

**RaneNote 144****Emerging Standards for  
Networked Audio  
System Control**

- **Hardware**
- **Network Planning & Design**
- **Software**
- **ActiveX**
- **Necessary Skills for Tomorrow's Installations**

**Devin Cook****Rane Corporation****Ethernet Links****John's Closet**<http://www.digitalmx.com/wires/index.html>

A down to earth set of networking how-to's

**Ethernet Tutorial**<http://www.lantronix.com/htmlfiles/mrktg/cataloget.htm>

A wonderful tutorial in plain English from Lantronix

**Macmillan's Personal Bookshelf**<http://www.mcp.com/personal/>

Free online books about programming and the internet

**Network Design Tutorials & Other Resources**<http://www.alaska.net/~research/Net/nwpages.htm>

An industrial strength link list of networking topics

© 1998 Syn-Aud-Con

Digital devices must be able to co-exist at the hardware, protocol and application level in order to create a seamless network of digitally controlled audio devices. Slowly these issues are being resolved so that we can soon look forward to larger, better integrated installations.

**Hardware**

Up to now, most audio manufactures have relied on either RS-232 or RS-485 to communicate with their units. This will continue for some time because these interfaces are easy to implement, with support built into most common low-cost microcontrollers designed into systems today. While adhering to electrical standards, both have their drawbacks.

RS-232 is common on computers, but can't be easily connected to multiple devices without some major kludging. RS-485 can more readily talk to multiple devices, but some sort of box must be used to convert the RS-232 to RS-485. Though you may be able to electrically interface to more than one device, there is no generally accepted protocol to talk to devices from different manufacturers. Even if you could overcome these limitations, speed starts becoming a problem as control for these devices becomes more complicated.

Ethernet has emerged as a viable electrical interface to solve these problems (along with all those I neglected to mention). In the past, hardware cost and software complication has kept this technology out of the hands of most small audio manufacturers. Now, microcontrollers are emerging with Ethernet controllers built-in, and the cost of external Ethernet chipsets has fallen to the point where it is practical for smaller companies to justify including the interface in their devices.

One pitfall of Ethernet is that the technology is a bit of a moving target. Just as 10Base-T (10 Megabits per second) Ethernet is becoming cheap enough for practical control applications, the computer industry is converting over to 100Base-T (100 Megabits per second), with work on a Gigabit standard currently being finalized. 10Base-T is more than adequate for control and monitoring, but digital audio distribution requires 100Base-T to be practical, and that's still a bit expensive.

**Network Planning & Design**

Surprising to many is the fact that not only can you connect devices together and control them from your computer via its NIC (Network Interface Card), you can control devices from *any* computer on the network. This opens up the possibilities for remote monitoring and control (and I haven't even mentioned control over the Internet yet!). If simple control and monitoring is all that is required, these devices may easily be incorporated into an existing data network, lowering the hardware and wiring costs. If you decide to take the big plunge and distribute audio on the network, along with control, a separate network is probably a must.

## Software

Now that an Ethernet hardware standard appears to be emerging, software remains a barrier to creating a truly integrated network. There already exists many excellent applications available from different manufactures to control a single device, or family of devices, but running a separate program for each device is nobody's idea of a "system", let alone providing the end-user with a simple network interface.

Though efforts to create an interoperable environment in the past have been disappointing, MediaLink being one notable disaster, they have pointed us towards the control environment of the future. To take full advantage of a network, control software must provide the following services:

- Control of ALL devices on the network
- Customizable interface to simplify user operation
- Complex device interactions to allow the above

## ActiveX

The most promising solution to the application interface problem appears to be ActiveX, Microsoft's evolution of OLE, into an Internet friendly programming model. ActiveX components can be used in web pages, visual programming languages such as Delphi, Power Builder and Visual Basic, and even tools such as Lab View. Each ActiveX control is made up of "Properties", values associated with the control which might include things as level settings and meter readings, and "Events", which tell the computer something significant has happened, such as a switch closer or clip detection.

ActiveX allows the manufacturer to create an object which fully encapsulates (describes) a device, while hiding the implementation details such as protocol from the programmer. For example, no longer would you need to know that "byte 5" of a 24 byte status message meant that the unit was limiting. With an ActiveX control, you might simply refer to the limiting status as "Device1.LimitStatus".

By hiding all the nitty-gritty communication details, there is no need for different manufacturers devices to *agree* on protocol. The lack of a protocol standard means that no cooperation between manufacturers is required. This is a good thing in light of the progress, or lack there of, on a common protocol such as AES-24. It allows manufactures to choose the best protocol for their device, just as you might choose JPEG versus GIF images for your web page, trading off size and speed versus quality. As a manufacturer I may choose a custom lightweight protocol versus a larger more standard protocol depending on the power of my device.

You might also add control of devices from outside the industry, such as lighting, as long as an ActiveX module is provided for you. Recently, the ESTA (Entertainment Services & Technology Association) has just published a standard title "Recommended Practice for Ethernet Cabling Systems in Entertainment Lighting Applications" and there are quite a few Ethernet to DMX512 (an ESTA standard control protocol for lighting equipment) bridges on the

market. Also available are Ethernet bridges for RS-232 and RS-485, although I haven't spotted any specific ActiveX controls available yet—but many bridges do use standard internet protocols for which there are controls available, such as telnet (the standard terminal protocol).

By linking various controls yourself and adding a bit of programming you can do things you can't imagine with just simple control software. For example, let's say you want your system to automatically turn down the output level of your speaker processor box whenever the amplifier clips. This might mean trying to get two devices from two different software interfaces to cooperate. This is not an easy task, but if each device was represented by an ActiveX control in a web page perhaps, a VBScript (Visual Basic Script) to control it might look something like:

```
Sub AMP1_OnClip()  
if DSP1.OutLevel >0 then DSP1.OutLevel = DSP1.OutLevel - 1  
end sub
```

This script is executed every time the AMP1 control generates a "Clipping" event. The script checks to see if the DSP box (DSP1) has its output level turned up (above 0) and if so, turns it down one step.

The example is simple, but illustrates staggering possibilities. Add a couple of buttons, graphics and text and you've got a customized installation interface tying together devices from separate manufacturers.

## Necessary Skills for Tomorrow's Installations

Building a network is not nearly as complicated as some would have you think. Start by getting some experience building a small home or office network. The cabling is simple, and as long as you don't need the equipment to talk to the Internet or a corporate network, you won't need switches, routers or any of the heavy-duty network equipment (which must be designed in carefully). A small hub will do, and this is not even necessary if you're simply trying to connect two Ethernet interfaces together. Getting two computers to talk together via TCP/IP (the de-facto king of network protocols) should give you enough exposure to figure out just what is involved with networking. The supplied links are good launching points for more details.

To make the most of a network, programming skills will be required. These skills may range from simply being able to use web publishing software to create web pages and a few simple scripts to tie controls together as in the earlier example, to creating custom applications in Visual Basic, Delphi or C++. Again, there are easy ways to get your feet wet. Programs such as Microsoft's *FrontPage* or Adobe's *PageMill* are more than adequate for building small web pages with ActiveX controls and VBScript.

Whatever level you wish to start, it is apparent that computers are not going away and that networks will be finding their way into new installations more and more often. It's time to get some exposure to the technology so your design options will grow *with* the technology.