AES 143

Network Audio Track

How to Make An AES70 Device

Session NA09

October 21, 2017

Agenda

- **Basic AES70 Concepts**

  Simon Jones, CTO, Focusrite

- **Implementation**

  Tom de Brouwer, Software Engineer, Bosch Communications

- **The OCA Alliance for Developers**

  Ethan Wetzell, Platform Strategist, Bosch Communications

- **Demonstrations**

AES143 NA09:

How to make an AES70 Device

Concepts & Design

Presented by Simon Jones:

Member, Board of Directors, OCA Alliance

Member, Technical Working Group, OCA Alliance

CTO, Focusrite Audio Engineering

Making an AES70 Device: *Concepts*

- Basic Elements of AES70
- Device Model
- Minimum Device (OCC-MIN)
- Designing an AES70 Device

Basic Elements of AES70

- Specifications
- Class System
- Protocols

Making an AES70 Device: *Concepts*

The Specification

- Divided into three Sections:
  - **AES70-1**: The Framework. Defines the models and mechanisms that form AES70. AES70 is a **control model**, not a **programming model**.
  - **AES70-2**: Class Structure. Specifies the control class structure, which defines the control and monitoring capabilities of AES70 classes.
  - **AES70-3**: Communication Protocol. Defines AES70 remote control and monitoring over a network.

AES70-2: Open Control Class Structure

- Open Control Class Structure, OCC.
- Based on object-orientated programming hierarchical **Class** methodology.
- Classes are program-code templates for creating objects, in this case controllable and monitorable objects.
- All OCC classes are based on the **base class, OcaRoot**.
- OcaRoot defines the basic functionality of all OCC class types
- Defines the entire repertoire of objects that an AES70 device can use (Annex A).
- Defines the mandatory objects an AES70 device must implement (Annex B).

Making an AES70 Device: *Concepts*

OCC is divided into four categories:

- Workers
- Managers
- Agents
- Networks

Making an AES70 Device: *Concepts*

- Workers: Classes that represent signal processing and monitoring functions
- Managers: Classes that represent device housekeeping functions
- Agents: Classes that represent control-flow processing functions
- Networks: Classes that represent the physical network (or networks) to which the device is connected

Making an AES70 Device: *Concepts*

- Workers are divided into three categories:
  - Actuators – Signal processing and routing functions, e.g. gain, mute, source selection
  - Sensors – Detectors and monitors of various types, e.g. signal level, gain reduction, temperature
  - Blocks and Matrices – Classes that aggregate objects into structured collections, generally used for modeling / managing complex devices, e.g. collecting objects into blocks of “channels”

Making an AES70 Device: *Concepts*

Signal Processing (Actuators)

Gain controls

Mutes

Switches (n-position)

Delays

Equalizers

Filters (IIR & FIR)

Limiters & Compressors

Expanders & Gates

Levelers

Signal generators

Arbitrary numeric and text parameters

Signal Monitoring (Sensors)

- Level sensors (meters)
- Frequency sensors
- Time interval sensors
- Temperature sensors
- Arbitrary numeric sensors

Making an AES70 Device: *Concepts*

Basic Actuators

OcaBooleanActuator

OcaInt8Actuator, Int16, Int32, Int64

OcaUint8Actuator, Uint16, Uint32, Uint64

OcaFloat32Actuator, Float64

OcaStringActuator

OcaBitStringActuator

Basic Sensors

- OcaBooleanSensor
- OcaInt8Sensor, Int16 …
- OcaUint8Sensor, Int16 …
- OcaFloat32Sensor, Float64
- OcaStringSensor
- OcaBitStringSensor

+ *Proprietary extensions as needed*

AES70-2 : Non-standard Classes

- Also termed “proprietary” classes
- Follow the same rules as the OCC class tree
- They are an extension [derivative] of a standard class
- Only derived from a single standard class
- Must have the same functionality as the derivative standard class
- Enhance the definitions of existing features
- Can have extra functionality and features beyond the standard class

AES70-2 : Non-standard Classes

- OCC Derivation Example:
- OcaSwitch (1.1.1.4)
  - OcaActuator (1.1.1)
    - OcaWorker (1.1)
      - OcaRoot (1)
- OcaSwitchAES (1.1.1.4.[MfrID].1):
  - Has all the features of the OcaSwitch
  - Plus the “extra” functionality required

Making an AES70 Device: *Concepts*

AES70-2, Mandatory Objects (Annex B):

Making an AES70 Device: *Concepts*

AES70-2, Mandatory Objects (Annex B):

- Mandatory objects have defined (fixed) object numbers (oNo).

Making an AES70 Device: *Concepts*

AES70-2, Mandatory Objects (Annex B):

- Two Managers

1. OcaDeviceManager (oNo:1) – Overall device manager, containing the Device Name, Manufacturer Name, Serial Number and ModelGUID etc

Making an AES70 Device: *Concepts*

AES70-2, Mandatory Objects (Annex B):

- Two Managers

1. OcaDeviceManager (oNo:1)
2. OcaSubscriptionManager (oNo:5) – Manages reporting of device data back to controllers. Not actually mandatory, but its absence would imply a polled system, which may be okay for small devices

Making an AES70 Device: *Concepts*

AES70-2, Mandatory Objects (Annex B):

- One Worker

1. OcaBlock (oNo:100) – The “root” block, which contains all the device’s worker objects

Making an AES70 Device: *Concepts*

AES70-3: Protocol for IP Networks

- Referred to as OCP.1.
- AES70 only uses *standard* transport protocols.
- Devices are “discovered” by interested controllers using DNS-SD service discovery.
- DNS-SD is often referred to by its common implementation, *Bonjour*

AES70-3: Protocol for IP Networks

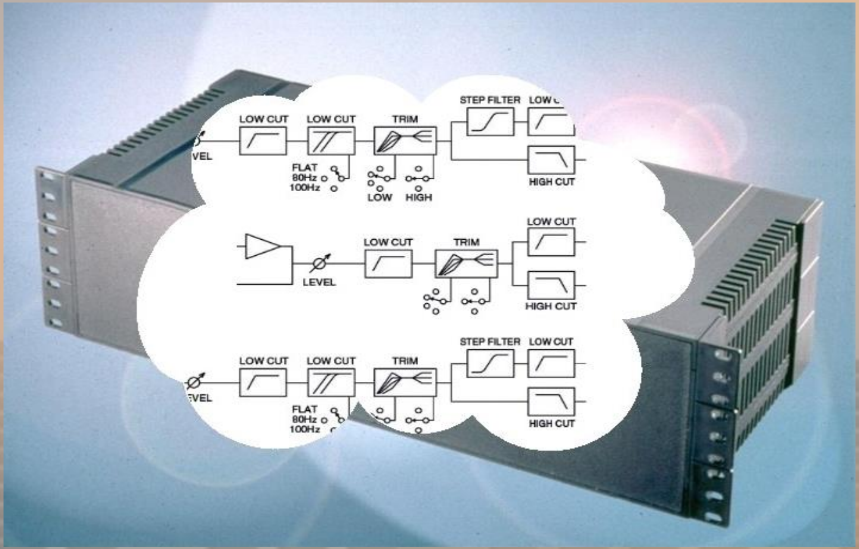
- Up to Four Supported Services:

1. TCP/IP (_oca._tcp)
2. UDP (_oca._udp)
3. Web-socket (_ocaws._tcp)
4. Secure via Pre-shared Key (_ocasec._tcp)

- At least one service must be supported

Making an AES70 Device: *Devices*

Implementation of AES70 Devices, three examples:



Making an AES70 Device: *Devices*

Implementation of AES70 Devices, three examples:

- A commercial audio device
- Demo Devices:
  - Simple non-audio control device
  - Development audio streaming device

Making an AES70 Device: *Audio Device*

A Commercial Implementation of AES70:

Making an AES70 Device: *Audio Device*

Focusrite RedNet4:

- Eight Channel Microphone Preamplifier
- IP audio product based on Audinate Dante / AES67
- Remote controlled via the IP network using a Focusrite proprietary protocol
- RedNet range is an ideal target for AES70

Making an AES70 Device: *Audio Device*

Implementation Requirements:

