A Digitally Controlled Automation System

[4CA1.03]
Preprint 3303

OLIVEIRA, Antonio Jose;
Acutron, Ela, Sacavem, Portugal

**Presented at
the 92nd Convention
1992 March 24–27
Vienna**

AES

AN AUDIO ENGINEERING SOCIETY PREPRINT

tomation System

A. J. Oliveira

Acutron Electroacustica Lda
Urbanizacao Quinta Nova Impasse 1 Lote 134
2685 Sacavem Portugal
Tel. 351-1-9414087 351-1-9450862 Fax 351 1 9412509

A flexible, bus-base architecture for a broadcast audio automation system is presented, which is capable of handling any machine format. Particular emphasis is given to the digitally controlled level setting and attenuator sections, where a modified R-2R MDAC topology was designed in order to implement a true log law, allowing for the optimization of the control range for a limited number of control bits.

0 Introduction

Modern broadcasting facilities often require that several program hours during the day are assured by an automatic control system to reduce personnel costs. It would be advisable that the very same automation system used for unattended operation could also be used for on-air operator assistance in order to ensure a correct program timing. Up to now, those functions have been assured by start-stop controllers using conventional machines with or without time-code facilities. Audio processing is normally performed through the on-air console, and often extensive machine modification is needed in order to interface the automation system. Finally user interfaces are usually unfriendly and the systems are useless with the new digital recording formats. There was then a necessity to rethink the concept of broadcast on-air automation in order to come up with an architecture which could, in a single box, provide all the needed features, allow for easy expansion and permit standard communications with other equipments. Audio quality, of course is a primary concern in such a device. This paper describes an attempt to such an open architecture.

1 Required functions

The system should do the following:

-Possibility of programming while running, so as not to have to interrupt normal operation for operator interfacing
-Have an extremely precise time-of-day reference, and a non-volatile configuration, program and time backup to prevent power failures
-Start and stop a certain number of stereo machines of any type, setting their relative levels so relative loudness level is preserved
-Machine control should be done either under absolute time or program-dependant control, with detection and replacement of a defective machine
-Crossfade the relaying machines as an human operator would do, including fade in/out at the start/end of program
-Provide for customized time signals, and allowing program insertion during the time signals
-Provide information to the operators in the form of large, easily readable displays.
-Provide for standard communications with a personal computer or modem, in order to allow remote program download.
-Have a simple, menu-driven user interface included so that the system would be entirely self-contained, needing no external controlling device to operate
-Allowance for the cue operations normally needed in an on-air console to setup machine program start, both by listening and by controlling the level with a peak-type meter
-Have an hardware that easily allows for future expansion, including totally digital audio in/out

Let's now examine the above exposed requirements in more detail:

2 Choice of controller

The IBM personal computer is a tempting choice for this type of application, because of the price, the easy user interface and compatibility. However, when we considered factors like the instability of the included real-time clock, the non multitasking nature of the DOS operating system and the failures associated with simulating multitasking operation by interrupting the central processor (mostly caused by interrupt priority and masking), we chose to build a dedicated control computer based on an 8 bit controller. This way the system functions are 100% under designer control, and not under BIOS or operating system control, either of which can obscurely play a few nasty tricks to the end user. The PC, if the user feels the necessity for an extended user interface, can be then used only for that purpose, and can communicate with the controller through a standard EIA-232D serial interface.

3 Time-of-day reference and battery backup

A standard computer real-time clock chip with local low power crystal-controlled oscillator was used, together with a large capacity static RAM to hold time, program and configuration information, the whole being backed up by a NiCd battery with built-in charging circuit. A power monitoring circuit checks for both high and low voltage in order to take the appropriate shutdown/backup action. The real-time clock chip is also responsible for time signal wave generation.

4 Machine hardware interfacing

We start from the assumption that the final user of such an automation system is also an user of conventional machines, such as carts, open-reel and cassete. We also assume that the user incorporated CD players, either single or multiple, and possibly DAT and hard-disk or optical disk recorders. The automation system shall be capable of communicating with these devices, which roughly means that we will have at least three categories of interfacing, if only start-stop control is needed:

A-Machines which require a pulse to start but stop by means of a cue-tone, such as carts

B-Machines that require both a start and a stop pulse. The majority of devices, such as CD and cassete recorders fall in this category.

C-Machines which require steady-state control signal for start-stop. Most open reels are included here.

The automation system must, through simple operator menu choices, allow the definition, over a per-channel base, of which type of machine is present on a given channel.

