

1 DESCRIPTION

The ASI2416 is a CobraNet™ audio interface in a 1U rack mount format providing 16 channels of CobraNet receive and transmit.

The unit can be populated with up to four function specific modules, allowing up to 16 channels of analog or AES/EBU I/O. Each module has an interchangeable connector that may be configured with either a pluggable terminal block, StudioHub+® RJ-45, or a 50pin Centronics connector with XLR breakout cables.

The ASI2416 features a powerful Texas Instruments 32bit floating point DSP that allows sophisticated switching/mixing. A graphics display on the unit's front panel shows peak meters and CobraNet status.

AudioScience provides application software that may be used to set up the ASI2416. ASIControl sets up all internal features of the unit. ASIControl or CobraNet Discovery allows CobraNet routing connections to be set up between the ASI2416 and any other CobraNet device on the network.

Note: This datasheet reflects 4.02 firmware updates to the ASI2416. See section 13.7.7 for updating firmware.

2 FEATURES

- 16 CobraNet™ input channels and 16 CobraNet output channels on 100Mbit Ethernet with redundant RJ-45 connectors
- 4 CobraNet Transmitters and 8 CobraNet Receivers.
- 1U rack-mount unit
- Modular architecture allows up to 4 I/O modules to be inserted into the back of the unit
- Module connector options include Terminal Block (Phoenix style), StudioHub+® RJ-45, or 50pin Centronics connector with XLR breakout cables
- Available modules include eight-channel analog I/O and eight channel AES/EBU I/O
- Powerful floating point DSP provides metering, level control and up to 20 dB gain on all signal paths
- Interoperable with all AudioScience CobraNet Sound Cards and other third party CobraNet equipment
- Built-in 90-260VAC power supply

3 MODULE INFORMATION

For ordering, the ASI2416 model number is formatted to include the module(s), connector(s), and placement(s) of them within the ASI2416: ASI2416-XXXX-YYYY (X=module, Y= connector). See also the "Modules and Audio Connections " section.

Module		Input/Output Type	Inputs	Outputs	Connector Type	
Name	X =				Name	Y =
ASI1431	1	Balanced analog	8	8	ASI1491 (50 pin Centronics) ASI1492 (StudioHub+ RJ-45) ASI1493 (Terminal block)	
ASI1433	3	Balanced analog	0	8		2
ASI1441	4	AES/EBU	8	8		3
ASI1442	5	AES/EBU	8	0		1
ASI1443	6	AES/EBU	0	8	ASI1491 (50 pin Centronics) ASI1492 (StudioHub+ RJ-45) ASI1493 (Terminal block) ASI1494 (1/4" TRS)	
ASI1432	2	Balanced analog	8	0		2
ASI1462	8	Mic/Line	8	0		3
ASI1451	7	GPIO Optos/Relays	16	16		1
					ASI1493 (Terminal block)	4
					ASI1493 (Terminal block)	1



CobraNet is a registered trademark of Cirrus Logic. StudioHub+ is a registered trademark of Radio Systems, Inc.

4 SPECIFICATIONS

COBRANET INPUT/OUTPUT			
Type	CS181022 silicon from Cirrus Logic		
Network	10/100BaseT Ethernet with dual redundant RJ-45 connectors		
Connector	Dual RJ-45		
Sample Resolution	16, 20 or 24bit PCM		
Sample Rate	CobraNet standard 48 kHz		
Connections	8 CobraNet receivers and 4 CobraNet transmitters		
Latency	1.33, 2.66 or 5.33ms		
Control Protocol	SNMP using standard CobraNet MIBs.		
SIGNAL PROCESSING			
DSP	Texas Instruments TMS320C6713 @ 225MHz		
Control Protocol	AudioScience HPI using UDP		
ANALOG INPUT/OUTPUT			
Type	Balanced		
Input Level	-10 to +24dBu in 1dBu steps		
Input Impedance	10K ohms		
A/D converter	24bit over sampling		
Output Level	-10 to +24dBu in 1dBu steps		
D/A converter	24bit over sampling		
Load Impedance	600ohms or greater		
Dynamic Range [1]	>105dB (input or output)		
THD+N [2]	<0.0015% (input or output)		
Sample Rates	48 kHz from CobraNet		
Frequency Response	20Hz to 20kHz +0.1/-0.3dB		
Connectors	3.81mm Terminal Block, 50pin Centronics type with XLR breakout cables or StudioHub+ compatible RJ45.		
AES/EBU INPUT/OUTPUT			
Type	AES/EBU (EIAJ CP-340 Type I / IEC-958 Professional)		
Sample Rates	Input: 32, 44.1, 48 and 96kHz - uses sample rate converter to convert to CobraNet sample rate. Output: 32, 44.1, 48 and 96kHz selectable from any input, CobraNet, or dedicated sync input.		
Dynamic Range [1]	140dB, any input to any output		
THD+N [3]	135dB, any input to any output		
Connectors	3.81mm Terminal Block, 50pin Centronics type with XLR breakout cables or StudioHub+ compatible RJ45.		
SAMPLE RATE CLOCK			
Internal	48kHz from CobraNet		
LATENCY (48kHz CobraNet)	5.33ms CobraNet latency	2.67ms CobraNet latency	1.33ms CobraNet latency
Analog input across network to analog out	8.60ms	5.94ms	4.60ms
CobraNet input to analog Out	6.93ms	4.33ms	2.99ms
Analog input to CobraNet output	6.97ms	4.29ms	2.95ms
Analog input to analog output	1.90ms		
REGULATORY			
FCC Class A (US)			
CE Mark (EN 55022 Class A, EN55024)			
GENERAL			
Dimensions	1 RU, 19"(482mm) W x 8"(203mm) L x 1.75"(44mm) H		
Weight	7 lb (3.2kg) max, with 16 analog inputs/outputs and terminal block connectors		
Operating Temperature	0C to 60C		
Power Requirements	100-240VAC, 47-63Hz, 25W max.		

NOTES

[1] – Dynamic Range measured using a –60dB 1kHz sine wave and A weighting

[2] - THD+N measured using a +20dBu 1kHz sine wave sampled at 48kHz, 22-22kHz b/w and A weighting filter

[3] - THD+N measured using a –1dBfs 1kHz sine wave sampled at 48kHz, 22-22kHz b/w and A weighting filter

5 CONTENTS

1	DESCRIPTION	1
2	FEATURES	1
3	MODULE INFORMATION	1
4	SPECIFICATIONS	2
5	CONTENTS.....	3
6	TABLE OF FIGURES	5
7	REVISIONS.....	6
8	IMPORTANT SAFETY INSTRUCTIONS.....	7
9	NOTICES.....	9
10	ARCHITECTS & ENGINEERS SPECIFICATION.....	10
11	INTRODUCTION	11
11.1	COBRANET™	11
11.2	COBRANET ROUTING	12
11.2.1	Audio Routing Channels	13
11.2.2	CobraNet Transmitters	14
11.2.3	CobraNet Receivers	14
11.2.4	CobraNet Sample Rate and Latency	14
11.2.5	CobraNet References	14
12	HARDWARE INSTALLATION.....	15
12.1	RACK MOUNTING	15
12.2	ETHERNET CONNECTION	15
12.3	AC POWER	15
12.4	MODULES AND AUDIO CONNECTIONS	16
12.4.1	Modules.....	16
12.4.2	Connectors.....	16
13	OPERATION.....	17
13.1	POWER UP SEQUENCE	17
13.1.1	AC Power.....	17
13.1.2	AudioScience Firmware images.....	17
13.1.3	Firmware loading sequence.....	17
13.1.4	Loading the factory firmware image.....	17
13.2	FRONT PANEL DISPLAY	18
13.2.1	CobraNet Transmit and Receive Peak Meters Display.....	18
13.2.2	CobraNet Bundle Number and Channel Settings Display	19
13.2.3	MAC and IP Address Display	19
13.2.4	Product Information Display	20
13.2.5	Module Input/output Meters – Analog Display.....	20
13.2.6	Module Input/Output Meters – AES/EBU Display.....	21
13.3	RESETTING THE ASI2416 CONFIGURATION TO DEFAULT SETTINGS.....	21
13.3.1	Resetting Audio Control Settings	21
13.3.2	Resetting Static IP Address	22
13.3.3	Resetting CobraNet Persistence.....	22
13.4	NETWORK INTERFACE	22
13.4.1	ASI2416 Network Mask Assignment	22
14	CONFIGURATION	23
14.1	ASICONTROL CONFIGURATION.....	23

14.1.1	<i>ASiControl Layout</i>	23
1.1.1	<i>Adapter Information</i>	24
1.1.2	<i>Interface</i>	24
14.1.2	<i>Level</i>	24
14.1.3	<i>Meter</i>	25
14.1.4	<i>Silence Detector</i>	26
14.1.5	<i>Setting AES/EBU Clocking</i>	27
14.1.6	<i>Input and Output Volume Adjustment</i>	31
1.1.1	<i>Access control using passwords</i>	31
1.2	AUDIOSCIENCE LUA IMPLEMENTATION	34
1.2.1	<i>Language Extensions</i>	34
1.2.2	<i>Script Layout</i>	38
1.2.3	<i>Examples</i>	38
1.2.4	<i>Using ASiControl to Write Scripts</i>	43
1.2.5	<i>Lua License</i>	46
1.2.6	<i>References</i>	46
14.2	COBRANET CONFIGURATION	47
14.2.1	<i>Setting Up AudioScience Software to Configure CobraNet</i>	47
14.2.2	<i>Configuring CobraNet Using ASiControl</i>	49
14.2.3	<i>References</i>	53

6 TABLE OF FIGURES

Figure 1.	ASI2416 block diagram.....	11
Figure 2.	Illustration of a CobraNet bundle going between 2 CobraNet devices.	12
Figure 3.	Audio routing channel details.....	13
Figure 4.	ASI2416 display of CobraNet peak meters and bar height mapping to dBFS range.	18
Figure 5.	ASI2416 bundle display for CobraNet receivers 1 and 2.....	19
Figure 6.	ASI2416 MAC and IP address display.....	19
Figure 7.	ASI2416 product information display.	20
Figure 8.	ASI2416 analog module display.	20
Figure 9.	ASI24216 AES/EBU module display.	21
Figure 10.	ASiControl layout	23
Figure 11.	Using ASiControl to select Line_Out 1.	24
Figure 13.	Level displayed by ASiControl for Line_Out 1	24
Figure 14.	A stereo peak meter display. The RMS is the green bar and the peak is the yellow bar.	25
Figure 15.	Silence Detector, disable and enable, as seen in ASiControl.....	26
Figure 16.	ASI2416 AES/EBU clocking for Module 1.....	27
Figure 17.	ASI2416 AES/EBU clocking for Module 2.....	28
Figure 18.	Selecting Line_Out 1 in ASiControl prior to setting Tx clock source.	29
Figure 19.	ASiControl display of AES/EBU Line_Out 1 node controls.....	29
Figure 20.	ASiControl selection of AES/EBU output clock source.	30
Figure 21.	ASiControl node displays with volumes.	31
Figure 23.	Laptop to ASI2416 connection.	50

7 REVISIONS

Date	Changes
21 October 2007	Updated format.
07 November 2008	Fix audio channel maps.
18 November 2008	Updated number of modules supported.
30 April 2009	Added information to Section 12.3; driver/firmware 3.12.00 or newer needed. "Save/Restore Configuration;" now Section 13.10. Added information to Section 13.10; sysName and sysDescription can be restored with driver 3.14.00 or later.
11 September 2009	Section 12.2.5 – added descriptive information for the front panel display.
03 December 2009	Section 4 - Adjust frequency response to +0.1/-0.3dB
04 June 2010	Page 1: Added Note that this datasheet presumes firmware 4.02 is installed; removed previous driver references. Added module/connector numbering information to Module Information section. Updated Input and Output Volume Adjustment section. Updated Silence Detector section. Updated Modules and Audio Connections section. Updated CobraNet section. Updated formatting.
25 January 2011	In ASiControl Configuration section, explicitly stated that ASiControl must be closed in order to save control changes to flash memory.
16 February 2011	Added lwIP copyright, etc., under Notices.
11 August 2011	Corrected module information table on page 1.
7 May 2015	Corrected formatting errors

8 IMPORTANT SAFETY INSTRUCTIONS

1. Read these instructions.
2. Keep these instructions.
3. Head all warnings.
4. Follow all instructions.
5. Do not use this apparatus near water.
6. Clean only with a dry cloth.
7. Do not block any ventilation openings. Install in accordance with the manufacturer's instructions.
8. Do not install near any heat sources such as radiators, heat registers, stoves, or other apparatus (including amplifiers) that produce heat.
9. Do not defeat the safety purpose of the polarized or grounding-type plug. A polarized plug has two blades with one wider than the other. A grounding type plug has two blades and a third grounding prong. The wide blade or the third prong is provided for your safety. If the provided plug does not fit into your outlet, consult an electrician for replacement of the obsolete outlet.
10. Protect the power cord from being walked on or pinched, particularly at plug ends, convenience receptacles, and the point where they exit from the apparatus.
11. Only use attachments/accessories specified by the manufacturer.
12. Use only with the cart, stand, tripod, bracket, or table specified by the manufacturer, or sold with the apparatus. When a cart is used, use caution when moving the cart/apparatus combination to avoid injury from tip-over.
13. Unplug this apparatus during lightning storms or when unused for long periods of time.
14. Refer all servicing to qualified service personal. Servicing is required when the apparatus has been damaged in any way, such as power-supply cord or plug is damaged, liquid has been spilled or objects have fallen into the apparatus, the apparatus has been exposed to rain or moisture, does not operate normally, or has been dropped.



This symbol is intended to alert the user to the presence of uninsulated dangerous voltage within the product's enclosure that may be of sufficient magnitude to constitute a risk of electric shock to humans



This symbol is intended to alert the users to the presence of important operating and maintenance (servicing) instructions in the literature accompanying the product.



CAUTION: To reduce the risk of electric shock, do not remove the cover. No user-serviceable parts inside.

WARNING:

1. To prevent fire or electric shock, do not expose this apparatus to rain or moisture.
2. This apparatus shall not be exposed to dripping or splashing and no objects filled with liquids, such as a vase, shall be placed on the apparatus.
3. This is a Class 1 apparatus, and as such must be connected to a mains socket outlet with a protective earthing connection.
4. The mains plug is used as the disconnect device and shall remain readily operable.

