Proceedings of the Institute of Acoustics

Serial Show Control - Methods for Audio and Video Equipment Control

S. G. M. Clarke.
School of Electronics, University College Salford, Peru Street M3 6EQ.

1. Introduction

There are a number of propriety audio and video control protocols. The advantages of networking these devices are many, from the remote control of equipment to automation of the complete production process.

The configuration of the network is dependent on the application. In particular, MIDI (the Musical Instrument Digital Interface), the most prevalent of the network standards in audio, can be used to automate and control not just synthesisers but all devices in audio and video production. This includes lighting, tape transports and even implements timecode for synchronisation.

In common with the other network protocols, it uses a serial interface. This determines the minimum complexity of any complexity of any controller on the network, as the controller must be able to give out the correctly formatted commands. By the use of PLC (Programmable Logic Controller) technology, it is possible to implement a cheap, easy to program controller for serial networks. This solution also addresses the problems of testing equipment for EMC emission, as the operation is altered through programming and hence only one box needs to be tested.

2. Background

The complexity of many audio and video devices, with the wealth of features that they offer, has created the need for sophisticated interfaces. The time of 'One pin per feature' has passed.

Audio and video control networks have found many applications, from the home to broadcasting and theatre. Some such as the *LANC* standard [1] and the *Home Bus* [2] provide automation of a limited number of features and are designed to allow a common method of remote control of domestic equipment. Nevertheless the command language is still powerful enough that it enables, for example, automated editing of video.

Other network types such as the *ES Bus* [3] are designed to remove some of the interface incompatibilities between different manufacturers machines in the broadcast environment. These define each class of device as a 'virtual' machine, each having an identical command protocol irrespective of the precise machine function. Clearly not all of these virtual machines will implement all functions - for example, not all VTR's have a record facility or have four audio tracks. Some instruments together but it has expanded to encompass nearly all audio equipment.

MIDI is a real-time network designed to introduce the minimum delay to the transmitted messages. As a consequence, most messages are two or three bytes long, taking around only a millisecond to transfer across the network. MIDI expects a 100% reliable link for all but *System Exclusive* messages. This is because the timing of the message determines, for instance, the start of a musical note, hence any delay would upset the rhythm of the music. For most applications, a one-way link is sufficient and so it acts as a broadcast network. Some processes may require a receiving device to add or *merge* its own message into the data stream an in some cases, a return link to the transmitting device is also necessary.

Features of MIDI Networks

The transmission rate of MIDI is 31.25 kbits/second, which is adaquate for most musical applications. The recommended maximum length of a single link, although it has been tested successfully at distances over 300m. The commands in the MIDI language describe a musical performance in terms of the note played, how loud it is also defining special features including pedals for sustain, pitch bends and modulation messages. So that several instruments can be played simultaneously over the network, it has 16 'virtual' channels which can have different sounds selected to them.

There are a number of specific control instructions such 'MIDI' timecode, sequencer 'start', 'stop' and 'continue', song position and 'MIDI clock'. These enable the correct synchronisation of devices. There is also a *System Exclusive* protocol which enables data specific to a particular type of device to be transmitted over the network. This last class of message is the most powerful as it allows control of any function within the device that the manufacturer chooses. By this nature it is also the longest type of message, as it requires a specific header to identify the device. Often system exclusive messages will facilitate control of parameters which cannot be accessed from the normal user interface provided on the device's front panel.

The simplicity of the MIDI language is its main strength as most messages are short and of a general nature. It is up to the listening device to determine whether it is interested in the received message. The network is able to cope with quite large volumes of data without significantly upsetting the timing of the music being performed. This simplicity can also lead to confusion as the sending device has no idea how the commands will be interpreted, leading to some very unusual renditions of pieces of music.

Audio and Video Control by MIDI

The non-specific nature can be put to good use in control systems. For example, the recently defined MIDI lighting control standards treats a lighting rig as if it were a synthesiser. The selection of an individual light is given by the note played and its brightness is set by the loudness of that note. This network standards, in particular MIDI [4], enables the control of features which cannot be accessed from the front panel. Often this is done to reduce manufacturing costs, although it may have the benefit of allowing automated control of certain parameters in real time, which in the case of MIDI may provide an extra aid to the production process.

