Control Your Audio/Video Connections with the AVMux

FEATURE ARTICLE

Steve Ciarcia

One of Steve's more popular Circuit Cellar projects from years ago was the AVMux. With today's more sophisticated home theater requirements and the availability of newer, better chips, it's time to revisit the AVMux project.

About nine years ago, the concept of an entertainment (a.k.a., media) room became a heavy topic in audiophile and videophile circles. Rec rooms lost their pool tables and pinball machines, and relatives expecting the same (free) svare bedroom were shown the Yellow Pages under "motels." Home builders who had just come to grips with jamming jacuzzis into the master bedroom closet now had to come up with a home theater to entice the sophisticated home buyer.

Aside from the physical room itself, the reality of a dedicated entertainment space was primarily a matter of assembling the components. Virtually all of the equipment (projection television, CD player, laserdisc, VCR, amps, etc.] was available off the shelf, but implementing the physical connections among the components presented a problem.

Virtually all of the equipment of sufficient quality for this purpose also had singular applications. Good stereo systems ("good" is interpreted as meaning audiophile quality) consisted of individual modules such as a tuner, preamplifier, amplifier, CD player, sound field processor, subwoofer amplifier, DAT player, and so forth. It was not uncommon to have to use eight or ten of these electronic subsystems to create the proper sound and video atmosphere in a true entertainment room.

The unfortunate truth back then was that while the concept of coordinated media was easily understood, an adequate means for channeling the audio and video signals among all these subsystems that would ultimately result in the desired effect was no small task. In fact, it was a wiring maze. To ease the wiring congestion and simplify overall interconnection, I designed a combination 8x8 audio and video multiplexer.

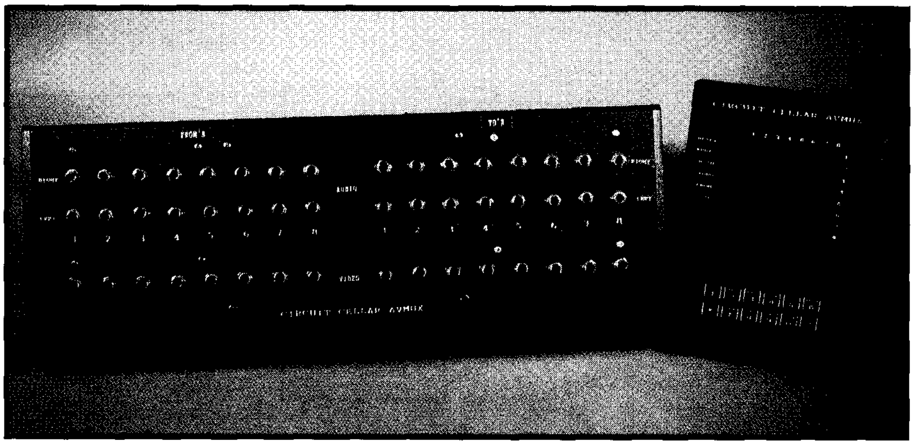


Photo l-The finished *AVMux* and hand-held control unit add both sophistication and class *to* any home *theater installation.*

The AVMux, as I called it, was (and is) exactly as the name implies. It was an electronic cross-matrix of switches which channels specific inputs to designated outputs. In the case of an 8x8 multiplexer, the inputs to it (commonly called the *froms*) come from the outputs of the various signal-generating subsystems, such as CD players, VCRs, laserdisc, tape recorders, and so forth (subsystems with both audio and video outputs would use the appropriate audio or video input side of the multiplexer). Similarly, the output side of the AVMux was connected to the inputs of various other system components (called tos): preamplifier, amplifiers, surround decoder, soundfield decoder, tape recorders, and so forth.

Using the AVMux and a series of push-button settings like "From 2 to 6" "From 3 to 5," and "From 3 to 7," would, for example, channel the laserdisc into a system combination of the preamp, amplifier, and subwoofer amp (we presume speakers are always connected to a particular amplifier). To change the source from the laserdisc to the CD player, we might enter "From 1 to 6" on the control. This "erases" the "From 2 to 6" connection and replaces it with "From 1 to 6" instead.

A SOLUTION NOW LACKING A PROBLEM?

Nine years later, the scenario is quite different. Consumer audio and video equipment have made significant advances such that the borderline between true "audiophile" and "high-end" consumer quality is fuzzy at best. To obtain superb sound and video in the past, the only solution was to use more expensive modules and externally connect things through an electronic multiplexer like the AVMux.

Today, stereo manufacturers have succeeded in providing some real power and quality in a single enclosure. It is not hard to find the equivalent of the tuner, preamplifier, two 100-watt stereo amplifiers, subwoofer amplifier, a Pro-Logic surround decoder, and a dedicated audio/video multiplexer in one enclosure. A Mitsubishi unit I looked at recently included all this for $1400. When I compare this to the $10,000 I spent on the equivalent separate boxes.. .

The greatest attribute of today's "integrated component" single-box stereos [provided it is of sufficient quality) is that they contain an integral dedicated multiplexer. With external input designations like VCR1, VCR2, TAPE 1, TAPE2, CD, AUX1, and AUX2 (remember, the tuner and surround decoder are already in the box, so they don't need external inputs), a simple push button or remote tells the internal multiplexer (frequently a relay) to channel the appropriate input through the preamp into the surround decoder and amplifiers. The outputs of this system are usually just speaker terminals and the direct video from the designated source. This video goes to a separate monitor such as a projection TV.