9 NOTICES

FEDERAL COMMUNICATIONS COMMISSION (FCC) INFORMATION

NOTE: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a commercial installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user will be required to correct the interference at his or her own expense.

lwIP

lwIP is licenced under the BSD licence:
Copyright (c) 2001-2004 Swedish Institute of Computer Science.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

10 ARCHITECTS & ENGINEERS SPECIFICATION

The CobraNet Audio Interface (interface) shall be available in various I/O configurations. Inputs/outputs shall be specified in channels in multiples of 8, up to a total of 32. Plug-in module options shall include analog input/output (ASI1431), analog input (ASI1432), analog output (ASI1433), AES/EBU input/output (ASI1441), AES/EBU input (ASI1442), AES/EBU output (ASI1443) and GPIO (ASI1445). The system shall support up to 4 plug-in modules. Analog inputs and outputs shall use 24-bit A/D and D/A and have adjustable level control. AES/EBU inputs and outputs shall have hardware sample rate converters. AES/EBU outputs shall support clocking from CobraNet, any AES/EBU input or from an internal sample clock running at any of the following rates 32, 44.1, 48, 64, 88.2, 96 kHz. Module connection options shall include XLR and Phoenix style terminal blocks.

CobraNet technology shall transport digital audio over Ethernet, allowing multiple units to share digital audio. The CS181022 CobraNet chip shall be used, supporting integration with any third party CobraNet system or device. Multiple unit installations shall require an external Ethernet switch. All Ethernet connections shall be via CAT5 cable or fiber-optic.

Each interface configuration shall include a 300MHz 32-bit floating point DSP and all internal audio processing shall be performed digitally in floating point format. The front panel of the system shall support displaying level meters for all audio inputs and outputs (including CobraNet channels), CobraNet bundle information, IP address, MAC address and module information.

Software shall be provided for configuring each system remotely across the network. Software shall operate on a PC computer, with network card installed, running Windows® 2000/XP Professional. Source code and programming information for controlling the system remotely shall be available to integrators upon request.

The CobraNet Digital Audio Interface shall be CE, FCC and UL / C-UL marked, and shall incorporate AES48-2005 Grounding and EMC best practices. The CobraNet Digital Audio Interface shall be compliant with EU Directive 2002/95/EC, the RoHS directive. Warranty shall be 3 years.

The CobraNet Audio Interface shall be ASI2416.

11 INTRODUCTION

The ASI2416 is a CobraNet™ audio interface in a 1U rack mount format providing 16 channels of CobraNet receive and transmit.

The unit can be populated with up to four function specific modules, allowing up to 16 channels of analog or AES/EBU I/O. Each module has an interchangeable connector that may be configured with either a pluggable terminal block, a StudioHub+® RJ-45, or a 50pin Centronics connector with XLR breakout cables

The ASI2416 features a powerful Texas Instruments 32bit floating point DSP that allows sophisticated switching and mixing. A graphics display on the unit's front panel shows peak meters and CobraNet status.

AudioScience provides application software that may be used to set up the ASI2416. ASIControl sets up all internal features of the unit such as levels also allows CobraNet routing connections to be set up between the ASI2416 and any other compliant CobraNet device on the network.

There are many combinations ASI2416 plus modules available. The most common subset is listed below:

ASI2416-1000-1000	8 channels of balanced analog in/out. Terminal block connectors
ASI2416-1100-1100	16 channels of balanced analog in/out. Terminal block connectors
ASI2416-4000-2000	4 AES/EBU in/out (8 channels in/out). XLR connectors on breakout cable.
ASI2416-4400-2200	8 AES/EBU in/out (16 channels in/out). XLR connectors on breakout cable.

The following is a block diagram of the ASI2416. Different models will have different configurations of the audio I/O, per the above table.

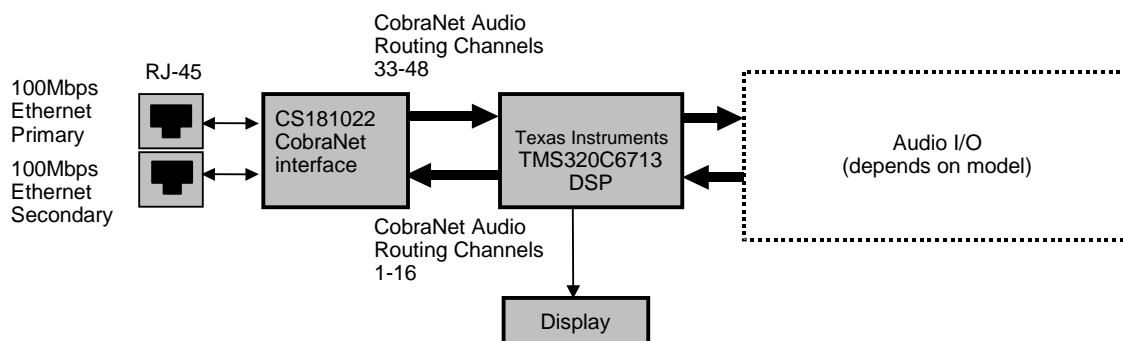


Figure 1. ASI2416 block diagram.

11.1 CobraNet™

CobraNet™ is a combination of software, hardware and network protocol that allows distribution of many channels of real-time, high quality, digital audio over an Ethernet network. It was developed by Peak Audio in the 1990s and is now owned by Cirrus Logic. Interoperability between CobraNet devices from different manufacturers is supported through a standard communications protocol. CobraNet compliant devices are based on a common silicon or hardware reference design from Cirrus Logic.

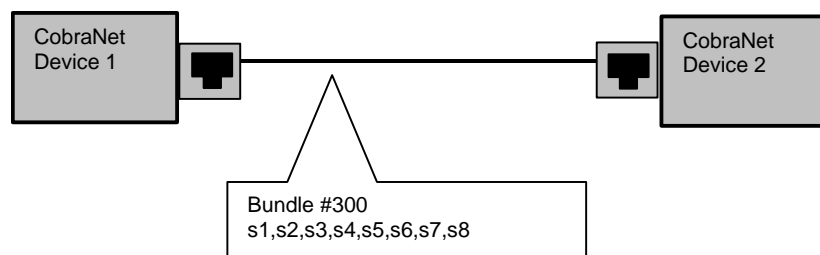
The Cirrus Logic website, www.cobranet.info, is dedicated to CobraNet.

CobraNet delivers audio in standard Ethernet packets over 100Mbit Fast Ethernet. Switches, hubs, media converters, and other gear that operate in compliance with the IEEE 802.3u specification for Fast Ethernet, will work with CobraNet. CobraNet does not support 10Mbit Ethernet varieties (10BASE-T, Coaxial) due to their limited bandwidth.

CobraNet operates at the Data Link Layer (also referred to as OSI Layer 2 or MAC layer). Because it does not use the higher IP layer for audio data transport, CobraNet does not suffer from IP latency limitations. In most cases data communications and CobraNet data can coexist on the same network without QOS issues. All audio is sent inside a custom Ethernet packet whose header tells network devices that the packet contains CobraNet audio, rather than plain data. The CobraNet term for an audio packet is "bundle". A bundle may contain from one to eight audio channels, each channel being composed of PCM samples of 16, 20 or 24 bits in length.

11.2 CobraNet Routing

The whole point of network audio is to route digital audio from point A to point B. CobraNet introduces a concept called a "bundle" to define virtual audio routes from one CobraNet device to another one. A bundle is a logical collection of up to 8 channels that can be sent from one device to another. Each bundle is assigned a unique number between 1 and 65535. Bundles form the heart of the CobraNet routing capability.



The bundle number 300 is used to describe this collection of channels coming from Device 1. s1 to s8 represent audio samples. The bundle shown above consists of 1 to 8 samples of audio each taken from different channels of Device 1.

Figure 2. Illustration of a CobraNet bundle going between 2 CobraNet devices.

The above figure illustrates a bundle of audio being sent from one CobraNet device to another. Device 1 is transmitting the CobraNet bundle, while Device 2 is receiving it. In this case, both devices need to be set to bundle 300 for the audio link to be made. The CobraNet mechanism for transmitting bundles is "transmitters." Similarly, the mechanism for receiving bundles is "receivers." Each CobraNet device has several transmitters and receivers and so can simultaneously send and receive audio channels using several different bundle numbers. This capability supports audio links between many different CobraNet devices.

11.2.1 Audio Routing Channels

Before further discussion of CobraNet transmitters and receivers, terminology useful for specifying audio channels within a bundle needs to be introduced. Somewhat obviously, these channels are called the Audio Routing Channels. On an ASI2416 audio routing channels 1 – 16 map to input channels 1 – 16. Similarly, on the output side audio routing channels 33-48 map to output channels 1 –16.

Audio Routing Channels

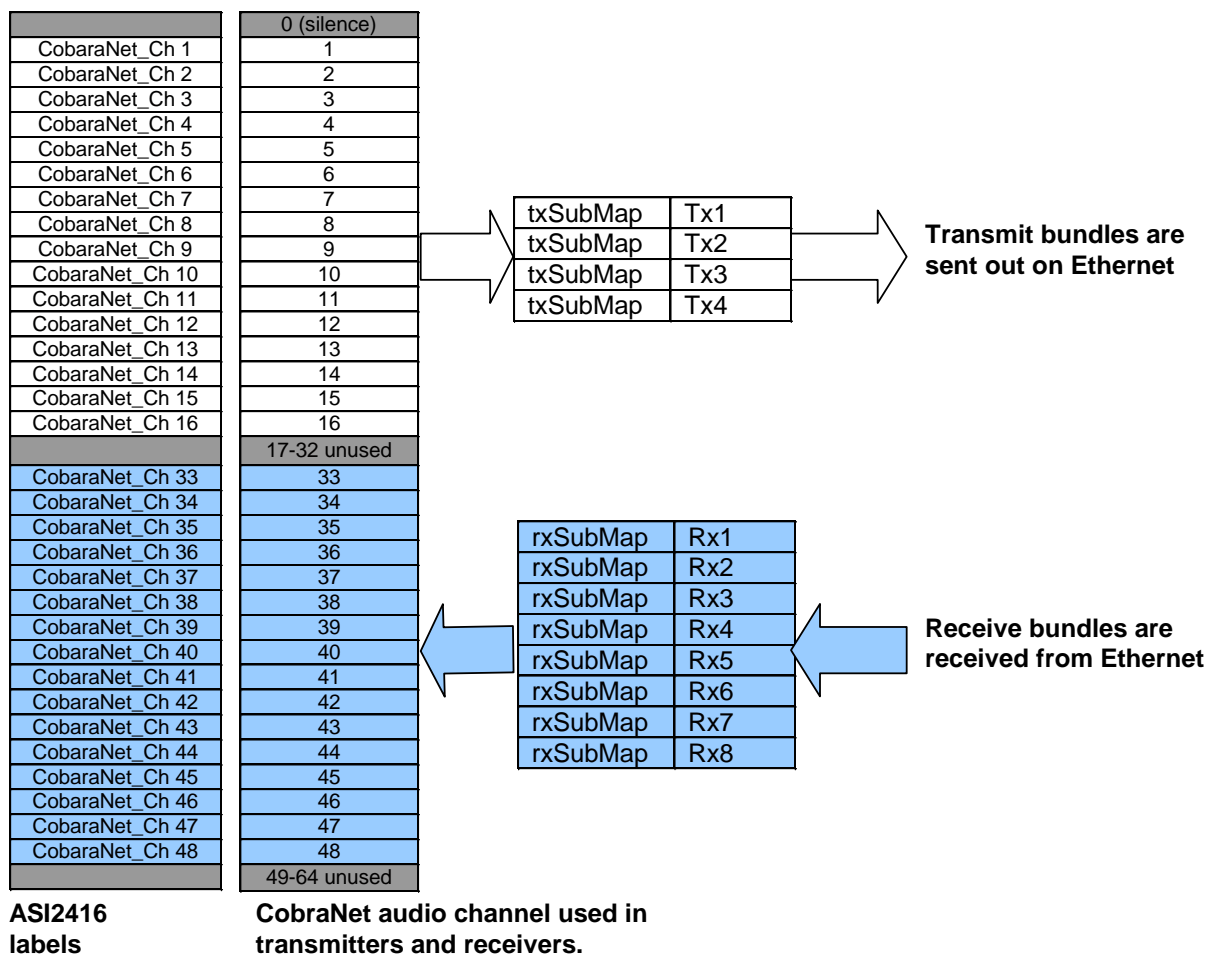


Figure 3. Audio routing channel details.

11.2.2 CobraNet Transmitters

A CobraNet transmitter is a logical entity in the CobraNet interface that has the ability to send a bundle of audio samples on the CobraNet network. CobraNet devices typically have multiple transmitters. The ASI2416, for example, has 4 transmitters. An incomplete list of transmitter routing variables follows:

- txBundle – this variable specifies the bundle number to transmit. A value of 0 indicates that the transmitter is disabled.
- txSubMap – a sequence of up to 8 audio routing channel numbers that specify which audio samples should be placed in the bundle. A value of 0 indicates an unused slot in the bundle.
- txSubFormat – a sequence of format specifiers that define how many bits per sample are placed in the bundle.
- txSubCount – the number of channels in this bundle.

11.2.3 CobraNet Receivers

A CobraNet receiver is a logical entity in the CobraNet interface that has the ability to receive a bundle of audio samples from the CobraNet network. CobraNet devices typically have multiple receivers. The ASI2416, for example, has 4 receivers. An incomplete list of receiver routing variables follows:

- rxBundle – the number of the bundle to receive. This should be the same bundle number being transmitted somewhere else on the network. A value of 0 indicates that the receiver is disabled.
- rxSubMap – a sequence of up to 8 audio routing channel numbers that specify where incoming bundle samples should be routed.

11.2.4 CobraNet Sample Rate and Latency

The CobraNet sample rate supported by the ASI2416 is fixed at 48kHz with three latency modes of 5.33ms (default), 2.67ms, or 1.33ms.

11.2.5 CobraNet References

This document is not intended to be an expansive guide to CobraNet networking and routing. The ASI2416 adheres to the CobraNet standard through the use of off-the-shelf CobraNet silicon from Cirrus Logic. More detailed CobraNet information is available from Cirrus Logic's website.

The following links may be helpful:

CobraNet Info:

<http://www.cobranet.info/en/support/cobranet/>

CobraNet CobraCAD and CobraNet Discovery:

<http://www.cobranet.info/dispatch/forms/sup/boardreg/breg/BregController.jspf>

CobraNet Audio Routing Primer:

http://cirrus.com/en/pubs/appNote/CobraNet_AudioRoutingPrimer.pdf

Hardware Manual and Programmer's Reference:

http://www.cobranet.info/en/support/cobranet/developer/tech_data_sheet.html


Switched Networks and Redundancy:

http://www.cirrus.com/en/support/cobranet/design/switched_networks.html

12 HARDWARE INSTALLATION

12.1 Rack Mounting

The ASI2416 is 1 RU (1 rack unit/space) high and mounts in a standard 19-inch equipment rack.