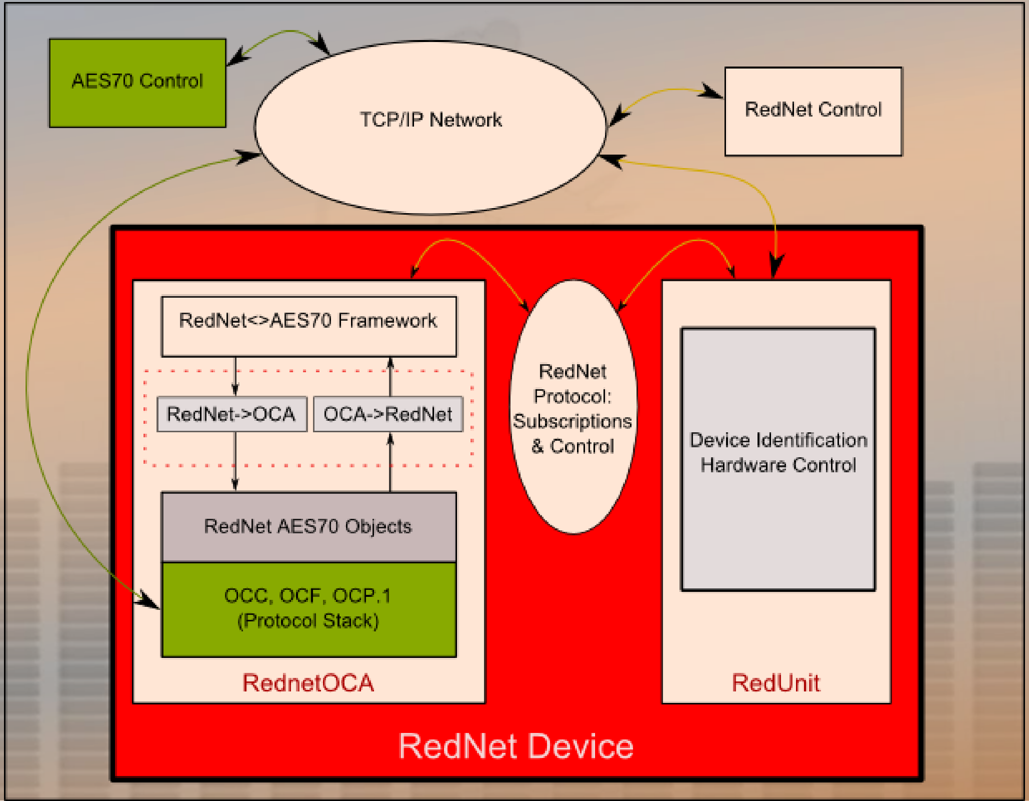
- Must be a simple firmware upgrade, no hardware changes
- Must be backwards compatible so that existing control software remains fully functional

Making an AES70 Device: *Audio Device*

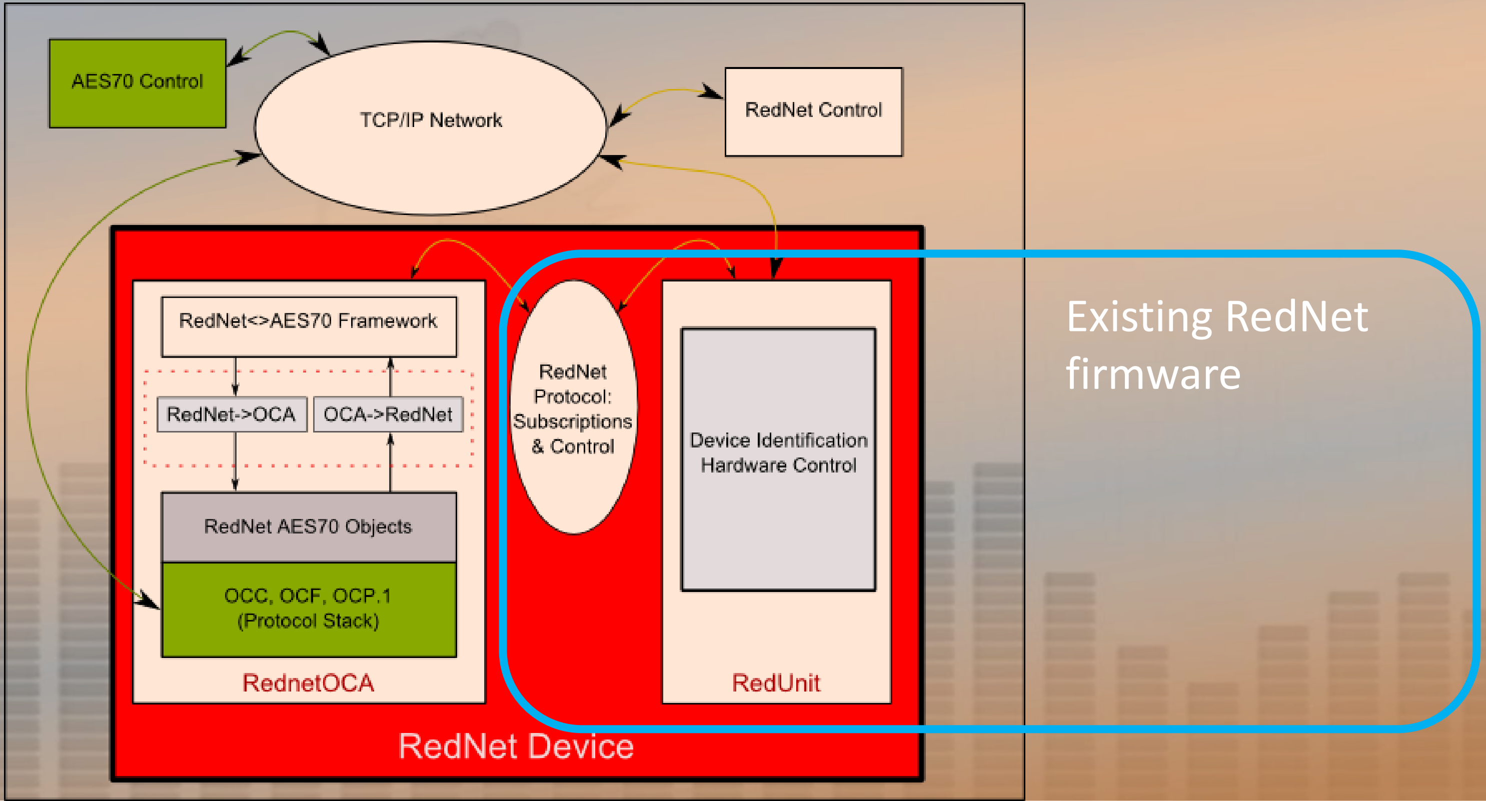
The Proxy Solution:

- Allow the existing firmware paradigm to continue to function, giving backwards compatibility
- Act as a “bridge” between the existing control protocol and AES70

Making an AES70 Device: *Audio Device*

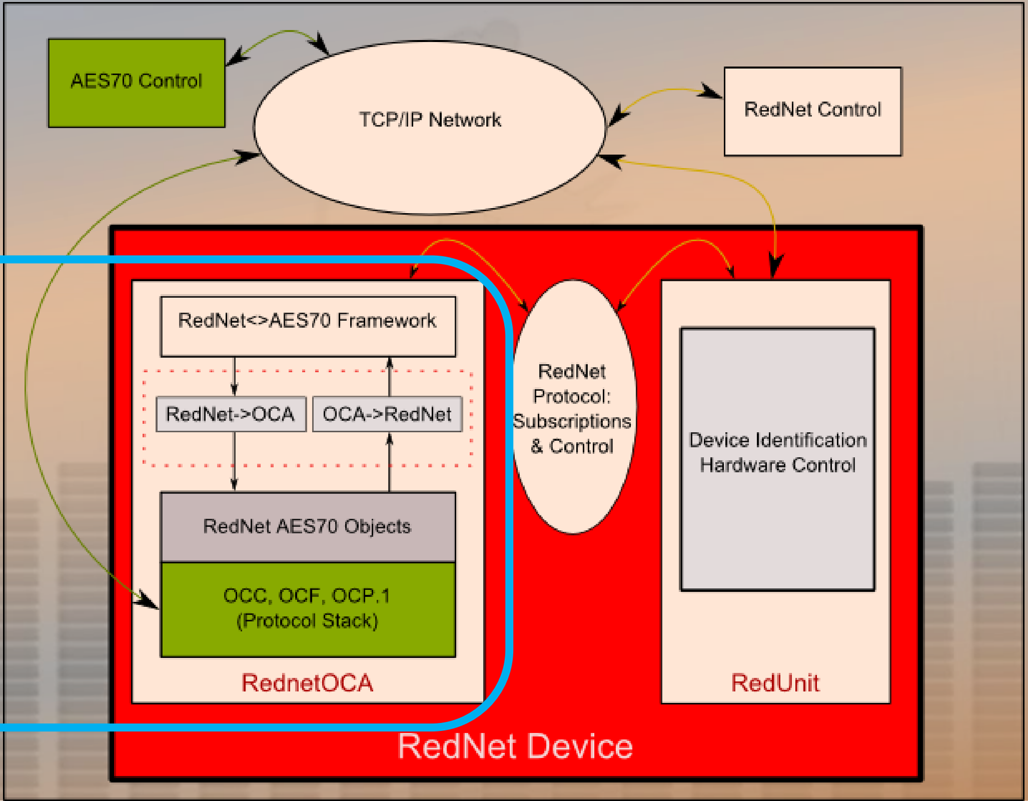


Making an AES70 Device: *Audio Device*



Existing RedNet firmware

Making an AES70 Device: *Audio Device*



Proxy solution. Developed on a Windows PC, with a simple port to the target: **Very Efficient and Quick**