Depending upon the brand and the cost, you can find a variety of input/output configurations. Because they have a dedicated application. the audio multiplexer is always a Nxl and the

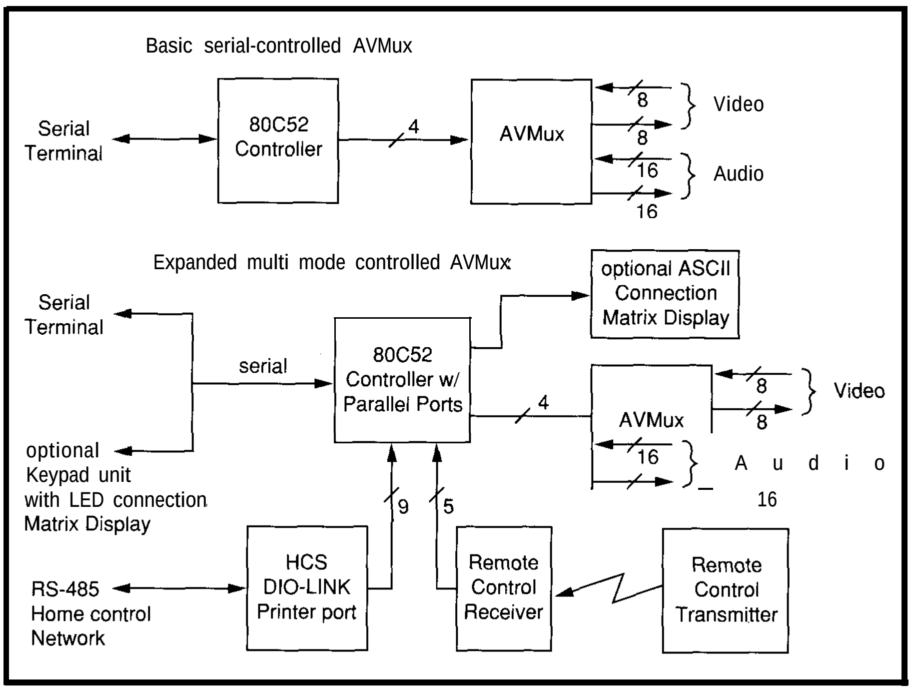
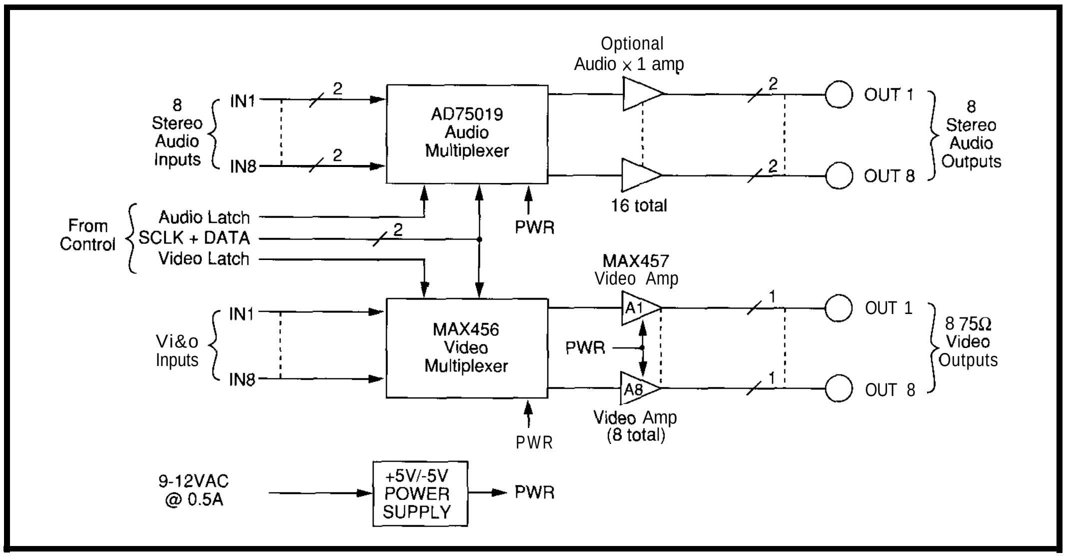


Figure l--The *AVMux setup* can be *as simple as a serial terminal controlling it, or as complex as using wireless controls, user displays, and the HCS II to make a truly powerful system.*

video rarely more than a 4x1 or 4x2. The Nxl audio configuration is because all the amplifiers are in the one box. Any sound created in the system has to be amplified by this box and all speakers are wired to it. If there are eight audio inputs and only a single output that goes to the amps, then we have an 8x1 audio mux. Similarly, since this integrated system does not incorporate its own monitor, the manufacturers facilitate an external connection through the mux. With four video sources and two outputs, we have a 4x2 video mux equivalent.



Figrue 2—*The Circuit Cellar A VMux uses a pair of chips (one from Analog Devices and the other from Maxim) designed specifically for doing audio and video multiplexing.*

OK, enough beating around the bush. If you want to put together an entertainment room and you are on a limited budget, these integrated systems are the only way to go. One hundred watts channeled through efficient speakers can sound quite good in any room.

The problem comes when you want to do something that is not part of the basic box. The instant you want to connect three video inputs where you only have two available, change the front surround speaker power output capability from 20 W per channel to 120 W per channel, or, using the same CD player, channel its output to a different set of speakers.

While the integrated system has greatly reduced the introductory cost of an entertainment room and satisfied bottom-up multiplexer execution to a great extent, it has done little for applications that don't fit the "mold."

As you might already have guessed, very little that I do fits any "mold." Let me explain.

Recently I decided to revamp our entertainment room. I thought of using an integrated surround system like those I described, but even the 350-W Mitsubishi unit didn't quite fit the bill. While it contained all the right elements, it didn't have anywhere enough power and its internal switch multiplexer (or that on virtually any integrated unit) was far too limited for the I/O selection I have.