- Use four mounting screws to fasten the front panel of the ASI2416 to the 19-inch rack rails.
- Support any cables that are attached to the back of the ASI2416 so that their weight does not put undue stress on the ASI2416's connectors.
-  **The ASI2416 has cooling vents on the side of the unit. Be careful not to obstruct these.**

12.2 Ethernet Connection



There are two RJ-45 Ethernet jacks on the rear of the ASI2416. If a redundant CobraNet network is not being used then plug a network cable into the PRIMARY jack. If a redundant network is being used then plug a second network cable into the SECONDARY jack. (See http://cobranet.info/support/design/switched_networks for information on redundant networks.)

A CAT-5 or better (CAT-5e, CAT-6 etc) network cable is required for 100baseT Ethernet operation. The cable length between the ASI2416 and a network switch should not exceed 100 meters (328 feet)

12.3 AC Power



The detachable AC power cord that comes with the ASI2416 plugs into the IEC connector on the chassis.

The ASI2416 operates with AC voltages from 90 to 260VAC, 47 to 63Hz. No selection of voltage or frequency is required, the ASI2416's power supply will automatically adjust.



Use only an AC power source with a protective earth ground.

The ASI2416 has no power switch. Detach the AC power cord to remove power to the ASI2416.

12.4 Modules and Audio Connections

The ASI2416 extended model number indicates the module(s) and where they are placed in the ASI2416, as well as the connector(s) and where they are placed in the ASI2416 (connected to the modules). The extended model number has the following format, where X=module and Y= connector: ASI2416-XXXX-YYYY. A 0 indicates empty slots/connectors.

For example, if purchasing an ASI1441 with connector ASI1492 for slot 1, the extended model number would be ASI2416-4000-3000.




12.4.1 Modules

The ASI2416 will ship with ordered modules and connectors already installed. Should a module need to be changed in the field, it is easily done. Use an anti-static strap when handling the modules. Power down the ASI2416 and remove the three, small screws that attach the module to the ASI2416 (two on the bottom, one on top). Slide the module(s) out. Slide the module(s) in the desired slot(s), replace the three screws, and power up the ASI2416. The connector is removed from the module in the same way; slide the two components apart.




Note that when installing modules, always populate the back of the ASI2416 from left to right, and do not leave any gaps.

For Example, looking at the back of an ASI2416:

Correct:

ASI1443	ASI1443	ASI1431	<empty>	  
---------	---------	---------	---------	---

Incorrect:

ASI1443	<empty>	ASI1431	<empty>	  
---------	---------	---------	---------	---

There are eight different modules for the ASI2416:

Analog:

ASI1431 – 8 channels in/out
ASI1432 – 8 channels in
ASI1433 – 8 channels out

AES/EBU:

ASI1441 – 4 in/out (8 channels in/out)
ASI1442 – 4 in (8 channels in)
ASI1443 – 4 out (8 channels out)

Other:

ASI1451 – GPIO, 16 Optos/16 Relays
ASI1462 – Mic/Line In, 8 channels in

For further information on the modules, datasheets for each one can be found [here](#) under Resources.

12.4.2 Connectors

The ASI2416 modules support a range of interchangeable I/O connectors. A choice of 50pin Centronics (ASI1491), StudioHub+™ (ASI1492), Phoenix (terminal block) (ASI1493), and ASI1493 (¼" TRS) allows the module to adapt to a wide variety of interconnection schemes with minimal custom wiring.

The ASI2416 will ship with the ordered connectors installed on the ordered modules. Should a connector need to be replaced in the field, remove the module as stated in the section above then grasp the module in one hand and the connector in the other and carefully slide the two pieces apart. Replace connectors as needed, and then slide the module/connector unit back in the ASI2416, as stated above.



ASI1491
50 pin Centronics



ASI1492
StudioHub+



ASI1493
Terminal Block



ASI1494
¼" TRS

For pinout information, refer to the rear label of the connector, or go to the ASI2416 overview web page [here](#) and under Resources click on the specific module's datasheet.

13 OPERATION

13.1 Power up sequence

This section describes the ASI2416 power up sequence.

13.1.1 AC Power

Connect AC power to the unit by attaching the power cord to the IEC connector on the rear of the ASI2416, as explained in section 12.4

13.1.2 AudioScience Firmware images

The ASI2416 boots from a firmware image stored in flash memory. There are two independent firmware images stored in every ASI2416. The two images are named “factory” and “update”. The “factory” image is a reference image that is in place should a “bad” image be downloaded to the device. The “update” image is the image that can be updated in the field if required.

13.1.3 Firmware loading sequence

When first powered up, the ASI2416 performs the following sequence:

1. Checks for a valid “update” firmware image.
2. Loads the update image and starts running it.
3. Loads any control settings that may have been stored to flash.

In the case where the “update” image is determined to be corrupt, the factory image is loaded.

13.1.4 Loading the factory firmware image

The ASI2416 can be forced to load the factory firmware image by depressing the SELECT button on front panel as power is applied to the device. Keep button depressed while power is applied.

13.2 Front Panel Display

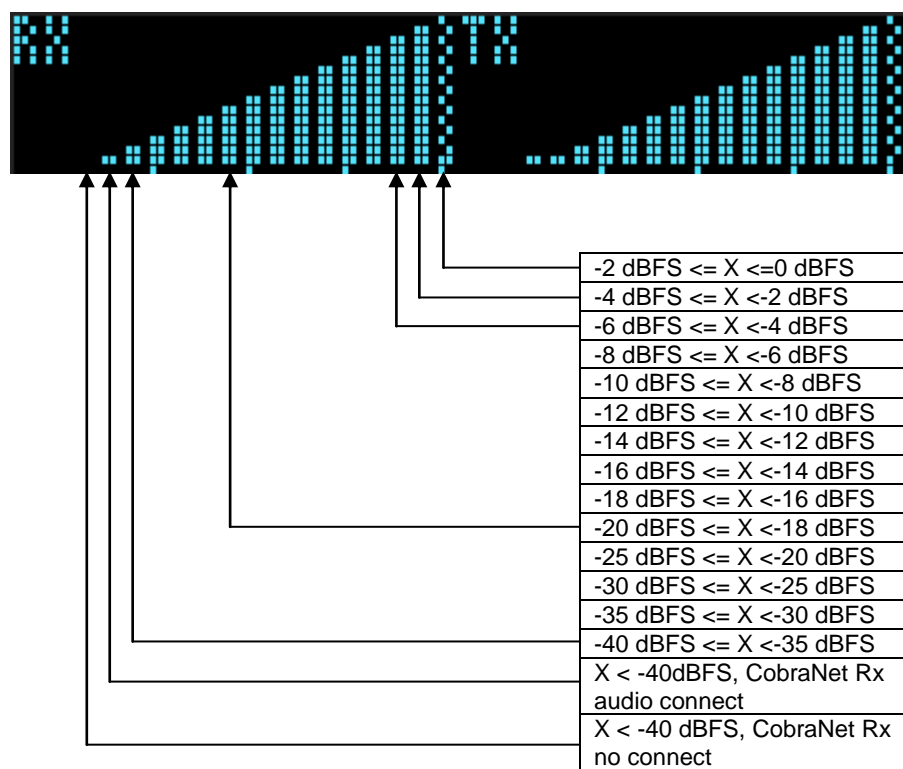
The ASI2416 display shows status readings from the ASI2416. The select button should be pressed to move to the next display. In order the displays are:

13.2.1 CobraNet Transmit and Receive Peak Meters Display

The first display shows peak meters for all 16 CobraNet receive and transmit routing channels

Peak Meter features:

- Ticks extending from the baseline to the bottom of the display mark every 4th peak meter, i.e. channels 4,8,12, and 16 are marked.
- A "checkered" bar indicates a peak level of -2dBFS or greater.
- Meter bars are 2 pixels wide.
- A missing bar on a receive channel indicates no CobraNet audio connect.
- A 1 pixel high bar indicates CobraNet audio connect, but audio level is below -40 dBFS.
- A 2 pixel high bar indicates CobraNet audio connect and an audio level greater than -40 dBFS but less than -35 dBFS.
- Audio peak levels in the range of -40 dBFS to -20 dBFS are displayed using 5 dB per pixel.
- Audio peak levels in the range of -20 dBFS to 0 dBFS are displayed starting at 6 pixels high and use addition pixels at a rate of 1 pixel every 2 dB.



In the above table X is a peak meter reading in dBFS.

Figure 4. ASI2416 display of CobraNet peak meters and bar height mapping to dBFS range.

13.2.2 CobraNet Bundle Number and Channel Settings Display

The bundle panels support displaying bundle and submap information for the device's active CobraNet transmitters and receivers. By default receivers 1 & 2 are displayed and transmitters 1 & 2 are displayed. Additional transmitters and receivers are displayed if they have been assigned a non-zero bundle number indicating that they are active.

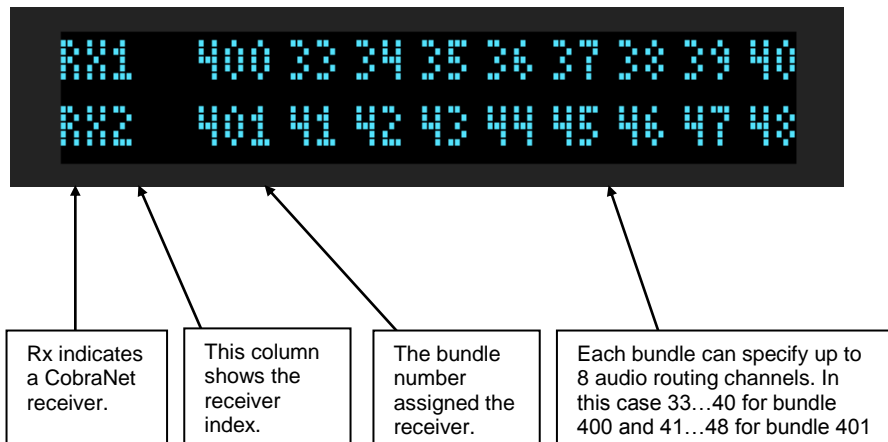


Figure 5. ASI2416 bundle display for CobraNet receivers 1 and 2.

In Fig. 5, the first receiver's submap is specifying the first sub-channel get output on audio routing channel 33. On an ASI2416-1100 (analog, 16 I/O), this corresponds to Line Out 1. The second sub-channel is output on Line Out 2 and so on. In this instance receiver 1 is taking in all 8 channels of bundle 400 and outputting them to Line Outs 1..8.

13.2.3 MAC and IP Address Display

This display shows the ASI2416s network MAC address and IP address. If the IP address has been assigned dynamically then DYNAM is displayed. If has been assigned statically then STATIC is shown.



Figure 6. ASI2416 MAC and IP address display.

13.2.4 Product Information Display

This display shows, top line, from left to right:

- Part number and revision (ASI2416C)
- Serial number (S:12345)
- Adapter index (I:1234)

This display shows, bottom line, from left to right:

- Installed modules: A=Analog, D=AES/EBU Digital (A:A:-:-).
- DSP firmware version (V:3.09.00)
- DSP utilization in % (0%)



Figure 7. ASI2416 product information display.

13.2.5 Module Input/output Meters – Analog Display

This display shows information and peak meters for an analog module's 8 input and outputs. The meters have the same scale as the CobraNet Transmit and Receive peak meters. From top to bottom, left to right:

- Number and type of installed module (1:Analog)
- Module hardware version (B)
- Input peak meters
- Output peak meters

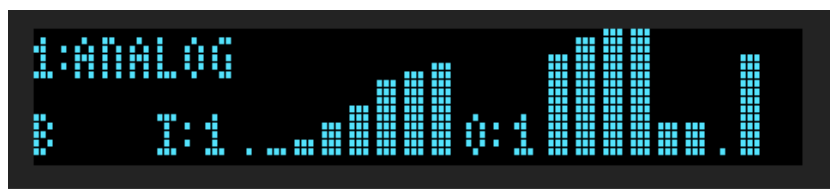


Figure 8. ASI2416 analog module display.

Peak Meter Features:

- A "checkered" bar indicates a peak level of -2dBFS or greater.
- Meter bars are 3 pixels wide.
- A missing bar on a receive channel indicates no CobraNet audio connect.
- A 1 pixel high bar indicates the audio level is below -40 dBFS.
- A 2 pixel high bar indicates an audio level greater than -40 dBFS but less than -35 dBFS.
- Audio peak levels in the range of -40 dBFS to -20 dBFS are displayed using 5 dB per pixel.
- Audio peak levels in the range of -20 dBFS to 0 dBFS are displayed starting at 6 pixels high and use additional pixels at a rate of 1 pixel every 2 dB.

13.2.6 Module Input/Output Meters – AES/EBU Display

This display shows information and peak meters for an AES/EBU digital module's 8 input and outputs. The meters have the same scale as the CobraNet Transmit and Receive peak meters

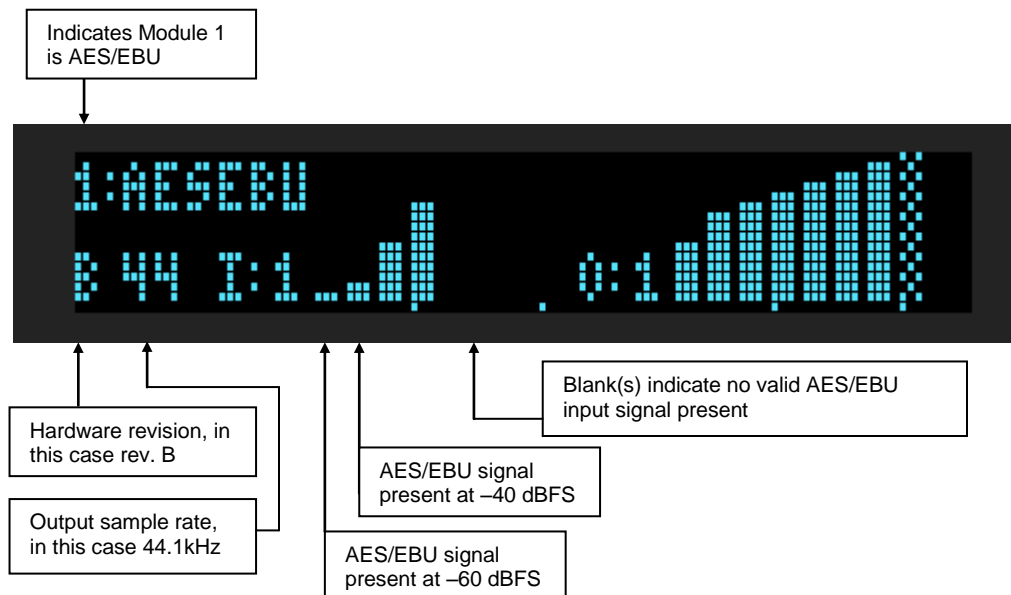


Figure 9. ASI2416 AES/EBU module display.