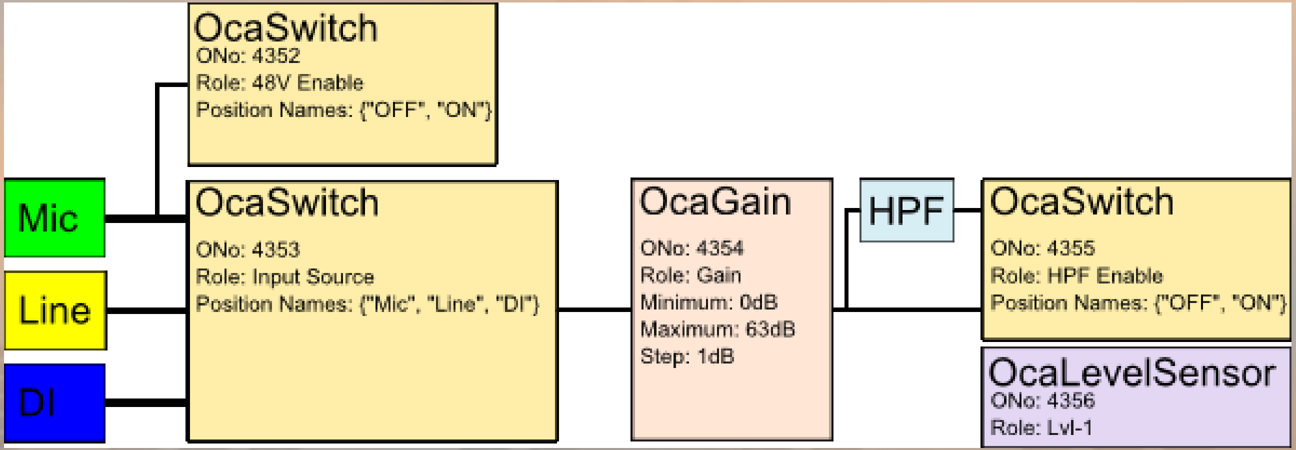
Making an AES70 Device: *Audio Device*

The Proxy Solution:

- Allow the existing firmware paradigm to continue to function, giving backwards compatibility
- Act as a “bridge” between the existing control protocol and AES70
- Allows for the creation of ***virtual devices***
- Virtual devices allow ecosystem development without needing multiple hardware units

Making an AES70 Device: *Audio Device*

AES70 Channel Control Classes: Channels 1 & 2



OcaSwitch
ONo: 4352
Role: 48V Enable
Position Names: {"OFF", "ON"}

Mic

Line

DI

OcaSwitch
ONo: 4353
Role: Input Source
Position Names: {"Mic", "Line", "DI"}

OcaGain
ONo: 4354
Role: Gain
Minimum: 0dB
Maximum: 63dB
Step: 1dB

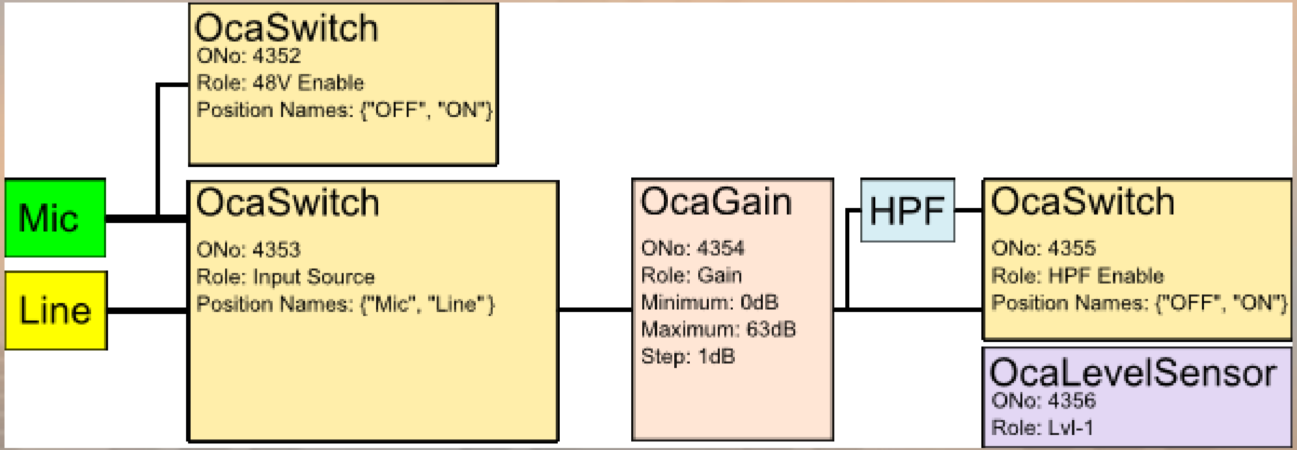
HPF

OcaSwitch
ONo: 4355
Role: HPF Enable
Position Names: {"OFF", "ON"}

OcaLevelSensor
ONo: 4356
Role: Lvl-1

Making an AES70 Device: *Audio Device*

AES70 Channel Control Classes: Channels 3 - 8



OcaSwitch
ONo: 4352
Role: 48V Enable
Position Names: {"OFF", "ON"}

OcaSwitch
ONo: 4353
Role: Input Source
Position Names: {"Mic", "Line"}

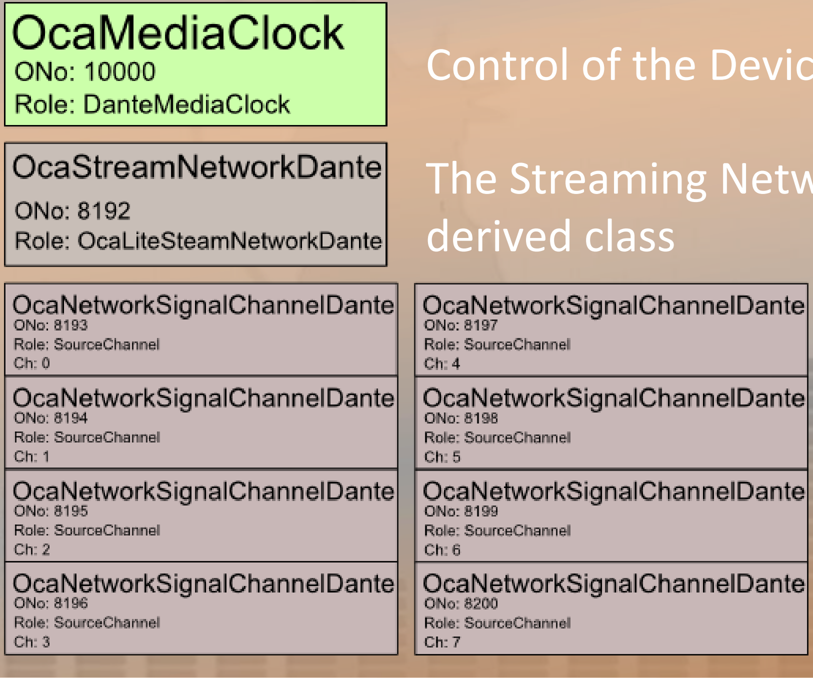
OcaGain
ONo: 4354
Role: Gain
Minimum: 0dB
Maximum: 63dB
Step: 1dB