In all truth, if I didn't already have all the amplifiers and many of the subsystem components, I might would have changed the rules to fit the solution, but the disparity in performance was still there. My attitude toward surround sound movies is to create an environment as close to the pictured event as possible. When we watch Top Gun, only the lack of exhaust fumes and salt spray keeps you from actually believing you are on a carrier deck. To achieve this sound envelope requires considerably more than 5 W through a 6" x 9" oval TV speaker. In my opinion, it also takes more than the 300-400 watts available from the current top-of-the-line integrated units.

My entertainment room technique involves using separate modules (preamp, surround decoder, etc.) instead of the integrated unit. The fact that a stand-alone surround decoder/sound field processor has all its signal outputs available, gives me the option of using any size amplifiers I want. Being brief, let me just say that I'm using about 1200 watts on the dozen speakers comprising the surround system. Included among them is a pair of 200-W subwoofers that guarantee you don't miss an F-14 fly by.

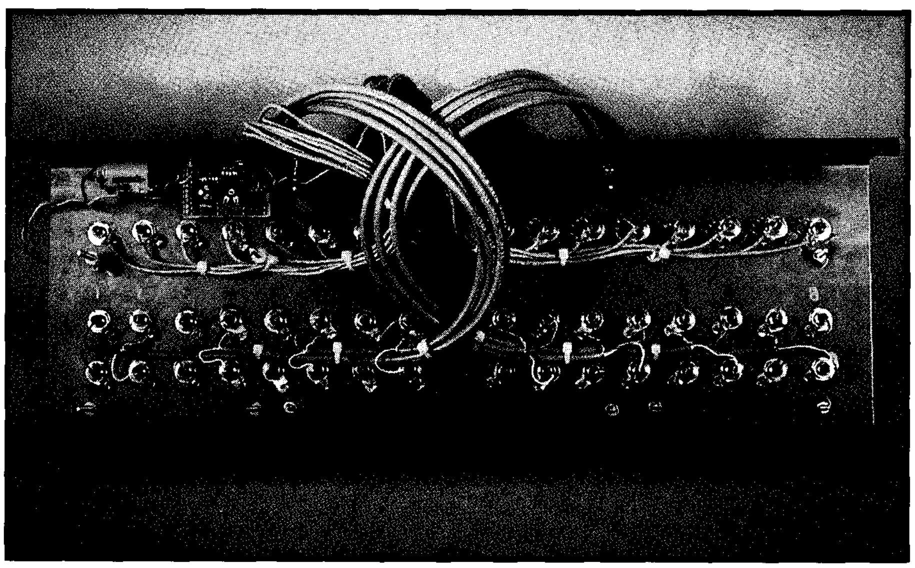
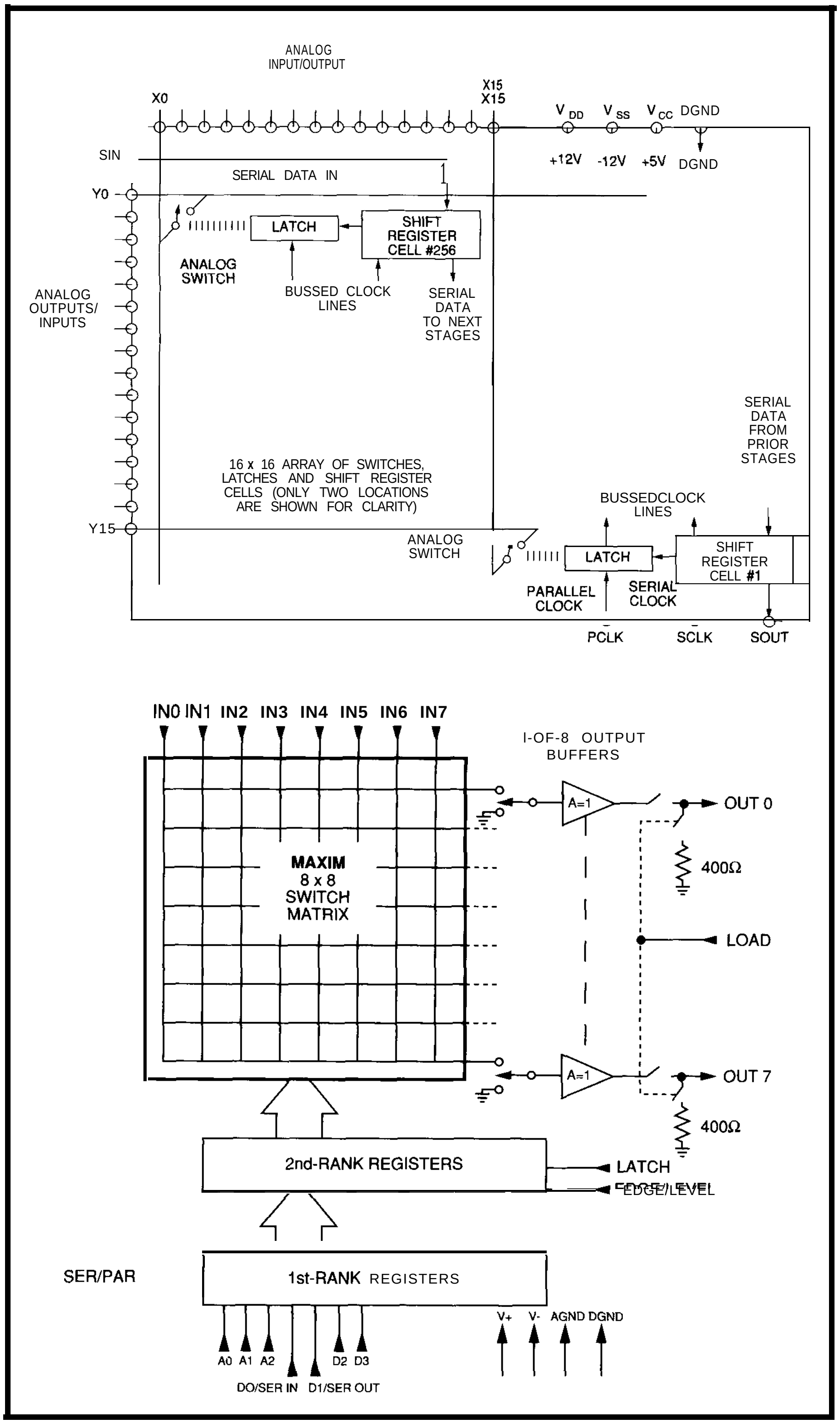