Peak Meter Features:

- A "checkered" bar indicates a peak level of -2dBFS or greater.
- Meter bars are 3 pixels wide.
- A missing bar on a receive channel indicates no CobraNet audio connected.
- A 1 pixel high bar indicates the audio level is below -40 dBFS.
- A 2 pixel high bar indicates an audio level greater than -40 dBFS but less than -35 dBFS.
- Audio peak levels in the range of -40 dBFS to -20 dBFS are displayed using 5 dB per pixel.
- Audio peak levels in the range of -20 dBFS to 0 dBFS are displayed starting at 6 pixels high and use addition pixels at a rate of 1 pixel every 2 dB.
- No meter bar indicates no valid input signal present on that AES/EBU stereo pair.

13.3 Resetting the ASI2416 Configuration to Default Settings

13.3.1 Resetting Audio Control Settings

Resets all mixer volumes and analog levels back to default settings.

The ASI2416 stores audio control settings in flash memory so that they may be restored upon power up. To restore all control settings to factory defaults perform the following steps:

```
ASI2416C  S:12345  I:1234
A:R-1-  U:3.09.00  0%
```

Navigate to the product information display using the select button.

1. Depress the select button. After 1 second you will see "RELEASE BUTTON TO: SAVE CONTROL SETTINGS". After about 3 seconds you will observe "RELEASE BUTTON TO: RESET CONTROL SETTINGS". Release the select button.
2. Powered cycle the ASI2416 to cause controls to be restored to default settings.

13.3.2 Resetting Static IP Address

If you have configured the ASI2416 for a static IP address, you can reset it back to a dynamically assigned IP address using the following steps:

```
MAC: 00.60.2B.06.D0.64
IP: 255.128.128.255 STATIC
```

1. Navigate to the "MAC and IP address" display using the select button.
2. Depress the select button for more than 1 second. After you observe "RELEASE BUTTON TO: CLEAR STATIC IP," release the select button.

13.3.3 Resetting CobraNet Persistence

Resets all CobraNet Tx and Rx bundles to 0.

If you somehow get your ASI2416 into a state where you cannot communicate with it, the CobraNet persistence variable can be reset from the front panel of the unit. This may allow communication with the ASI2416.

```
MAC: 00.60.2B.06.D0.64
IP: 255.128.128.255 STATIC
```

1. Navigate to the "MAC and IP address" display using the select button.
2. Depress the select button for more than 3 seconds. After you observe "RELEASE BUTTON TO: TURN PERSISTENCE OFF" release the select button.
3. Now that persistence is turned off, power cycle the ASI2416. You should now be able to communicate with the device.

13.4 Network Interface

ASI2416 audio settings are configured over the network via UDP messages using port 44600. The ASI2416 CobraNet networking settings are set using SNMP messaging. Both of these protocols require an IP address to be assigned to the ASI2416.

13.4.1 ASI2416 Network Mask Assignment

The networking stack used for UDP messaging on the ASI2416 uses the following rules to automatically assign a NetMask:

Class A subnet: 1.0.0.0 - 127.0.0.0, Private allocation range: 10.0.0.0 - 10.255.255.255
NetMask : 255.0.0.0

Class B subnet: 128.0.0.0 - 191.255.0.0, Private allocation range: 172.16.0.0 - 172.31.255.255
NetMask : 255.255.0.0

otherwise
NetMask: 0.0.0.0

14 CONFIGURATION

14.1 ASIControl Configuration

ASIControl is available from the AudioScience Windows Driver's web page at [AudioScience](http://www.audioscience.com). It is embedded in AudioScience's standard audio drivers. If you are using an AudioScience CobraNet adapter or an AudioScience non-CobraNet audio adapter in the PC, choose the Wave, WDM, or Combo driver to download and install (the driver type to install will be dependant upon the application used with the AudioScience adapter; consult the manufacturer of the application). If you do not have an AudioScience adapter installed in the PC, simply download the Wave driver.

Run the driver .exe to install the driver components and be sure to select the second install option: "Standard PCI + Network Driver".

After driver installation, ASIControl can be run from either the desktop icon or from Start → All Programs → AudioScience → ASIControl.

If there is more than one NIC in the PC, upon startup, ASIControl will first prompt the user for which network interface to use to communicate with CobraNet devices. Next, to invoke the CobraNet Object Tools COM object to assign IP addresses to any CobraNet devices that do not currently have valid IP addresses, select Options→Configure CobraNet IP address range and fill in the range. (See the CobraNet section for further information.) ASI2416 communication with the host is via UDP messaging using port 44600.

To preserve control changes made to the ASI2416, ASIControl must be shut down. This will save control settings to ASI2416's flash memory, allowing the settings to be restored after a power cycle.

14.1.1 ASIControl Layout

List of CobraNet devices discovered on the network. Highlighted device is shown in the topology pane.

Topology pane showing the inputs and outputs of the selected adapter.

The node Line_Out 1 has been selected by left clicking with the mouse. Controls show up in the node pane.

Node pane shows the controls on the selected node; in this case Line_Out 1.

Type	Index	Name	Serial No.	Rev	Firmware	MAC Address	IP Address	sysDescription	sysName	sysLocation
	1100	ASI2416	26752	C0	3.07.19	00.60.28.06.D0.64	192.168.0.7	AudioScience ASI2416 Modular Inter...	ASI2416_M2	Tom's Office
	1132	ASI2416	33698	D0	3.07.19	00.60.28.06.D0.84	192.168.0.50	AudioScience ASI2416 Modular Inter...	ASI2416	

ASI2416

CobraNet_Ch 33 → Line_Out 1
CobraNet_Ch 34 → Line_Out 2
CobraNet_Ch 35 → Line_Out 3
CobraNet_Ch 36 → Line_Out 4
CobraNet_Ch 37 → Line_Out 5
CobraNet_Ch 38 → Line_Out 6
CobraNet_Ch 39 → Line_Out 7
CobraNet_Ch 40 → Line_Out 8
CobraNet_Ch 41 → Line_Out 9
CobraNet_Ch 42 → Line_Out 10
CobraNet_Ch 43 → Line_Out 11
CobraNet_Ch 44 → Line_Out 12
CobraNet_Ch 45 → Line_Out 13
CobraNet_Ch 46 → Line_Out 14
CobraNet_Ch 47 → Line_Out 15
CobraNet_Ch 48 → Line_Out 16

CobraNet_Ch 1 ← Line_In 1
CobraNet_Ch 2 ← Line_In 2
CobraNet_Ch 3 ← Line_In 3
CobraNet_Ch 4 ← Line_In 4
CobraNet_Ch 5 ← Line_In 5
CobraNet_Ch 6 ← Line_In 6
CobraNet_Ch 7 ← Line_In 7
CobraNet_Ch 8 ← Line_In 8
CobraNet_Ch 9 ← Line_In 9
CobraNet_Ch 10 ← Line_In 10
CobraNet_Ch 11 ← Line_In 11

SampleClock on Line_Out_1
Clock Source: Cobranet Sample Rate: 48000

AES/EBU Transmitter on Line_Out_1
AES/EBU Format
AES/EBU

Meter on Line_Out_1
Peak: -150.00dB RMS: -150.00dB

Volume on CobraNet_Ch_33 to Line_Out_1
0 -100 20 Autofade

Volume on Line_In_1 to Line_Out_1
-100 -100 20 Autofade

ASX V3.07.19, System V3.07.19

Figure 10. ASIControl layout

1.1.1 Adapter Information

This control displays information about the installed AudioScience product.

1.1.2 Interface

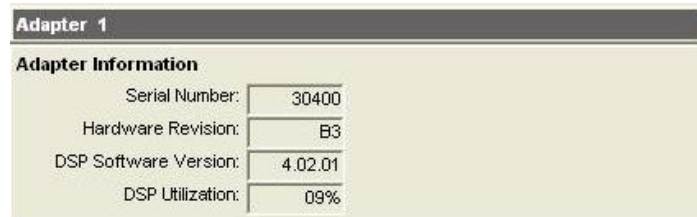


Figure 11. Figure 1. Adapter information seen in right side of ASiControl.

Serial Number:

The serial number is displayed here.

Hardware Revision:

This lists the hardware revision of the AudioScience product.

DSP Software Version:

The DSP software version is displayed; usually the same as the driver version installed.

DSP Utilization:

This shows the loading of the AudioScience product's DSP in percent.

Note: Utilization should be kept below 90%.

14.1.2 Level

The levels in dBu for the adapter's line_outs and line_ins can be adjusted here.

In the example below, the Line_Out 1 node in the topology view of ASiControl has been selected. Its Level will show up on the right side of ASiControl. The same is done for a Line_In to see its Level.

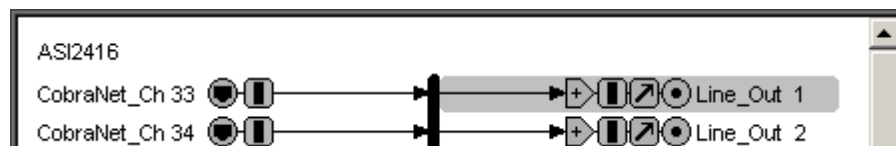



Figure 12. Using ASiControl to select Line_Out 1.

14.1.2.1 Interface



Figure 13. Level displayed by ASiControl for Line_Out 1

Level:

The line out level can be adjusted by clicking the  arrows or by typing values in to set the appropriate level. Consult the specification section of this datasheet for the range of supported levels.

14.1.2.2 Developer

14.1.2.2.1 Windows APIs

Wave/Mixer – Analog levels are controlled using mixerSetControlDetails() on a control of type signed and with the name Level/Trim.

HPI – Analog levels are controlled using the [HPI_LevelSet\(\)](#) API.

ASX – Analog level are controlled using the [ASX_Level_Set\(\)](#) API.

DirectSound – TBD.

14.1.2.2.2 Linux APIs

HPI – Analog levels are controlled using the [HPI_LevelSet\(\)](#) API.

ASX – Analog level are controlled using the [ASX_Level_Set\(\)](#) API.

ALSA – TBD.

14.1.3 Meter

Meters in ASIControl are located on audio nodes and display the audio level as the audio signal passes through the node. Most AudioScience devices return both RMS and peak level readings and ASIControl displays both simultaneously. The ASI2416 has mono (single channel) peak meter, so only a single bar is displayed in that instance.

14.1.3.1 Interface

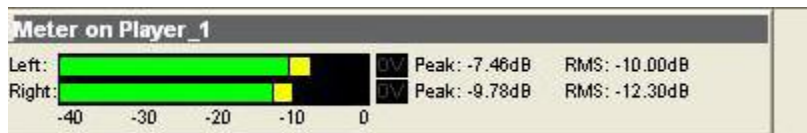


Figure 14. A stereo peak meter display. The RMS is the green bar and the peak is the yellow bar.

To the right of the peak meter is the absolute readings in dBFS. These can be useful when testing input tones of a specific known level.

14.1.3.2 Developer

14.1.3.2.1 Windows APIs

Wave/Mixer – Meters are read using mixerGetControlDetails() on a control of type signed and with type “Peak” the name “Peak Meter”. A minimum value is 0 and maximum is 32767. The interface returns the peak readings only, not the RSM level. It confirms to expected Windows functionality.

HPI – Meters are read using the [HPI_Meterxxx\(\)](#) API.

ASX – Meters are read using the [ASX_Meter_xxx\(\)](#) API.

DirectSound – TBD.

14.1.3.2.2 Linux APIs

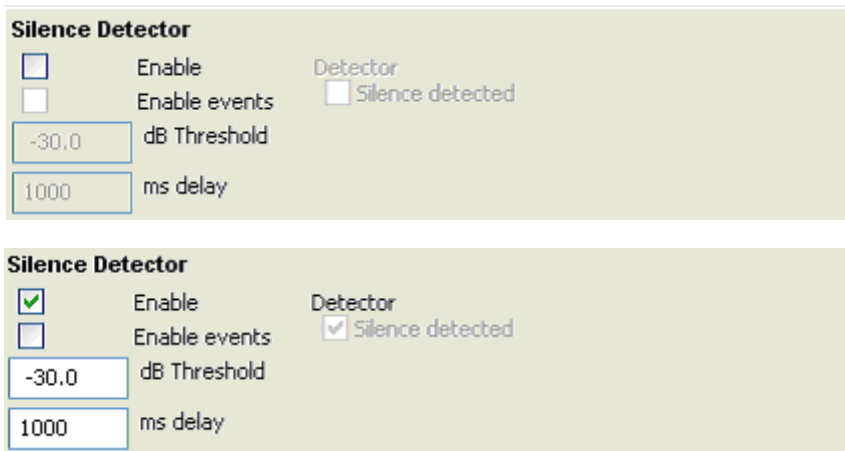
HPI – Meters are read using the [HPI_Meterxxx\(\)](#) API.

ASX – Meters are read using the [ASX_Meter_xxx\(\)](#) API.

ALSA – TBD.

14.1.4 Silence Detector

For analog out and AES/EBU out modules, a silence detector is available. Clicking on an Analog_Out or AES/EBU_Out in the node pane of ASIControl will bring up a list of volumes in the node control pane. Scroll to the bottom of the volumes for the silence detector, shown in the image below. Check Enable to allow silence detection and to input dB threshold and ms delay levels, and to enable events, as shown in the image below.



Silence Detector

☐ Enable ☐ Enable events ☐ Silence detected

dB Threshold

ms delay

Silence Detector

☒ Enable ☒ Enable events ☒ Silence detected

dB Threshold

ms delay

Figure 15. Silence Detector, disable and enable, as seen in ASIControl

14.1.5 Setting AES/EBU Clocking

The following diagram shows the sample rate-clocking scheme for ASI2416 models configured with AES/EBU modules. For example, the ASI2416–4400 has inputs and outputs 1...8, while the ASI2416–4000 model has inputs and outputs 1...4. By default AES/EBU output clocking is set to the local module clock running at 44.1 kHz.

NOTE: Each AES/EBU input and output carries two channels of audio.

