ELM624

Control L (LANC) to RS232 Interpreter

Description

The Control L or LANC interface is an industry standard introduced by the Sony Corporation for controlling audio and video devices. It uses a bit serial data format, and requires that the controller be synchronized to the controlled device – something which is difficult to do using standard serial interfaces. The ELM624 is an 8 pin integrated circuit that performs the synchronizing function for you.

All user interaction with the ELM624 is by standard ASCII characters over an RS232 interface. There is no special formatting required, other than perhaps an understanding of the hexadecimal numbering system, nor is there a need for a powerful PC - virtually any model with a serial port will do.

Since the ELM624 was designed to provide a cost-effective way for people to experiment with the Control L system, many features typically found in commercial devices, such as RS232 handshaking, variable baud rates, extra buffering of signals, etc. have not been implemented. Responses are kept to a minimum as well (eg. a single question mark is returned for a misunderstood command), but the general principles of operation are demonstrated and for many applications, this is all that is required.

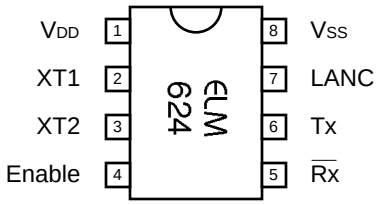
Applications

- Video editors
- Time-lapse recording controllers
- Programmed control of A/V equipment
- Remote camera controls

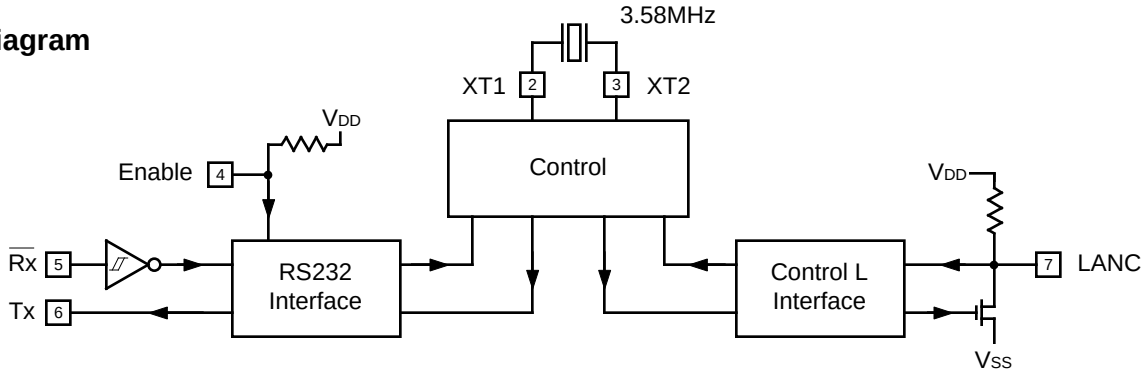
Features

- Low power CMOS design
- Enable input allows control of multiple devices
- Configurable with simple AT commands
- ASCII output formatted as standard hex digits
- Oscilloscope trigger pulse output
- Power control pulse output
- Crystal controlled for timing accuracy
- Works with 50 Hz and 60 Hz systems
- Power up to monitor mode

Connection Diagram
PDIP and SOIC
(top view)



Block Diagram



Control L and LANC

The terms Control L and LANC mean the same thing, and are used interchangeably throughout the following.

Pin Descriptions

V_{DD} (pin 1)

This pin is the positive supply pin, and should always be the most positive point in the circuit. Internal circuitry connected to this pin is used to provide power on reset of the microprocessor, so an external reset signal is not required. Refer to the Electrical Characteristics section for further information.

XT1 (pin 2) and XT2 (pin 3)

A 3.579545MHz NTSC television colourburst crystal is connected between these two pins. Crystal loading capacitors (typically 27pF) will also normally be connected between each of the pins and Vss.

Note that this frequency is used for both 50Hz and 60Hz systems. The ELM624 automatically adjusts to the system that you are using.