Photo **2**—*While the circuit isn't complex, wiring the project can be a major task since all the cab/es must be* shielded.

What finally convinced me to redesign and further employ an external AVMux was the implementation of dual systems. I'm sure you've heard the saying, "If it ain't broke, don't fix it!" Well, that can apply to stereo systems as well.

Take away all the sound effects processing, decoding, and electronic signal manipulation and true sound quality of any stereo is simply an audio source, amplifier, and a pair of speakers. In my opinion, the sound produced by a pair of B&W 808



Figrue *3-The Analog Devices AD75019 (top) is a 16x16 switch matrix idea/ for use in audio applications while the Maxim MAX456 (bottom) is an 8x8 switch meant for video.*

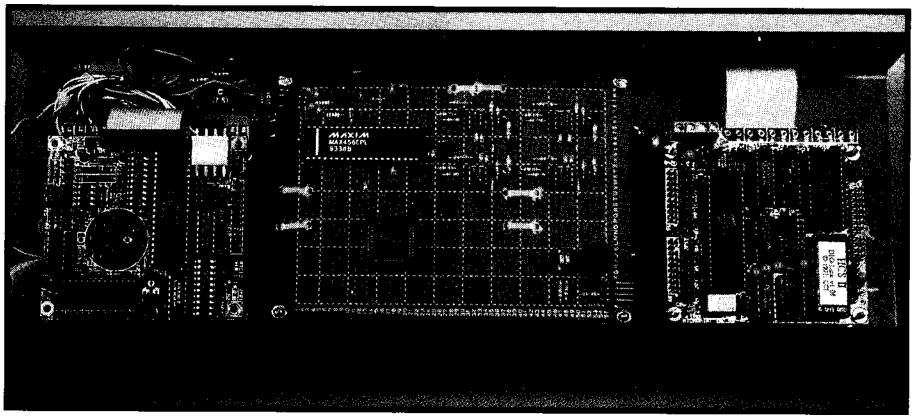


Photo 3—*Inside the finished AVMux you can see the controller (left), multiplexer electronics (center), and HCS II DIO-Link (right).*

Reference Speakers [running at another 1200 watts) comes close to perfect audio. I like classical music straight, while rock music seems better with all the effects. That means I have one system configuration for classical and a completely different one for rock.

Keeping a dual redundant system like this requires a sophisticated multiplexer to allow a common set of AV sources to be routed among all input combinations.

A SIMPLE MUX DESIGN WITH MANY OPTIONS

The new AVMux has improved performance primarily because it uses newer technology. Like the original unit, this is an 8x8 stereo audio and 8x8 video multiplexer. Unlike its predecessor, however, this mux allows the user to select independent or coordinated control of the audio and video sections. You can have eight audio sources with only three corresponding video signals and still use the other five video inputs for unrelated activities. Audio and video channels can be routed independently.

Due to its particular architecture, which allows independent audio and video operation, you can choose to make just a video-only or audio-only multiplexer. Control is realized with an 80C52-based microcontroller which allows considerable expansion options. As outlined in Figure 1, the new Circuit Cellar AVMux can operate through simple, direct, serial RS-232 control; operate with a wireless infrared remote control; operate with a hand-held keypad and LED connection matrix display; display an ASCII list of connected points during operation; and allow network connection (using a DIO-Link) with functional control provided by the Circuit Cellar HCS II.

As you can see, this is a long list of functions. It also incorporates features that involve considerably more software than I cared to write. To finish in a reasonable time, I drafted Jeff Bachiochi to help. I'm glad I did and I'd further have to say that this new AVMux has as many features as it does because of his added expertise. Since we worked together, Jeff will be covering certain AVMux aspects in his column as well.

For this month, I'll describe the basic AVMux components and the controller design. Next month, Jeff will fill in more on how the software works, with particular emphasis on the remote keypad/LED connection matrix unit. I'll conclude next month with the wireless remote control interface specifics.

NEW SINGLE-CHIP MULTIPLEXERS

Any rational person would not even attempt to make an 8x8 multiplexer if it were not for the availability of some new highly integrated multiplexer chips from Analog Devices and Maxim. The configuration of the basic AVMux using these chips is outlined in Figure 2. Regardless of all the other optional features, this basic circuit stays the same. The two sections—audio and video-are also divisible and you could build each separately.

The audio multiplexer is the simpler of the two, both in concept and construction. Employing an Analog Devices CMOS AD75019 16x16 crosspoint switch array (its functional block diagram is in Figure 3) allows the complete circuit to be fabricated with just that single chip.

Unlike its video companion, an 8x8 stereo audio multiplexer involves switching 16 signals (8 left channel and 8 right channel). I chose the AD75019 specifically so I would not have to use two 8x8 devices. The AD75019 contains 256 analog switches in a 16x16 array. Any of the X or Y pins may serve as inputs or outputs. Any of the X pins can be programmed to connect to any Y pins and vice versa. The switches allow amplitudes up to the supply voltages and have a typical resistance of 150 ohms.