The hardware must lend itself to let the software assume complete control of the functions of a given machine, if that is required by the user, in order to be able to cue through the media. This suggests a multi-function card that can control either several start-stops in different machines, or several functions inside the same machine. We chose to limit the number of the functions to be controlled by each card to 8. Of course, galvanic isolation and ruggedness of the interface are essential to avoid both ground loops and control faults within the system, which suggests the use of relays as control elements.
In what concerns analog audio interface with the machines, a balanced input is mandatory in the system, and gain of that input must be easily changed, not through conventional pots, but through software, in order to allow for remote control. A typical span of nominal output levels of today's machines lead us to consider that a range of -9 to +6dBu would be adequate. Resolution of the software "potentiometer" should be 1dB, with an accuracy of 0.1dB in order to provide repeatable level settings and a correct stereo image.
Intuitively, and since most controllers and available MDACs work on an 8-bit word basis, one could be tempted to do it with a conventional 8-bit MDAC. The required decimal control word for a generalized n bit converter and for an attenuation A expressed in dB relative to the all 1's converter code would be:

$$W=(int)(2^{n}-1)10^{-A/20}$$
(1)

To correctly apply the above expression, an approximation method is required which call for floating-point arithmetic. This is of course out of question with current 8 bit controllers, so integer arithmetic will be used instead, yelding a worst-case 1 bit error in the computation itself.
Computation error E, expressed in dB, will then be:

$$E=|20\log_{10}(W/(2^{n}-1))/A|$$
(2)

If we consider that at a nominal level of +6dBu (which corresponds to an attenuation of 15dB relative to the maximum control word of $FF, so to roughly $2D) a one bit computational error corresponds to 0.2dB gain error, we conclude that, even for this simple attenuator function, an 8 bit linear MDAC is completely inadequate. It would also require increased software overhead in the controller in order to be able to divide and calculate powers, a time-consuming process considering that normal controllers can only perform addition, subtraction and multiplication. Furthermore, a survey of commercially available IC showed that noise performance is normally not specified.
An alternative could consist on the use of a converter using a wider control word, with the additional circuitry to perform the conversion of the 8-bit digital word to a larger one, perhaps in the form of a lookup table stored in ROM. Such a principle is commercially used at least by one manufacturer [1]. In that case, a 9 bit converter would be adequate and a 4 bit control word would be required to search the lookup table.
A simpler solution is described in 13 - "Log multiplying converter design", which uses a 4 bit control word and features an error well within the 0.1dB target specification, requiring no computational power and as such not suffering from finite precision errors.

5 Machine control modes

To implement the first mentioned operating mode, absolute time control, information is derived from the internal real time clock and compared to a programming table stored in non volatile RAM. Time required for crossfade is taken into account, so that the stopping machine is kept working till complete fade-out is effected, and is then disconnected. Of course to avoid lengthy list manipulations provision is made in the controlling software for automatic repetition, either on an hour basis or daily based.
The user is given the option of choosing a time-out in the end of which, if the chosen machine has not delivered audio above a certain threshold (corresponding to -20dBu output) in that time interval, it will be considered defective and replaced by a default one by the controller. Level detection is effected by a precision rectifier with a time constant of roughly 1 second followed by a comparator which compares the rectifier output voltage against a reference and delivers the information to the CPU. The CPU then uses upon signal failure an internal timing sequence to check, each second up to the time-out if the so considered normal signal has been restored or not, at least in one of the stereo channels. Signal is detected before the crossfade section, so the defective machine can even be replaced while still not on-air (see fig. 1).
The described mixed hardware/software method of checking the operating machine's audio can also be applied to sequence programming. In this consists the second mode of operation, where a machine sequence can be defined, instead of an absolute timing list, so on audio failure of the currently operating machine the next one in the list is selected.

6 Crossfade

It is desirable that the system has the capability of fade-in/out and crossfading, simulating an human operator. To do that, three factors are important:

-The fade control range must be at least 60dB before complete cutoff
-The attenuation must follow a logarithmic function of time (fixed dB/s)
-The fade time must be programmable, say between 0s and 64s, to accommodate any requirements

Taking into account the stated requirements, we see that using an analog VCA to do the job with reasonable audio quality results in very complicated circuitry [2] and, since we defined similar requirements for a 15dB attenuator in section 4, we can perform a similar analysis for a linear MDAC-based 60dB attenuator. Time control is then easily effected using a microcontroller-implemented timer to change the control word accordingly.
If we define again 0.1dB for the precision of the attenuator, and define that an inaudible gain change step would be around 0.25dB, we get, for an 8 bit wide control word a 63dB range, which is more than adequate. Let's look, using equations 1 and 2, what would be the requirements if we used a linear MDAC to do the job. We can deduct easily from those equations that the required MDAC would have to be 17 bit wide, and the control word could be 8 bit.
Again, we resorted to the new architecture described in section 13 in order to solve the problem, using an 8 bit wide logarithmic MDAC. Used control word is reversed and $FF will be used to completely cut-off the attenuator.