Enable (pin 4)

This is an active high input that controls RS232 data flow to and from the ELM624. If at a high level, the RS232 data communications are enabled, while a low input disables them. When disabled, all RS232 data sent to the IC is ignored, and no RS232 output is generated by the ELM624. See the Example Applications section for more on this.

$\overline{Rx}$ (pin5)

The computer's RS232 transmit signal is directly connected to this pin through a single current limiting resistor (typically about 47KΩ). Internal signal inversion and Schmitt trigger waveshaping provide the necessary signal conditioning.

Tx (pin 6)

This is the RS232 data output pin. The signal polarity is compatible with most interface ICs, and the drive is sufficient to allow interfacing using only a single PNP transistor if desired. See the Example Applications section for more details.

When selected for Trigger Pulse outputs, this pin provides the active low output pulses.

LANC (pin 7)

This is the open drain Control L (LANC) interface pin. An internal pullup resistor is provided for a nominal drain load.

V_{SS} (pin 8)

Circuit common is connected to this pin. This is the most negative point in the circuit.

Ordering Information

These integrated circuits are available in either the 300 mil plastic DIP format, or in the 200 mil SOIC surface mount type of package. To order, add the appropriate suffix to the part number:

300 mil Plastic DIP..... ELM624P 200 mil SOIC..... ELM624SM

Absolute Maximum Ratings

Storage Temperature..... -65°C to +150°C

Ambient Temperature with
Power Applied.....-40°C to +85°C

Voltage on V_{DD} with respect to V_{SS}..... 0 to +7.5V

Voltage on any other pin with
respect to V_{SS}..... -0.6V to (V_{DD} + 0.6V)

Note:

Stresses beyond those listed here will likely damage the device. These values are given as a design guideline only. The ability to operate to these levels is neither inferred nor recommended.

Electrical Characteristics

All values are for operation at 25°C and a 5V supply, unless otherwise noted. For further information, refer to note 1 below.

Characteristic	Minimum	Typical	Maximum	Units	Conditions
Supply Voltage, V_{DD}	4.5	5.0	5.5	V	
V_{DD} rate of rise	0.05			V/ms	see note 2
Average Supply Current, I_{DD}		1.0	2.4	mA	see note 3
Input low voltage	V_{SS}		0.15 V_{DD}	V	
Input high voltage	0.85 V_{DD}		V_{DD}	V	
Output low voltage			0.6	V	Current (sink) = 8.7mA
Output high voltage	V_{DD} - 0.7			V	Current (source) = 5.4mA
Internal pullup resistances (see note 4)	300 20	500 30	600 50	KΩ KΩ	Pin 4 (Enable) Pin 7 (LANC)
$\overline{Rx}$ pin input current	-0.5		+0.5	mA	see note 5
RS232 Baud Rate		9600		baud	see note 6

Notes:

1. This integrated circuit is produced with a Microchip Technology Inc.'s PIC12C5XX as the core embedded microcontroller. For further device specifications, and possibly clarification of those given, please refer to the appropriate Microchip documentation.
2. This spec must be met in order to ensure that a correct power on reset occurs. It is quite easily achieved using most common types of supplies, but may be violated if one uses a slowly varying supply voltage, as may be obtained through direct connection to solar cells, or some charge pump circuits.
3. Device only. Does not include any load currents.
4. The value of the internal pullup resistance is both supply and temperature dependent.
5. This specification represents the current flowing through the protection diodes when applying large voltages to the $\overline{Rx}$ input (pin 5) through a current limiting resistance. Currents quoted are the maximum continuous.
6. Nominal data transfer rate. Assumes that a 3.58 MHz crystal is used as a frequency reference. Data is transferred to and from the ELM624 with 8 data bits, no parity, and 1 stop bit (8N1).

Overview

The following describes how to use the ELM624 to control, and to obtain information from your LANC device. We begin by discussing just how to talk to the IC, then how to adjust some options through the use of 'AT' commands, and finally go on to actually talk to the LANC device, both obtaining status codes and sending commands. For the more advanced experimenters, there are also sections on how to use some of the other features of this product as well.