Figure 4 shows the actual circuit connections. To keep things simple, we designated all X pins as inputs and all Y pins as outputs: X0–X7 are right channel inputs l-8, X8–X15 are left channel inputs l-8, YO-Y7 are right channel outputs l-8, and Y8–Y 15 are left channel outputs l-8. I know that chip designations use "0" instead of "1" as the first signal line, but try to tell someone unfamiliar with electronics that and you'll get a blank stare. Zero to seven is for the builder and one to eight is for the user.

Connecting input X2 to output Y7 is accomplished by closing the crosspoint switch at their intersection. Each of the 256 crosspoint switches is controlled by a shift register cell. If the cell contains a 0, the switch is open. If the cell contains a 1, the switch is closed. The contents of all the cells, describing the crosspoint matrix map, is a 256-bit data packet.

The data is loaded serially via the SIN input (DATA) and clocked into the 256-bit shift register via SCLK (CLK). When all 256 bits have been entered, data is transferred in parallel to the switch latches on the high-to-low leading edge transition of PCLK (ALD). One caution to note here: the serial shift register in the AD75019 is dynamic and the minimum clock rate is 20 kHz (the maximum rate is 5 MHz). With a high-level language controller, an assembly language routine is often used to send data to

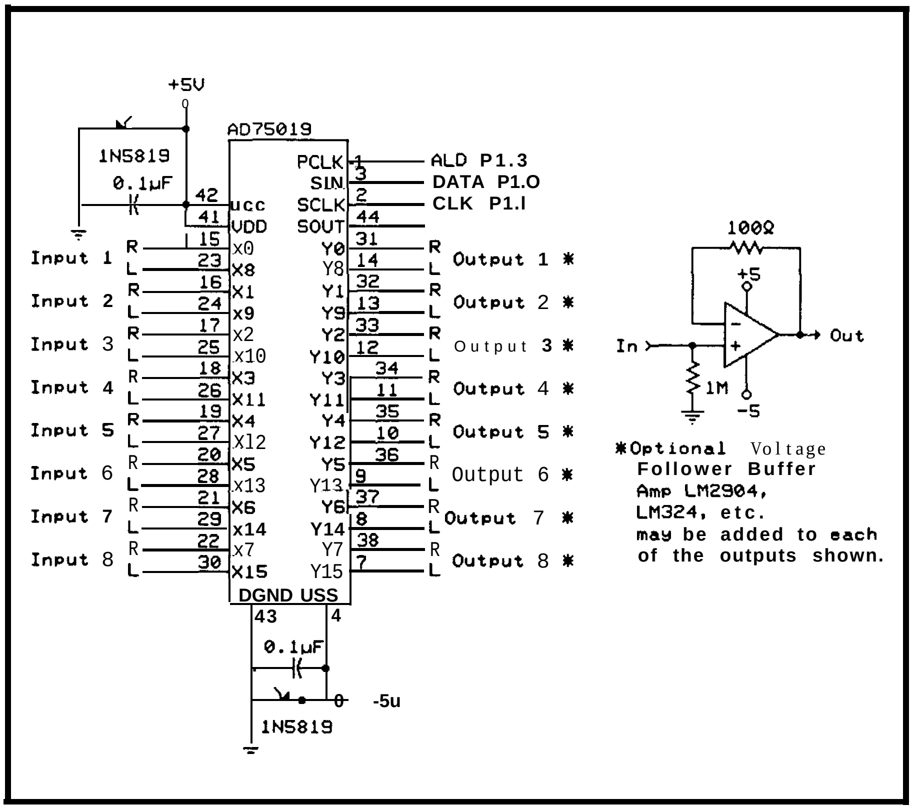


Figure 4—*The actual connections to the AD75019 are numbered l-8 rather than O-7 to make it easier on the user.*

this chip. Once programmed, however, operation is static and the switches stay programmed for as long as the power is on.

One final note on this chip. When powered on ±5 V, the AD75019 has a typical switch resistance of 300 ohms. Many applications may be unaffected by this added resistance and you can use the circuit as is. If you are unsure or desire extra output protection, add the voltage follower buffer amplifier to each output channel (16 amps total) as noted in the schematic.

A NEW MAXIM MULTIPLEXER

If you look closely at the spec sheet on the AD75019, you'll note that it has a switch frequency response of 20 MHz. While I could have used it for both sections, I opted to use a chip specifically designed for video.

I decided to make the AVMux video section using the new Maxim MAX456 CMOS 8x8 crosspoint switch (functional block diagram in Figure 3) specifically designed for video. It contains a similar register, latch, and switch arrangement as the audio mux, but the MAX456's connection logic prohibits illegal switch connections.

The MAX456 is an 8x8 matrix, and there are indeed 64 separate switches, but in video applications, not all crosspoint connections are allowed. Video outputs should not be tied together, for example.

In the MAX456, each output connection command is thought of as a combination of the binary address of the selected l-of-8 inputs and data such as whether the buffer is on, off, or grounded. This data combination results in a 4-bit value for each output channel. To program 8 channels, therefore, only requires 32 bits.

The MAX456 is unique in that it can load this switch register data by either parallel or serial means. To keep programming consistency, I chose the latter. The 32 bits are applied to the D0/SER IN (DATA) line as the ● WR (CLK) line is clocked (these are the same two lines that are connected to the audio mux). At its conclusion, the data is latched on a high-to-low leading-edge transition of the LATCH pin (VLD). Using common data and clock control lines with separate video load (VLD) and audio load (ALD) allows just four control lines to control the whole AVMux.

