcome

At Axia, we love to hear your feedback. If you find anything in this manual that you feel needs clarification or correction, please let me know: e-mail [cnovak@AxiaAudio.com](mailto:cnovak@AxiaAudio.com).

## About This Manual

This manual covers setup and use of the Axia IP-Audio driver software for Windows. It is assumed in this document that you are familiar with Livewire's basic concepts, as outlined in the companion *Introduction to Livewire: Systems Primer*.

If you have not done so, please review that material first. In it we explain the ideas that motivated Livewire and how you can use and benefit from it, as well as nitty-gritty details about wiring, connectors, and the like. Since Livewire is built on standard networks, we also help you to understand general network engineering so that you have the full background for Livewire's fundamentals. After reading *Introduction to Livewire* you will know what's up when you are speaking with gear vendors and the network guys that are often hanging around radio stations these days.

As always, we welcome your suggestions for improvement. Contact Axia Audio with your comments:

### **Axia Audio, a Telos Company**

2101 Superior Avenue

Cleveland Ohio 44114 USA

Phone: +1.216.241.7225

Web: [www.AxiaAudio.com](http://www.AxiaAudio.com)

E-Mail: [Inquiry@AxiaAudio.com](mailto:Inquiry@AxiaAudio.com)

# Table of Contents

<i>Customer Service</i> . . . . .	iii	<b>Appendix A: Livewire Ports</b> . . . . .	12
<i>Warranty</i> . . . . .	iv	<b>Appendix B: Windows and GPIO Interfaces</b> . . . . .	13
<i>Service &amp; Support</i> . . . . .	iv	GPIO Bridge Service . . . . .	13
<i>About This Manual</i> . . . . .	iv	<i>What is the Windows GPIO Bridge Service?</i> . . . . .	13
<i>A Note From The President of Telos</i> . . . . .	vi	<i>GPIO Profile File Format</i> . . . . .	13
<i>A Note From The President of Axia</i> . . . . .	vii	<i>Profile File Format</i> . . . . .	13
<b>Chapter One: Introducing the Axia IP Audio Driver</b> . . . . .	1	<i>Windows Class Name</i> . . . . .	13
<i>Description</i> . . . . .	1	<i>Sending Windows Messages</i> . . . . .	13
<i>Axia IP-Audio Driver vs Axia IP-Audio Multi Channel (OEM)</i> . . . . .	1	<i>Sending UPD messages</i> . . . . .	13
<b>Chapter Two: Installation and Configuration</b> . . . . .	3	Windows GPIO/TCP-IP . . . . .	
<i>Installing the Axia IP-Audio Driver</i> . . . . .	3	Direct Interface . . . . .	14
<i>Configuring Axia IP-Audio Driver</i> . . . . .	4	<i>Connection</i> . . . . .	14
<i>NIC (Network Interface Card) Selection</i> . . . . .	5	<i>Commands</i> . . . . .	14
<i>Enable and Name Sources</i> . . . . .	5	<b>Appendix C: Troubleshooting Tools</b> . . . . .	16
<i>Destinations</i> . . . . .	6	<i>Livewire Statistics Window</i> . . . . .	16
<i>GPIO – (OEM Version Only)</i> . . . . .	6	<i>Clock Sync Information:</i> . . . . .	16
<i>Statistics</i> . . . . .	6	<i>Transmit (to the network) information:</i> . . . . .	17
<i>PC Audio Nominal Level</i> . . . . .	7	<i>Receive (from the network) information:</i> . . . . .	17
<i>Keep Channel Active</i> . . . . .	7	<i>Windows Performance Monitor</i> . . . . .	18
<i>Windows XP Firewall Settings</i> . . . . .	7	<b>Appendix D: Warranty</b> . . . . .	19
<i>Using and Configuring The Axia IP-Audio Driver as a Playback Device</i> . . . . .	7	<b>Tech Note Section</b> . . . . .	20
<i>Mixing Multiple Sources to an IP-Audio Playback Device</i> . . . . .	8	Controlling AudioVAULT using GPIO IP interface . . . . .	20
<i>Using IP-Audio for Multi-Channel Recording and Playback</i> . . . . .	8	Controlling ENCO DAD using Axia GPIO IP interface . . . . .	25
<b>Chapter Three: Advanced Features</b> . . . . .	9	Controlling Prophet NexGen using Axia GPIO IP interface . . . . .	29
The IP-Audio Driver GPIO Bridge . . . . .	9	Using Axia Windows Driver GPIO . . . . .	32
<i>Starting the GPIO Module</i> . . . . .	9		
<i>Loading a GPIO Profile</i> . . . . .	9		
. . . . .	9		
Livewire Statistics . . . . .	9		
Using Axia Driver With The Standard Windows™ Multi-media Interface . . . . .	10		
<i>Setting Sources and Destinations for Software Applications</i> . . . . .	10		
<i>Setting Recording and Playback Levels</i> . . . . .	11		

## A Note From The President of Telos

It's been a tradition since Telos' very first product, the Telos 10 digital phone system, that I share a few words with you at the beginning of each manual. So here goes.

In radio broadcast studios we're still picking up the pieces that have fallen out from the digital audio revolution. We're not using cart machines anymore because PCs are so clearly a better way to store and play audio. We're replacing our analog mixing consoles with digital ones and routing audio digitally. But we're still using decades-old analog or primitive digital methods to connect our gear. Livewire has been developed by Telos to provide a modern PC and computer network-oriented way to connect and distribute professional audio around a broadcast studio facility.

Your question may be, "Why Telos? Don't you guys make phone stuff?" Yes, we certainly do. But we've always been attracted to new and better ways to make things happen in radio facilities. And we've always looked for opportunities to make networks of all kinds work for broadcasters. When DSP was first possible, we used it to fix the ages-old phone hybrid problem. It was the first use of DSP in radio broadcasting. When ISDN and MP3 first happened, we saw the possibility to make a truly useful codec. We were the first to license and use MP3 and the first to incorporate ISDN into a codec. We were active in the early days of internet audio, and the first to use MP3 on the internet. Inventing and adapting new technologies for broadcast is what we've always been about. And we've always been marrying audio with networks. It's been our passion right from the start. In our genes, if you will. As a pioneer in broadcast digital audio and DSP, we've grown an R&D team with a lot of creative guys who are open-eyed to new ideas. So it's actually quite natural that we would be playing marriage broker to computer networks and studio audio.

What you get from this is nearly as hot as a couple on their wedding night: On one RJ-45, two-way multiple audio channels, sophisticated control and data capability, and built-in computer compatibility. You can use Livewire as a simple sound card replacement – an audio



interface connecting to a PC with an RJ-45 cable. But add an Ethernet switch and more interfaces to build a system with as many inputs and outputs as you want.

Audio may be routed directly from interface to interface or to other PCs, so you now have an audio routing system that does everything a traditional "mainframe" audio router does – but at a lot lower cost and with a lot more capability. Add real-time mixing/processing engines and control surfaces and you have a modern studio facility with many advantages over the old ways of doing things. OK, maybe this is not as thrilling as a wedding night – perhaps kissing your first lover is a better analogy. (By the way, and way off-topic, did you know that the person you were kissing was 72.8% water?)

While were on the subject of history... you've probably been soldering XLRs for a long time, so you feel a bit, shall we say, "attached" to them. We understand. But no problem – you'll be needing them for microphones for a long while, so your withdrawal symptoms won't be serious. But your facility already has plenty of Ethernet and plenty of computers, so you probably already know your way around an RJ-45 as well. It's really not that strange to imagine live audio flowing over computer networks, and there's little question that you are going to be seeing a lot of it in the coming years.

The 20<sup>th</sup> century was remarkable for its tremendous innovation in machines of all kinds: power generators, heating and air conditioning, cars, airplanes, factory automation, radio, TV, computers. At the dawn of the 21<sup>st</sup>, it's clear that the ongoing digitization and networking of text, audio, and images will be a main technology story for decades to come, and an exciting ride for those of us fortunate to be in the thick of it.

Speaking of years, it has been a lot of them since I wrote the Zephyr manual intro, and even more since the Telos 10 – almost 20 years now. Amazing thing is, with all the change around us, I'm still here and Telos is still growing in new ways. As, no doubt, are you and your stations.

Steve Church, January 2004

## A Note From The President of Axia

Nearly 20 years ago, I designed my first broadcast console for PR&E. I look back on that time with great fondness; we were building bullet-proof boards for the world's most prestigious broadcasters, making each new console design bigger and fancier to accommodate a wider variety of source equipment and programming styles. The console was the core of the studio; all other equipment was on the periphery.

Then things changed: the PC found its way into broadcast audio delivery and production. At first, PC audio applications were simple, used only by budget stations to reduce operating expenses. But soon the applications evolved and were embraced by larger stations. Slowly, the PC was taking center stage in the radio studio.

Like many, I was captivated by the PC. Stations retired carts, phonographs, open-reel decks, cassettes — even more modern digital equipment such as DAT and CD players, replacing all with PC apps. Client/server systems emerged and entire facilities began using PCs to provide most — or all — of their recorded audio. Yet consoles continued to treat PCs as nothing more than audio peripherals. I knew that we console designers were going to have to rethink our designs to deal with computer-centric studios.

During this time, traditional broadcast console companies began producing digital versions. But early digital consoles were nearly identical in form and function to their analog predecessors. It took a fresh look from a European company outside broadcasting to merge two products — audio routing switchers and broadcast consoles — into a central processing engine and attached control surface. Eventually nearly every console and routing switcher company followed suit, and a wide variety of digital “engines” and control surfaces flooded the market.

But, advanced as these integrated systems were, they still handled computer-based audio sources like their analog ancestors. Sure, the router and console engine were now integrated, but the most important studio element — the PC — was stuck in the past, interfaced with

100-year-old analog technology. The PC and console couldn't communicate in a meaningful way — strange, considering that PCs everywhere were being networked, fast becoming the world's most popular and powerful communication tool.

Then a group of Telos engineers developed a method of using Ethernet to interconnect audio devices, allowing computers and consoles, controllers and peripherals to interact smoothly and intelligently. Powerful, flexible networks had finally come to our studios. As with the transition from carts to computers, the benefits are many and impressive. A few networked components can replace routing switchers, consoles, processing peripherals, sound cards, distribution amps, selector switches and myriad related devices.

This deceptively simple networked system costs a fraction of other approaches, yet has capabilities surpassing anything else. The system is modular and can be used to perform discrete functions in a traditional environment. Concurrently, it easily scales to serve both the humblest and the very largest of facilities. Console, router, and computer work in harmony.

So, equipped with this new technology and countless ideas, we launch *Axia*, the newest division of Telos. *Axia* is all about delivering innovative networked audio products to future-minded broadcasters. On behalf of our entire team, I welcome you as a charter client. *Axia* is the culmination of nearly 40 man-years of some of the most ambitious R&D ever applied to the radio industry. And this is only the beginning. We have more products, innovations, and partnerships in the pipeline.

You already know your *Axia* system is unlike anything else. So it shouldn't be surprising that your new system is loaded with new thinking, new approaches, and new ideas in virtually every conceivable area. Some concepts will challenge your traditional ideas of studio audio systems, but we're certain that once you have experienced the pleasures of the networked studio, you'll never want to go back. And now, for something completely different...

Michael “Catfish” Dosch, February 2004



*Bits, bytes, megabytes*

*Cascading beautifully.*

*Audio river.!*



# Chapter One:

## Introducing the Axia IP Audio Driver

---

This manual is written with the assumption that you have read our *Introduction to Livewire* document. While the Axia Livewire technology is easy to use in powerful ways, it does represent a radical new way of thinking for broadcasters. That document will serve to get your feet wet and to orient you. We highly recommend you review that document first, before building a Livewire audio system or reading this manual.

If you are already one of the thousands of Livewire users, congratulations! Axia's IP-Audio driver will make your experience with networked audio even better.

### Description

The Axia IP-Audio Driver is the software interface between your PC's audio applications and the Livewire network. It provides the following functions:

- Software interface for audio “sources” to be sent to the Livewire network from PC/Windows audio applications such as multichannel delivery systems and other audio players.
- Software interface to receive audio from the Livewire network to “destinations” on the PC/Windows system such as audio recording applications.
- A ‘GPIO’ function to convey “button-press” data from the Livewire network to “destination” applications. For example, a control surface fader start button will command a PC/Windows-based audio player to start playback (OEM version only)

There are two versions of the IP-Audio driver. The first version is the single stereo (or surround) driver that is used in much the same manner as a standard sound card in a Windows PC. This version of the driver is sold to our customers and supported as an end-user product like most of our other products.