For experimenting, all that is required is a PC or a PDA with a terminal program (such as HyperTerminal or ZTerm), and knowledge of one or two LANC commands, which we provide in the following…

Communicating with the ELM624

The ELM624 relies on a standard RS232 type serial connection to communicate with the user. The data rate is fixed at 9600 baud, with 8 data bits, no parity bit, 1 stop bit, and no handshaking (often referred to as 9600 8N1). All responses from the IC will be terminated with a single carriage return character and by default, a line feed character as well. Make sure that your software is configured for this type of communication.

Properly connected and powered, the ELM624 will initially display the message:

```
ELM624 v3.0

>
```

In addition to identifying the version of the IC, receipt of this string is a convenient way to be sure that the computer connections and the settings are correct. However, at this point no communications have taken place with the LANC device, so the state of that connection is still unknown.

The '>' character displayed above is the ELM624's prompt character. It indicates that the device is in its idle state, ready to receive characters on the RS232 port. Characters sent from the computer can either be intended for the ELM624's internal use, or for reformatting and passing on to the LANC device.

Commands for the ELM624 are distinguished from those to the LANC device by always beginning with the characters 'AT' (as is common with modems), while commands for the LANC bus must contain only the ASCII characters for hexadecimal digits (0 to 9 and A to F). This allows the ELM624 to quickly determine where the received characters are to be directed.

Whether an 'AT' type internal command or a hex string for the LANC bus, all messages to the ELM624 must be terminated with a carriage return character (hex '0D') before it will be acted upon. The one exception is when an incomplete string is sent and no carriage return appears. In this case, an internal timer will automatically abort the incomplete message after about 20 seconds, and the ELM624 will print a single question mark to show that the input was not understood (and was ignored).

Messages that are not understood by the ELM624 (syntax errors) will always be signalled by a single question mark ('?'). These include incomplete messages, invalid hexadecimal digit strings, or incorrect AT commands. It is not an indication of whether or not the message was understood by the LANC device. (The ELM624 is a protocol interpreter that makes no attempt to assess LANC messages for validity – it only ensures that either four or eight hex digits were received, combined into bytes, and sent out the LANC port. It cannot make judgement on the actual bytes that were sent.)

Incomplete or misunderstood messages can also occur if the controlling computer attempts to write to the ELM624 before it is ready to accept the next command (as there are no handshaking signals to control the data flow). To avoid a data overrun, users should always wait for the prompt character ('>') before issuing the next command.

Finally, a few convenience items to note. The ELM624 is not case-sensitive, so 'ATZ' is equivalent to 'atz', and to 'AtZ'. Also, the device ignores space characters as well as control characters (tab, linefeed, etc.) in the input, so they can be inserted anywhere in order to improve readability.

AT Commands

The ELM624 can accept internal configuration commands at any time, in much the same manner that modems do. Any command sent to the ELM624 which begins with the letter 'A' followed by the letter 'T' is assumed to be an internal configuration (or 'AT') command. These commands are executed upon receipt of the terminating carriage return character, and are acknowledged with some form of response, which may be as simple as the characters 'OK'.

The ELM624's factory default settings should be appropriate for most applications, but some users may wish to customize their settings, such as turning the character echo off, or perhaps quiet mode on, by issuing these commands. Doing this is as easy as sending AT E0, or AT Q1, followed by the return character. Note that this version of the ELM624 is only able to accept one command per line, so changing multiple settings requires one line for each change.

The following summarizes the 'AT' commands that are recognized by this version of the ELM624. Note that the character '0' is the number 'zero':

2D [use version 2 Defaults]

This command is used to set several of the options to their default (or factory) settings, as if this were a v2.x IC. In the 2D mode, linefeed characters are not sent automatically after each carrige return, and there is no prompt character generated when the IC is ready to accept the next command. Other than that, the E, C, D, Q and R options are set the same as those discussed here for the v3.0 IC.

C0 and **C1** [send all (0) or only Changes (1)]

These commands specify when the Control L status bytes are to be returned on the RS232 bus. With the C1 command, values are only sent when there is a change from the previous eight bytes, while with C0 the response bytes are always sent. For most devices, setting C1 will have little noticeable effect, as the LANC responses usually alternate between status and time-code values, so they do continually change. The default is C1, send on change.