OcaSwitch
ONo: 4355
Role: HPF Enable
Position Names: {"OFF", "ON"}

OcaLevelSensor
ONo: 4356
Role: Lvl-1

Making an AES70 Device: *Audio Device*

AES70 Media Networking Related Classes



OcaMediaClock
ONo: 10000
Role: DanteMediaClock

OcaStreamNetworkDante
ONo: 8192
Role: OcaLiteSteamNetworkDante

OcaNetworkSignalChannelDante
ONo: 8193
Role: SourceChannel
Ch: 0

OcaNetworkSignalChannelDante
ONo: 8194
Role: SourceChannel
Ch: 1

OcaNetworkSignalChannelDante
ONo: 8195
Role: SourceChannel
Ch: 2

OcaNetworkSignalChannelDante
ONo: 8196
Role: SourceChannel
Ch: 3

OcaNetworkSignalChannelDante
ONo: 8197
Role: SourceChannel
Ch: 4

OcaNetworkSignalChannelDante
ONo: 8198
Role: SourceChannel
Ch: 5

OcaNetworkSignalChannelDante
ONo: 8199
Role: SourceChannel
Ch: 6

OcaNetworkSignalChannelDante
ONo: 8200
Role: SourceChannel
Ch: 7

Control of the Device Sample Rate

The Streaming Network: Dante adaption derived class

Advertisement of the streaming source audio channels

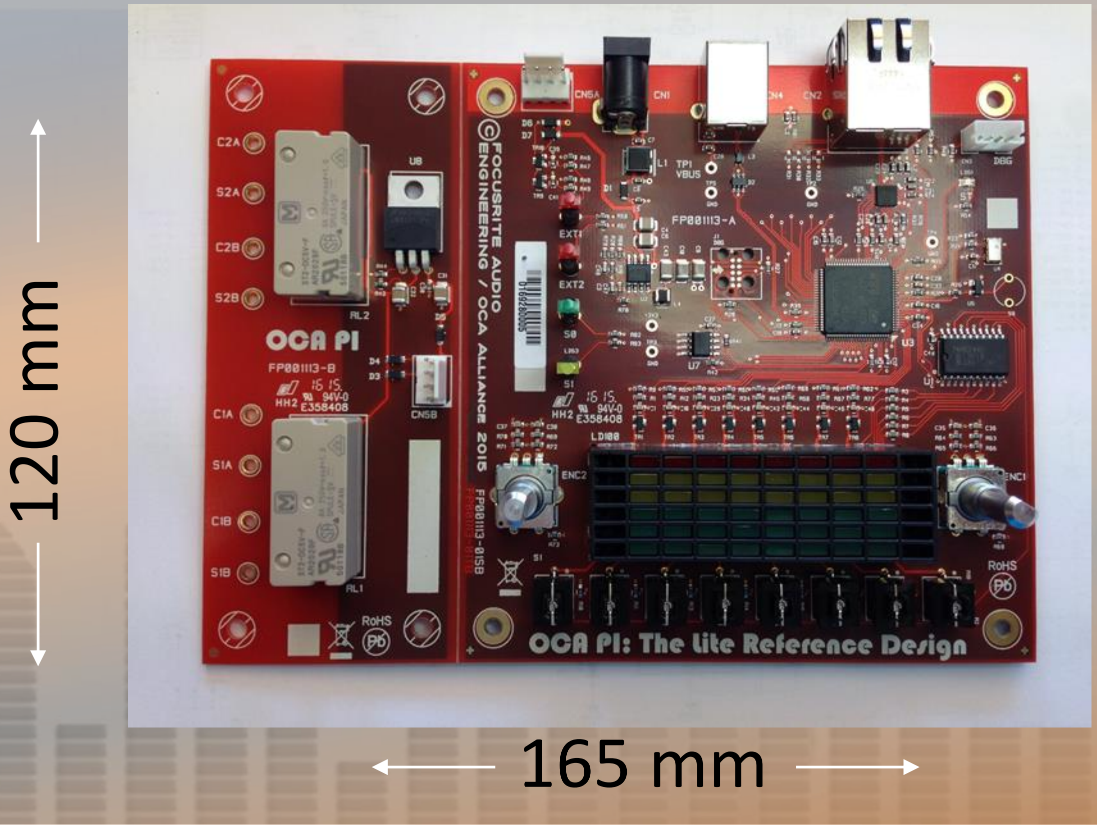
Making an AES70 Device: *Devices*

Implementation of AES70 Devices, two examples:

- An audio device
- **A simple non-audio control device**

Making an AES70 Device: *Non-Audio Device*

The OCA Microdemo


120 mm
165 mm
OCA PI
FP001113-B
© FOCUSRITE AUDIO ENGINEERING / OCA ALLIANCE 2015
FP001113-A
OCA PI: The Lite Reference Design

Making an AES70 Device: *Non-Audio Device*

Brief:

- To show that it’s possible to implement OCA in small embedded processor environments

Making an AES70 Device: *Non-Audio Device*

Hardware Overview:

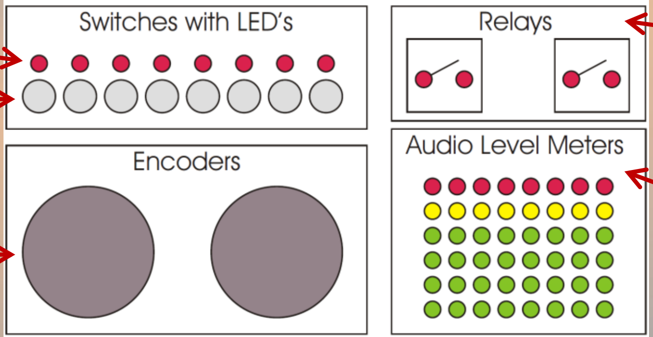
- CPU: ST Microelectronics STM32F207VET6 (512kB flash, 128kB SRAM, 120MHz Cortex M3)
- 10/100 baseT Ethernet
- Eight switches with LED’s
- Two rotary encoders
- Eight LED bargraph meters, six segments
- Two GPO outputs, controlling relays for isolated control
- USB 2.0 full speed, for future use.

Making an AES70 Device: *Non-Audio Device*

AES70 Implementation:

- Make use of the repertoire of “simple” actuators and sensors, keeping it as generic as possible
- Only four different worker classes required for all functions

AES70 Objects:



OcaBitStringActuator (8W)

OcaBitStringSensor (8W)

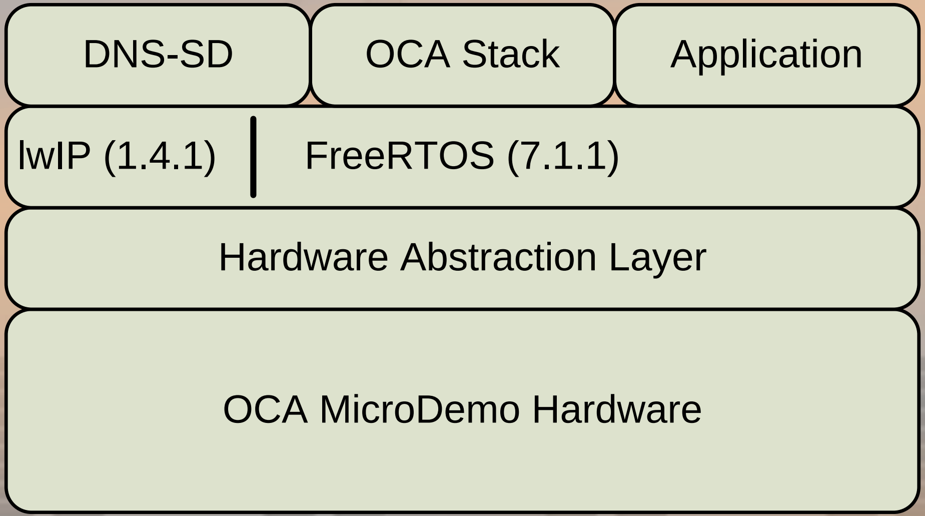
2 of OcaInt8Sensor (-128 to 127 with wrap)

2 of OcaBooleanActuator

8 of OcaBitStringActuator (6W)

Making an AES70 Device: *Non-Audio Device*

Core Firmware Implementation:



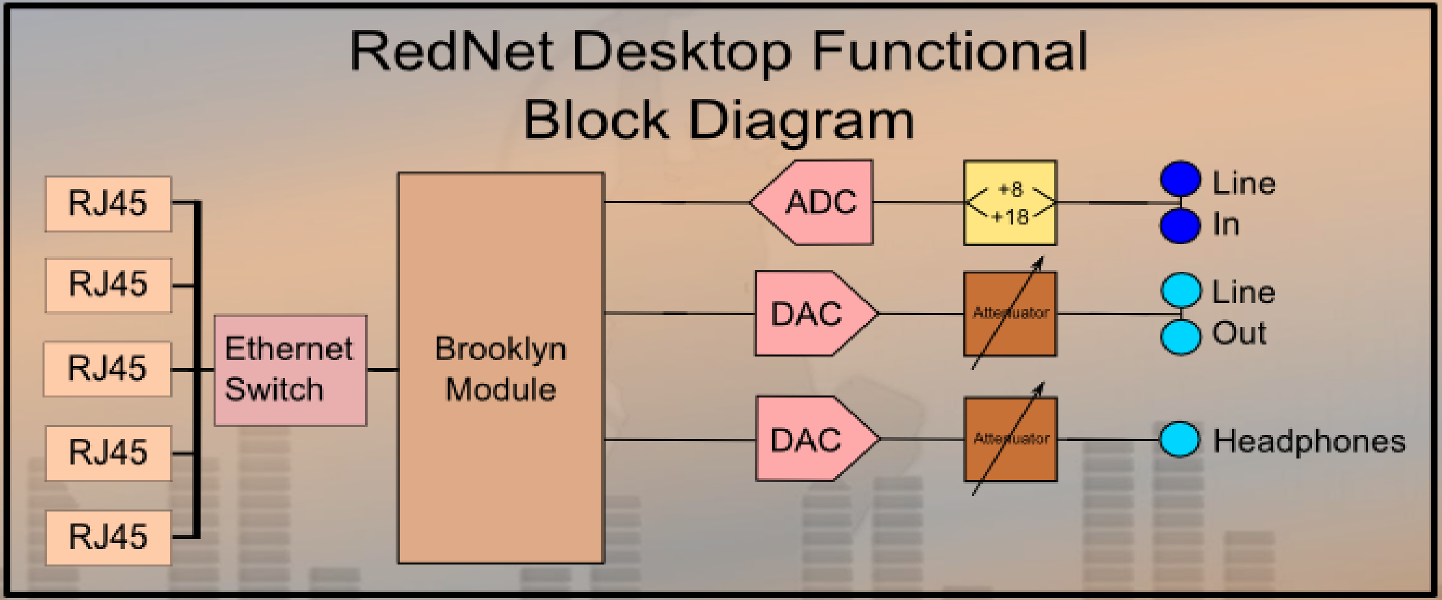
The firmware architecture is straightforward and familiar to many embedded developers