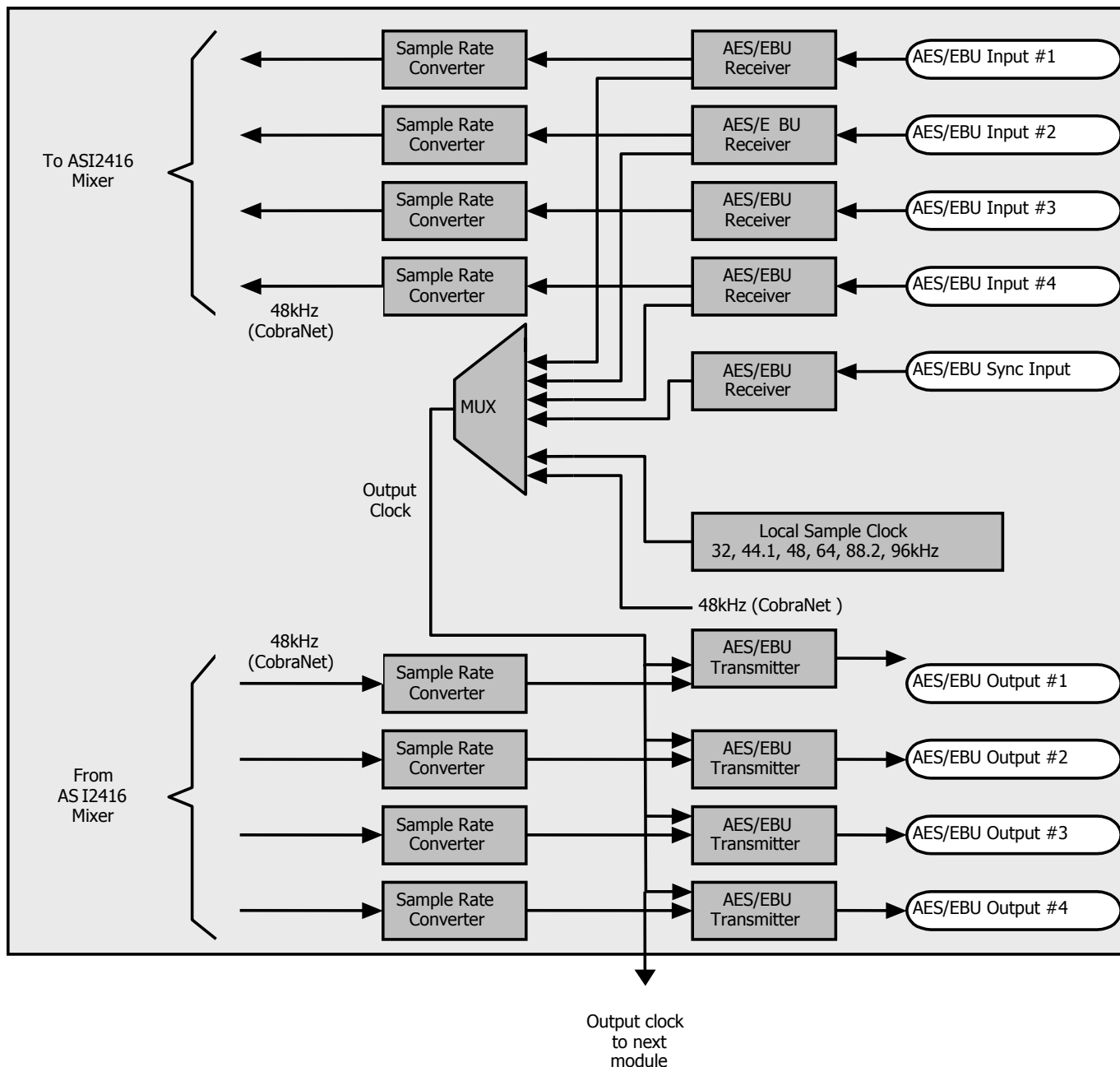


Figure 16. ASI2416 AES/EBU clocking for Module 1.

The figure below highlights the fact that the ASI2416 can receive and transmit AES/EBU at almost any sample rate.

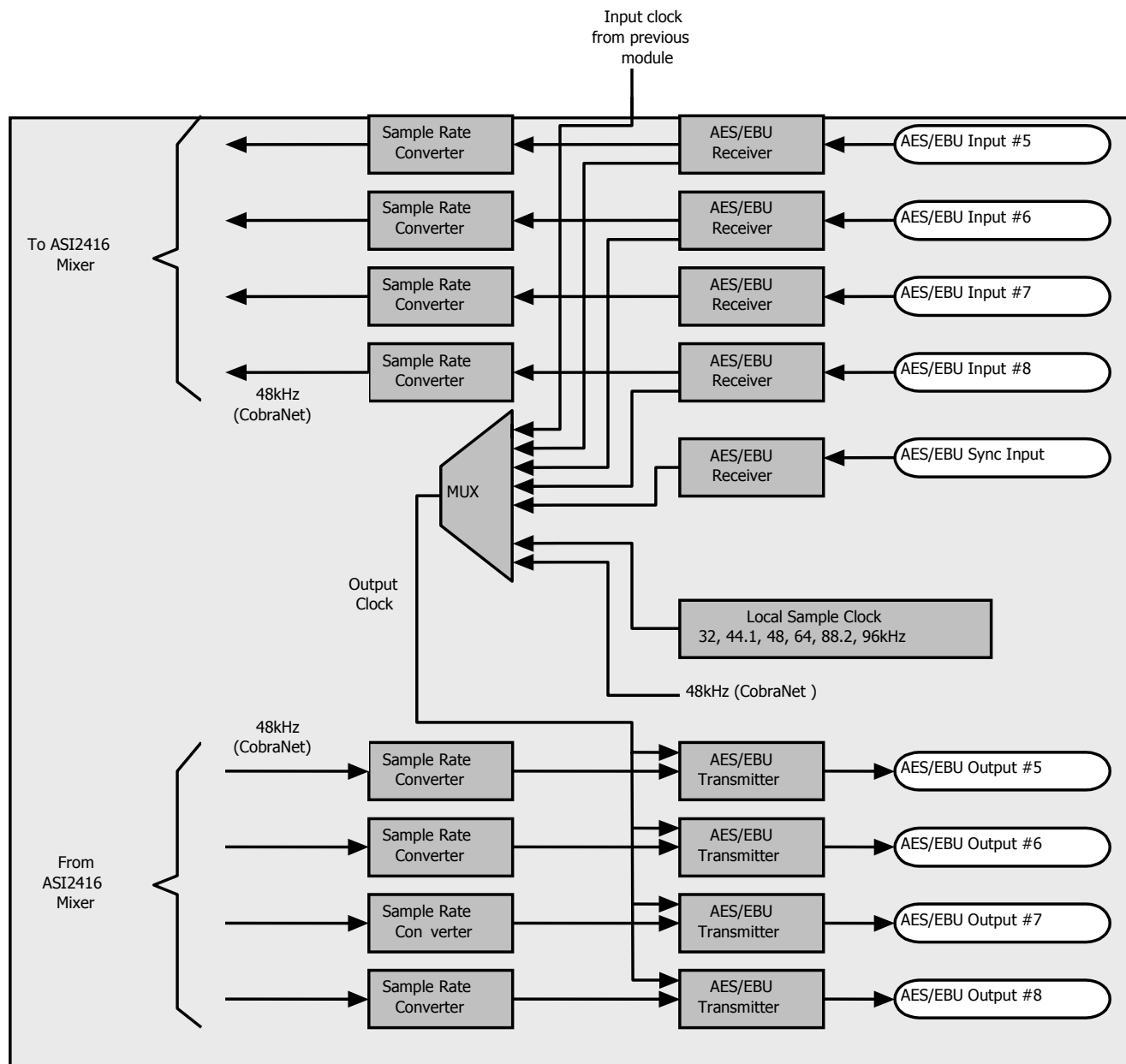


Figure 17. ASI2416 AES/EBU clocking for Module 2.

14.1.5.1 AES/EBU Inputs

Each AES/EBU input has a sample rate converter (SRC) on it and so may have a sample rate that is asynchronous to the rest of the system. Valid sample rates are 32, 44.1, 48, 64, 88.2 and 96kHz.

There is nothing to set for the AES/EBU inputs since the sample rate conversion to the reference CobraNet clock takes place automatically in hardware.

14.1.5.2 AES/EBU Outputs

There are the five sample rate sources for clocking the AES/EBU outputs:

1. A sample rate from the ASI1441 (AES/EBU) module in the previous slot.
2. A sample rate derived from any of the four AES/EBU inputs on that module (32, 44.1, 48, 64, 88.2 or 96kHz)
3. A sample rate derived from the AES/EBU Sync input (32, 44.1, 48, 64, 88.2 or 96kHz)
4. A local sample rate clock that may be set to 32, 44.1, 48, 64, 88.2 or 96kHz.
5. The ASI2416 CobraNet clock (48kHz)

14.1.5.3 Setting the AES/EBU Tx clock source

The ASI2416 AES/EBU output clock is controlled by a Sample Clock control on the first output of each AES/EBU module. The first step in setting the AES/EBU output clocking is to select Line_Out 1 in ASIControl.

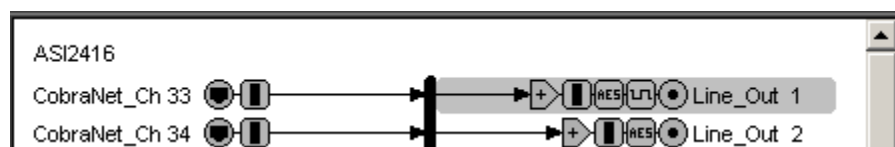


Figure 18. Selecting Line_Out 1 in ASIControl prior to setting Tx clock source.

The node pane should now display the following:

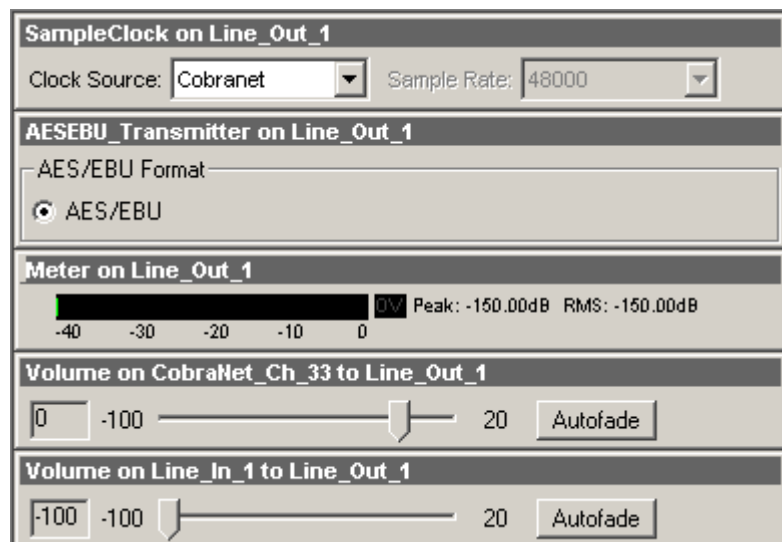


Figure 19. ASIControl display of AES/EBU Line_Out 1 node controls.

Click on the “Clock Source” combo box to select the desired source. See illustration below.

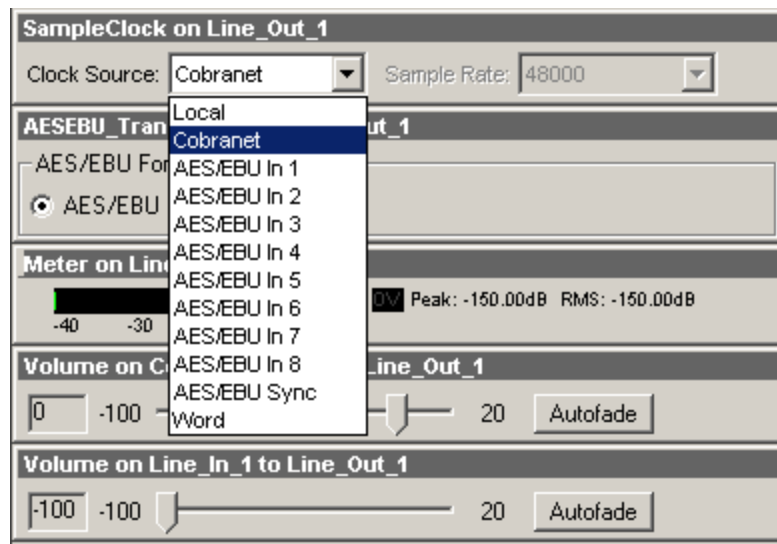


Figure 20. ASIControl selection of AES/EBU output clock source.

Choose the desired clock source. Since AES/EBU inputs are actually stereo and come in on the same wires, the following inputs are paired together, and will use the same clock (1 & 2), (3 & 4), (5 & 6), (7 & 8).

The output clocking for the second set of 8 AES/EBU outputs is controlled independently from the first 8 outputs. It is accessed via Line_Out 9 (the first output on the second module). When clicking the “Clock Source” combo box for Line_Out 9 an additional option for “Previous Module” will be presented.

14.1.5.4 Common AES/EBU clocking configurations

This section outlines some common AES/EBU clocking configurations.

14.1.5.4.1 Everything running at 44.1 kHz local clock – no sync required

This is the default ASI2416 configuration. There is nothing to do.

14.1.5.4.2 All outputs running at AES/EBU house sync

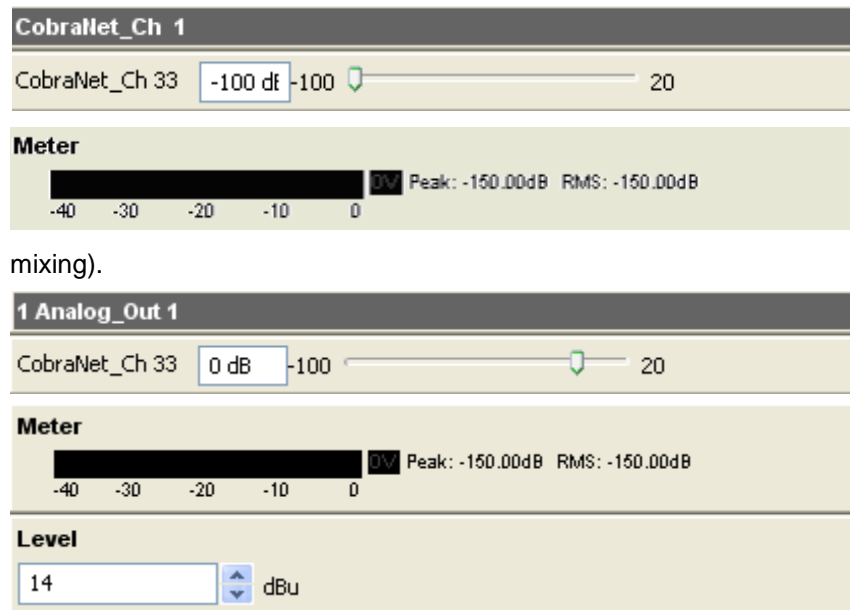
Connect the house AES/EBU to the AES/EBU sync input of the first AES/EBU module. Using ASIControl set the SampleClock source to be “AES/EBU Sync”. This will sync all outputs on the first module to AES/EBU. To sync the outputs on the second module as well, go to Line_Out 9 and set the SampleClock source to “Previous Module”.