CS [Check Sync]

This command is used to determine (check) if the connected LANC device is sending synchronizing signals or not. If the signals are correct, a 'SYNC OK' message will be returned. If there is a problem, either 'NO SYNC' or 'SYNC ERROR' will be returned. This provides a quick way to see if a device is powered and ready to accept commands.

D [set all to Defaults]

This command is used to reset the E, C, D, L, Q, and R options to their default (or factory) settings, as when power is first applied. It provides a fast and efficient way to restore the settings without having to wait through the time delay of a power up reset.

D0 and **D1** [Duplicate off (0) or on (1)]

While the LANC message structure allows for four command bytes to be sent, most devices only use the first two, and ignore the others. For convenience, the ELM624 can be provided with either two or four bytes to send, but if it is provided with only two, it needs to know what to insert into the other two positions of the four byte command field.

If the duplicate option is turned on, the two command words that were provided to the ELM624 will be inserted into the word 0 and word 1 positions, and then will be duplicated and used for the other two positions (words 2 and 3). If the duplicate option is off, no duplication will occur, and 0's will be sent in positions 2 and 3 instead. Setting the duplicate option to off may be useful if you suspect that a unique device is providing status bytes in positions 2 and 3 and you do not want to overwrite them. The default setting is D1, duplicate on.

E0 and **E1** [Echo off (0) or on (1)]

These commands control whether all characters received on the RS232 port are retransmitted (or echoed) back to the host computer. To reduce traffic on the RS232 bus, and perhaps simplify some computer software, users may wish to turn echoing off by issuing E0. The default is E1, echo on.

AT Commands (cont'd)

FD [send Formatted Data]

This command requests that all responses from the LANC device be sent as standard ASCII characters which are readable with virtually any terminal program. The four status bytes (words 4, 5, 6 and 7) will be sent as eight hexadecimal digits, with two ASCII characters representing each byte. There is insufficient time at 9600 baud to insert spaces to separate these bytes, so they are simply sent as a block of eight characters. Every line will end with a carriage return character and (optionally) a linefeed character, ensuring that all responses appear on a new line. This is the default mode.

I [Identify yourself]

Issuing this command causes the chip to identify itself, by printing the startup product ID string (this is currently 'ELM624 v3.0'). Software can use this to determine exactly which integrated circuit it is talking to, without resorting to resetting the entire IC.

L0 and **L1** [Linefeeds off (0) or on (1)]

Many computer terminal programs expect a linefeed character (hex 0A) to be sent after each carriage return character (hex 0D). The sending of this linefeed character is controlled by this option. Users may find that for general use, leaving linefeeds on is preferrable, but for some computer controlled applications, they may not require it (and could find that it only serves to slow processing down). The default setting is L1, linefeeds on.

MA [Monitor All messages]

Using this command places the ELM624 into a bus monitoring mode, in which it displays all messages as it sees them on the LANC bus. This continues indefinitely until stopped by activity on the RS232 input. To stop the monitoring, one should send any single character then wait for the ELM624 to respond with a prompt character ('>'). Waiting for the prompt is necessary as time to respond is unpredictable, and depends on what the IC was doing when interrupted. If it were in the middle of printing a line, it would first complete that line before sending the prompt character, but if it were simply waiting for input, it would return immediately. The character which stops the monitoring will always be discarded, and will not affect subsequent commands.

Q0 and **Q1** [Quiet mode off (0) or on (1)]

This is a convenient means to stop the continuous flow of status messages that occur while you are experimenting. If quiet mode is selected, the chip will function normally in all respects, except that the sending of the LANC status messages will be stopped. The default setting is Q0, quiet mode off.

RD [send Raw Data]

There may be times when one would like to see all eight of the LANC words that are in a message instead of just the four status bytes. There is not enough time to send sixteen hexadecimal digits, a carriage return and perhaps a linefeed character while working with 60Hz systems, however. In order to send this information, it must be kept in it's un-converted or 'raw' form.