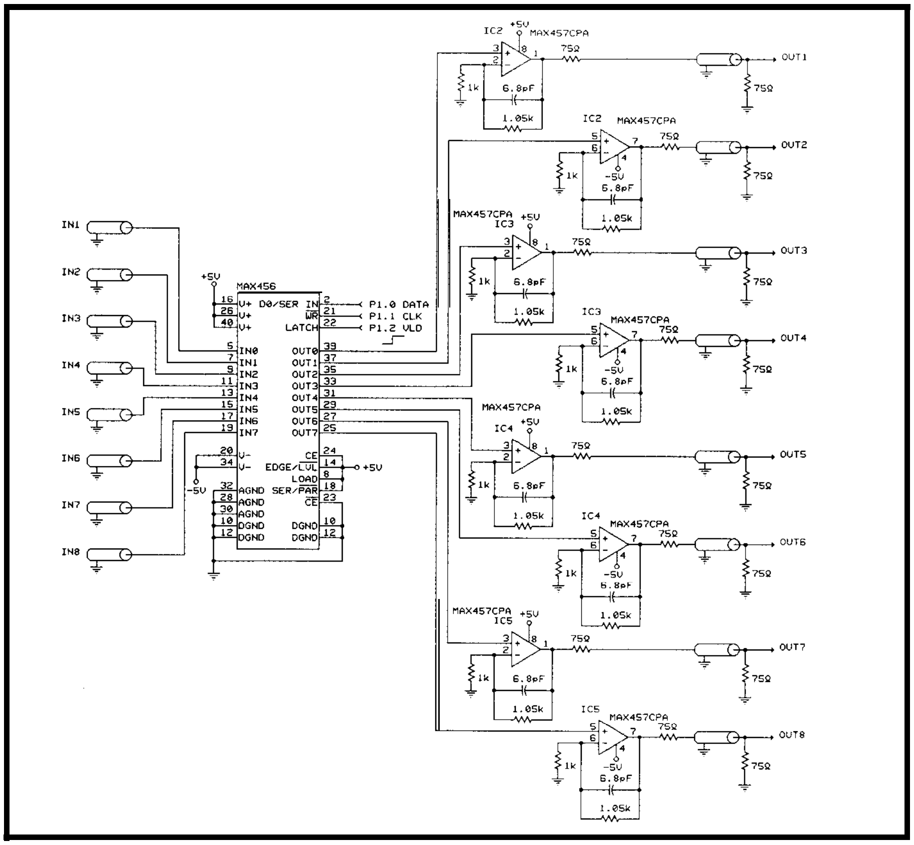


Figure **5**—*While the circuitry around the MAX456 video mux is simple, having to use shielded cable complicates wiring. The MAX457 video amps are used for impedance matching.*

One additional note on the MAX456. While it does have internal video buffers, they can only drive 400-ohm loads. Since most video applications require 75-ohm impedance, additional external buffer amplifiers have to be added. The dual-amp MAX457, which has a 70-MHz unity-gain bandwidth, is the perfect choice. As connected in Figure 5, the MAX457 is configured for a closed-loop gain of 2. The gain selection resistor is set at 1.05k instead of I k to make up for open-loop gain loss. The 6.8-pF capacitor helps eliminate phase delay at high frequencies.

We have to be careful to properly match the cable impedances even in a hand-wired prototype. This particular circuit allows us to drive a doubly terminated 75-ohm line. The combination of the series 75-ohm resistor and the 75-ohm termination at the output connector produces a low-noise 75-ohm external connection with the appearance of unity-gain amplification.

A FAMILIAR CONTROLLER FOR THE AVMUX

As I mentioned, it takes just 4 wires from the controller to the AVMux chips to do everything. Figure 6b is an 80C52-BASIC-based microcontroller circuit which is more than adequate for the task. Together, they represent the configuration outlined in Figure 1. I used a Micromint RTC52 and RTCIO as the controller rather than spending the time and effort to build one. At the price I get them, it wasn't a hard choice.

We chose to use BASIC (with assembly language calls) here because this is not a speed-intensive control application. Using a high-level language also expedited development and made certain expansion options easier. For example, since BASIC-52 supports a serial printer output, presenting an ASCII connection list for terminal display was as easy as adding an LPRINT (PRINT#) command.

The controller circuit has two sections: basic controller with one serial port and one parallel port (Figure 6a), and the same basic configuration with more parallel ports and nonvolatile memory (Figures 6a and 6b}. Since this circuit has been presented on numerous occasions, I will not belabor it. The initial circuit is suitable for operating the AVMux through a serial terminal. A simple connection display and command menu prompts the user to designate the desired From-To connections. More later.

PROTOTYPING

Success here is not just a function of interpreting the schematic correctly. Experience counts.

As you can see from the photos, even though this is only a six-chip circuit, it is a wiring nightmare. To keep signals from radiating or picking up the radiation from adjacent signals, all input and output wires should be shielded cable [now you might understand the reason for doubly terminated video buffers). I even used shielded cable from the MAX456 to the buffer amps. The enclosure should be metal to serve both as an EM1 shield and as a large noise-reducing common ground. It can also function as the heatsink for your power supply.

Ordinarily, I wouldn't discuss the power supply in any detail. In an application where limiting electrical noise is critical, I believe it a noteworthy exception. First of all, I suggest you use a linear supply only and not a switching regulated unit. Without the benefits of printed circuit board layout, ground planes, single-point termination, and proper shielding (all of which are incorporated in a production unit), a switching power supply is just another source of unwanted noise (herringbone pattern on the video and buzz in the audio).

Getting from here to there can be an interesting journey, however. I designed my AVMux as a totally enclosed unit. My intention was to use one of those cheap wall-module power sources and an internal regulator. The AVMux operates on +5 V and -5 V. Unfortunately, all the cheap wall-modules seemed to be 2-wire output.