The other version of the IP-Audio driver is a multi-

channel version that is sold only through Axia's industry partners such as the developers of broadcast automation or playout systems.

Both versions of the IP-Audio driver interface through Windows to appear as audio input/out devices. The Axia IP-Audio Driver for Windows allows integrating Windows multimedia system with an Axia Livewire audio network. There are wide variety of applications where Axia Livewire hardware and the Windows software are used together. The driver provides all software components needed to record and play Livewire streams using standard recording software for Windows.

The Axia IP driver package also contains a GPIO module that allows controlling many Windows audio applications from Smart Surface or Element console (Supported in the multichannel OEM version only). This feature is commonly used for “device start” commands initiated by pressing the Channel ON button on the Axia control surface.

### Axia IP-Audio Driver vs Axia IP-Audio Multi Channel (OEM)

The Axia IP-Audio Driver end-user version emulates a single sound card, with one stereo (or surround) audio output device and one stereo (or surround) audio input device. This version is suitable for typical two-channel (stereo) playback or recording applications.

Axia IP-Audio Multichannel OEM version emulates up to 24 stereo sound cards\*, with one stereo (or surround) audio output device and one stereo (or surround) audio input device per “sound card”. It is intended for more complex professional applications. The multichannel driver is provided only as an OEM product to Axia's partners and is not available to users as a standalone product.

Other than the number of audio channels and the addition of GPIO to the multichannel driver, the two drivers are the same. The following information applies to both drivers, with the understanding that the number of inputs and outputs is different as mentioned above.

Figure 1-1 shows how PC's equipped with the IP-Audio driver might be interconnected with other Livewire studio equipment.

In this example we show two PC's that have the multi-channel OEM driver. They might be delivering several channels of stereo audio to the Livewire network while, at the same time, recording other streams - maybe for voice tracking or satellite recording.

A couple of laptop users are also shown. They might be using the single-channel IP-Audio driver in an audio editing application where they could edit phone calls or material downloaded from the Internet.

Axia nodes are also shown here. They are used to convert analog or AES audio into Livewire streams that are made available to the network. Some of these streams

would be sources such as microphones or phone lines. The nodes also convert Livewire audio streams to analog or AES audio. These are connected to external devices such as headphone amplifiers, cue speakers, etc. The Studio Mix Engine and Element Control Surface perform the mixing and processing functions in the system.

All sources created by the IP-Audio are available to any device in the Livewire network and in a similar manner, all sources created by other devices are available to the IP-Audio users.

Of course, it is all tied together with one or more high-quality Ethernet switches such as the Cisco 3560 shown here.

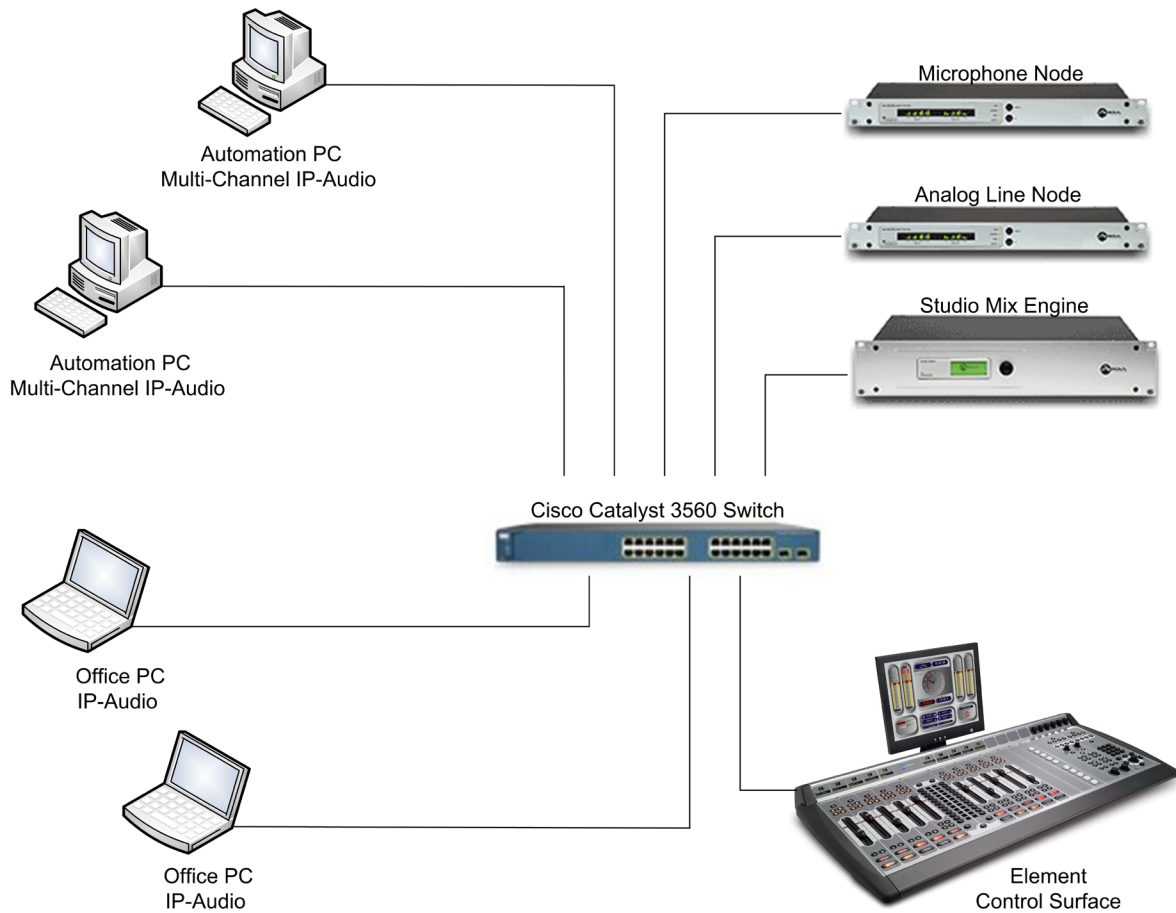


Figure 1-1: Basic Livewire Network

# Chapter Two:

## Installation and Configuration

### Installing the Axia IP-Audio Driver

The Axia IP-Audio driver supports the following Microsoft operating systems:

- Microsoft Windows 2000 Professional
- Microsoft Windows XP Pro (32-bit, standard)
- Microsoft Windows XP Home Editions
- Microsoft Windows 2003 Server R2

Microsoft Vista (32-bit versions) can be used however some items specific to Vista are currently under development. Vista compatibility is somewhat version dependent. Please contact Axia support for assistance if required.

There are versions of the IP-Audio driver available for some Macintosh and Linux operating systems. Those products are covered by separate documentation.

If you are running a recent version of the operating systems, or have installed the latest “service pack” from Microsoft, installation will go smoothly. In the rare case where the installer gives an error message indicating that the installer is outdated, you must install the latest service pack from Microsoft. Alternatively you can download the required Microsoft Installer 2.0 from <http://www.microsoft.com/downloads/>.

If you are upgrading from a previous version of the IP-Audio driver, it is recommended that you uninstall the old version before installing the updated version.

The steps to install the IP-Audio driver follow. And yes - there are pictures!

1. Insert your CD and let auto-run launch the installer OR if your have disabled auto-run, explore the CD and double-click the AxiaIPDR.MSI file to start the installation process. You will see the Welcome screen

and the version of the driver you are installing. Use NEXT button to proceed with the installation.

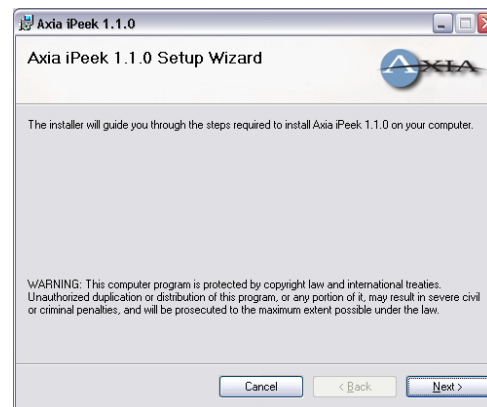


Figure 2-1: Installation Wizard

2. Read, understand and accept the End-User License agreement. Click Next.

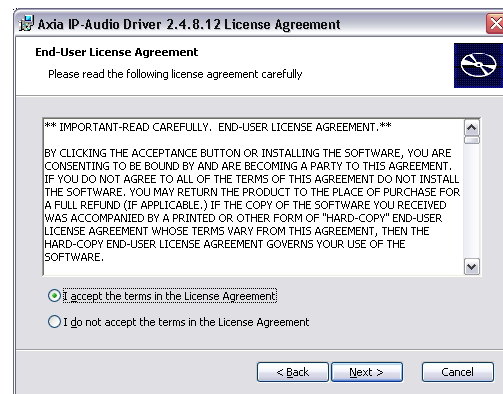


Figure 2-2: License Agreement

3. Enter your User information, License Number and License Key. Note that each license is for a single installation if the IP-Audio driver. If you install the driver on multiple machines, each will require its own license number and key.

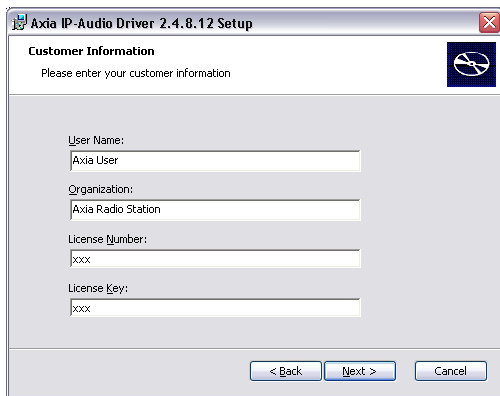


Figure 2-3: User Information

- The next step is to choose the number of devices that will be installed with the IP-Audio driver (OEM version only). The number of devices chosen will be dependent on the software you intend to run as well as the capabilities of your PC. Of course, the more channels, the more resources it will take to process the IP-Audio. We suggest installing the minimum number of devices. If you find that you need more in the future, you can always update the driver.

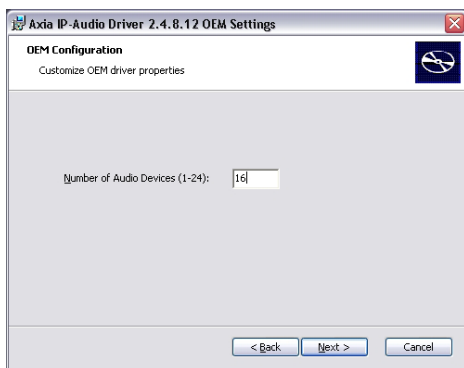


Figure 2-4: Livewire Devices

- Choose the setup type. “Typical” is recommended and should be used. It will install Axia IP-Audio driver, Livewire terminal and GPIO modules.
- You will be prompted with the “Ready to install” screen. Select “Install” to proceed.

- You may be presented with a Windows Logo warning as shown below. Please select “Continue Anyway” to continue with the installation.

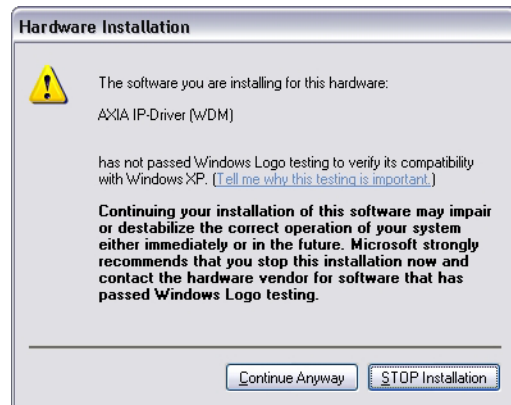


Figure 2-5: Windows Logo Notification

- Your installation should now be complete and you are ready to configure the IP-Audio driver .

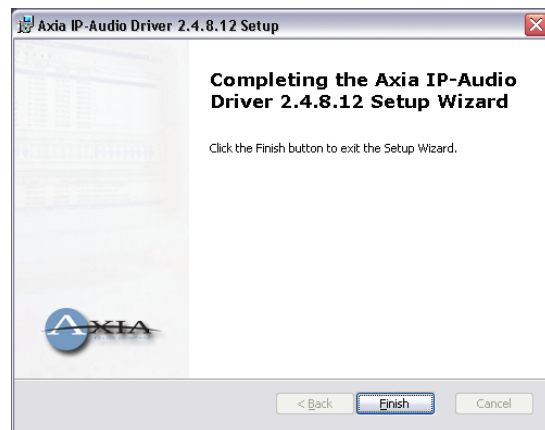


Figure 2-6: Installation Complete

## Configuring Axia IP-Audio Driver

After all files are installed on your system, Axia IP-Audio Configuration window will appear, as in Figure 2-7. You can access this configuration screen at any time from the Control Panel. In the following section, each of the configuration functions will be discussed in detail.