In the raw data mode, the ELM624 performs no translation of the received LANC data. It simply leaves each byte as the raw value which was received, and resends them to the connected PC along with a single terminating carriage return character. No linefeed is sent after the carriage return, regardless of the AT L0/L1 setting. In the Raw Data mode then, each eight byte LANC message will always appear as a nine byte RS232 message.

This option will likely find limited use by many users since many of the received values will represent unprintable characters on a terminal screen, so will require special capturing and processing for use. If it is absolutely necessary to see what the value of the four command words are, however, this is a means to do so. Note that the values shown in the raw data response are the result of actual bus reads, and not simply a regurgitation of what is in the transmit buffer. If there are bus conflicts or wiring problems, the values may differ. By default, this mode is off.

R n [Repeat commands n times]

This sets the LANC command repeat value. Although commands are only sent from the computer to the ELM624 once, they must be sent to the Control L device multiple times in order to be recognized. The repeat value supplied ('n') can be any single hex digit, which allows values in the range from 0 to 15 (hex F). Sending a 0 as the parameter (AT R0) is a special case, which causes the command bytes to

AT Commands (cont'd)

be sent continually, until interrupted by the user.

Initially, the Control L/LANC standard required that commands be repeated at least four consecutive times to be valid (even though some machines respond with less). The AT Rn command allows this parameter to be adjusted as you experiment. The default value for this setting is five (R5).

SP [Send a Pulse]

The power to some Control L/LANC devices can be controlled by sending a short pulse on the LANC data bus. When issued, the AT SP command sends a 150msec wide pulse for this purpose. See the Power Control section for more information.

TP n [Trigger Pulse output on word n]

Selecting this mode causes a short, negative-going pulse to be output on pin 6 at the beginning of the LANC word/byte selected by 'n'. These pulses will continue to be sent until interrupted by the user, as long as there are synchronizing pulses to lock to. Allowable values of 'n' are from 0 to 7.

The output pulse width is nominally one bit wide at 9600 baud (104µsec), and is meant to be used for triggering an oscilloscope or logic monitor, in order to view word 'n' in more detail. Note that the RS232 transmit circuitry can remain connected while using this option, as the short pulse will simply be seen as a start bit for a byte of value 0xFF. Normal RS232 output is stopped while in this mode.

Z [reset all]

Sending AT Z causes the ELM624 to perform a complete reset, as if power were turned off, and then on again. All settings we be returned to their default values, and the IC will be waiting for user input.

? [status ?]

Issuing this command causes the current status bytes to be continually obtained from the LANC device, without sending any command to it. This is an alternate way to issue the Monitor All command, and is functionally identical to it.

AT Command Summary

Figure 1 shows all of the ELM624 commands in one convenient chart. In order to help with the understanding of these, we have grouped the commands into three functional areas, but this has no bearing on how the commands should be used, it is only for clarity. You may find this chart to be useful when experimenting with the IC.

ELM624 AT Commands

general

2D	use v2 Defaults
D	use Default settings
I	show the ID string
Z	reset all

responses

C1/0	Show on Change
E1/0	Echo on/off
L1/0	Linefeeds on/off
FD	use Formatted Data
Q1/0	Quiet mode on/off
RD	use Raw Data

requests

CS	Check for Sync
D1/0	Duplicate to w2/w3
MA	Monitor All
Rn	Repeat n times
SP	Send (power) Pulse
TP n	Trigger Pulse on n
?	Monitor All

Figure 1. ELM624 v3.0 AT Commands

The Control L Data Format

In a ‘Control L’ or ‘LANC’ system, data is sent in groups of eight bytes at a time, with all groups repeating at either a 50Hz or a 60Hz rate depending on where the system was purchased. Information flows both ways using these eight bytes, with the first four bytes usually being sent to the device being controlled, and the remaining four being responses from that device. Except for voltage level differences, each byte is transferred in a manner which is identical to that of the RS232 standard (that is common throughout the computer industry). If the signal were inverted, it could likely be input directly to a computer’s serial port, and read as 9600 baud data. The data could not be displayed directly by a terminal program however, as it would be raw data and would have to be converted to ASCII characters for display on a screen.