Making an AES70 Device: *Devices*

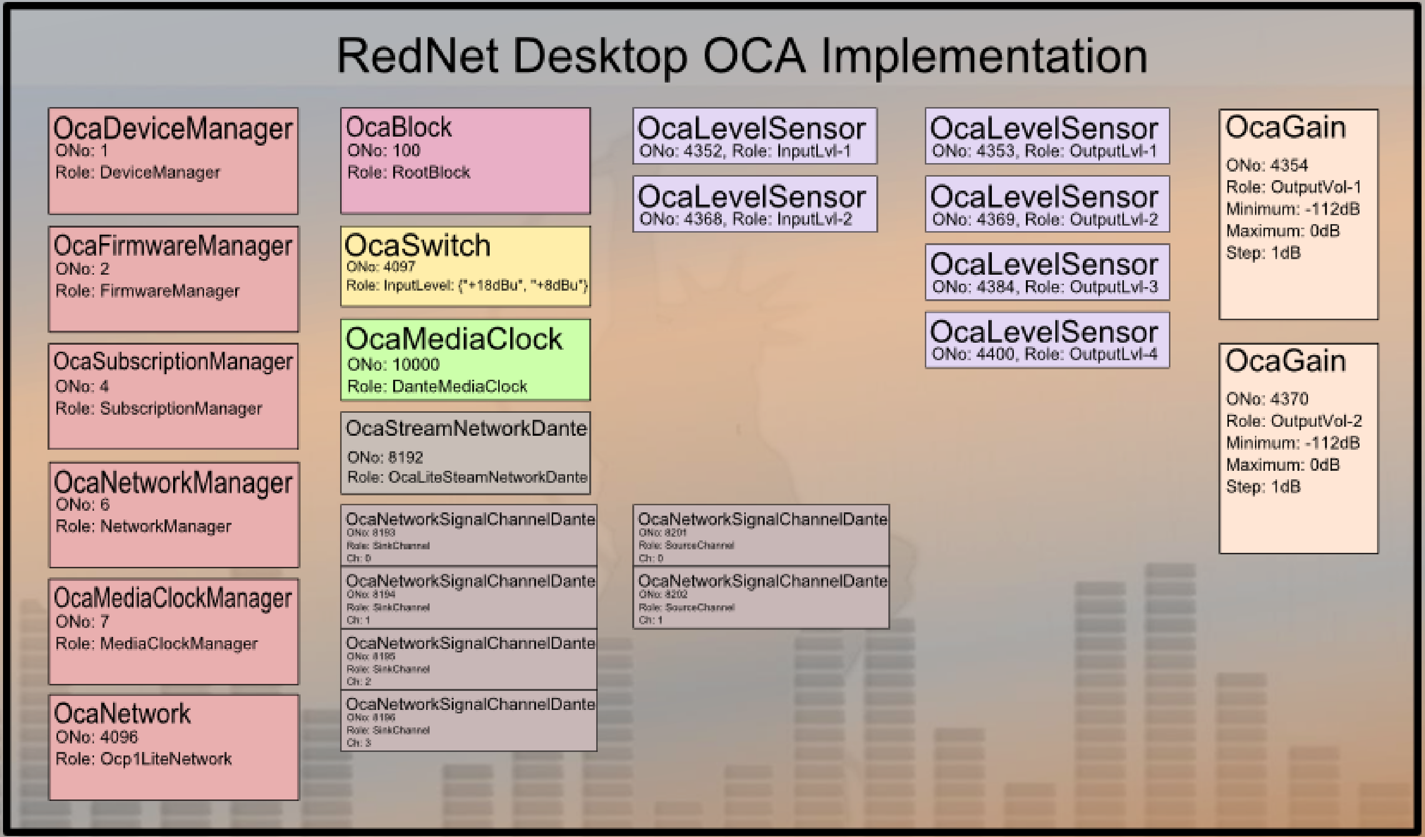
Implementation of AES70 Devices, two examples:

- An audio device
- A simple non-audio control device
- **A development streaming device**

Making an AES70 Device: *Audio Device*



Making an AES70 Device: *Audio Device*



RedNet Desktop OCA Implementation

OcaDeviceManager
ONo: 1
Role: DeviceManager

OcaFirmwareManager
ONo: 2
Role: FirmwareManager

OcaSubscriptionManager
ONo: 4
Role: SubscriptionManager

OcaNetworkManager
ONo: 6
Role: NetworkManager

OcaMediaClockManager
ONo: 7
Role: MediaClockManager

OcaNetwork
ONo: 4096
Role: Ocp1LiteNetwork

OcaBlock
ONo: 100
Role: RootBlock

OcaSwitch
ONo: 4097
Role: InputLevel: {"+18dBu", "+8dBu"}

OcaMediaClock
ONo: 10000
Role: DanteMediaClock

OcaStreamNetworkDante
ONo: 8192
Role: OcaLiteSteamNetworkDante

OcaNetworkSignalChannelDante
ONo: 8193
Role: SinkChannel
Ch: 0

OcaNetworkSignalChannelDante
ONo: 8194
Role: SinkChannel
Ch: 1

OcaNetworkSignalChannelDante
ONo: 8195
Role: SinkChannel
Ch: 2

OcaNetworkSignalChannelDante
ONo: 8196
Role: SinkChannel
Ch: 3

OcaLevelSensor
ONo: 4352, Role: InputLvl-1

OcaLevelSensor
ONo: 4368, Role: InputLvl-2

OcaNetworkSignalChannelDante
ONo: 8201
Role: SourceChannel
Ch: 0

OcaNetworkSignalChannelDante
ONo: 8202
Role: SourceChannel
Ch: 1

OcaLevelSensor
ONo: 4353, Role: OutputLvl-1

OcaLevelSensor
ONo: 4369, Role: OutputLvl-2

OcaLevelSensor
ONo: 4384, Role: OutputLvl-3

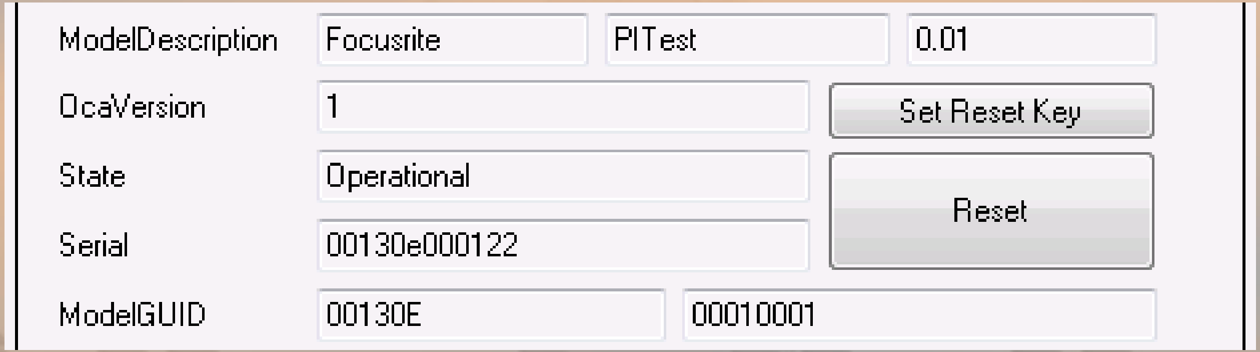
OcaLevelSensor
ONo: 4400, Role: OutputLvl-4

OcaGain
ONo: 4354
Role: OutputVol-1
Minimum: -112dB
Maximum: 0dB
Step: 1dB

OcaGain
ONo: 4370
Role: OutputVol-2
Minimum: -112dB
Maximum: 0dB
Step: 1dB

Making an AES70 Device: *Devices*

Device Manager and Identification:



ModelDescription	Focusrite	PITest	0.01
OcaVersion	1		
State	Operational		
Serial	00130e000122		
ModelGUID	00130E	00010001	

Tom de Brouwer

- Software Architect for Bosch Security Systems
- Involved in programming AES70 products:
  RTS Intercoms, Eletrovoice, Dynacord, Bosch
- Based in The Netherlands

Agenda

- AES70 Implementations available
- Resources and performance
- Test tools for development
- Techsite

AES70 implementations available

- Micro demo code
  - https://ocaalliance.github.io/downloads.html

  - Contains a lite AES70 device implementation found in actual products

    - Platform ports are included:
      - Windows
      - uClinux
      - STM32 (with FreeRTOS/LwIP) with GCC compiler

How to make an AES70 device

- Limited set of objects implemented
  - 3 agents
  - 5 managers
  - 8 workers (5 actuators, 3 sensors, 1 block)
- Supports OCP.1 TCP connections
- OCA Alliance EULA is Apache like license

How to make an AES70 device

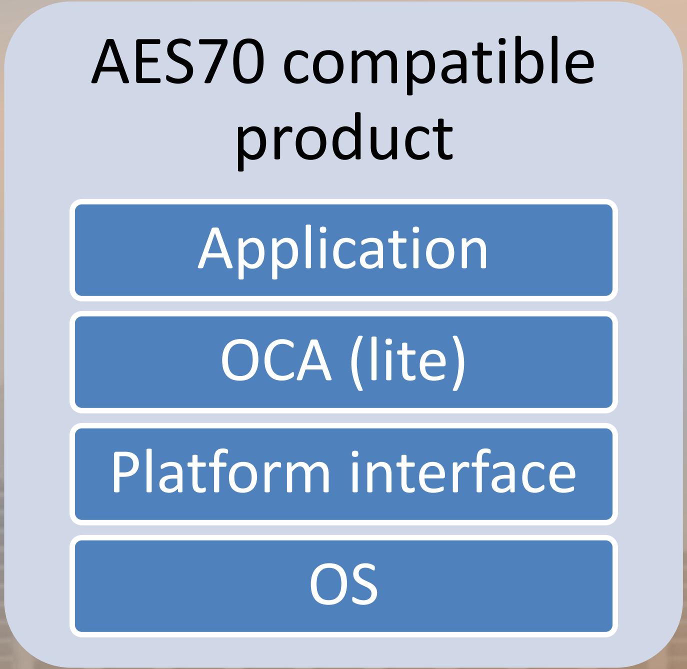
- Commercial implementation
  - Available from a commercial party
  - Device and controller code
  - Fully functional with all specified objects implemented, OCA 1.4 specification in progress
  - Support for Windows platform, has been ported to other platforms.
  - Supports OCP.1 over TCP / “Secure TCP” and UDP
  - Used by multiple manufacturers for critical applications

Programming environments

- Since AES70 is object oriented C++
- Platform interface is C

Visual Studio

Make files (ARM GCC)



AES70 compatible product

Application

OCA (lite)

Platform interface

OS

Resources and performance

- Depends highly on the implementation:
  - Micro demo is based on (**STM32F207VE**)
    - ARM® 32-bit Cortex®-M3 CPU (120 MHz max)
    - 1 Mbyte of Flash memory
    - 128 Kbytes of SRAM

- Number of simultanous controllers (TCP buffers / OCA connection buffers)
- Number of objects (code / heap)
- Device functionality
- Binary protocol / event
- OCA Micro (FreeRTOS / LwIP / OCA Stack) uses 195 Kbytes flash

Protocols

- DNS-SD
  - mDNSresponder / Bonjour (Registration / Browsing)
  - Tinysvcmdns (Registration)
- DHCP / IPv4 LL

OCP.1 selection

- TCP
- UDP

- Throughput / Buffers / Platform support / etc..