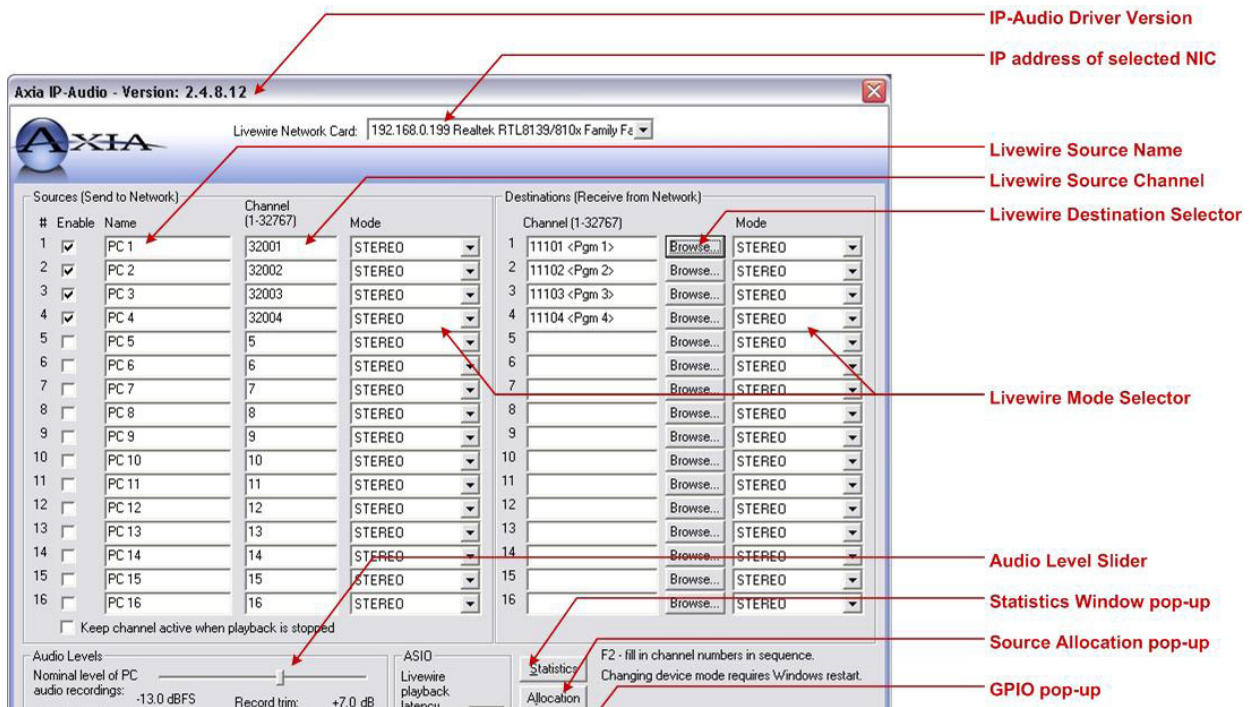


Figure 2-7: IP-Audio Driver - Control Panel

### NIC (Network Interface Card) Selection

If your computer has multiple Network Interface Cards (NICs) installed make sure that network card connected to the Livewire audio LAN is selected from the pull down menu for Livewire Network Card shown in figure 1-8. We recommend that this be a “wired” connection as opposed to a wireless NIC. Even a single Livewire stream requires more bandwidth than provided by most wireless connections.

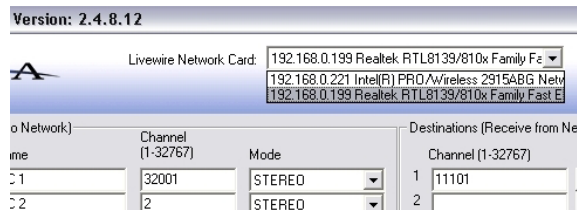


Figure 2-8: Network Card Selection

### Enable and Name Sources

The IP-Audio sources are the streams that your PC is delivering to the Livewire network - your Windows playback devices. This is an important concept. For the multi-channel driver, enable only as many devices as you intend to use by checking the enable box. The single

channel version permits only one input and one output audio device. The remaining devices will be greyed out and cannot be selected. At this time you can also provide names for these IP-Audio sources.

Try to be as descriptive as possible so this name will tell you which machine and which studio you are viewing when these sources are displayed as a part of a much bigger list. Even a small Livewire network may have hundreds of sources.

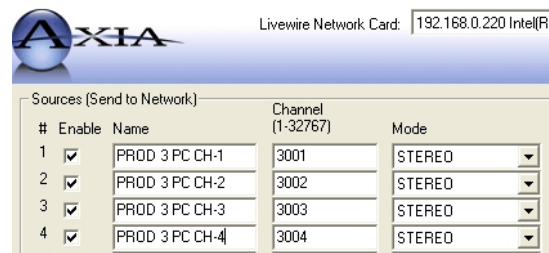


Figure 2-9: Specifying Sources

The IP-Audio driver setup includes an auto-fill feature to save you some typing and avoid some channel numbering errors. For multiple sources (OEM version),

you can fill in the Livewire channel number for the first source and automatically fill in the remaining channels with consecutive channel numbers by simply pressing the F2 key. WHichever method you choose - make sure the Livewire channel numbers are unique. No duplication is permitted.

With all this talk about Livewire Channels, you may wonder exactly how this translates to networking terminology.

---

**DEEP TECH NOTE:** Livewire channel is an abstraction simplifying user setup. Each channel number is translated to a multicast addresses for internal use (239.192.0.0/13 range for Stereo Audio, 239.194.0.0/13 for Surround). The user need only assign a Livewire Channel. For more on this see the document ***Introduction to Livewire***.

---

By default, the source of the IP-Audio driver is configured to send audio to Livewire channels 1 (1 though 24 for the Multi-Channel driver). For simple applications those settings do not have to be changed unless you have another device assigned with those channel numbers. Of course, if you have more than one IP-Audio driver (using default settings), you will instantly have a channel conflict. For this reason, you should consider changing the channel numbers to a unique higher range.

In a very basic application consisting of only a node and a PC, one or more ports of a Livewire Analog or AES 8x8 node can be connected to the PC as an audio input and output devices. For the Multi-Channel driver all 8 output ports can be configured to receive audio on channels 1 through 8. Individual inputs and outputs can be assigned non-contiguous channels using the Axia node's browser interface. See ***Introduction to Livewire*** for more on channel assignments.

### Destinations

The IP-Audio destinations are the Livewire streams that will be delivered to your Windows record device(s). The standard IP-Audio driver has one destination while the OEM version can have as many as 24 destinations. We recommend that you specify only those destinations

that you actually intend to use.

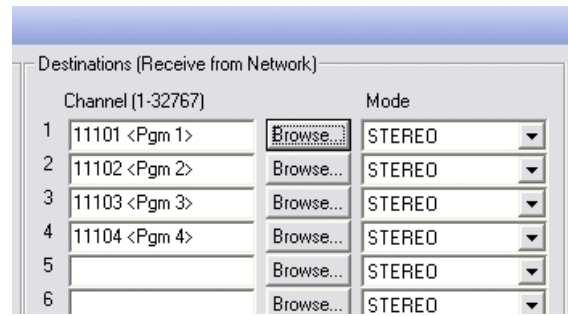


Figure 2-10: Destination Selection

You can select destinations by directly typing the Livewire channel number or by using the Browse button which opens Destination window shown in Figure 2-11.

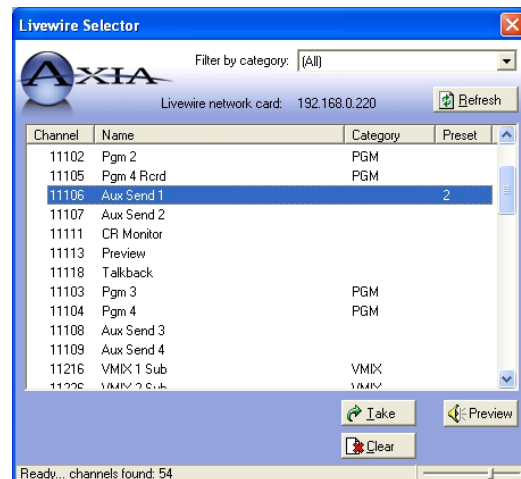


Figure 2-11: Destination Browse Window

### GPIO – (OEM Version Only)

This button allows configuration of the GPIO module that allows sending and receiving commands to control the console and fire events on the PC. If you will be interfacing software on this computer to be remotely controlled by an Axia audio mixer control surface (such as the Element) you will need to configure GPIO. For more on this see Chapter 3.

### Statistics

This button opens a window with packet counters and other information that may be used to verify that

Livewire driver works properly. This is described more fully in Appendix C of this manual. There is nothing to configure in this window as it is a viewer used for diagnostic and troubleshooting purposes.

### PC Audio Nominal Level

The slide fader adjustment is a global setting for both input and output levels. It is used only as a one time trim setting and denotes the NOMINAL Operating Level of the audio into and out of the PC. Axia Devices operate at a NOMINAL -20dBFS level. The default setting of -8dBFS implies that the audio from the PC is very high. This would increase the input audio by 12dBFS to the PC and decrease the output of the PC audio to the Axia network by 12dBFS in order to obtain the -20dBFS nominal level. If your PC audio out appears too low on nodes or on a console, slide the fader LEFT to change the nominal level more towards the actual nominal level of the Axia network.

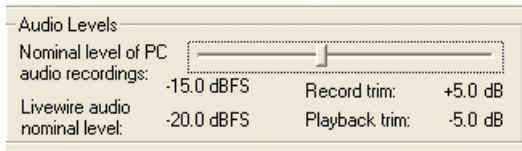


Figure 2-12: Nominal Level Setting

For adjusting input and output levels individually to the PC application use the Windows Volume controls/Mixer options.

### Keep Channel Active

This option is to eliminate truncating of audio from certain playback applications that could result in pops and clicks of the audio when monitored. This will keep the stream from the PC actively sending a valid stream even when the playback device is NOT playing, hence it will send a stream that is silent audio. It is typically enabled for PCs running an On-Air automation system.

### Windows XP Firewall Settings

Installation to Windows XP may require some added firewall settings to work properly even if the firewall is DISABLED. XP still does filtering of ports as a security measure. The current version of the IP-Audio driver

will automatically enter the required Firewall settings in XP. If you need to manually enter exceptions to ports, or verify the installation of the exceptions, the list of port requirements are below:

Navigate to My Network Places and view the available network connections. Select the NIC interface used by the Livewire Driver and enter the PROPERTIES of the NIC. Go to the ADVANCED tab and FIREWALL Settings of the NIC and go to the EXCEPTIONS tab:

The following ports will need to be exempted from being blocked:

Livewire Advertisement	4001	UDP
Livewire Advertisement Req	4000	UDP
Livewire Clock	7000	UDP
Livewire Source Allocation	2060	UDP
Audio Streams	5004	UDP

A more complete listing of ports used by Livewire can be seen in Appendix A.

### Using and Configuring The Axia IP-Audio Driver as a Playback Device

Configure your player application to use Axia Wave 01 through 24. Most playback applications use a drop down menu to list all available audio output devices; simply choose Axia Wave01 from the list. If you are using Axia IP-Audio OEM Multichannel driver, you can configure up to 24 players to simultaneously stream audio to the Livewire network.

**Note:** Standard version allows only one input and one output audio device.

You may also want to specify one of Livewire Axia IP-Audio Output devices as the “preferred device” in Windows Sound and Multimedia Properties control panel, as shown in Figure 2-13.

Once you start playback on a selected Axia IP-Audio Output device, the driver will start streaming network packets to one of the 32767 Livewire channels, as configured above. Livewire nodes configured to receive audio on that channel will receive the audio data. If the lowest

(bottom) segment of the output audio meters are illuminated this indicates that the stream is present, even if the audio source is currently silent. For more on Livewire Channels see the document *Introduction to Livewire*.