7 Time signal generation

A clock derived from the real-time clock chip is software gated in order to provide a series of time signal bursts. The number of bursts and their relative time location can thus be easily altered, or the time signal can be totally inhibited. Provision is also made to optionally suppress the bursts if any machine is programmed to start during the time-signal, allowing for example the insertion of commercials. In the latter case, fade in/out will be inhibited, to allow for fast operation, in order to keep the critical time schedule of the signal. The frequency of the bursts can be also manipulated to generate different time-signal tones.

8 External displays

Communication with external displays for operator information is effected through an EIA422 interface, which allows for fast communications over balanced lines up to 1200m. A generalized information packet is sent to the cable, containing both time and machine control information. At each tap along the line, a microcontroller-driven large LED display unit extracts from the information packet the one it needs. An address can be incorporated in the packet to send information only to a particular display.

9 External computer communications

EIA232D is by far the most common interface for computer control of devices used in broadcast environments in Europe, because of its availability virtually in every PC. However, it suffers from the drawback of allowing only a 15m cable and of being designed for single point-to-point communication. Modems can alleviate the first drawback, at the expense of a decrease in throughput, and the second one can be bypassed by making the data terminal pass the information to another interface, if a proper address is included in the packet. In spite of being the most used interface, the EIA232D is manipulated by almost every manufacturer in a different fashion, what renders difficult to group different controllable devices within the same setup. Discussion of a possible standard for information packets is beyond the scope of this paper, but will be the object of a future one. We then adopted the EIA232D, with software controlled communications parameters, but with a default configuration consisting on the following parameters:

Bit rate - 9600
Word width - 8 bits
Parity - None
Stop bits - One

The employed packets follow the XON-XOFF protocol, allowing connections through a normal microphone-type cable, so without any hardware handshake. Each command is preceded by the ESC character, and device identification can be included in it.
The above exposed working conditions can serve as a basis for future discussions on the theme, such as the ones that derived on the present PA422 standard [3] for sound reinforcement equipment.

10 User interface

The proposed automation system uses a miniature 4X4 keyboard and a backlighted LCD display for user interface, when it is used as a stand-alone system. A menu-driven software interface is provided to implement every function previously described, and provision has been made to take control of the hardware through the include EIA232D interface, as described in the previous section. This dual-mode control allows both a simplified user interface for a minimum cost stand alone system and a full pop-up window user interface implemented on a PC. It is interesting to note that, on what concerns firmware code embedded into the microcontroller, 80% of that code is actually dedicated to the user interface, and only the remainder is actually working code. This fact is known by the majority of software designers, but is worth as an example to novice audio automation system designers, so they can estimate the actual code after the full user interface has been developed!

11 Cue operations

Instead of operating directly with individual cue buttons, one for each channel, we choose to put the cueing operations under software control. Being so, an automatic cue circuit consisting on a digitally controlled analog multiplexer outputs the on-air signal when not selected, both to a pair of peak indicators and to an headphone amp, and, when selected, waits that the user inputs one channel number or more to cut the on-air output and insert the channels corresponding to the user-provided channel numbers.

12 Hardware architecture

Now we will look at the hardware structure, which is of fundamental importance in automation systems because it controls not only what the system can do in the present, but also what we can make it do in the future. We choose to implement the automation system in the form of a mother board with 14 slots, containing the power supplies, separated for digital logic and audio circuitry, reset circuits for power-up, low voltage and user-requested reset, a temperature compensation circuit to drive the LCD backplane, a crowbar circuit to cut power on power supply failure, the battery backup and, finally, input/output routing to the bus-bars.
As main cards in the system we have, as shown in fig.1:

One containing the main microprocessor, and address controller, the two serial interfaces and the real-time clock.

One containing the user program memory and an extension EPROM to be used for messages to be sent to the display for user interface purposes.