Since I couldn't just throw in a switching regulator or charge pump inverter to get -5 V, I concluded that my only option was an elaborate dual tracking regulator with a concocted ground. Would I have to resort to an external power supply and pipe everything in instead?

The answer turned out to be really low tech. Shown in Figure 7, the AVMux power supply is just a regulated-output AC-to-DC converter. We can get by with a 2-wire voltage source if it is AC. The AC is then half-wave rectified and stored as separate positive

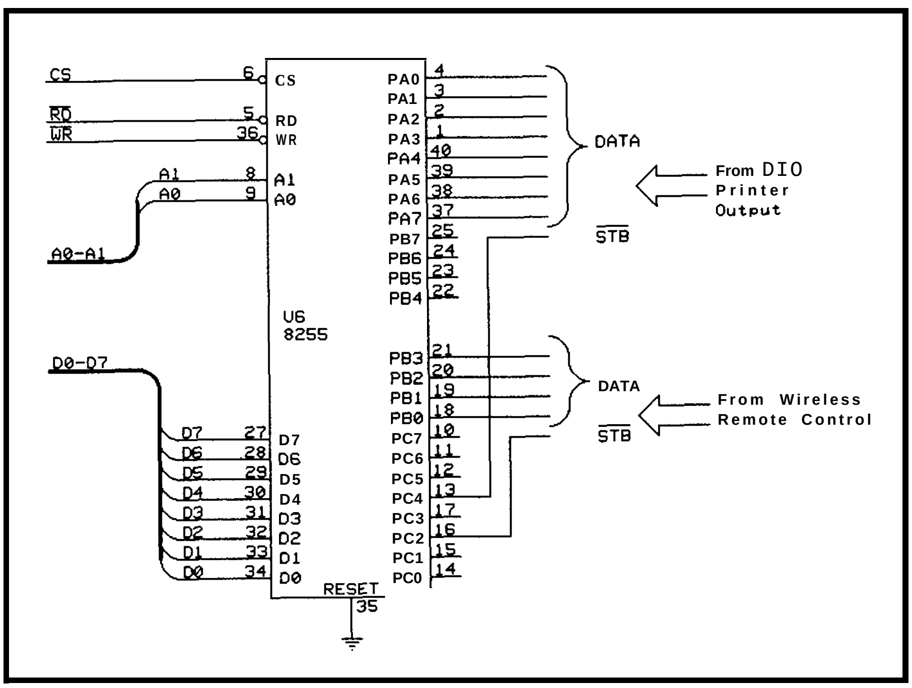
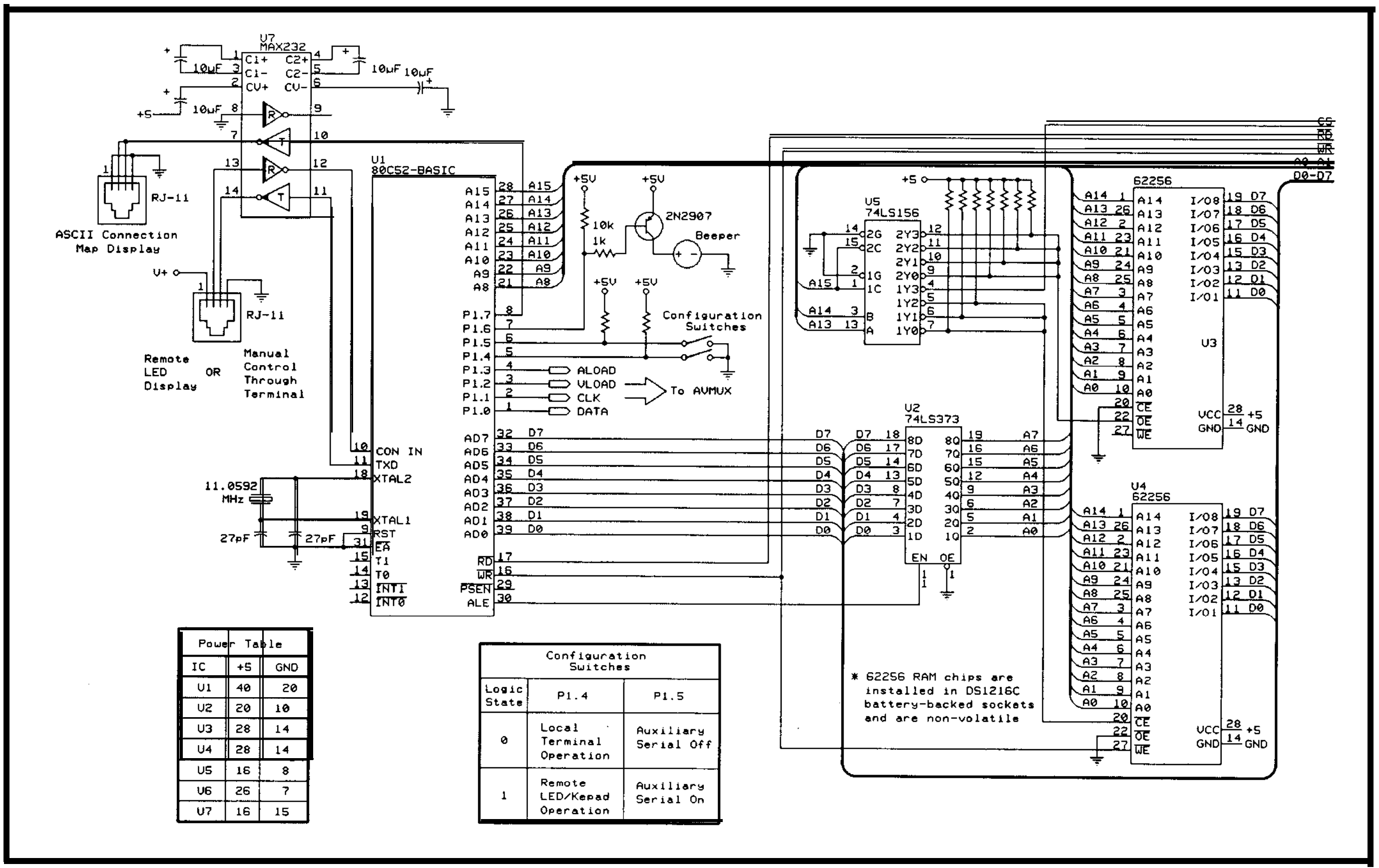