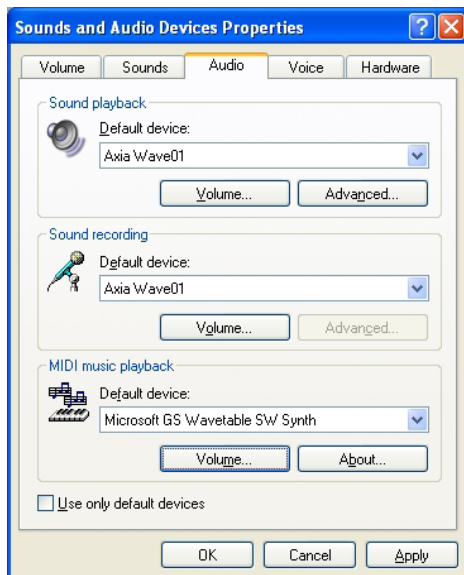


Figure 2-13: Windows Audio Devices

## Mixing Multiple Sources to an IP-Audio Playback Device

Windows provides for mixing multiple sources to a specified playback channel when more than one application uses the same Audio Device. In a similar manner, the Livewire driver Line Input can be mixed to the Livewire driver Line Output to allow monitoring of recorded audio. Standard tools can be used to control the mixing, such as the Windows™ mixer application. See Setting Recording and Playback Levels in Chapter 3 for additional information.

## Using IP-Audio for Multi-Channel Recording and Playback

Numerous multi-channel audio applications can be used with Axia's IP-Audio driver. One example is the Telos iProfiler that can be used for simultaneous logging up to 24 different stereo audio channels.

Setup is not difficult and most applications that use “standard” Windows multi-channel sound cards can also

use the Axia multi-channel IP-Audio driver. Make sure that your player application is configured to use Axia IP-Audio Input devices 01 through 24. Many recording applications use a drop down menu to list all available audio output devices for both recording and playback; simply choose from the appropriate list.

You may also wish to specify one of the Axia IP-Audio Input Devices as the preferred “Sound Recording” device in Windows Sound and Multimedia Properties control panel, as shown above. This specifies the Axia IP-Audio Input device for recording on applications that do not allow input device selection.

If you have not already done so, configure the IP-Audio Destination Channel with the desired Livewire channel number from which the Audio Input device will receive audio. Remember, a Livewire Destination is the “destination” to which an IP audio stream is directed. This will correspond to a record device in your audio software. To do so, click on the Browse button and a Livewire Selector window will appear as shown below, allowing you to choose a source stream.

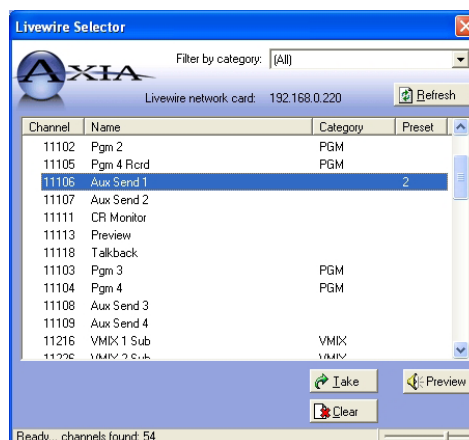


Figure 2-14: Destination Browse Window

When you start recording, the Axia IP-audio driver will pick up the designated Livewire stream from the audio network.



# Chapter Three:

## Advanced Features

This section contains information on the GPIO function, Livewire performance statistics, using standard players to listen to Livewire streams, and how the driver interacts with standard Windows™ devices.

### The IP-Audio Driver GPIO Bridge

Clicking on the GPIO button from the Axia IP-Audio Configuration screen permits configuration of the GPIO module. (Only available on the OEM Multi-Channel Driver) This allows remote devices such as the SmartSurface or Element consoles to send commands to applications on the Windows™ computer running the Axia IP-Audio Driver. This is a one-way communication from Axia consoles to the IP-Audio driver. If you will be interfacing software to be remotely controlled by an Axia audio mixer control surface you will need to configure this option.

#### Starting the GPIO Module

Pressing the GPIO button will display the window shown in Figure 3-1. To use a profile from your software vendor, enter the location of the file in the appropriate field. If you do not have one, we have some sample profiles archived on our FTP site. Contact our Axia tech support team if you need help locating or downloading these files. Text examples of some these files are also found in Appendix B of this manual.

To activate the GPIO component place a check mark to **Enable** the **GPIO Control Surface Bridge** application. Leaving this option enabled will start this program automatically as a service whenever Windows starts. Clicking the **Close** button will close the Configuration Window however the GPIO module will remain running.

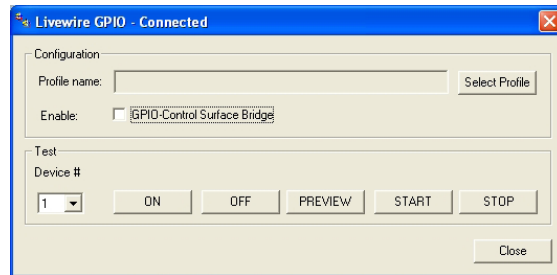


Figure 3-1: GPIO Service Configuration

#### Loading a GPIO Profile

If the manufacturer of your PC software is an Axia partner they will have provided you with a GPIO profile to configure the GPIO module for you. To load a profile:

1. Click on the **Select Profile** button. A standard Windows Open file dialog will be displayed.
2. Using the usual Windows procedures locate the GPIOVK file (on the manufacturer's CD-ROM for example) then click on **Open**.

For more information regarding the GPIO Bridge service see the Appendix B at the end of this manual.

### Livewire Statistics

This section provides some basic information on the Statistics section of the Windows IP-Audio driver. If you are encountering problems, please refer to **Appendix C: Troubleshooting Tools** for more detailed information.

Playback applications send audio data to the Axia IP-Audio driver. The driver packages the audio into Livewire audio packets and sends them out the selected network interface to the Livewire LAN. The Livewire Statistics window provides counters for all Livewire Output and Input devices.

To assure reliable low-delay audio delivery, Livewire provides synchronization between terminals. The Axia IP-Audio Driver receives clock information from hardware nodes on the Axia Livewire LAN (you must have at least one hardware node on the network). To make sure

the Axia IP-Audio driver is on-line and receiving this synchronization (e.g. the driver's PLL is locked) check that the Synchronization Lost value is not incrementing in the Livewire Statistics window. When the clock signal is not available, this value will *increase* and the Packets Received value will *stop increasing*.

Additional information is made available to you in order to troubleshoot the connection of the IP-Audio-driver to the network. Jitter value must be below 25 ms. Higher values of Jitter can indicate problems with the hardware or the driver.

**TxPkts** - Transmitted Packets to the Axia Network. When local audio is streamed to the network TxPkts counter will increment at 200pkts/s rate.

**TxErr** - Transmitted Errors to the Axia Network. TxErr counter should never increment under normal conditions.

**RxPkts** - Received Packets from the Axia Network. RxPkts shows receive packet count. It should be 200pkts/s when receiving Standard Streams, and 4000pkts/s when receiving Livestreams. Note the difference in packet rate. Some systems may have performance problems with receiving Livestreams.

**RxBytes** - Received Bytes from the Axia Network.

**RxUr** - Received UnderRuns from the Axia Network. Notes malformed packets received.

**RxOr** - Received Overruns from the Axia Network. Notes malformed packets received.

**RxSeqErr** - Received Sequenced Errors from the Axia Network. Every time a Livewire stream starts to receive a new Livewire channel, a new packet sequence will be coming into the receiver. You will see RxSeqErr incremented once for each new stream. It should not increment while receiving a constant stream.

It is normal to see an occasional error in the Errors

columns, after all we are using Windows, but excessive errors in this field indicates a serious problem. Examine your networking cabling and consider a problem with your NIC card. Remember that not all NIC cards are created equal.

Axia Audio will routinely test commonly-used NIC cards and we will do our best to let you know when we have good or bad news to report about specific cards. Please visit our web site at [www.axiaaudio.com](http://www.axiaaudio.com) and check out Support section for current tech notes on this topic.

## Using Axia Driver With The Standard Windows™ Multimedia Interface

Livewire sources (e.g. audio sources on the Livewire Network) are mapped to Sound Playback devices. To Windows, the Axia IP-Audio driver appears just like any other Sound Playback Device. Playback applications use these devices. Your audio software may have a way of selecting a Playback Device or it may simply use the Preferred Device selected in the Windows Sounds and Multimedia Properties control panel.

Livewire destinations (e.g. audio software capable of accepting audio on the Livewire-enabled computer) are mapped to Sound Recording devices. Various recording, audio editing and logging applications use those devices to determine the audio source to be recorded. Again, your software may have a way of selecting a Sound Recording Device or it may simply use the Preferred Device selected in the Windows Sounds and Multimedia Properties control panel.

### Setting Sources and Destinations for Software Applications

Often playback or recording software have their own custom dialog used to assign playback and recording devices. If they do not, the default settings for Sound Playback and Sound Recording devices can be configured using the Windows Sounds and Multimedia Properties Control Panel.

## Setting Recording and Playback Levels

Windows also offers tools for setting Playback and Record levels. To adjust these audio levels the, standard Windows Mixer application can be used, see below. Note that the Windows multimedia mixer can display either Record or Playback device at a given time. To change between these two functions choose options/properties to select either the Playback or Recording function. To make adjustments to the volume for the option you have chosen, you will see level controls similar to those as shown here in Figure 3-2.

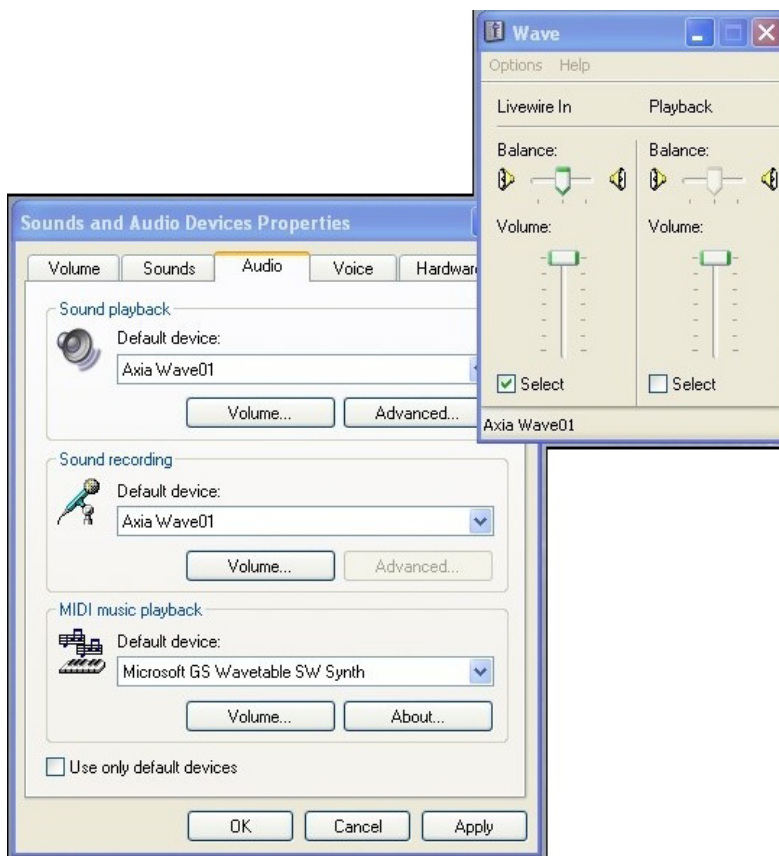


Figure 3-2: Windows Record Level

Some software chooses not to support the standard Windows Multimedia interface. Unfortunately, you won't be able to use Livewire IP-Audio with this software unless it is modified. Please contact your software vendor.

# Appendix A: Livewire Ports

---

Standard ports are used for specific functions within the Livewire system. Here is a summary of the ports and their use. If you have installed an Axia application with your Windows firewall enabled, the applications and ports will be automatically added to your Windows firewall exceptions. If your installation has been performed with the Windows firewall disabled and you have subsequently enable the firewall, you may have to perform some tweaking of the firewall to get things operating normally. This information will be helpful in that case or in the event that you are using some other firewall.