While receiving data is relatively easy, the sending of data to the controlled device is not. All transfers must be sent in synchronism with the eight pulses which are generated by the controlled device, and this must be done with only a few microseconds of error. Responding to a real-time signal such as this is very difficult to do with many modern computer systems due to the complexity of their operating systems. The ELM624 can handle this easily however, as it is a dedicated device. It receives the data from the computer or PDA, checks for errors, then synchronizes to the time signals and sends the data out. Data received during the frame is reformatted and sent to the controlling computer during the next frame. There is an idle period between frames (typically about 5msec) when very little activity actually occurs, so the IC uses this to prepare for the next frame, and to monitor the RS232 Rx line for any new commands.

Occasionally, the ELM624 will be asked to send a Control L signal on the bus, and it will be unable to sense any synchronizing pulses being sent by the controlled device. If the device is not providing these pulses for any reason, the ELM624 will simply display ‘NO SYNC’, and will return to the ready mode awaiting another command. If this error does occur, check your connecting cables and power supplies. The Check Sync (AT CS) command can be used to verify that all is well before proceding.

The figure below is representative of a typical frame of data that is sent on the Control L or LANC bus. Commands to the device (from the ELM624) are sent during the first half of the frame, while the second half (the last four bytes) are generally for feedback from the controlled device. Often, four bytes are not enough for the device to send all of the information, so data is multiplexed over several frames. An example of this is the time codes which will be discussed in the next section.

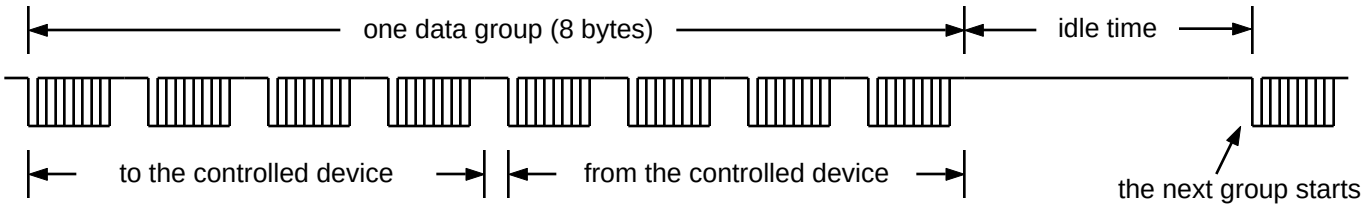


Figure 2. Control L data grouping

Talking to Your LANC Device

The ELM624 allows for two types of messages. When you send a string of characters to the ELM624 that begins with the letters ‘A’ and ‘T’, the message is used internally as discussed before. If the message is a series of hexadecimal digits, it will be assumed that you are trying to talk to the LANC device.

Hexadecimal digits can take on values from 0 to 9, and from A to F, representing decimal values 0 to 15. Two hex digits are combined to form a byte, and are usually shown that way (in pairs). It is very common for charts of Control L/LANC codes to use hex digits to show the control commands, so for that reason, the ELM624 was designed to use them as well.

When the ELM624 receives hexadecimal digits, it will also check to be sure that you have provided either four or eight of them. If there were four hex characters received, they will be combined in pairs to form the first two control bytes, and the two remaining control bytes will be set depending on the AT D0/1 setting. If there were eight hex digits received, they will be combined into four pairs and used for the four command bytes. Once the ELM624 has the commands ready, it will synchronize to the LANC device, and send them. Note that the use of four control bytes (eight hex digits) is quite rare but is a feature provided by the ELM624 in case you wish to experiment. For most devices, only the two command bytes (four hex digits) are needed, and that is what we show here.

After sending a command, all responses received from the LANC device are reported back to the user, just as if an AT MA command had been issued. If the default formatted data option is selected, all data will be shown as hex digits, using standard ASCII characters. Control characters are never sent – the hexadecimal digit ‘A’ for example, is transmitted as the decimal value 65 (a letter ‘A’), and not as 10 (a linefeed character).