3. Common Features of Networks

All the networks here use a serial interface in order to reduce cost and complexity. A number of different data rates are used, but most range between 9.8 and 38.4 kbaud. This does not necessarily determine the response speed or capacity of the network as some protocols have quite heavy overheads in terms of establishing an active link. Most networks use either the RS232 electrical standard (for short connections) or the balanced version RS422 for larger networks, allowing communication over typically 1200m. MIDI is the exception as it uses a 5mA current loop.

All of the networks allow for quasi real-time control of certain parameters within the device. For the simple protocols this may be features such as 'Stop', 'Start' or 'Record'. The more advanced protocols may allow complete control of *all* parameters and in certain cases may also offer some opportunity to re-program the internal function of the device.

4. Types of Network

The transport layer of the network, ie the electrical connections and the types of connection between device, varies upon the intended application. There are three main classes used in audio and video networks:

i) Broadcast and master/slave
ii) Controlled Bus or *arbitrated*
iii) Peer to Peer *non-arbitrated*

Broadcast networks tend to be used for the simplest applications where there is no requirement for acknowledgement of the transmitted message, nor is much need for two-way communication. Most of the 'intelligence' is resident in the main controller, although in the case of MIDI devices, the slave may sometimes add its own message to the broadcast to be passed on to other devices further down the chain. Furthermore, two-way communication is possible with MIDI by connecting the devices in a loop.

Master /Slave networks are a special case where there is a separate two-way link for each device between it and the controller as used in the Sony 422 interface for professional equipment. This standard is designed particularly for video editing systems, where the control information originates from the master controller. The slave device supplies only enough information to enable satisfactory feedback for control, such as 'Play' or 'Stop' tallies and tape counter or timecode.

Controlled-Bus networks share a common *bi-directional* bus where access to the network is strictly controlled by by a bus master. In most cases a device must wait to be polled before it may transmit over the network. This system implicitly avoids bus contention but increases the complexity of the network, as it requires at least three devices on the network (two ordinary devices plus the bus master) before an meaningful communication can take place. The bus master will usually maintain a database about devices on the network. Both the ESBus and the PA-422 Interface [5] are examples of this, but while both are comparatively old systems, they remain uncommon owing in part to the need for the intelligent bus controller.

Peer to peer networking of audio and video devices does not need a bus master for arbitration if it can rely on 'smart' devices to prevent the conflict of messages. The LANC standard uses a dedicated IC to perform this. The LANC bus is intended mainly for the control of domestic and semi-professional VCR's and so has a restricted command set. The AES standard [6] under development for audio equipment control goes much further in terms of the information which may be passed down the network. This is made possible only through the increased processing power available in most professional equipment. Even so, practical implementations are not likely to become common for some time, as each device must have a substantial resident knowledge base about the protocol used. It is intended that devices should be able to both automate the control of their parameters as well as report on their current state.

Practical Problems Caused By Serial Interfaces

A number of difficulties arise, such as the speed restrictions of the bus used causing the delay or even the loss of messages. Other problems may be caused by devices on the bus being poorly connected or even turned off, resulting in the loss of part of the network.

One principal problem of serially controlled networks is that each device, whether it is the controller or the controlled equipment, must have a certain amount of intelligence. It is not possible, for example, to wire a simple switch to initiate an event. The switch must send a serial command to the appropriate device which enables the desired function.

MIDI in Control

Whilst MIDI was revolutionary at its introduction in 1982, it is not especially sophisticated in terms of the facilities it directly provides. It has a limited range of commands available in its language and is designed mainly for one-way communication. It is however (with the exception of LANC), the most widespread of all the audio/video networks, having many times the number of devices in use than other types.

The popularity stems from its ability to enable a wide variety of audio equipment to talk to each other, with or without a computer. MIDI was designed as standard for connecting electronic musical means that a cheap musical keyboard can now be used to control quite complex rigs and can also be used to program a computer to sequence the lights in any desired pattern. This augments the existing 'machine control' protocol which defines the messages used to interface to tape transports. These together make up the *Show Control* and *Machine Control* standards.

Musical Keyboards are not necessarily the easiest devices by which to implement fine control of a system. Further enhancements are available to improve the interface. By making extensive use of mapping of device parameters it may be possible to use a short messages to control certain system exclusive commands. These may come from physical sliders or from a computerised control panel, thus the automation of equipment such as mixing desks becomes practical.