OCP.1 selection – future

- Websockets / JSON formatting

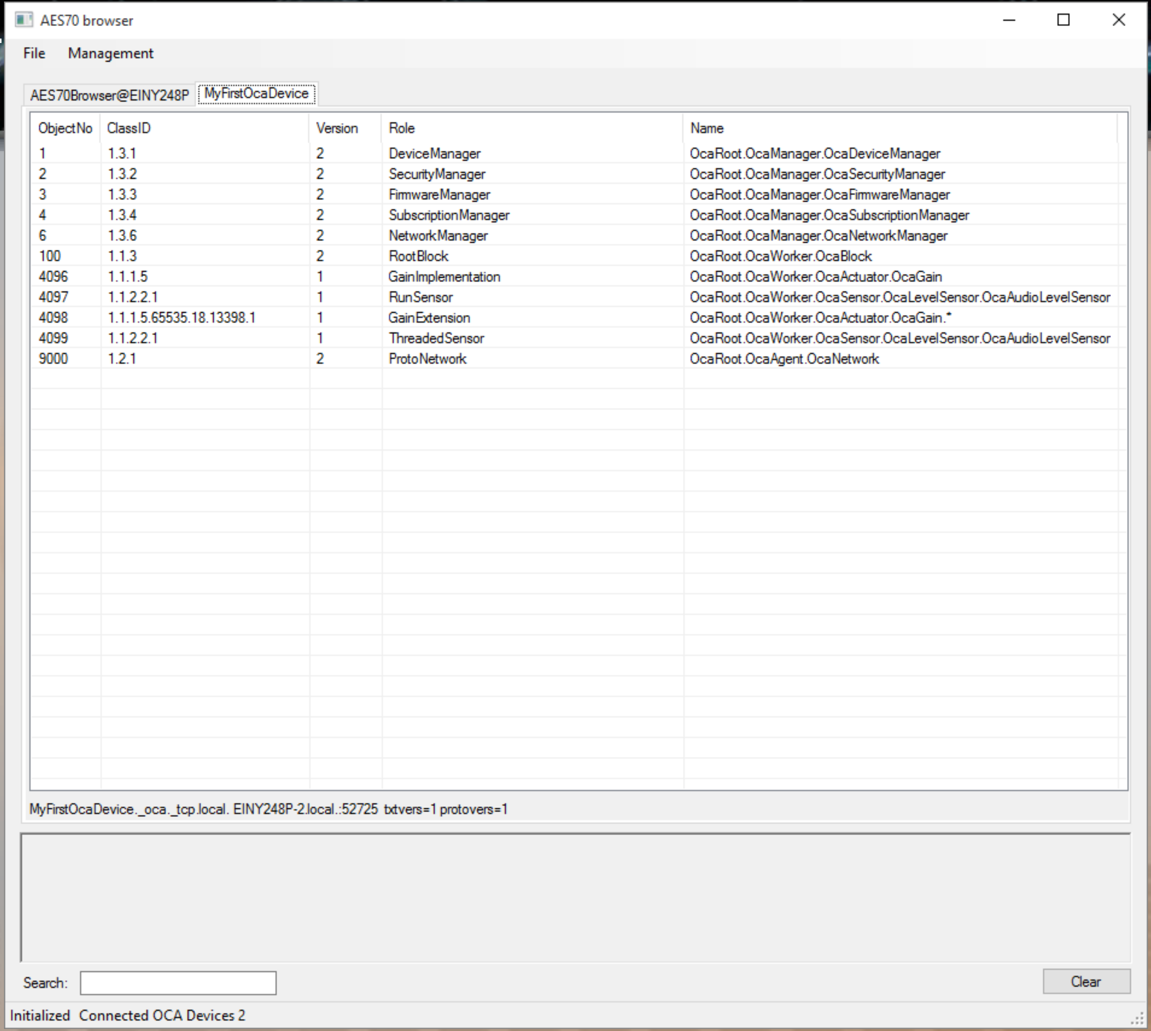
How to make an AES70 dev

Test tools

- AES70 Browser

- Based on the commercial implementation

- Only binaries available



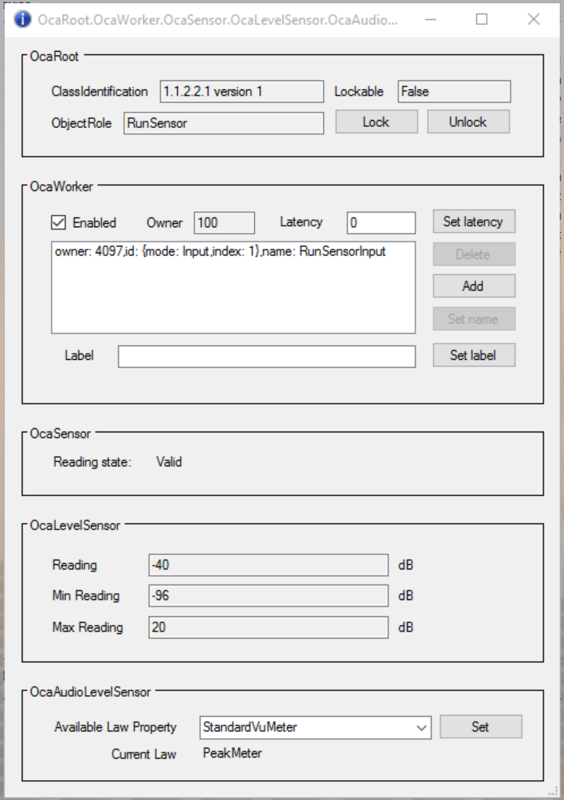
ObjectNo	ClassID	Version	Role	Name
1	1.3.1	2	DeviceManager	OcaRoot.OcaManager.OcaDeviceManager
2	1.3.2	2	SecurityManager	OcaRoot.OcaManager.OcaSecurityManager
3	1.3.3	2	FirmwareManager	OcaRoot.OcaManager.OcaFirmwareManager
4	1.3.4	2	SubscriptionManager	OcaRoot.OcaManager.OcaSubscriptionManager
6	1.3.6	2	NetworkManager	OcaRoot.OcaManager.OcaNetworkManager
100	1.1.3	2	RootBlock	OcaRoot.OcaWorker.OcaBlock
4096	1.1.1.5	1	GainImplementation	OcaRoot.OcaWorker.OcaActuator.OcaGain
4097	1.1.2.2.1	1	RunSensor	OcaRoot.OcaWorker.OcaSensor.OcaLevelSensor.OcaAudioLevelSensor
4098	1.1.1.5.65535.18.13398.1	1	GainExtension	OcaRoot.OcaWorker.OcaActuator.OcaGain.*
4099	1.1.2.2.1	1	ThreadedSensor	OcaRoot.OcaWorker.OcaSensor.OcaLevelSensor.OcaAudioLevelSensor
9000	1.2.1	2	ProtoNetwork	OcaRoot.OcaAgent.OcaNetwork

MyFirstOcaDevice._oca._tcp.local. EINY248P-2.local.:52725 txtvers=1 protovers=1

Search:

Initialized Connected OCA Devices 2

How to make an AES70 device


OcaRoot.OcaWorker.OcaSensor.OcaLevelSensor.OcaAudio...