One design choice was to put only one audio channel in each audio processing boards, to facilitate future expansion. So as audio processing boards we have:

One card containing a double dual-channel selector, a double gain controlling section and a double level detector as described in section 5.

One card with a dual digitally controlled fader, with one fader control word complemented in relation to the other, so cross fade between channels is effected with just one control word. This board also carries the balanced output circuits.

Two fixed eight channel boards for input conditioning before driving the system bus. These are not slot boards, but instead couple directly to input XLR connectors, in order to avoid digital signal coupling over possibly high input impedances, at frequencies where common mode rejection ratio of the balanced inputs could be impaired.

Finally, any combination of machine control cards, containing a microcontroller addressable latch, and associated relays and connectors can be used, under software control, to provide the combination of functions that the user will need.

13 Log multiplying converter design

Note - Parts of the whole of the principle described in this section may, at the time of printing be the subject of international patent application.

Taking the voltage-mode version R-2R digital to analog converter model as an example, which basic topology is shown in fig. 2, we readily see that to transform it to a law different from linear using only hardware, we normally need to provide buffer isolation between stages, because the converter relies on accurate current division between fixed ratio resistors to do its job. That procedure will lead, if the number of stages (or bits, if you prefer) in the converter is sufficiently high, to serious noise and distortion degradation of the device performance when used as an audio MDAC, not to speak of the added circuit complexity, cost and power consumption. As mentioned in section 4, others did the modification by software, using a wider bit width DAC coupled to a lookup table stored in ROM and a reduced bit width used as an address generator. Still another possibility that could be envisaged might consist in using a weighted resistor summing D/A, but the involved coefficients for each bit add instead of multiplying, rendering a log law impossible to implement. Furthermore, impedances can become excessively high for bit widths in excess of 4.
Our proposal is to use a series correcting impedance between each MDAC R-nr converter stage, where n is the multiplying coefficient for each bit in order to implement the required law, keeping the series impedance as much constant as possible from stage to stage. Let's check in detail how that can be accomplished:
If we look to the topology of fig. 3, which describes a digitally controlled attenuator, we note that, instead of using a single switch to connect each node to ground, we use an SPDT switch to insert a compensation series resistor when, in each stage, the corresponding weighting resistor is in circuit.
We will write the equations describing the circuit behavior, taking into account that each switch has a finite series resistance R_S, which will appear in parallel with the compensation resistor R_2 when the stage is not selected, and in series with R_3 when the stage is selected. We will also take into account that the reference series attenuator arm R for the first section is the resistor R_1, being corrected from stage to stage as described by equation 9. We shall therefore designate the corrected series resistor by the letter R.
First thing to do is to determine the value of R_3 for each stage. This can be done bit by bit with the following procedure:
Assuming an attenuation range A, expressed in dB for a number of bits n, we get, calling G_x the gain corresponding to bit x, where x=1..n:

$$G_x=\log^{-1}(-A/(5(2^{(3-x)}(2^n-1)))) \qquad (3)$$

and the parallel attenuator arm will have an equivalent resistance:

$$R_T=R_1G_x/(1-G_x) \qquad (4)$$

As said before, the calculated equivalent resistor value will include the finite series resistance of the switch. Then, the actual value of R_3 will be:

$$R_3=R_T-R_S \qquad (5)$$

Now, we calculate the equivalent series resistor seen from the stage output terminals, assuming the considered stage is unselected:

$$R=R_1+R_SR_2/(R_S+R_2) \qquad (6)$$

The compensation resistor R_2 we have to insert will then be, assuming that we previously calculated R_3, and hence R_T:

$$R_2=R-R_1R_T/(R_1+R_T) \qquad (7)$$

Substitution of equation 6 into equation 7 and solving for R2 results in:

$$R_2=(R_1^2\pm(R_1^4+4R_1^2R_S(R_T+R_1))^{0.5})/(2(R_T+R_1)) \qquad (8)$$

Obviously equation 8 has two solutions, one positive and the other negative, so we will disregard the negative one.
Applying equation 6 to correct R1 in order to perform identical calculations for the following stage is equivalent to set:

$$R_1=R \qquad (9)$$

Calculation will proceed from stage to stage by successive iterations of equations 3, 4, 5, 6, 8 and 9.
The above exposed conducts to a digitally controlled attenuator which precision is ultimately dependant from the resistor matching to theoretical values.
Results obtained with standard E96 values are shown in figures 4 and 5. Figure 3 shows the results for the 63 dB fader attenuator, where the thick trace shows deviation from linearity on a ±0.2dB scale when the MSB is untrimmed and the thin trace depicts similar performance when the MSB is trimmed (approximately 5% deviation from theoretical value). Figure 5 shows the performance of the input 15 dB attenuator, which requires no trimming, bearing the theoretical performance (thick line) and the actual measured one (thin line).