14.1.5.4.3 All outputs sync'd to one of the AES inputs

Connect the house AES/EBU to the AES/EBU sync input of the first AES/EBU module. Using ASIControl set the SampleClock source to be “AES/EBU In 1”. This will sync all outputs on the first module to AES/EBU In 1. To sync the outputs on the second module as well, go to Line_Out 9 and set the SampleClock source to “Previous Module”.

14.1.6 Input and Output Volume Adjustment

All outputs from the ASI2416 have volume adjustments in their path that support a range of -100 to $+20$ dB. The nodes that support this are CobraNet_Ch 1-16 and Line_Out 1-16.



Clicking on CobraNet_Ch 1 in the topology pane of ASIControl will show a list of volumes in the node view pane. Below is an image of the first volume shown in the node pane. The meter is found after the list of volumes (the ASI2416 incorporates AudioScience's 'anything to anywhere' mixing).

Clicking on Analog_Out 1 in the topology pane of ASIControl will show a list of volumes in the node view pane. Below is an image of the first volume shown in the node pane. The meter and level is found after the list of volumes (the ASI2416 incorporates AudioScience's 'anything to anywhere' mixing).

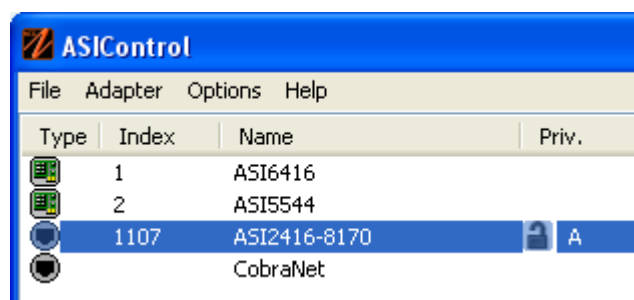
Figure 21. ASIControl node displays with volumes.

The volumes are self-explanatory. Just drag the sliders. All lineouts also have an audio path (with volume) from the corresponding line in. This can be use useful in verifying the correct operation of the audio modules without having to send the audio across a CobraNet network.

C:\asi\doc\common\asicontrol_cobranet_clocksource.doc

1.1.1 Access control using passwords

Beginning with driver 4.10.00, some AudioScience adapters support password protected access to adapter controls. In ASIControl, an adapter that supports passwords shows a padlock in it's adapter information line of the adapter list window. For example see



By default, if a password has not been set, the adapter operates as if there is no active password. Any user has complete access to all the device functionality.

The access control system supports 3 different “user” login levels. They levels their associated privileges are outlined in the following table.

Username	Controls	Scripting	Configuration Save/restore	Passwords
Admin	Read/write	Read/write	Read/write	Write
User	Read	No access	No access	No access
Guest	Read	No access	No access	No access

Password information is stored on the adapter itself, not the host computer, so if a different computer is used to control the adapter, the same passwords should be used.

1.1.1.1 Login in states

1.1.1.1.1 Admin

This is the default state if no passwords have been set on the device. Or, the user has logged in using the Administrator password. This is indicated in the ASIControl as:



1.1.1.1.2 User

The user successfully has logged in using the User password. This is indicated in the ASIControl as:



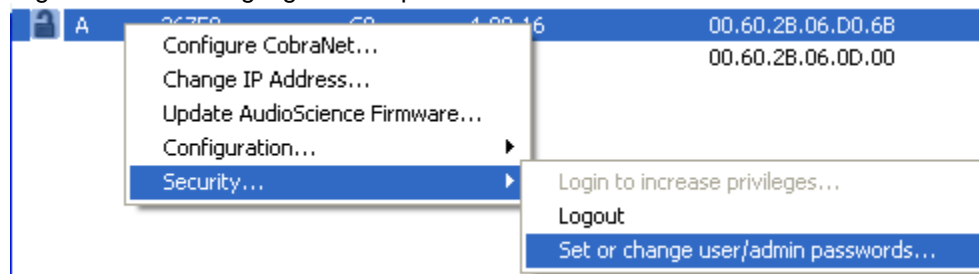
1.1.1.1.3 Guest

If passwords have been set, but the user has not logged on, the Guest privilege level is invoked. This is indicated in the ASIControl as:

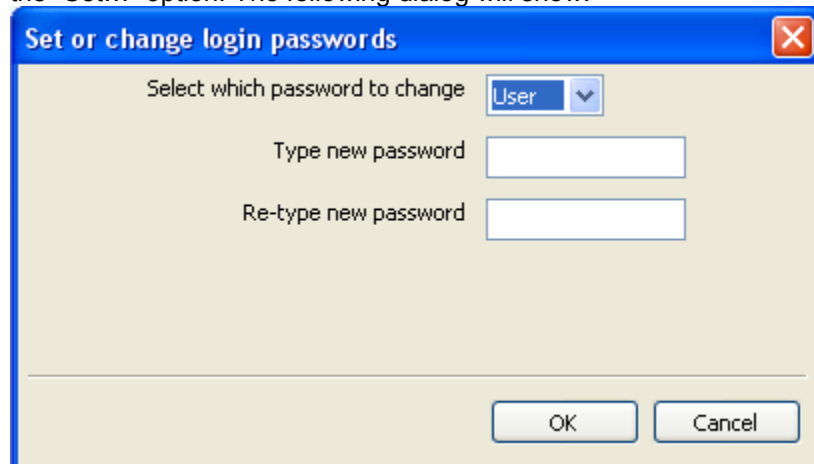


1.1.1.2 Setting passwords

Right click on the highlighted adapter and follow the menu tree to



the “Set...” option. The following dialog will show.



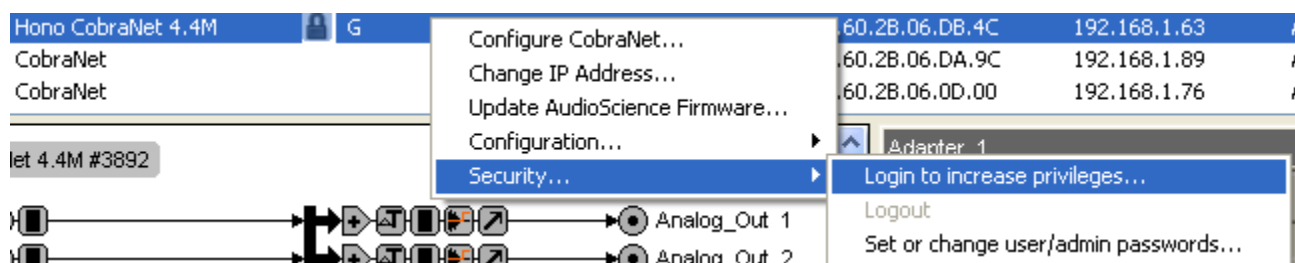
Select User or Admin from the first combo box and enter the new password in the two password fields.

Upon setting a password for the first time on a device that has never had a password before, Admin rights are assumed to be in place for the current user, so after setting the password the user should logout if they wish the adapter to be secured.

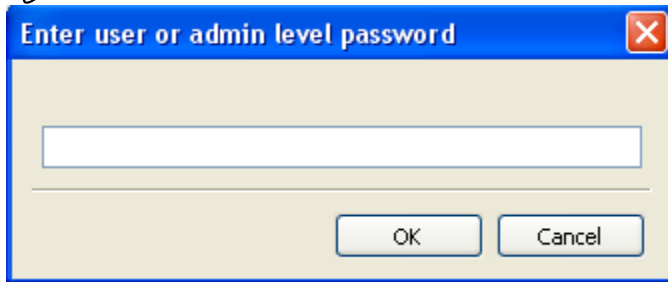
Note that on an PCI or PCIe adapter that has password support, the user should not power down or restart the PC within 30 seconds of changing the passwords. In other words, permanent storage of the updated passwords settings can take up to 30 seconds.

1.1.1.3 Logging in

Right click on the adapter in the list to login and increase user rights.

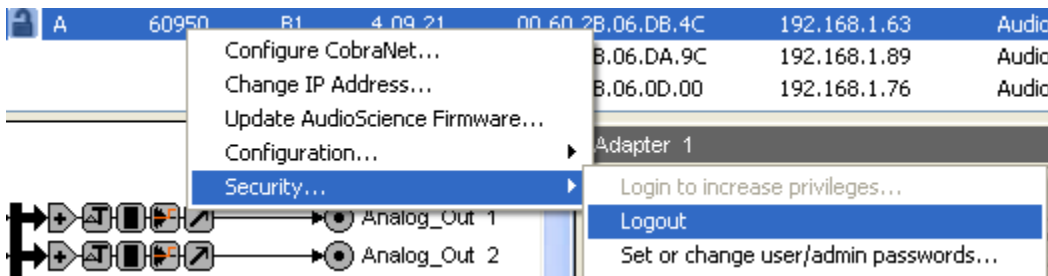


and enter the password for the access privilege level that you wish to use.



1.1.1.4 Logging out

After completing privileged operations, logout is performed by right clicking on the adapter and selecting the logout option.



1.2 AudioScience Lua implementation

Starting with driver release 4.10.00, Lua scripting is supported on many AudioScience networked audio interfaces.

The AudioScience lua implementation supports the standard lua environment (see References section). However, the following libraries are not available:

- mathematical functions (sin, log, etc.)
- package library
- string manipulation
- table manipulation
- input and output
- operating system facilities
- debug facilities

Note1: print() is supported when running a script in ASIControl. On the AudioScience network node the print() command does not perform any action.

Note2: All lua indices start at “1” so the first opto would be referred to as opto[1].

1.2.1 Language Extensions

AudioScience’s lua implementation adds a library called “asi” to the runtime environment. Since the library is loaded by AudioScience’s runtime environment, there is no need to use the lua module command to load the asi library.

1.2.1.1 Method asi.new()

Creates a handle to the specified object.

1.2.1.1.1 Parameters

A URI string specifying the object to use.

1.2.1.1.2 Return

A lua object of type metatable. Optionally, a lua table of metatables is returned if the URI specifies an indexed object (like GP In optos).

1.2.1.1.3 Example

```
local_ip = "192.168.1.53"
opto = asi.new("hpi://"..local_ip.."/adapter/gpio/inputs")
silence_detector =
    asi.new("hpi://"..local_ip.."/analog_out/1/silence_detector/state")
```

1.2.1.2 Method asi.get()

Reads the current setting of the object.

1.2.1.2.1 Parameter

An object (metatable) that was created using asi.new() as specified above. Using lua's implied reference to self object, asi.get(obj) can be expressed as obj:get().

1.2.1.2.2 Return

The current state of the object and a Boolean changed flag. The type of state depends on the object referenced.

1.2.1.2.3 Example

```
state,changed = opto[1]:get()
```

1.2.1.3 Method asi.set()

Sets the value of specified object.

1.2.1.3.1 Parameters

An object (metatable) that was created using asi.new() as specified above and the value to set. Using lua's implied reference to self object, asi.set(obj,value) can be expressed as obj:set(value).

1.2.1.3.2 Return

None.

1.2.1.3.3 Example

```
relay[1]:set(state)
```

1.2.1.4 Method asi.events()

Returns true while there are events to process. This is used to bracket an infinite while loop that reads inputs, performs logical operations, and sets outputs.

1.2.1.4.1 Parameter

None.

1.2.1.4.2 Return

True or false.

1.2.1.4.3 Example

```
while ( asi.events() ) do
-- read inputs
-- perform logic
end
```

1.2.1.5 Method `asi.time()`

Reads the current time in milliseconds. This returns a 32-bit number (counter) that wraps around, so care must be taken when doing operations on the returned time value. See `asi.difftime()` and `asi.addtime()` methods. Although the timer has a resolution of a millisecond, lua scripts are not run every millisecond. Greater than a duration comparisons should therefore be used, not equal to comparisons.

1.2.1.5.1 Parameter

None.

1.2.1.5.2 Return

Time stamp in millisecond ticks.

1.2.1.5.3 Example

```
current_time = asi.time()
```

1.2.1.6 Method `asi.difftime()`

Takes the difference between time objects that both represent a timestamp in milliseconds.

1.2.1.6.1 Parameters

Two timestamps, A and B.

1.2.1.6.2 Return

The time difference in milliseconds. If input parameters are called A and B, then the returned time difference is A – B.

1.2.1.6.3 Example

```
time_reference = asi.time()

while ( asi.events() ) do

    current_time = asi.time()

    -- have 3 seconds passed ?
    if asi.difftime( current_time, time_reference) > 3000 then
        -- take action(s)
        -- update the time reference for next period
        time_reference = asi.addtime(time_reference,3000)
    end
end

end
```

1.2.1.7 Method `asi.addtime()`

Add two millisecond timestamps together and return the result.

1.2.1.7.1 Parameter

Two timestamps, A and B.

1.2.1.7.2 Return

The sum of 2 timestamps in milliseconds. If parameters are A and B, returned time sum is A + B. The return timestamp may be less than both A and B if the sum causes the 32-bit unsigned timestamp presentation to wrap.

1.2.1.7.3 Example

```
time_reference = asi.time()
while ( asi.events() ) do

    current_time = asi.time()

    -- have 3 seconds passed ?
    if asi.difftime( current_time, time_reference) > 3000 then
        -- take action(s)
        -- update the time reference for next period
        time_reference = asi.addtime(time_reference,3000)
    end
end
end
```

1.2.2 Script Layout

Language extensions outlined in the previous section shape the structure of lua script running on an AudioScience network node. Scripts are broken down into an opening section followed by a looping section.

1.2.2.1 Opening section

The opening section uses the `asi.new()` call to map AudioScience control objects to lua objects that can be manipulated later on the script. All calls to create lua objects should be completed before the input processing section.

Any required initial operating conditions can also be set before the following looping section is entered. For example, all input microphones can be muted.

1.2.2.2 Looping section

The looping section is where the processing occurs. Inputs are read and actions taken according to the rules and conditions written into the lua script. Looping continues until the script is halted.

1.2.3 Examples

1.2.3.1 Single opto input to relay output

Define the IP address of the ASI2416 that will run the script.

```
ip = "192.168.1.53"
```

Open the opto and relay handles.

```
opto = asi.new("hpi://"..ip.."/adapter/gpio/inputs")
relay = asi.new("hpi://"..ip.."/adapter/gpio/outputs")
```

Create the processing loop.

```
while ( asi.events() ) do
    state, changed = opto[1]:get()
    if changed then
        relay[1]:set(state)
    end
end
```

Specify opto #1

Specify relay #1

Final code:

```
-- example, single opto input to relay output
local_ip = "192.168.1.53"
opto = asi.new("hpi://"..ip.."/adapter/gpio/inputs")
relay = asi.new("hpi://"..ip.."/adapter/gpio/outputs")
while ( asi.events() ) do
    state, changed = opto[1]:get()
    if changed then
        relay[1]:set(state)
    end
end
end
```

1.2.3.2 Single opto input to multiple relay outputs

This example uses the same object handles as the previous example. The action part of the processing loop is altered to set relay outputs 1-4 instead of just relay output number 1.