Figure 6a—*To add either HCS II or wireless remote control to the AVMux, an 82C55 must be added to the core controller in Figure 6b.*



Power Table		
IC	+5	GND
U1	40	20
U2	20	10
U3	28	14
U4	28	14
U5	16	8
U6	26	7
U7	16	15

Configuration Switches		
Logic State	P1.4	P1.5
0	Local Terminal Operation	Auxiliary Serial Off
1	Remote LED/Kepad Operation	Auxiliary Serial On

Figure **6b**—*The control section of the AVMux uses the same 80C52-BASIC core seen in many other projects. Connections to the outside world are done through RJ1 1 jacks*

and negative supplies with a common ground. Two three-terminal regulators simply convert this to +5 V and -5 V, respectively. The one notable circuit peculiarity is that each regulator has been configured for voltage output adjustment with a pot. By referencing these pots to the opposite supplies, the overall input-output differential necessary for these three-terminal regulators is reduced. Rather than a 12-VAC input, a 9-VAC will work just fine. The benefit is reduced power dissipation and a cooler AVMux box.

AVMUX OPERATION

The AVMux is designed to operate in a variety of modes. With just the basic serial controller, all user interaction is through a terminal. When initialized, the program displays a connection matrix. To set a connection, the user designates whether it is (A)udio, (V)ideo, or (B)oth. Answering (A) displays the audio matrix while (V) or (B) displays the video matrix. The next entries are the From and To channel numbers, or cancel. The program then repeats the display, showing it with the new connections (Jeff's program is on the Circuit Cellar BBS for anyone building the AVMux).

As I mentioned, by using BASIC we have a serial printer output also available. One program function physically documents the entire From-To connection list, by component names (such as From ADS CD Player To Nakamichi Preamp Aux 2 input). This is for people who don't want to carry a component list to remember channel numbers or for those who want to have space in their stereo cabinet and want a unique terminal display.

EXPANDING AVMUX FEATURES—THE HCS CONNECTION

Adding the 8255 PPI to the basic controller circuit facilitates other AVMux features. Initial operation is through a serial ASCII terminal. Considering that the communication is simply the 12 characters typically on a telephone keypad, other methods can be employed to communicate these characters.

The first option we thought of was a hand-held remote that presented a display of the connected channels on an 8x8 LED array. It also had an integral keypad to enter commands.

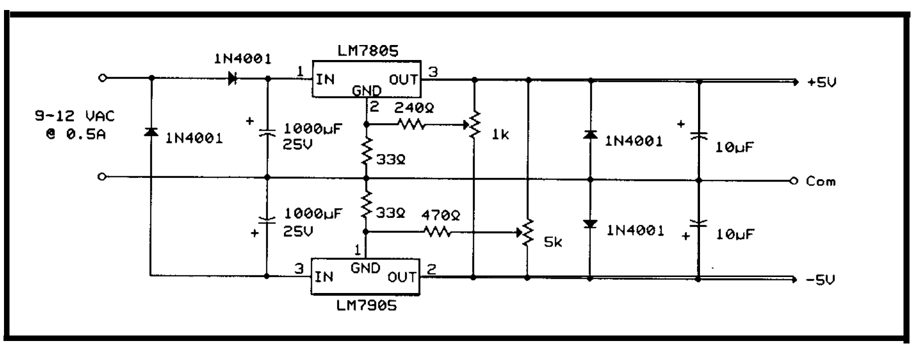


Figure **7**—*The power supply section creates quiet positive and negative supplies for the amplifiers from an AC source.*

This construction of this unit and a description of its software will be presented by Jeff next month.

The next brain storm we had was to connect the AVMux so it could work independently from, or in conjunction with, the HCS II and the trainable infrared MCIR-Link. The connection is accomplished using the DIO-Link.

Normally, the DIO-Link is used as an 8-bit, bit-programmable I/O port on the HCS II's RS-485 network. The DIO-Link can also be set to drive a parallel printer. In this configuration, all bits are outputs and each character transfer is accompanied by a strobe pulse. Character transfer timing is programmable within the DIO-Link.

Port A and one bit of Port C are used to receive the parallel data. The HCS can command the AVMux by simply sending an ASCII string, such as "A35" (make an Audio connection from 3 to 5), to the DIO-Link printer port. It's as easy as that.

Since we were accepting parallel data inputs anyway, it didn't seem outrageous to consider an additional source. The total 12-character vocabulary could also be represented as 12 combinations within a 4-bit code. It shouldn't be too hard to communicate these codes using readily available remote control transceiver chips. In fact, next month I'll demonstrate such a wireless remote interface design. ■

Steve Ciarcia is an electronics engineer and computer consultant with experience in process control, digital design, and product development. He may be reached at steve.ciarcia@circellar.com.

SOURCE

Some components of this project may be purchased from

Pure Unobtainium
13109 Old Creedmoor Rd.
Raleigh, NC 27613
Phone/fax: (919) 676-4525

I R S

401 Very Useful
402 Moderately Useful
403 Not Useful