Port	Protocol	Use
67,68	UDP	BOOTP Server/Client - Remote IP address assignment
93	TCP	Livewire Routing Protocol
123	UDP	NTP
514	UDP	Syslog logging
2055	UDP	Multicast based GPIO (CMsg2 protocol) (mcast on 239.192.255.4) - GPIO commands from GPIO node to console-type endpoint
2060	UDP	Multicast based GPIO (CMsg2 protocol) (mcast on 239.192.255.4) - GPIO commands from console-type endpoint to GPIO node
4000	UDP	Livewire Advertisement and Source Allocation Protocol - Full info advertisement requests and source allocation requests
4001	UDP	Livewire Advertisement and Source Allocation Protocol (mcast on 239.192.255.3) - Periodic announcements and full info advertisements - Source allocation state announcements and responses
4002	UDP	Engine supervision protocol (mcast on 239.192.255.3)
4010	TCP	Livewire Control Protocol
4011	UDP	LWCP for Accessory Modules (from module to console) (mcast on 239.192.255.4)
4012	UDP	LWCP for Accessory Modules (from console to module) (mcast on 239.192.255.4)
5004	UDP	RTP Livewire Audio
7000	UDP	Livewire Clock
9997	TCP	Protocol logging (for debug) in Element

# Appendix B: Windows and GPIO Interfaces

## GPIO Bridge Service

### What is the Windows GPIO Bridge Service?

The Axia GPIO hardware unit is a device that provides a number of physical GPIO ports, each with 5 input and 5 output pins for sending and receiving commands. The OEM Windows driver provides a separate GPIO port associated with each audio device hence it supports up to 24 ports. When the Axia Driver is used with Smart Surface or Element, the meaning of the GPO pins is defined in the control surface manuals, in the GPIO section. Automation systems should use “Line Source” type or “Computer Player” type.

Virtual GPO pins can control the automation system via the GPIO Bridge Service. The GPIO Bridge Service is a one way communication service designed to provide a simple user interface to accept the incoming logic commands from the Axia network to help to translate commands to customer’s or OEM PC applications in order to initiate events on the PC.

The Bridge service is one-way and thus cannot send messages from the PC back to the Axia Network.

### GPIO Profile File Format

The following technical information on sending Windows and UDP message is included for OEM’s developing GPIO Profiles for interfacing to their software packages.

#### Profile File Format

These are ASCII test files.

#### Windows Class Name

If you wish to send Windows™ messages the profile must contain a line that determines Window Class Name of the automation system main window. See WINCLASS= in example below.

### Sending Windows Messages

The file must contain a line that determines Windows Class Name of the automation system main window. See WINCLASS= in example below.

Every event is represented by string in the format DEV<d>.<e>, where <d> is audio device number (0-24), and <e> is event number (0-4) as specified in example below.

Example:

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

```
WINCLASS=TPlayerMainForm  
DEV0.3 MSG=0x40A WPARAM=0 LPARAM=1  
DEV0.4 MSG=0x40A WPARAM=0 LPARAM=0  
DEV1.3 MSG=0x40A WPARAM=1 LPARAM=1  
DEV1.4 MSG=0x40A WPARAM=1 LPARAM=0  
[...]
```

### Sending UDP messages

The file must contain a line that determines the destination UDP port (UDP\_DSTPORT=<port>).

Every event is represented by string in the format DEV<d>.<e>, where <d> is audio device number (0-24), and <e> is event number (0-4)

An example of a simple GPIO Profile definition file is shown below.

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

UDP_DSTPORT=2002

DEV0.3 COMMAND="<DADCMD><ID>99999</ID><COMMAND>play pbk1</COMMAND></DADCMD>"

DEV0.4 COMMAND="<DADCMD><ID>99999</ID><COMMAND>stop pbk1</COMMAND></
DADCMD>"<DADCMD><ID>99999</ID><COMMAND>next pbk1</COMMAND></DADCMD>"
```

## Windows GPIO/TCP-IP Direct Interface

The Axia GPIO hardware unit is a device that provides a number of physical GPIO ports with 5 input and 5 output pins each for sending and receiving commands. The OEM Windows driver provides a separate GPIO port associated with each audio device hence it supports up to 24 ports.

When the Axia Driver is used with Smart Surface or Element, the meaning of the GPI and GPO pins is defined in the control surface manuals, in the GPIO section. Automation systems should use “Line Source” type or “Computer Player” type.

Virtual GPO pins control the automation system. Automation system can use virtual GPI pins to deliver GPIO events to Axia network.

Note that the GPIO Bridge Service only listens for the incoming messages and is not bi-directional. If the Windows application can connect directly to port 93 of the local PC then bi-directional communication can be available and you don’t have to use the GPIO Bridge service.

### Connection

Client application opens a TCP/IP connection to the Axia server running locally on the PC and NOT to the Axia console. That server accepts connections on port 93.

### Commands

ADD GPO - Enables indications of GPO pin changes. Those are control messages for the automation system.

ADD GPI - Enables indications of GPI pin changes. Those are control messages sent from the automation system. This is useful when multiple clients control one GPI port.

GPI <state> - Automation system emulates input pin state changes. A GPIO event will be sent to Axia network. Note: client must send LOGIN command before issuing GPI.

LOGIN - Enables commands that change state of the device from client. Required before GPI.

BEGIN .. END - Define a block of commands that are sent at the same time. It is optional. Client can use them to optimize user interface updates for example,

by delaying refresh operation after receiving BEGIN, until END is received.

State of GPIO port is encoded using character string containing: 'l', 'L', 'h', 'H', 'x' letters. The meaning of those symbols is the following:

- l steady low state
- L pin changed state from high to low
- h steady high state
- H pin changed state from low to high
- x can be used in GPI commands where pin state is not supposed to change

GPIO contact closures are active low. This implies that application typically would react to state changes from high to low – symbol 'L'.

As the GPIO has 5 pins the state of each pin is expressed by two characters and the case of the character. In the example shown in Figure B-1, several GPO status changes are illustrated. The lower case letters indicate the status of the pins. The upper case letters indicate the updated status of items that have changed.

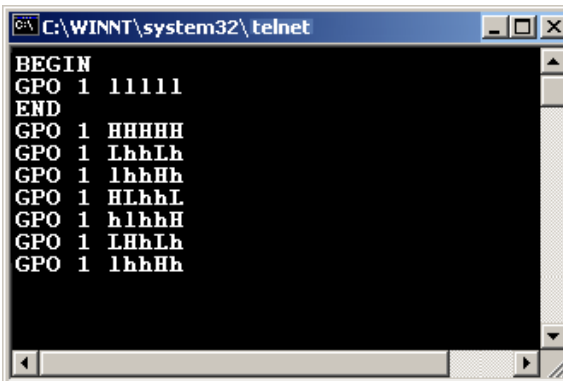


Figure B-1: GPO Status Change Example

The specific case in Figure B-2 shows Pin 1 of GPO port 1 changing from state of steady LOW to HIGH:

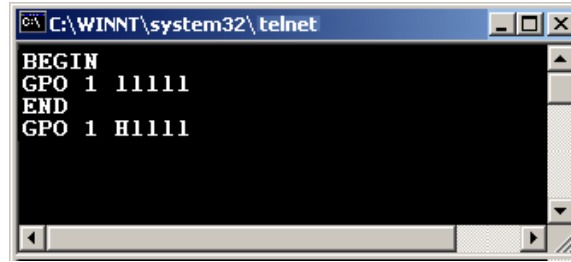


Figure B-2: GPO Pin 1 Change

# Appendix C: Troubleshooting Tools

## Livewire Statistics Window

The Livewire Statistics window can be used to view input/output streaming statistics. This window is opened from the Control Panel >> Axia IP-Audio configuration window and then clicking on the Statistics button.

Playback applications send audio data to the Axia IP-Audio driver. The driver packages the audio into Livewire audio packets and sends them out the selected network interface to the Livewire LAN. The Livewire Statistics window provides counters for all Livewire Output and Input devices.

*Note: Wireless network adaptors do NOT provide sufficient performance to be used with the IP-Audio driver. If you have multiple NICs in your PC, make sure that the connection to your Axia LAN is a hard-wired connection.*

### Clock Sync Information:

Synchronization is an essential part of Livewire system. A clock signal is distributed to all devices in the net-

work, including the Windows Driver. The Axia IP-Audio Driver receives clock information from hardware nodes on the Axia Livewire LAN (you must have at least one Hardware node on the network). Problems with Livewire synchronization will cause audio dropouts.

The **Packets Received** counter (RxPkts) should increment continuously. If it does not, it means one or more of the following:

- There is either no clock master on the network (all Axia Nodes are set as slaves-only)
- A firewall is stopping clock stream (UDP port 7000 must be opened)
- Multicast forwarding is not working properly in your in the network segment

The **Synchronization lost** counter should only increment when the system starts, when the clock master changes, or some other network related changes are made. If it increments more frequently, it may indicate some problems with the PC system clock or other network problems.

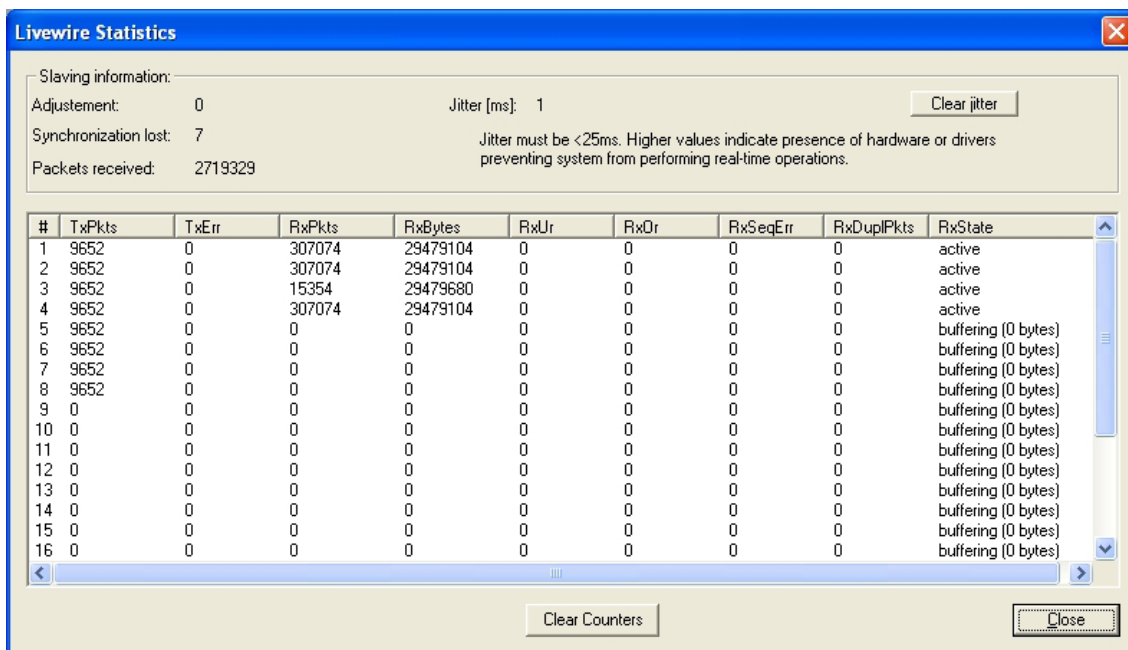


Figure C-1: IP-Audio STatics Window



The **Jitter** value should be below 25 msec. Higher values indicate inadequate performance caused by hardware or limitations of your system.

If Jitter is high, you can reduce the number of source and destination streams in order to minimize the load on your PC. Another tip is to convert any destination Livestreams to Standard Streams by using VMODE in your Mix Engine. Remember a Livestream is 20 times more packets per second than a Standard Stream. It is also important to investigate potential hardware problems. As a first indicator - check the Task Manager to ensure CPU utilization and Network Activity are in normal ranges. See the sections that follows on the Windows Performance Monitor for more Windows troubleshooting tips.

**Transmit (to the network) information:**

When local audio is streamed to the network, the **TxPkts** counter will increment at 200pkts/s rate.

The **TxErr** counter should never increment. If it does, there are some serious performance problems with Windows system or network hardware or software stack. Check to ensure your network card has 100Mbit full-duplex connection established to the Ethernet switch.

**Receive (from the network) information:**

**RxPkts** shows receive packet count. It should be 200pkts/s when receiving Standard Streams, and 4000pkts/s when receiving Livestreams. Note the difference in packet rate. Some systems may have performance problems with receiving Livestreams. If this is the case with your system, a possible workaround is to utilize VMODE in your Studio Mix Engine to convert such streams from Livestream to Standard Stream.

If there are no packets being received and the receiver is configured properly, there may be a firewall on the PC that is blocking the audio stream. In that case, UDP port 5004 should be opened.

A new Livewire stream starts every time a receiver is set to receive a Livewire channel and a new packet sequence is coming into the receiver. Under normal con-

ditions, you will see **RxSeqErr** incremented once.

The **RxSeqErr** counter should never increment while receiving continuous, unchanging streams. If it does, it means that there is a problem with the network and packets are being dropped. This can be a bad Ethernet cable connected to the transmitting device, the receiver PC or anywhere else in the chain. This condition can also be caused by a faulty or inadequate network card. Check to ensure your network card has a 100Mbit full-duplex connection established to the Ethernet switch and that it not one of the cards on our black list.

Dropped packets can be also caused by performance problems originating from bottlenecks in the PC motherboard and network card hardware or software. It is always good to check the CPU utilization to make sure the PC is running at a reasonable performance level.

The **RxDuplPkts** (Receive Duplicate Packets) counter shows duplicate packets which typically occur under the following conditions:

- Duplicate Livewire channel numbers. This condition can be eliminated by examining the device configurations and channel scheme used for nodes and other IP drivers. iProbe is of assistance to track this down. A duplicate channel may result in recording silence, noise or intermittent audio.
- Redundant network paths. This can occur in larger networks where there may be redundant links between routers. More than one copy of Livewire packets may be delivered continuously or periodically. We recommend using PIM (Protocol Independent Multicasting) Sparse Mode rather than Dense Mode to avoid periodic flooding.

The next section discusses Windows performance issues in more detail.

## Windows Performance Monitor

Additional troubleshooting information can be obtained using the Windows Performance Monitor. This is a standard Windows tool, which allows reading statistical information from the system and drivers.

You can access the Performance Monitor through the Windows Control Panel. Navigate to: Control Panel >>Administrative Tools>>Performance. You will probably see the performance monitor graph similar to the one shown in Figure C-2. There are three graphs being continuously plotted.

While this may be interesting, it is not all that useful to us since we are not viewing any parameters of interest. The performance monitor also has the ability to add other parameters, including network adaptor performance and other Axia processes. We can add these items and display them as well.

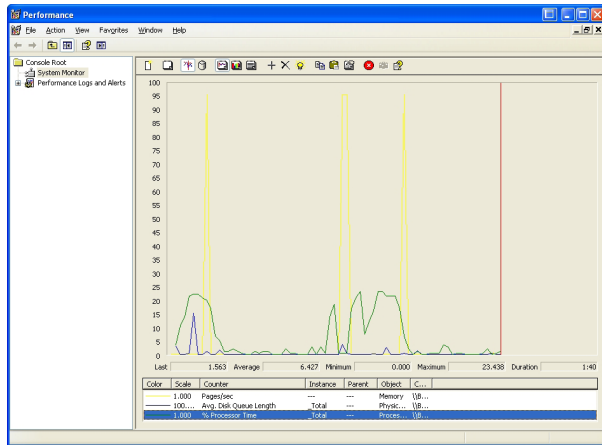


Figure C-2: Windows Performance Monitor - Default

As a useful example, here is how you could add some items related to NIC performance. Place your mouse cursor within the graph window and right click to bring up some options. Select ADD COUNTERS and select all the Error counters related to Ethernet interface being used by Axia driver.

You can use the Graph or Report view to examine the data. You should see the new error counters corresponding to the network interface configuration. In a similar manner you can view Axia Driver parameters.

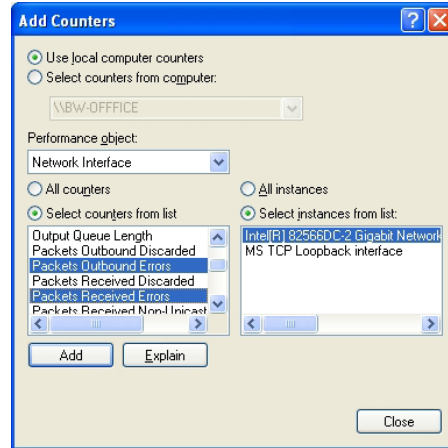


Figure C-3: Adding Counters

In this example, if there are indications of any packets with errors or discarded packets, it may indicate some problems with the network card or network cable.

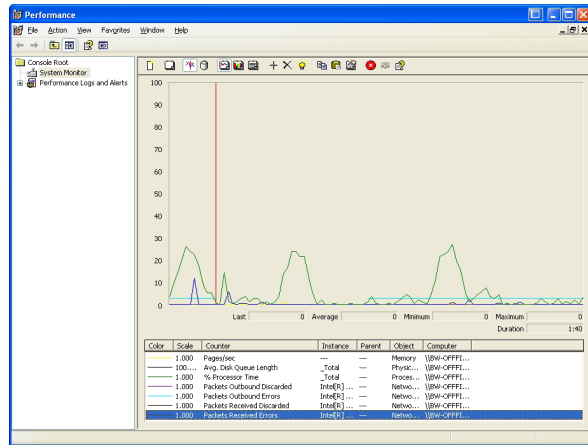


Figure C-4: Performance Monitor - Additional Counters

When troubleshooting network problems, you should also check the port statistics on the Ethernet switch for the port to which your PC is connected. Ensure that the port is not flooded with data from some other sources.

# Appendix D: Warranty

---

## Axia Audio Limited Warranty

This Warranty covers “the Products,” which are defined as the various audio equipment, parts, software and accessories manufactured, sold and/or distributed by TLS Corp., d/b/a Axia Audio (hereinafter “Axia Audio”).

With the exception of software-only items, the Products are warranted to be free from defects in material and workmanship for a period of five (5) years from the date of receipt by the end-user. Software-only items are warranted to be free from defects in material and workmanship for a period of 90 days from the date of receipt by the end-user.

This warranty is void if the Product is subject to Acts of God, including (without limitation) lightning; improper installation or misuse, including (without limitation) the failure to use telephone and power line surge protection devices; accident; neglect or damage.

EXCEPT FOR THE ABOVE-STATED WARRANTY, AXIA AUDIO MAKES NO WARRANTIES, EXPRESS OR IMPLIED (INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE).

In no event will Axia Audio, its employees, agents or authorized dealers be liable for incidental or consequential damages, or for loss, damage, or expense directly or indirectly arising from the use of any Product or the inability to use any Product either separately or in combination with other equipment or materials, or from any other cause.

In order to invoke this Warranty, notice of a warranty claim must be received by Axia Audio within the above-stated warranty period and warranty coverage must be authorized by Axia Audio. If Axia Audio authorizes the performance of warranty service, the defective Product must be delivered, shipping prepaid, to: Axia Audio, 2101 Superior Avenue, Cleveland, Ohio, USA 44114.

Axia Audio at its option will either repair or replace the Product and such action shall be the full extent of Axia Audio’s obligation under this Warranty. After the Product is repaired or replaced, Axia Audio will return it to the party that sent the Product and Axia Audio will pay for the cost of shipping.

Axia Audio’s authorized dealers are not authorized to assume for Axia Audio any additional obligations or liabilities in connection with the dealers’ sale of the Products.

Axia Audio’s products are to be used with registered protective interface devices which satisfy regulatory requirements in their country of use.



## Controlling AudioVAULT using Axia GPIO IP interface

Version: 0.9.1, 21 February 2007  
By Maciej Szlapka, Axia Audio

### Axia GPIO signals:

The Axia IP-Audio Driver provides GPIO interface over TCP (port 93). The driver handles routing of the control to make sure the GPIO follows fader/channel assignment on the control surface.

All commands correspond to signals available on the DB connectors on Axia GPIO Nodes. Each port has 5 input pins and 5 output pins. Function of those pins is determined by source profile defined in Element Control Surface configuration.

For computer playback devices "GPIO Line Input Logic" is used most commonly. This profile defines the following function:

#### INPUTS

ON Command	GPI pin 1	Turns channel ON
OFF Command	GPI pin 2	Turns channel OFF & sends 100 msec STOP pulse
PREVIEW Command	GPI pin 3	Turns preview ON
RESET Command	GPI pin 4	Turns channel OFF, does not send a STOP pulse
READY Command	GPI pin 5	Illuminates OFF lamp to indicate source's readiness

#### OUTPUTS

ON Lamp	GPO pin 1	Illuminates when channel is ON
OFF Lamp	GPO pin 2	Illuminates when channel is OFF and READY is active
PREVIEW Lamp	GPO pin 3	Illuminates when PREVIEW is ON
START Pulse	GPO pin 4	Sends 100 msec pulse when the channel status changes from OFF to ON
STOP Pulse	GPO pin 5	Sends 100 msec pulse when the channel status changes from ON to OFF



## **Audio Communication Setup:**

AudioVAULT needs to open a connection to the Axia IP-Audio Driver. A section like the following can be added in the audiovau.ini file for the application that will be sending the commands:

```
[REMOTECONTROL.UDP1]
DeviceType=Socket
Protocol=TCP
SocketType=caller
HostName=127.0.0.1
ServiceName=93
```

If the AudioVAULT application (e.g. AVAir or Audiovau.exe) runs on a different machine than the IP driver, you need to change "127.0.0.1" to the IP address of the machine where the driver is installed.

## **Audio VAULT Events:**

Please refer to Audio VAULT Programming Manual for details on using remote control interface.

We are going to use the following commands from an [AVAir] section of the audiovau.ini file.

```
IDD_StartN - Events and input control associated with start of deck N
IDD_StopN - Events associated with loading and unloading the deck and playback stop control
```

Each command can contain multiple mappings, which are separated with a semicolon. Command mappings take format of <REMOTECONTROL\_NAME>:[<prefix>]<text>, where REMOTECONTROL\_NAME is a name specified in audiovau.ini section (e.g. "UDP1"), text is the actual string sent over TCP connection, prefix defines type of the command.

Specification of commands sent by Audio Vault differ from the external triggers by a prefix which is either "+" or "-". The "+" tells AV to send a command when an event occurs (like Channel ON), while "-" tells AV send a command when the opposite action happens (e.g. Stop).

Comma separates the function lamp indicator from the function circuit board input assignment.

Some commands tend to be very long. Make sure that single line does not exceed 250 characters. You may consider using shorter REMOTECONTROL identifiers.

```
IDD_StartN=,<Start control><Channel ON/OFF Event>
IDD_StopN=<Deck Loaded Event>,<Stop control>;<Deck Unloaded Event>
```

<Start control> specifies the external GPIO command, which triggers start on the deck. In Axia 5-pin gpio convention, it is Pin 1 going Low, i.e. "GPO N L".

<Stop control> specifies external GPIO command which triggers stop on the deck. In Axia 5-pin gpio convention it is Pin 1 going High, i.e. "GPO N H".



<Deck Loaded Event> is a command, which is sent out by Audio Vault once deck is loaded. To make it light up the OFF button on the control surface, we need to send READY (Pin 5 Low): "GPI N xxxxL". This command needs to be sent with "+" prefix.

<Deck Unloaded Event> is a command, which is sent out by Audio Vault once deck is cleared. To make the OFF light go off on the control surface, we need to send READY (Pin 5 High): "GPI N xxxxH". This command needs to be sent with "-" prefix.

<Channel ON/OFF Event> defines commands to be sent when channel changes state fo ON or OFF. The ON command will be indicated by the "+" prefix: "+GPI N LH", while the OFF command is indicated by "-" prefix: "-GPI N HL".

Axia protocol requires LOGIN before sending to it any commands changing pin state. We are going to add "LOGIN\r\n" to every command sent by Audio Vault. Technically LOGIN is required only once, but it can be as well sent with each command.

In order to activate GPIO indications, Audio Vault must send a subscription message "ADD GPO". It can be done as a part of <Deck Loaded> event.

The following example demonstrates full control implemented for Decks 1 and 2. Each deck uses a different GPIO port. In Axia setup, GPIO ports are associated with the Livewire channel corresponding to the Audio Source (Wave Out Device) with the same index. In other words "GPI 1" will control "Axia Wave01" playback device. It is assumed that Audio Vault is using "Axia Wave01" for Deck 1, "Axia Wave02" for Deck 2, etc.

### Examples

```
IDD_Start1=UDP1:+LOGIN\r\nGPI 1 LH\r\n,UDP1:GPO 1 L;UDP1:-LOGIN\r\nGPI 1 HL\r\n;  
IDD_Stop1=UDP1:+ADD GPO\r\nLOGIN\r\nGPI 1 xxxxL\r\n,UDP1:GPO 1 H;UDP1:-  
LOGIN\r\nGPI 1 xxxxH\r\n;  
  