Final code:

```
-- example, single opto input to multiple relay outputs
ip = "192.168.1.53"
opto = asi.new("hpi://"..ip.."/adapter/gpio/inputs")
relay = asi.new("hpi://"..ip.."/adapter/gpio/outputs")
while ( asi.events() ) do
    state, changed = opto[1]:get()
    if changed then
        relay[1]:set(state)
        relay[2]:set(state)
        relay[3]:set(state)
        relay[4]:set(state)
    end
end
end
```

1.2.3.3 Single opto input with pulse extended relay output

Pulse extending an opto input requires the use of timing information within the lua script. The current time stamp is queried using:

```
current_time = asi.time()
```

When the opto is triggered we record the current time along with the fact that we are now pulse extending using the following code snippet:

```
time_reference = current_time
pulse_extending = true
```

Future iterations of the script loop examine the elapsed time and take appropriate actions using the following code (assuming a 10 second pulse extension):

```
if asi.difftime( current_time, time_reference ) > 10000 then
    pulse_extending = false
    relay[1]:set(false)
end
```

Final code:

```
-- example, single opto input with 10 second pulse extended relay output
ip = "192.168.1.53"
opto = asi.new("hpi://"..ip.."/adapter/gpio/inputs")
relay = asi.new("hpi://"..ip.."/adapter/gpio/outputs")
pulse_extending = false
while ( asi.events() ) do
    current_time = asi.time()
    state, changed = opto[1]:get()
    if changed then
        if state then
            time_reference = current_time
            pulse_extending = true
            relay[1]:set(true)
        end
    else
        if pulse_extending then
            if asi.difftime( current_time, time_reference ) > 10000 then
                pulse_extending = false
                relay[1]:set(false)
            end
        end
    end
end
end
```


1.2.3.4 Single opto input to relay output on remote device

Interfacing to a remote device requires only an IP address change in the string passed in to `asi.new()`. For this example, the IP address of the remote device is defined as:

```
ip_other = "192.168.1.70"
```

An object handle to the GP Out on the remote device is obtained by going:

```
relay = asi.new("hpi://"..ip_other.."/adapter/gpio/outputs")
```

Putting it all together, final code becomes:

```
-- example, single opto input to single relay output on remote
device
ip = "192.168.1.53"
ip_other = "192.168.1.70"
opto = asi.new("hpi://"..ip.."/adapter/gpio/inputs")
relay = asi.new("hpi://"..ip_other.."/adapter/gpio/outputs")
while ( asi.events() ) do
    state, changed = opto[1]:get()
    if changed then
        relay[1]:set(state)
    end
end
end
```

1.2.3.5 Silence detector to relay output

A silence detector returns a true or false state, exactly the way an opto does. A handle to the silence detector state for analog output number 1 is obtained using:

```
silence_detect = asi.new("hpi://"..ip.."/analog_out/1/silence_detector/state")
relay = asi.new("hpi://"..ip.."/adapter/gpio/outputs")
```

Note, ASIControl should be used to enable the silence detector on as it is disabled by default. The control loop becomes:

```
while ( asi.events() ) do
    state, changed = silence_detect:get()
    if changed then
        relay[1]:set(state)
    end
end
end
```

1.2.3.6 Push to talk microphone

Read an opto and when the opto is trigger unmute the microphone.

```
opto = asi.new("hpi://"..ip.."/adapter/gpio/inputs")
mute = asi.new("hpi://"..ip.."/analog_in/1/volume/mute")
mute:set(false)
```

Note, ASIControl should be used to enable the silence detector on as it is disabled by default.

```
while ( asi.events() ) do
  state, changed = opto[1]:get()
  if changed then
    mute:set(state)
  end
end
```

1.2.3.7 Voice activated microphone(s)

When the input meter level measured in RMS exceeds a specified level, un-mute the microphone.

```
meter = asi.new("hpi://"..ip.."/analog_in/1/meter/rms")
mute = asi.new("hpi://"..ip.."/analog_in/1/volume/mute")
db_FS_mute_threshold = -20
```

Processing loop.

```
while ( asi.events() ) do
  rms = meter:get()
  mute:set( rms < db_FS_mute_threshold )
end
```

1.2.4 Using ASIControl to Write Scripts

Notes

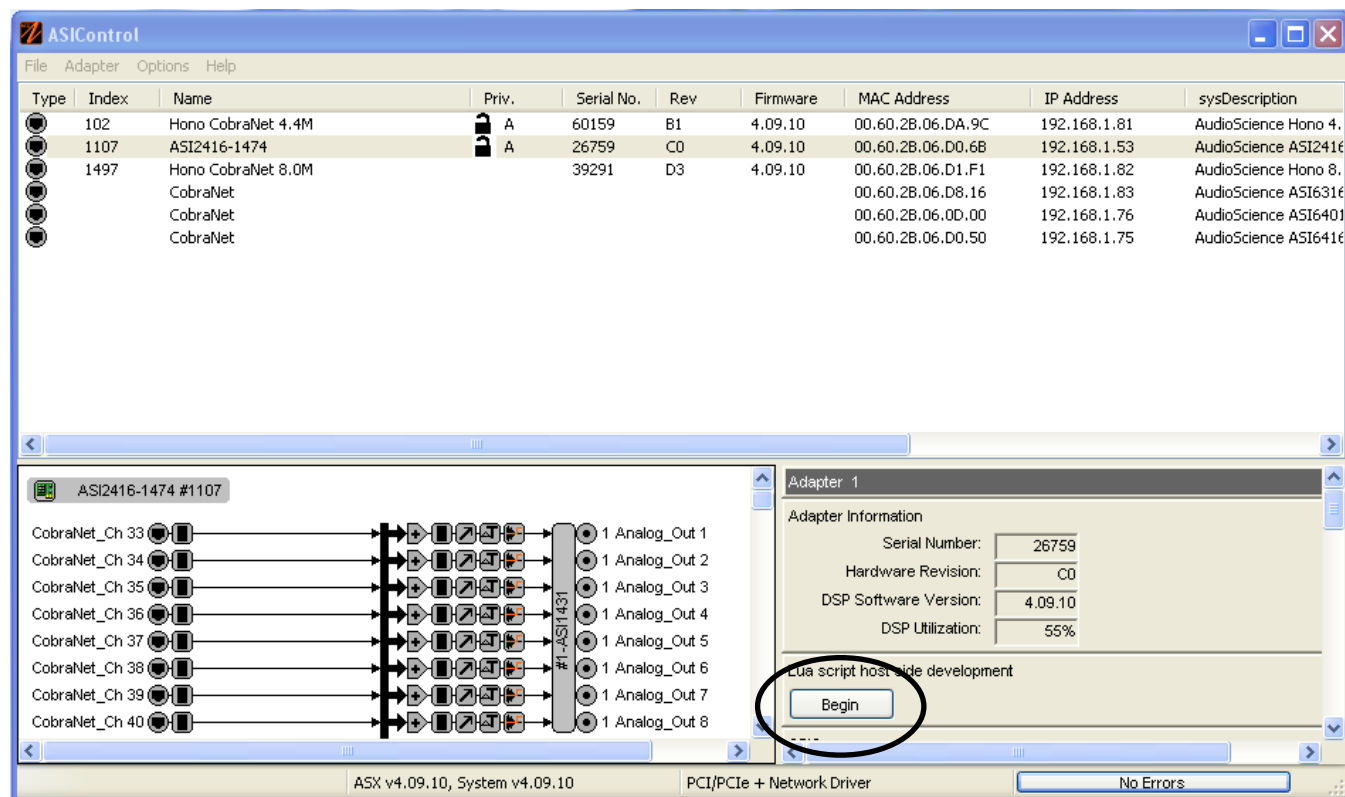
1. Static IP addresses are required for reliable lua script operation!
See the device's datasheet for instructions.
2. Lua bytecode needs to be generated from a 32-bit version of ASIControl. When using W7 64-bit, navigate to Programs→AudioScience→ ASIControl.

ASIControl embeds a text editor that can be used to create and edit lua scripts. The typical development sequence proceeds as follows.

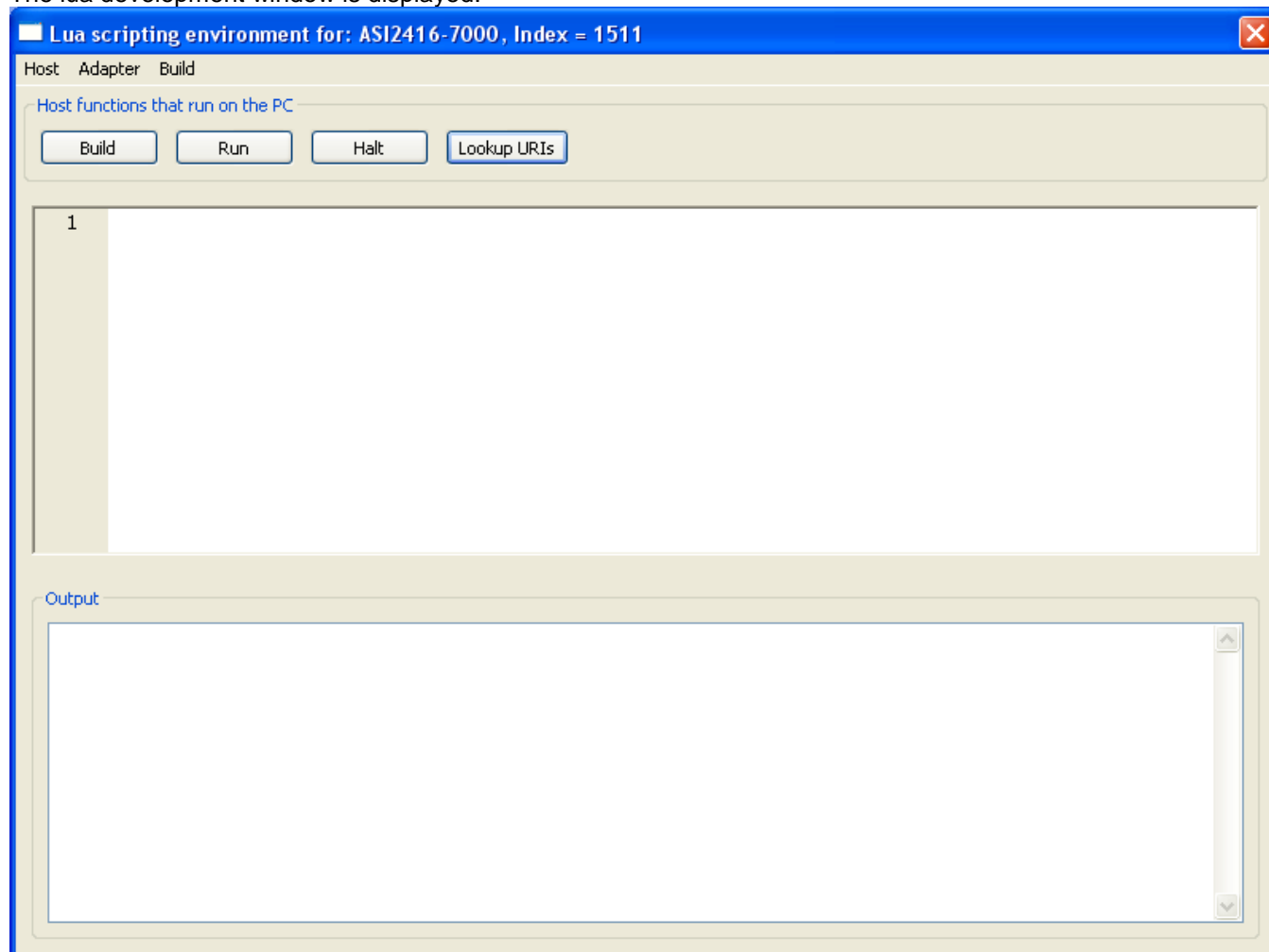
1. Design on paper what inputs should control what outputs for the entire system.
2. Design on paper what inputs should control what outputs for each device.
3. Create the .lua scripts for each device and run them on the host for testing purposes before downloading and running them on the network node.
4. Optionally, save the .lua source code on the network node.

1.2.4.1 Opening the script editor

Open the lua development environment by clicking on the Begin button as shown in the below figure.



The lua development window is displayed.



Starting from the top left we have a “File” menu for loading and saving lua source files to and from the host PC hard-drive. Next we have the “Adapter” menu which operates the device’s lua file storage. It supports saving lua source files and lua byte code to the network interface.

The “Host functions...” section controls host operations for script development. Operations are :

- Compile - converts lua source code to byte code and produces a .luac output file with the same name as the .lua script file. A compile operation must be performed before the script can be run.
- Run – starts a compiled lua script running on the host.
- Halt – stops a lua script running on the host. Since the structure of all 2416 lua scripts contains a repeating loop so that they keep running, a “Halt” operation is always required to terminate script execution.
- Lookup URIs – supports looking up uri strings for the currently attached network interfaces.

The upper of the 2 large windows above is the text editor, while the lower is output from the engine that runs the lua script.

1.2.4.2 Opening an existing lua source file from disk

Choose Host → File Open to open a lua source file from the local host PC's hard drive. lua source files are assumed to use the extension .lua.

1.2.4.3 Saving to disk

Choose Host → File Save to save the lua source file being edited to the local hard drive. Note that compiling the source will automatically save the source to the hard drive.

1.2.4.4 Saving script to AudioScience audio network node

Choose Adapter → Save source to save the local lua script to the remote network node

Note that there is no requirement to save the lua source to the network node for lua scripting to work correctly. Only the bytecode need be saved.

1.2.4.5 Reading script to AudioScience audio network node

Choose Adapter → Load source to retrieve lua script source from the remote node.

1.2.4.6 Running the script on the PC for testing purposes

The steps for developing a lua script on an AudioScience network interface are:

1. Start with a lua example that is close to what you are trying to implement. Load it into the editor.
2. Save it under a new name using the "Save As" option.
3. Take care to adjust IP address(es) .
4. Make functional changes as required. Use the "Lookup URIs" option as required to obtain object uris.
5. Compile the script.
6. Run the script. If there are syntax errors go back to step 4.
7. Try to manipulate inputs so that you can verify the script is working.
8. If all is working press "Halt".
9. Save the script to the network node.

1.2.4.7 Running the script on the network interface

ASiControl shows a “lua” block on network interfaces that support lua scripting. The names of lua source files and lua byte code files that are stored on the network interface are displayed:



Most lua control block functions are obvious. The Script Status option is used to control the script status and the Messages section shows message from the lua runtime engine running on the network node. If a script is selected to be run, the setting will be preserved through a power down cycle of the network interface.