OcaRoot

ClassIdentification: 1.1.2.2.1 version 1

Lockable: False

ObjectRole: RunSensor

OcaWorker

☑ Enabled

Owner: 100

Latency: 0

owner: 4097,id: {mode: Input,index: 1},name: RunSensorInput

Label:

OcaSensor

Reading state: Valid

OcaLevelSensor

Reading: -40 dB

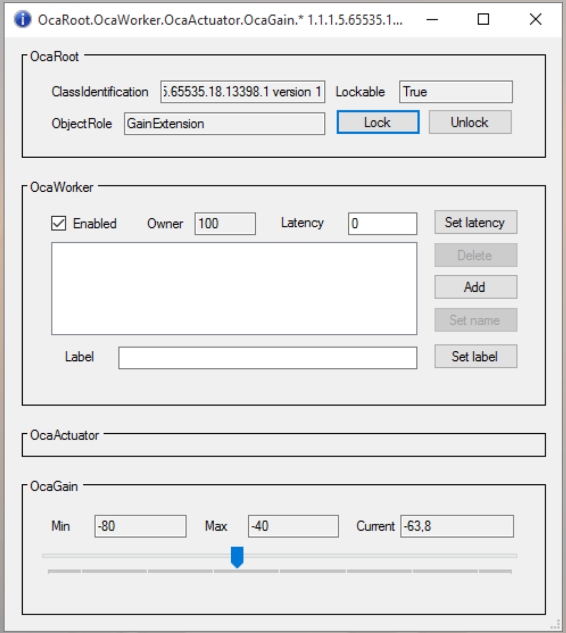
Min Reading: -96 dB

Max Reading: 20 dB

OcaAudioLevelSensor

Available Law Property: StandardVuMeter

Current Law: PeakMeter


OcaRoot.OcaWorker.OcaActuator.OcaGain.* 1.1.1.5.65535.1...

OcaRoot

ClassIdentification: ;.65535.18.13398.1 version 1

Lockable: True

ObjectRole: GainExtension

OcaWorker

☑ Enabled

Owner: 100

Latency: 0

Label:

OcaActuator

OcaGain

Min: -80

Max: -40

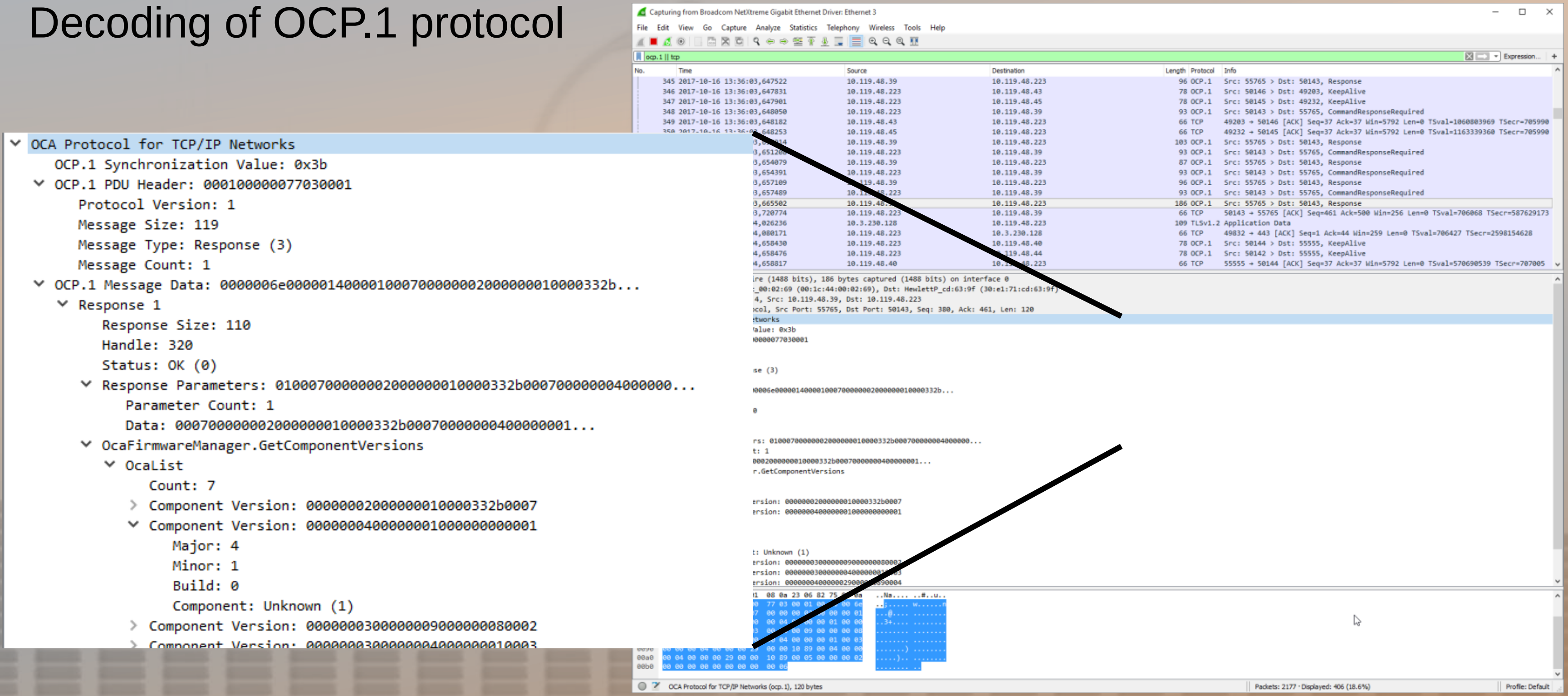
Current: -63,8

AES70 compliancy test tool

- CLI tool
- Checks for interface compliancy
- Checks for implementation of required functionality
- Can be extended for your proprietary extensions to verify the interface is stable

Wireshark plugin

- Decoding of OCP.1 protocol



Capturing from Broadcom NetXtreme Gigabit Ethernet Driver: Ethernet 3
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ocp.1 || tcp
Expression...

```
No.  Time                            Source          Destination     Length Protocol Info
345 2017-10-16 13:36:03,647522       10.119.48.39    10.119.48.223       96 OCP.1   Src: 55765 > Dst: 50143, Response
346 2017-10-16 13:36:03,647831       10.119.48.223   10.119.48.43        78 OCP.1   Src: 50146 > Dst: 49203, KeepAlive
347 2017-10-16 13:36:03,647901       10.119.48.223   10.119.48.45        78 OCP.1   Src: 50145 > Dst: 49232, KeepAlive
348 2017-10-16 13:36:03,648050       10.119.48.223   10.119.48.39        93 OCP.1   Src: 50143 > Dst: 55765, CommandResponseRequired
349 2017-10-16 13:36:03,648182       10.119.48.43    10.119.48.223       66 TCP     49203 → 50146 [ACK] Seq=37 Ack=37 Win=5792 Len=0 TSval=1060803969 TSecr=705990
350 2017-10-16 13:36:03,648253       10.119.48.45    10.119.48.223       66 TCP     49232 → 50145 [ACK] Seq=37 Ack=37 Win=5792 Len=0 TSval=1163339360 TSecr=705990
                                     10.119.48.39    10.119.48.223      103 OCP.1   Src: 55765 > Dst: 50143, Response
         3,651208                    10.119.48.223   10.119.48.39        93 OCP.1   Src: 50143 > Dst: 55765, CommandResponseRequired
         3,654079                    10.119.48.39    10.119.48.223       87 OCP.1   Src: 55765 > Dst: 50143, Response
         3,654391                    10.119.48.223   10.119.48.39        93 OCP.1   Src: 50143 > Dst: 55765, CommandResponseRequired
         3,657109                    10.119.48.39    10.119.48.223       96 OCP.1   Src: 55765 > Dst: 50143, Response
         3,657489                    10.119.48.223   10.119.48.39        93 OCP.1   Src: 50143 > Dst: 55765, CommandResponseRequired
         3,665502                    10.119.48.      10.119.48.223      186 OCP.1   Src: 55765 > Dst: 50143, Response
         3,720774                    10.119.48.223   10.119.48.39        66 TCP     50143 → 55765 [ACK] Seq=461 Ack=500 Win=256 Len=0 TSval=706068 TSecr=587629173
         4,026236                    10.3.230.128    10.119.48.223      109 TLSv1.2 Application Data
         4,080171                    10.119.48.223   10.3.230.128        66 TCP     49832 → 443 [ACK] Seq=1 Ack=44 Win=259 Len=0 TSval=706427 TSecr=2598154628
         4,658430                    10.119.48.223   10.119.48.40        78 OCP.1   Src: 50144 > Dst: 55555, KeepAlive
         4,658476                    10.119.48.223   .119.48.44          78 OCP.1   Src: 50142 > Dst: 55555, KeepAlive
         4,658817                    10.119.48.40    10.  48.223         66 TCP     55555 → 50144 [ACK] Seq=37 Ack=37 Win=5792 Len=0 TSval=570690539 TSecr=707005
```

```
ire (1488 bits), 186 bytes captured (1488 bits) on interface 0
:_00:02:69 (00:1c:44:00:02:69), Dst: HewlettP_cd:63:9f (30:e1:71:cd:63:9f)
4, Src: 10.119.48.39, Dst: 10.119.48.223
ocol, Src Port: 55765, Dst Port: 50143, Seq: 380, Ack: 461, Len: 120
```

```
v OCA Protocol for TCP/IP Networks
    OCP.1 Synchronization Value: 0x3b
  v OCP.1 PDU Header: 000100000077030001
      Protocol Version: 1
      Message Size: 119
      Message Type: Response (3)
      Message Count: 1
  v OCP.1 Message Data: 0000006e000001400001000700000002000000010000332b...
    v Response 1
        Response Size: 110
        Handle: 320
        Status: OK (0)
      v Response Parameters: 0100070000000200000001000332b000700000004000000...
          Parameter Count: 1
          Data: 000700000002000000010000332b000700000004000000001...
      v OcaFirmwareManager.GetComponentVersions
        v OcaList
            Count: 7
          > Component Version: 000000020000000010000332b0007
          v Component Version: 0000000400000001000000000001
              Major: 4
              Minor: 1
              Build: 0
              Component: Unknown (1)
          > Component Version: 000000030000000900000080002
          > Component Version: 000000030000000400000010003
```

```
t: Unknown (1)
ersion: 00000003000000090000000080002
ersion: 000000030000000400000001  03
ersion: 0000000400000029000  890004
```