IDD_Start2=UDP1:+LOGIN\r\nGPI 2 LH\r\n,UDP1:GPO 2 L;UDP1:-LOGIN\r\nGPI 2 HL\r\n;  
IDD_Stop2=UDP1:+ADD GPO\r\nLOGIN\r\nGPI 2 xxxxL\r\n,UDP1:GPO 2 H;UDP1:-  
LOGIN\r\nGPI 2 xxxxH\r\n;
```

Examples above use READY signal. You need to make sure that this function is enabled in "Source Profile" on your Element:

## Element Configuration

Source profile

Primary source:

Source profile name:

Source availability:  Channel 1  
 Channel 2

GPIO Ready enabled/disabled:  Disabled  
 Enabled

### Extra outputs

In order to control extra GPIO outputs from Audio Vault on an external GPIO unit, you need to specify additional TCP connection parameters:

```
[REMOTECONTROL.AXIAGPIOAUX]
DeviceType=Socket
Protocol=TCP
SocketType=caller
HostName=192.168.2.210
ServiceName=93
```

In this example, 192.168.2.210 is an IP address of the external GPIO unit.

We are going to extend the previous example with control of output pins on port 1 on AXIAGPIOAUX. Pin 1 will go Low when channel is turned ON, Pin 2 will go Low when channel is turned OFF.

Example of controlling multiple outputs from single Audio Vault event:

```
IDD_Start1= UDP1:+LOGIN\r\nGPI 1 LH\r\n,UDP1:GPO 1 L;UDP1:-LOGIN\r\nGPI 1
HL\r\n;AXIAGPIOAUX:+LOGIN\r\nGPO 1 LH\r\n;AXIAGPIOAUX:-LOGIN\r\nGPO 1
HL\r\n;
```

### Multiple buttons per function

Sometimes Audio VAULT Deck needs to be activated by two different GPIO signals, e.g. regular ON/OFF control from Element Control Surface and external RIOT box. This can be accomplished by separating multiple assignments with a semicolon.

Example:

```
IDD_Start1=,UDP1:GPO 1 L;UDP1:+LOGIN\r\nGPI 1 LH\r\n;UDP1:-LOGIN\r\nGPI 1
HL\r\n;,TCP43:IN0.1+;TCP43:+OUT0.1+OUT0.2-;TCP43:-OUT0.1-OUT0.2+;
```



```
IDD_Stop1=UDP1:+ADD GPO\r\nLOGIN\r\nGPI 1 xxxxL\r\n,UDP1:GPO 1 H;UDP1:-  
LOGIN\r\nGPI 1 xxxxH\r\n;,TCP43:IN0.2+;TCP43:+OUT0.2+;TCP43:-OUT0.2-;
```

In this example, we assume TCP43 is a connection to BE RIOT GPIO device.





## Controlling ENCO DAD using Axia GPIO IP interface

Version:1.0, 26 June, 2006  
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 301)*
SRC 8 ADDR:"239.192.0.105"   (sets source 1 to Axia channel 105)
```

*\*NOTE: You must understand how Axia uses multicast addresses to be able to compute the addresses to include in these commands. See Axia Document: Livewire Routing Protocol for detailed command information.*

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 LXXXXX'
```

and to open it:

```
SEND TEXT A 'GPO 1 HXXXXX'
```

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.

# Controlling RCS/Prophet NexGen using Axia GPIO IP interface

Version: 0.9.1, 29 February 2008  
 By Maciej Szlapka, Axia Audio

This document describes how to configure GPIO remote control between Axia Consoles and RCS/Prophet NexGen. This configuration uses IP network and replaces need for hardware GPIO card in the PC as well as hardware wiring.

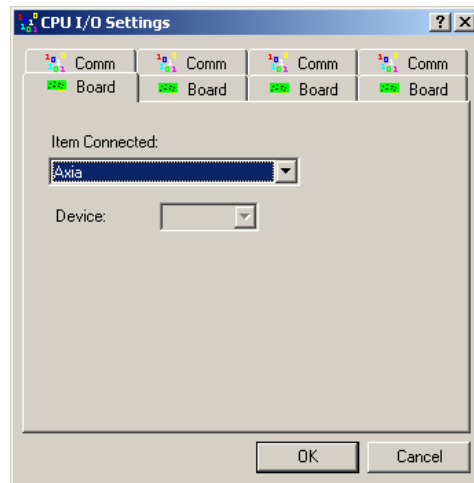
NexGen is aware of Axia GPIO interface and provides user-friendly configuration interface.

## Global CPU configuration

This configuration step enables the Axia GPIO module in NexGen.

Go to "Config" / "CPU" screen. Select CPU to configure and hit the "IO" button. On the "Board" tab, select Axia.

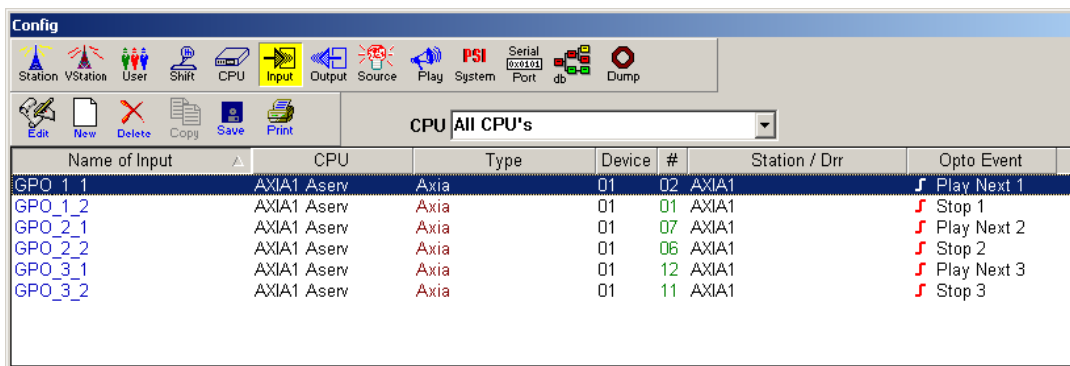
You may need to close NexGen at this point and start it again to make sure the new configuration is loaded.



## Configuring inputs

Input configuration enables control flow from Axia console to NexGen. This allows remote start and stop of selected playback machines as well as activation of other functions provided by NexGen.

On the "Config" / "CPU" screen select "Input". The screenshot below shows list of 6 defined events. Those events control "Control room" playback module.



Name of Input	CPU	Type	Device #	Station / Drr	Opto Event
GPO_1_1	AXIA1 Aserv	Axia	01 02	AXIA1	↵ Play Next 1
GPO_1_2	AXIA1 Aserv	Axia	01 01	AXIA1	↵ Stop 1
GPO_2_1	AXIA1 Aserv	Axia	01 07	AXIA1	↵ Play Next 2
GPO_2_2	AXIA1 Aserv	Axia	01 06	AXIA1	↵ Stop 2
GPO_3_1	AXIA1 Aserv	Axia	01 12	AXIA1	↵ Play Next 3
GPO_3_2	AXIA1 Aserv	Axia	01 11	AXIA1	↵ Stop 3

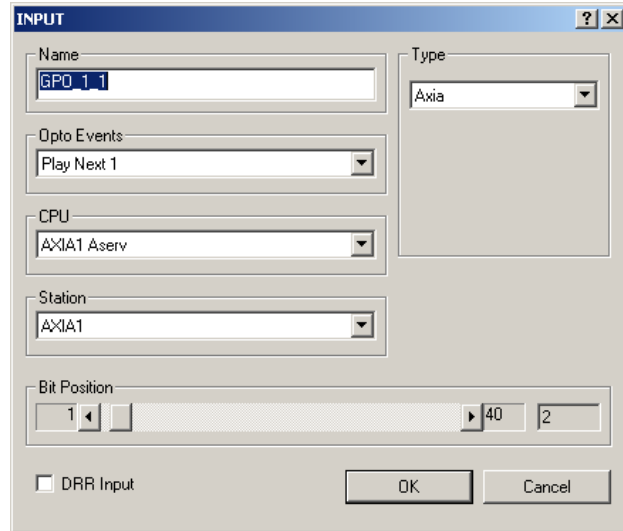
Use "New" button to create new event.

GPO\_1\_1 event corresponds to Axia Audio device 1, GPIO logic pin 1. It starts playback on machine 1. The detailed configuration of this event is the following:

Note pin numbering. Pins 1-5 correspond to livewire GPIO port 1 pins 1-5. Pins 6-10 correspond to Livewire port 2, etc.

You may need to close NexGen at this point and start it again to make sure the new configuration is loaded.

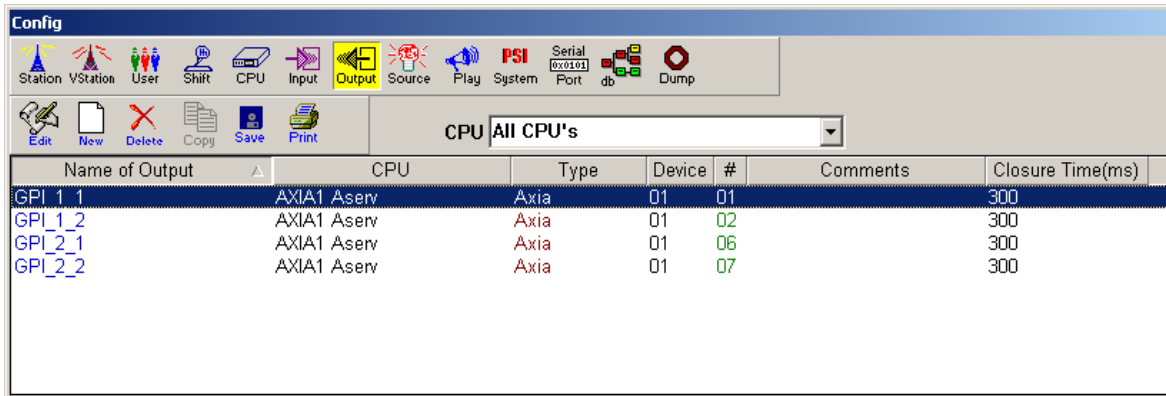
You should be able to start and stop playback on machines in control room module by pressing channel ON and OFF buttons on the console.



## Configuring outputs

Output configuration allows sending control messages from NexGen to the console. Such setup can be used to automatically turn on fader channels corresponding to current playback state of automation system.

On the "Config" / "CPU" screen select "Output". The screenshot below shows list of 6 events. Those events control "Control room" playback module.

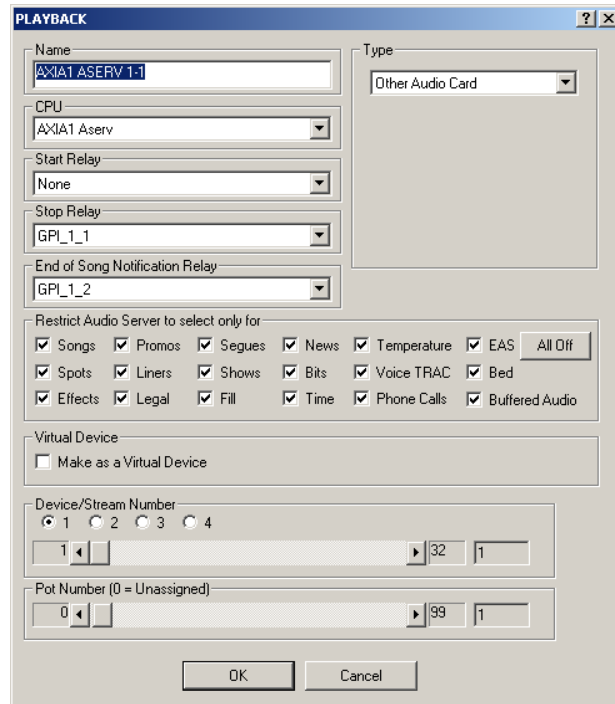


To enable defined GPIs in selected station, use "Config" / "Station" screen. Open station window, hit "Relay" button to build a list of Relays (GPIs).

Using the “Config” / “Play” screen we change options of playback machines. Here, we can assign a selected relay to Start and Stop player events.

Close NexGen and start it again to make sure the new configuration is loaded.

Once NexGen is restarted, you will be able to control NexGen using Axia GPIO.

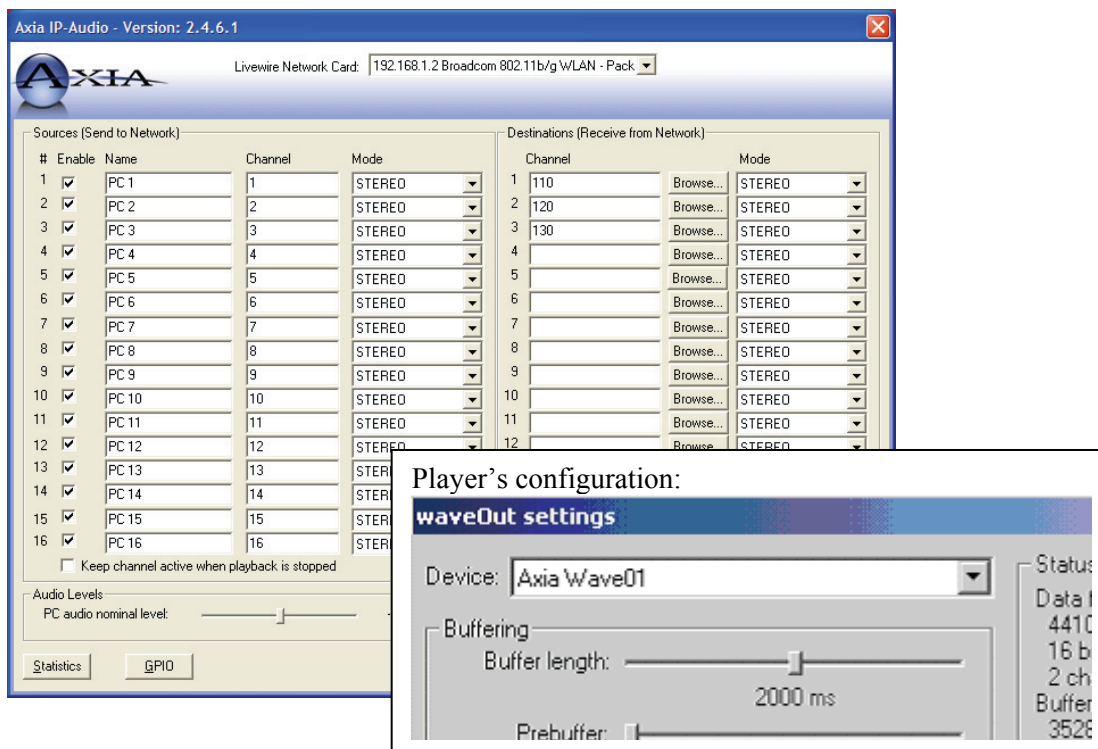


# Using Axia Windows Driver GPIO

31 January 2007

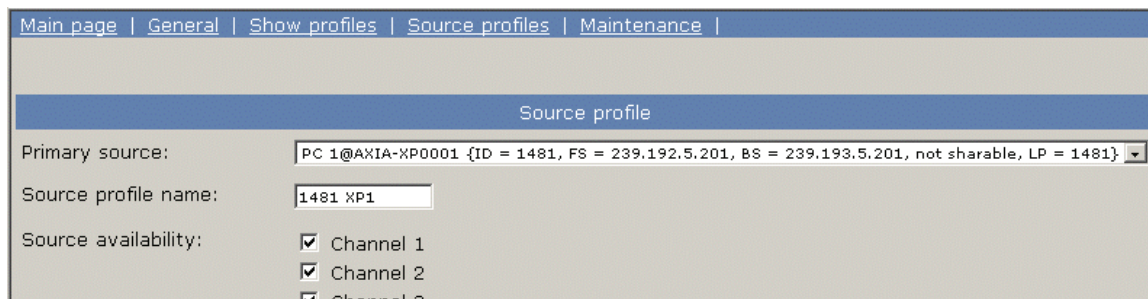
## Setup

The driver has the following sources configured. In our example, we are going to use source 1, which corresponds to audio device: “Axia Wave01”



The screenshot shows the Axia IP-Audio software interface. The main window displays two columns: 'Sources (Send to Network)' and 'Destinations (Receive from Network)'. The sources are numbered 1 through 16, each with a 'Name' (PC 1 to PC 16), a 'Channel' (1 to 16), and a 'Mode' (STEREO). The destinations are numbered 1 through 12, each with a 'Channel' (110 to 120) and a 'Mode' (STEREO). A 'Player's configuration' window is overlaid on the bottom right, showing 'waveOut settings' for 'Device: Axia Wave01'. The 'Buffer length' is set to 2000 ms. Other settings include 'Prebuffer' and 'Status' (Data 4410, 16 b, 2 ch, Buffer 3528).

Axia Element or SmartSurface consoles will use this source, so a Source Profile must be configured, as shown below (See the Element or SmartSurface users manual for instructions on how to construct a Source Profile).



The screenshot shows the 'Source profile' configuration window. The 'Primary source' is set to 'PC 1@AXIA-XP0001 {ID = 1481, FS = 239.192.5.201, BS = 239.193.5.201, not sharable, LP = 1481}'. The 'Source profile name' is '1481 XP1'. The 'Source availability' section has three checked options: 'Channel 1', 'Channel 2', and 'Channel 3'.





For automation systems, the “Computer Player” source type should be used (older software does not have this type, so “Line Input Logic” should be selected instead). Logic port must be enabled. GPIO signals available for every source type are defined in Element and SmartSurface manuals available at [www.AxiaAudio.com/manuals](http://www.AxiaAudio.com/manuals) .

Source type:	<input type="radio"/> CR host
	<input type="radio"/> CR producer
	<input type="radio"/> CR guest
	<input type="radio"/> Studio guest
	<input checked="" type="radio"/> Line
	<input type="radio"/> Phone
	<input type="radio"/> Codec
Fader mode:	<input checked="" type="radio"/> Normal
	<input type="radio"/> Fader start
Preview switching mode:	<input checked="" type="radio"/> Normal, auto switching disabled
	<input type="radio"/> CHANNEL ON turns Preview OFF
Hybrid answer mode:	<input checked="" type="radio"/> Normal, auto answer disabled
	<input type="radio"/> Channel ON answers hybrid
	<input type="radio"/> Channel ON or Preview ON answer hybrid
Backward feed enabled/disabled:	<input type="radio"/> Disabled
	<input checked="" type="radio"/> Enabled
Logic port enabled/disabled:	<input type="radio"/> Disabled
	<input checked="" type="radio"/> Enabled

In the example above, the channel is configured as “Line” source type. The Element and SmartSurface manuals refer to hardware pin numbers; in the GPIO protocols those pins are directly mapped to logical inputs and outputs.

All GPIO pins are active Low. This means that the automation system will react to state changes from High to Low.

### ***GPIO protocol***

Computer player opens a TCP/IP connection to port 93. Developers can use telnet to test commands and troubleshoot the system.