1.2.5 Lua License

Copyright © 1994–2011 Lua.org, PUC-Rio.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2.6 References

For more information on lua, see <http://www.lua.org/>.

14.2 CobraNet Configuration

ASiControl can be used to configure both AudioScience CobraNet devices and third party CobraNet devices, via its CobraNet Configuration dialog box. This dialog box can be used to configure CobraNet devices and assign names, locations, bundle numbers, channel assignments, etc. Third party applications such as CobraNet Discovery can also be used to configure AudioScience CobraNet devices.

Below lists how to set up AudioScience's software so ASiControl may be used to configure CobraNet.

NOTE: Under some Windows operating systems, ASiControl must be run as administrator in order to see all networked CobraNet devices. In Windows 7 this is done by right clicking on the ASiControl shortcut on the desktop and selecting "Run as administrator."

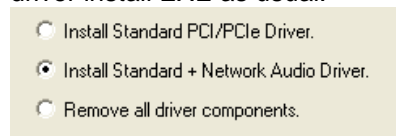
NOTE: Some antivirus programs and/or firewalls can prevent ASiControl from showing all CobraNet devices on a CobraNet network. If all CobraNet devices are not showing in the top pane of ASiControl, please check settings and/or turn off any antivirus programs or firewalls that may be running on the PC.

14.2.1 Setting Up AudioScience Software to Configure CobraNet

To configure an AudioScience CobraNet devices using ASiControl, the following must be in place: the AudioScience network driver must be installed, the networked adapters option must be selected in ASiControl, the AudioScience CobraNet device must be connected to the CobraNet network via an Ethernet cable, and the PC that is running ASiControl must also be connected to the same CobraNet network.

14.2.1.1 Install the Network Driver

When an AudioScience driver install EXE is run, three choices are made available as shown in the image below. Select "Install Standard + Network Audio Driver" when installing AudioScience CobraNet products, then run the driver install EXE as usual.



14.2.1.2 Select Networked Adapters in ASiControl

In ASiControl, go to the menu item Options, then select Configure adapter interface, as shown below.



Since ASiControl is used for both AudioScience's non-CobraNet adapters and CobraNet adapters, the correct interface must be selected when using ASiControl for CobraNet configuration. Click on menu item Options, then Configure adapter interface. The Interface Selection dialog box opens; the default selection is "Local PCI(e) adapters. Select 'Local PCI(e) + Networked adapters.' then click OK.



ASiControl will restart, looking for all CobraNet devices on the network, assigning IP addresses where needed, and showing CobraNet devices on the network in its top pane.

14.2.1.3 Connect CobraNet Device to the Network

Plug one end of an Ethernet cable into the back of the AudioScience CobraNet device. Plug the other end into the CobraNet or test network.

14.2.1.4 Connect PC to the Network

CobraNet control is via SNMP. In order for ASIControl to establish an SNMP connection to an AudioScience (or other) CobraNet device(s) for configuring, a valid network path must be established between the PC running ASIControl and the AudioScience (or other) CobraNet device(s). This is done by plugging one end of an Ethernet cable into the back of the PC's network interface card (NIC) and plugging the other end into the CobraNet or test network.

If the PC running ASIControl must be connected to an office network and if the CobraNet network data is required to be separate from the office network, another NIC must be added to the PC; one for the office and one for CobraNet.

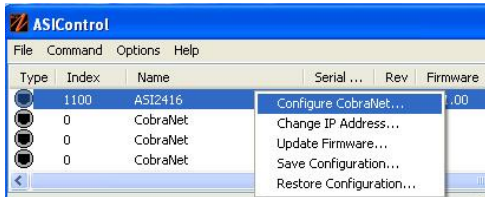
If the PC is on an office network and the CobraNet network data is not required to be separate from the office network, the PC can temporarily be removed from the office network and plugged into the CobraNet or test network. After the CobraNet configurations are made, the PC can be taken off the CobraNet network and placed back on the office network. **Note:** If any CobraNet configurations need to be changed in the future, the PC must be placed back on the CobraNet network in order to establish the SNMP connection.

If the PC does not need to be on an office network, plug it's NIC into the CobraNet or test network and either leave it there or remove it after configuring CobraNet. If it's removed, the PC must be connected to the CobraNet or test network again if CobraNet configurations need to be changed in the future.

Note: If plugging an AudioScience CobraNet device into an already existing CobraNet network, the PC must have the same IP address range as the CobraNet network in order for it to assign the AudioScience CobraNet device an IP address in the same range. Alternatively, if another CobraNet set-up application was used such as CobraNet Discovery, the same application may be used to assign an IP address and configure AudioScience CobraNet devices.

14.2.2 Configuring CobraNet Using ASIControl

The CobraNet Configuration dialog box is accessed via ASIControl, which is installed when the driver EXE is run. Double click on the ASIControl icon on the desktop, then in the top section right click on the CobraNet device and select Configure CobraNet... (see further for information on the other selections available):



The CobraNet Configuration dialog box opens:

Transmitter	Bundle	1	2	3	4	5	6	7	8	Multicast Mode	Format	Dropouts	Receivers
1	1000	1	2	3	4	5	6	7	8	Unicast 1	20	1	1
2	1001	9	10	11	12	13	14	15	16	Unicast 1	20	1	1
3	0	1	2	3	4	5	6	7	8	Unicast 1	20	0	0
4	0	9	10	11	12	13	14	15	16	Unicast 1	20	0	0

Receiver	Bundle	1	2	3	4	5	6	7	8	Status	Dropouts
1	0	33	34	35	36	37	38	39	40	Not Receiving	
2	0	41	42	43	44	45	46	47	48	Not Receiving	
3	0	0	0	0	0	0	0	0	0	Not Receiving	
4	0	0	0	0	0	0	0	0	0	Not Receiving	
5	0	0	0	0	0	0	0	0	0	Not Receiving	
6	0	0	0	0	0	0	0	0	0	Not Receiving	
7	0	0	0	0	0	0	0	0	0	Not Receiving	
8	0	0	0	0	0	0	0	0	0	Not Receiving	

14.2.2.1 CobraNet Configuration Interface

There are three main sections of the dialog box: the top shows the configuration section of the CobraNet device, the transmitter section is in the middle, and the receiver section is at the bottom. There are also four user control buttons on the bottom right.

By default the ASI2416 comes up with the following default CobraNet settings:

sysDescription: AudioScience ASI2416 Modular Interface CobraNet version 2.11.4(3) CS181022 (read only)

sysName: ASI2416

sysLocation: blank

sysContact: blank

Mode and Rate: 20bits/sample, 48 kHz, 5 1/3 ms

Persistence: off

Transmitter defaults:

Transmitter	Bundle	Ch 1 map	Ch 2 map	Ch 3 map	Ch 4 map	Ch 5 map	Ch 6 map	Ch 7 map	Ch 8 map	Format	Count	UniCast Mode
TX1	0	1	2	3	4	5	6	7	8	20 bits	8	Never
TX2	0	9	10	11	12	14	14	15	16	20 bits	8	Never
TX3	0	1	2	3	4	5	6	7	8	20 bits	8	Never
TX4	0	9	10	11	12	14	14	15	16	20 bits	8	Never

Receiver defaults:

Receiver	Bundle	Ch 1 map	Ch 2 map	Ch 3 map	Ch 4 map	Ch 5 map	Ch 6 map	Ch 7 map	Ch 8 map	Format
RX1	0	33	34	35	36	37	38	39	40	undefined
RX2	0	41	42	43	44	45	46	47	48	undefined
RX3	0	0	0	0	0	0	0	0	0	undefined
RX4	0	0	0	0	0	0	0	0	0	undefined
RX5	0	0	0	0	0	0	0	0	0	undefined
RX6	0	0	0	0	0	0	0	0	0	undefined
RX7	0	0	0	0	0	0	0	0	0	undefined
RX8	0	0	0	0	0	0	0	0	0	undefined

14.2.2.1.1 Persistence

In most real world deployments the persistence bit should be set to on so that the ASI2416 will remember its CobraNet settings and IP address in the event of a power cycle. If the persistence bit is left off, bundle numbers, sysName and sysLocation will be set back to default values on power cycle.

14.2.2.1.2 Simple crossover example

The ASI2416 can be configured using a laptop and an Ethernet crossover cable. The laptop's network adapter must have a static IP address.

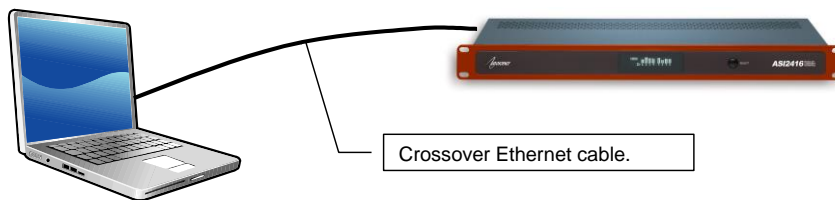


Figure 23. Laptop to ASI2416 connection.

14.2.2.1.3 CobraNet Device Configuration Section
MAC Address:

Read only; displays the MAC address of the CobraNet device.

IP Address:

Read only, displays the assigned IP address of the CobraNet device.

SysName:

Editable; a meaningful name can be given to a CobraNet device here.

SysLocation:

Editable; a physical location can be listed here for ease of reference.

SysDescription:

Read only; lists the type of CobraNet interface.

Sample Rate:

48k (default) or 96 kHz. A CobraNet device must run at one sample rate only. Two CobraNet devices on the same network can each operate at a different sample rate.

Latency:

Choose between 1.33, 2.67, or 5.33 (default) ms.

Persistence:

On or off. When set to on, settings are saved to flash and will be restored after a restart.

Conductor Priority:

Editable; set a low number to ensure the CobraNet device will be a Performer or set it to a high number to ensure it will be a Conductor. The following are Cirrus numbering conventions:

- 0 = never conductor
- 32 = normal conductor priority (the CobraNet device will either be the Conductor or the Performer)
- 128 = high conductor priority (the CobraNet device will be the Conductor)

Conductor Status:

Read only; displays if the CobraNet device is a Conductor or Performer.

Error Code:

Currently blank. Will show CobraNet-specific error information in future firmware versions.

Count:

Displays the number of errors that have occurred on the device. Some errors result from normal operation; disconnecting cables, rerouting, etc. The text field to the right of Count displays the Cirrus-related text.

14.2.2.1.4 Transmitter Section**Transmitter:**

Sends an audio bundle, made up of up to 8 audio channels, to a CobraNet receiver on the network.

Bundle:

The basic CobraNet audio routing component, can have 0 to 8 audio channels. Assign specific bundle numbers here. Bundle numbers can be 0 and 1 – 65535

- 0 for turning transmitter off or for silence
- 1-255 for multicast
- 256-65279 for unicast
- 65280-65535 for private

1-8 Audio Channels:

The number of mapped channels from mixer to transmitter. Channels 0 (inactive) and 1-32 can be assigned. A 0 mapping of end channels 'shortens' the bundle; a mapping of 0 in the middle of mapping denotes silence.

Multicast Mode:

Double click to change setting in the dropdown list; use the following table to determine what setting to select:

Settings	Number of Receivers $\leq N$	Number of Receivers $> N$
Multicast Bundle # < 256	Multicast	Multicast
Always	Multicast	Multicast
Unicast N	Unicast	Unicast if Receiver = 1-N Fail if Receiver $> N$
Multicast Over N	Unicast	Multicast

Format:

Double click to change the audio sample format to 16, 20, or 24 bits.

Dropouts:

The number of times the channel transmission has been interrupted. Interruptions can be caused by loss of transmit permission from conductor (i.e. conflict with another transmitter set to the same bundle number) or by changes to txBundle.

Receivers:

The number of receivers requesting this bundle.

14.2.2.1.5 Receiver Section

Receiver:

Can receive from a CobraNet transmitter on the network one bundle made up of up to 8 audio channels.

Bundle:

The basic CobraNet audio routing component; assign a specific Bundle number here. A Receiver can carry 0 (channel will not be mapped to mixer) to 8 audio channels.

1-8 Audio Channels:

The number of audio channels in the bundle to be routed to the particular Receiver; channels 0 (inactive) and 33-64 can be assigned.

Status:

Displays 'Receiving' when a bundle is successfully routed to it.

Dropouts:

Displays number of times bundle reception has been interrupted. Interruptions can be caused by transmitter failure or by reconfiguring the receiver.

14.2.2.1.6 User Control Buttons

Refresh:

Updates fields and statuses. If clicked before Apply, clears any changes made.

Apply:

Accepts changes and leaves dialog box open.

OK:

Accepts changes and closes dialog box.

Cancel:

Does not accept any changes made and closes dialog box.

14.2.2.2 How To Configure and Transmit a Bundle

The minimum configuring needed to test basic functionality of sending and receiving a bundle is to set a transmitter bundle and channels on one CobraNet device, and then set the receiver of another CobraNet device to the same bundle number, plus assign its channel numbers.

For example, an ASI6416 is installed in a computer, ASIControl is opened and is used to play a stereo audio file in the ASI6416's Player 1. The Repeat box is checked to have continuous audio playing.

ASIControl is then used to set the ASI6416's first transmitter to have bundle number and channel assignments. Double clicking in receiver 1's bundle field makes it editable and 255 (for example) is typed in. Double clicking in the receiver 1, channel 1 box makes it editable and 1 is typed in (numbers 0 and 1-32 can be used). Hitting the Tab key auto fills the rest of the channels. Since only two channels are needed, double clicking in the channel 3 box, typing in 0, and then hitting the Tab key changes channels 3-8 to have 0s in them. The OK button is clicked and the dialog box closes.

On the same CobraNet network is an ASI2416. Its CobraNet Configuration dialog box is opened using ASIControl. Using the same procedures as above, 255 is typed in (since 255 was used in the example above) as the bundle number for the ASI2416's receiver 1, and its first two channels are assigned numbers 33 and 34 (numbers 0 and 33-64 can be used). This time the Apply button is clicked so the dialog box remains open. Immediately in the CobraNet Configuration dialog box the first two channels in receiver 1 turn green and the Status field changes to 'Receiving' to indicate a bundle has been successfully routed to the receiver. If any errors, the receivers would turn red.

Clicking OK closes the CobraNet Configuration dialog box. ASIControl is still open and meter movement is seen in CobraNet_Ch 33 and 34, as well as Line_Outs 1 and 2.

14.2.3 References

http://cirrus.com/en/pubs/appNote/CobraNet_AudioRoutingPrimer.pdf

http://cirrus.com/en/pubs/manual/CobraNet_Programmer_Manual_PM25.pdf

http://www.cirrus.com/en/support/cobranet/design/switched_networks.html

C:\as\doc\common\asicontrol_cobranet_network_config.doc

<end>