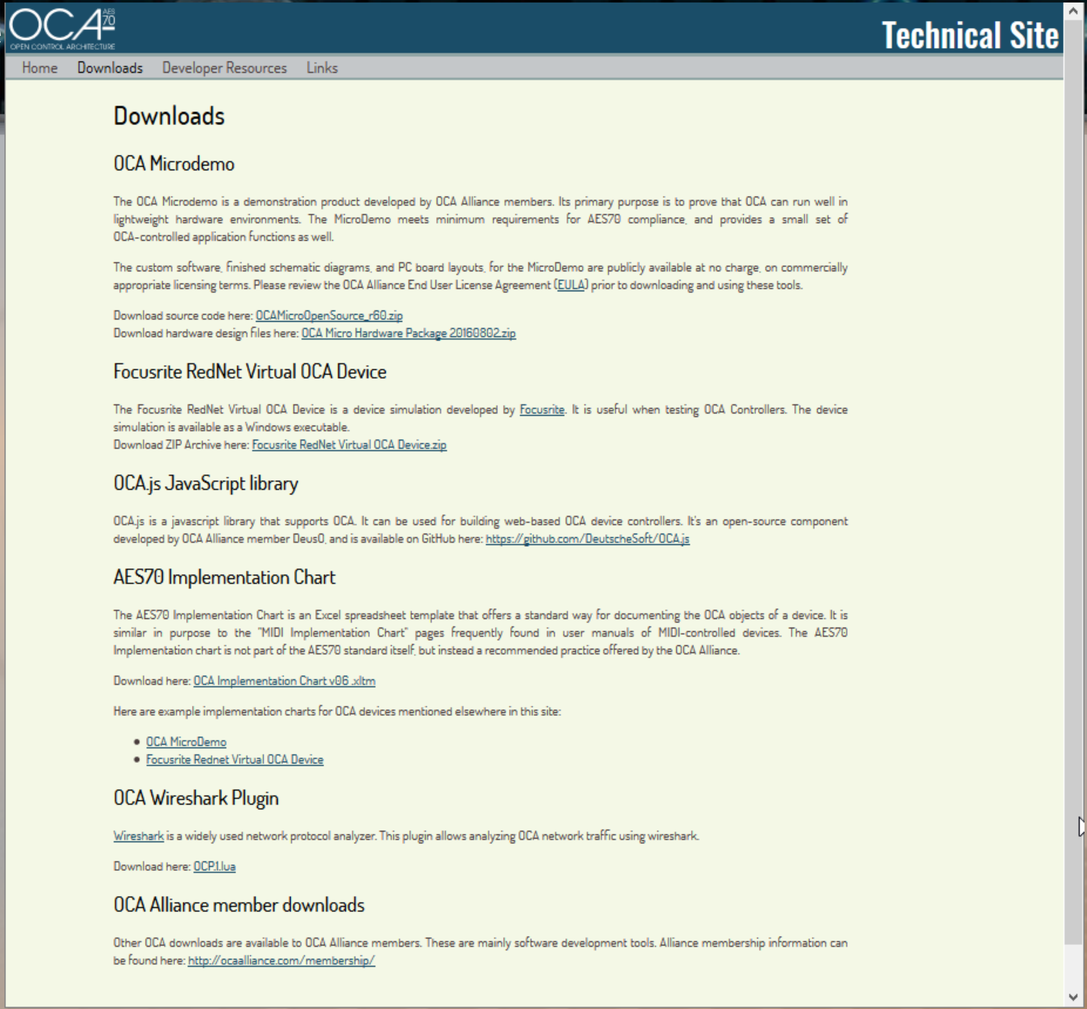
```
01  08 0a 23 06 82 75    0a    ..Na.... ..#..u..
0  77 03 00 01 00    00 6e    ..;..... w......n
7  00 00 00 0    00 00 01    ...@.... ........
0  00 04    00 00 01 00 00    ..3+.... ........
3  00    00 09 00 00 00 08    ........ ........
0     04 00 00 00 01 00 03    ........ ........
00a0  00 04 00 00 00 29 00 00  10 89 00 05 00 00 00 02    .....).. ........
00b0  00 00 00 00 00 00 00 06    ........ ..
```

OCA Protocol for TCP/IP Networks (ocp.1), 120 bytes
Packets: 2177 · Displayed: 406 (18.6%)
Profile: Default

Where to find tools

- Members only area
- Public techsite

https://ocaalliance.github.io/



Downloads

OCA Microdemo

The OCA Microdemo is a demonstration product developed by OCA Alliance members. Its primary purpose is to prove that OCA can run well in lightweight hardware environments. The MicroDemo meets minimum requirements for AES70 compliance, and provides a small set of OCA-controlled application functions as well.

The custom software, finished schematic diagrams, and PC board layouts, for the MicroDemo are publicly available at no charge, on commercially appropriate licensing terms. Please review the OCA Alliance End User License Agreement (EULA) prior to downloading and using these tools.

Download source code here: OCAMicroOpenSource_r60.zip
Download hardware design files here: OCA Micro Hardware Package 20160802.zip

Focusrite RedNet Virtual OCA Device

The Focusrite RedNet Virtual OCA Device is a device simulation developed by Focusrite. It is useful when testing OCA Controllers. The device simulation is available as a Windows executable.
Download ZIP Archive here: Focusrite RedNet Virtual OCA Device.zip

OCA.js JavaScript library

OCA.js is a javascript library that supports OCA. It can be used for building web-based OCA device controllers. It's an open-source component developed by OCA Alliance member DeusO, and is available on GitHub here: https://github.com/DeutscheSoft/OCA.js

AES70 Implementation Chart

The AES70 Implementation Chart is an Excel spreadsheet template that offers a standard way for documenting the OCA objects of a device. It is similar in purpose to the "MIDI Implementation Chart" pages frequently found in user manuals of MIDI-controlled devices. The AES70 Implementation chart is not part of the AES70 standard itself, but instead a recommended practice offered by the OCA Alliance.

Download here: OCA Implementation Chart v06 .xltm

Here are example implementation charts for OCA devices mentioned elsewhere in this site:

- OCA MicroDemo
- Focusrite Rednet Virtual OCA Device

OCA Wireshark Plugin

Wireshark is a widely used network protocol analyzer. This plugin allows analyzing OCA network traffic using wireshark.

Download here: OCP1.lua

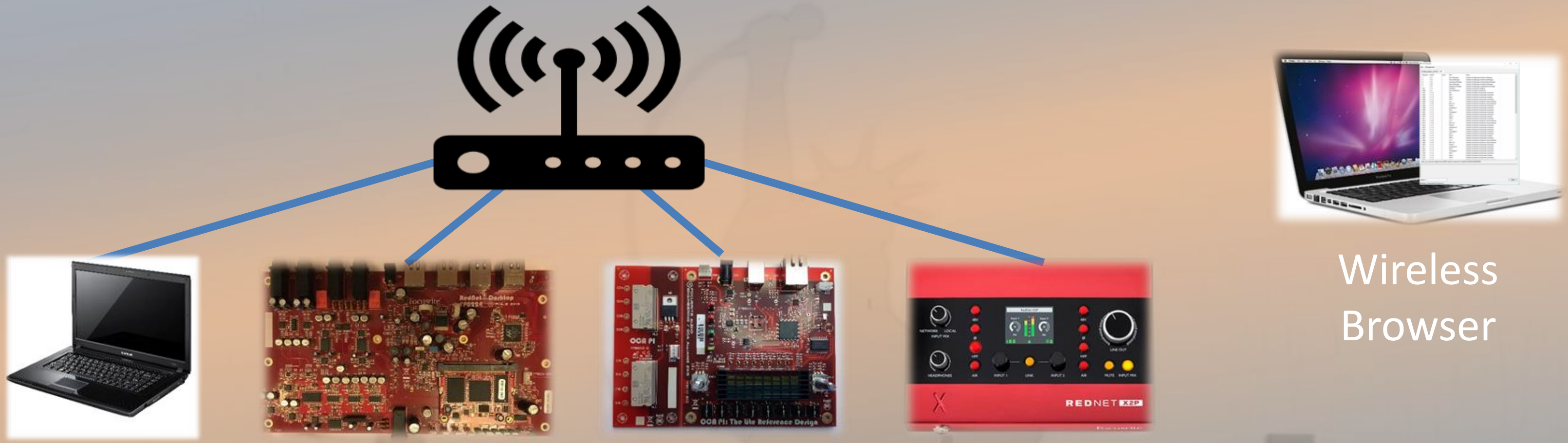
OCA Alliance member downloads

Other OCA downloads are available to OCA Alliance members. These are mainly software development tools. Alliance membership information can be found here: http://ocaalliance.com/membership/

Membership

Demo setups

Making an AES70 Device: *Demo Setup*



Virtual Device

Desktop Device

Microdemo Device

AES70-UDP Device

Wireless Browser