```
telnet 127.0.0.1 93
```



Commands sent by automation system (or user from telnet session) are in bold. Indications sent by GPIO module to the automation system are in normal text.

The automation system listens to GPO control messages from the console. From the console's point of view, the automation system is a playback device, so the console controls its "virtual" general-purpose outputs.

The automation system will send GPI commands to trigger actions on the console. Through the protocol, the PC triggers changes on "virtual" console's inputs.

First, the automation system subscribes for GPI and GPO updates using 'ADD GPI' and 'ADD GPO' commands, as illustrated below:

```
ADD GPI  
BEGIN  
GPI 1 IIIII  
GPI 2 IIIII  
[...]  
GPI 16 IIIII  
END
```

```
ADD GPO  
BEGIN  
GPO 1 hhhhh  
GPO 2 hhhhh  
[...]  
GPO 16 IIIII  
END
```

Operator turns the channel OFF by pressing button on the Console:

```
GPO 1 hhhhL      from GPIO logic table: STOP Pulse (Low)  
GPO 1 hLhhl     from GPIO logic table: OFF Lamp (Low)  
GPO 1 hlhhH     from GPIO logic table: STOP Pulse (High)
```

Operator turns the channel ON by pressing button on the Console:

```
GPO 1 hHhHh     from GPIO logic table: OFF Lamp (High), START Pulse (Low)  
GPO 1 Lhhlh     from GPIO logic table: ON Lamp (Low)  
GPO 1 lhHhH     from GPIO logic table: START Pulse (High)
```

An automation system turns the channel ON, remotely:

```
GPI 1 HHHHH  
GPI 1 Lxxxxx
```

An automation system turns the channel OFF, remotely:

```
GPI 1 HHHHH  
GPI 1 xLxxxx
```



## Custom GPIO messages

This extension to GPI/GPO commands allows user messages to be routed along established GPIO routes. Livewire Routing Protocol needs to be used to establish a GPIO route from one device to another, and PathFinder PC software provides a user interface to allow such configurations. The functionality can be tested using standard telnet client (TCP/IP connection to port 93).

To send custom message use the following syntax (must LOGIN before):

**GPI <source port number> CMD:"<custom command>"**

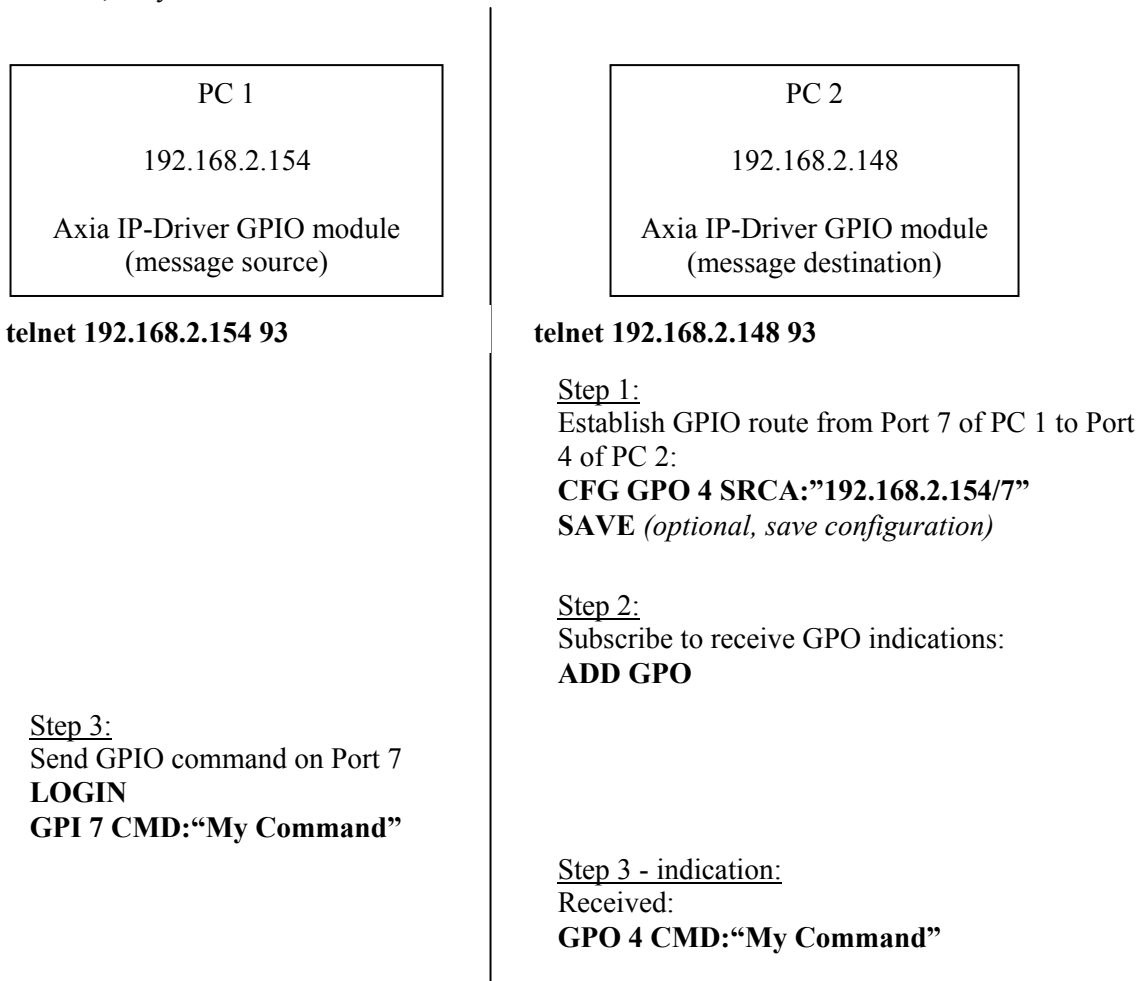
Indications syntax (must ADD GPO before):

**GPO <destination port number> CMD:"<custom command>"**

### Test example:

Let's connect two PC work stations with Axia Driver installed.

Scenario below, demonstrates LWRP message exchange between GPIO clients. Custom command, "My Command" is sourced at Port 7 of PC1 and delivered to Port 4 of PC2.



Custom GPIO messaging requires Axia IP-Audio Driver v2.4.6.1 or later.



Axia Audio, a Telos Company • 2101 Superior Ave. • Cleveland, Ohio, 44114, USA • +1.216.241.7225 • [www.AxiaAudio.com](http://www.AxiaAudio.com)