5. Introducing the MIDI PLC

It not always practical or desirable to use a keyboard and/or a computer to control devices over a network. Some applications may demand a semi-permanent installation, or may need a layout of controls not easily mimicked using mouse-driven computer interfaces. In some cases it may be necessary to apply some simple pre-processing of information before being put into a computer or device. It may be that a simple start/stop button is needed, which is easy to implement with a older interfaces with one wire per feature. It is not a trivial task with a serial interface as the switch must launch a serial command to the appropriate device which enables the desired function. These problems are common to all serial interface standards.

In order to solve the problems caused by using a serial network, it is necessary to use a programmed controller. It *is* possible to perform some of the simple tasks by designing a dedicated hardware solution - however this may only implement very simple functions. Complex tasks will require a microprocessor system and a number already exist to perform a range of tasks such as MIDI to CV (Control Voltage) converters for interfacing pre-MIDI analogue synthesisers.

Features of the MIDI PLC

The aim of the MIDI PLC is to offer a more flexible approach to the interface. It provides user with a number of programmable controller facilities in MIDI. The most common application is to provide a programmable control panel which would either assist a computer on the network by providing a more user-friendly set of controls, or to replace the computer completely where only a relatively simple set of controls is necessary. For this second application, this increases the usefulness of MIDI networks for non-musical applications, as well as removing the need for an expensive dedicated musical keyboard and/or a computer. It may also provide a more reliable interface, as MIDI computer programs are notorious (slightly unfairly) for crashing.

It would also be possible to provide a MIDI receive panel, where incoming commands are converted into, say, voltage levels and contact closures. This type of application would be less common, as there are already enough MIDI programmable devices on the market that should be able to perform the required function. It would however enable existing non-MIDI devices to be used over the network. This could be extended further by adding plug in modules to implement commonly used features, eg. audio level control.

A further application would be translation and processing of MIDI data, where for example, scaling or compression of incoming velocity data could be applied. This type of feature is often implemented by computers, but it could be useful for performing musicians.

There are two modes of operation for the MIDI PLC:

Program Mode
The analogue controls and switches can have command sequences assigned to them. For example, a potentiometer is programmed as a volume remote volume control for an audio device. The three-byte MIDI message to do needs the following commands:

Control Change on channel n
Select Volume Adjust
Send new volume level (0-127)

This would have the effect of sending a string of bytes each time the volume control setting is altered ie:

$B0_h$ 07_h #value

The PLC could be programmed from the front panel, although it is intended that it would be programmed over the MIDI network from a computer. This can be done using the user's existing software, keeping costs of the MIDI controller to a minimum and allows the greatest flexibility.

Operation Mode
In the operational mode, the controller is set to send out the programmed to send out the commands as programmed, where it might have for example volume and pan information sent from its controls to the remote device. The programmed state of the controller is retained when powered down.

Hardware

There is not sufficient space here to discuss detailed hardware requirements. The current MIDI PLC is based around the PIC16C71 microcontroller chip [7], although almost any small microcontroller with analogue to digital conversion and serial interface capabilities should be suitable. The system has been designed for minimum production costs as the device should be cheap enough to be used on a 'fit and forget' basis. Total hardware cost targets for low production volumes are around £10.

6. Conclusions

The MIDI PLC project is still in its early development stages and more work needs to be done to evaluate the optimum design. The solution proposed here need not be restricted only to MIDI networks and can be applied equally to any network standard through appropriate hardware and software changes to the PLC.

The use of networks for audio and video control will undoubtedly increase. Serial control PLC's should provide a simple, cheap and friendly way of interfacing to audio and video networks.

7. References

[1] H. Hirose, Y. Matsumura, K. Shimada, S. Takada, K. Nakano, 'Local Control Bus System for Consumer VCRs'. IEEE Transactions on Consumer Electronics, Vol. 35, No. 3 August 1989, p589-595.

[2] Hiroshi Yamazaki et al.,'A Home Video Bus System'. IEEE Transactions on Consumer Electronics, Vol. 34, No. 3 August 1988, p728-733.

[3] A. M. Woodhouse,'An Introduction to the ESBus'. BBC Engineering Training Information Sheet, February 1989.

[4] A. Armstrong and S. Clarke, 'The Essentials of Music Technology'. Publication due Spring 1995.

[5] Robert Rogers, 'The PA-422 Communications Interface and Device Control Language'. Journal of the Audio Engineering Society, Vol. 38 No. 9, 1990 September, p619-640.

[6] Audio Engineering Society, 'Draft AES-24: Application for Controlling and Monitoring Audio Systems', 1993.

[7] Microchip Technology Incorporated, 'PIC16C71 microcontroller'. Microchip Technology Data Book p2-146