14 Conclusions

A new architecture for a self-contained automation system was presented, having the final user in mind in all respects, and fully using the available software/hardware resources, as well as a few tricks and techniques from the old all analog world that dictated the standards for many years. The proposed architecture was a mixed analog digitally controlled, with provision for future expansion and modification. Special emphasis was given to the implementation of special-law digital-to-analog converters, having as a final result ultimate performance in terms of amplitude linearity. Furthermore, the presented technique lends itself to easy integration.

The described architecture is being manufactured as a commercial product by Acutron under the designation TM1-Broadcast Automation System.

---

15 References

[1] Analog Devices AD7111 logarithmic D/A converter data sheet, Analog Devices data conversion products data book, 1989/90, pages 2-183/2-188

[2] Oliveira, A. J., A feedforward side-chain Limiter/Compressor/De-esser with improved flexibility, Journal of the Audio Engineering Society, vol 37 no. 4, April 1989

[3] AES recommended practice for sound reinforcement systems-Communications interface (PA-422), Journal of the Audio Engineering Society, vol 39 no. 9, September 1991

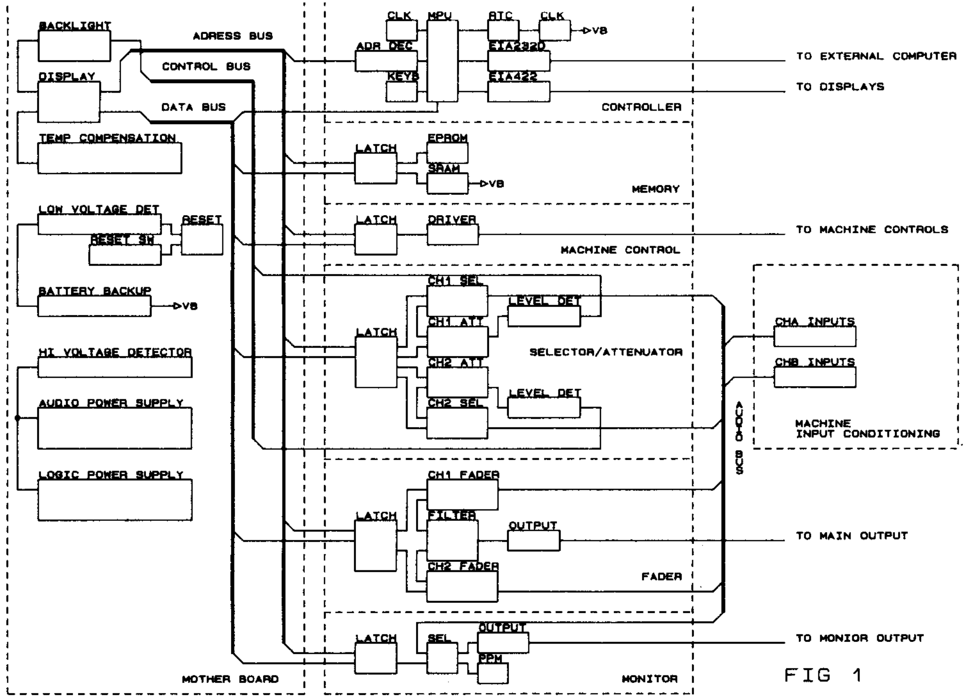
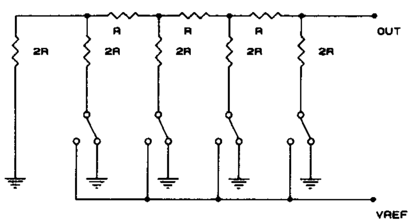
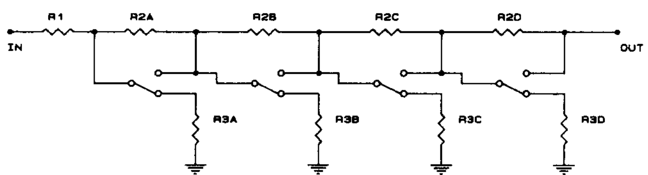


FIG 1



R-2R MDAC

FIG 2



NEW TOPOLOGY R-nR ATTENUATOR

FIG 3