In order to send a command to the LANC device then, all the user need do is type four hex digits then press return, and all of the data conversion, formatting, synchronizing, etc. is done by the ELM624.

There are several sources of information on the web for command codes, and the manufacturer of your camcorder may be able to provide some information as well. For the purpose of this discussion, we will look at some of the more generic codes that might typically be found.

In the Control L standard, the first byte sent usually identifies the device being spoken to. The first nibble is often the type of device (eg. 1 for VTR, or 2 for camera), and the second nibble is a unique identifier for that device (ie. the device number). Often this second digit is shown as an ‘8’, but it does not need to be. These two digits correspond to the first and second hex digits that need to be sent to the ELM624.

The second command byte (the third and fourth hex digits) is the actual command for the device being controlled. Some typical control codes are:

30	Stop	36	Rewind
32	Pause	38	Fast Forward
34	Play	3A	Record
2C	Eject	8C	Counter Reset

It is not the purpose of this data sheet to show all possible command codes, as they vary by device and manufacturer. The above ones seem to be very common however, and should get you started.

As our first example, assume that the camcorder is on, and selected for play (VTR) mode. The first thing we might want to do is be sure that it is ready to accept commands. At the prompt, ask the ELM624 to check the sync:

```
>AT CS
SYNC OK
```

If all is well, you should get the OK message as shown above. If we wish to play the tape, we might now enter the four digit command:

```
>1034
```

This command can better be understood if we separate the first few characters, and read them as follows:

1	0	3 4
VTR	#0	Play

The command is ‘VTR #0 Play’ (as the code for play is 34). The above talks to device #0, but much of the literature often shows commands being sent to #8. No problem with the ELM624 - for this example, you would simply use 1834 instead of 1034.

Similarly, 2130 could be sent to the ELM624, to mean ‘Camera #1 Stop’. Again, note that there does not seem to be a completely standard set of commands, so you will have to check with your manufacturer to be sure of which ones are supported by your device.

Talking to Your LANC Device (cont'd)

Once the controlled device has received a valid command, it will respond with a sequence of 4 status bytes which the ELM624 resends as a series of 8 hexadecimal digits. Although the addition of space characters could make this data more readable, there is not sufficient time to do this with 60Hz systems, without losing data. The initial byte (the first two digits) will usually be a status byte, having typical values as follows:

01	No Tape	02	Stopped
03	Fast Forwarding	04	Recording
83	Rewinding	06	Playing
72	Stopped - tape beginning		

The next nibble (third digit) is normally used to identify the meaning of the final two bytes (the fifth to eighth hex digits). For example, if the third digit is a 1, it generally means that the final two bytes are status bytes, while if the third digit is a 3 or 4, the last bytes are time codes. A 3 might mean that the bytes are seconds and minutes, while a 4 may mean they are hours and days.

The fourth hex digit (least significant nibble of the second byte) generally gives various system status information such as (in this case):

b3- Counter memory on
b2- Low battery
b1- Record tab status
b0- Invalid command received

Often it is easier to follow an actual example to see how the codes look. Assume that the command 1034 (VTR#0 Play) has just been sent to the ELM624. It will format the data, transmit it the number of times that is specified by the ATRn command, and begin reading and sending status bytes back to the controlling computer. The first few lines of the (lengthy) response might look like:

0 2 1 2 2 4 5 0
0 2 3 2 4 9 1 1
0 2 4 2 0 1 0 0

Many, many more bytes would actually be sent by the ELM624, as each successive line differs from the previous, and the IC always transmits a line when it differs from the previous one. Sending a single space character to the ELM624 will stop the endless stream of data and place the integrated circuit into the ready state, able to receive more commands. Almost any character will serve to stop the data stream, but the space character is the most convenient, and the most likely to be ignored by other devices connected to the system. When the prompt appears, you may want to stop the tape, if it is still running. Simply send:

>1030

Again, you'll need to send an interrupting character to stop the stream of status bytes. If you are using an older computer system, there may seem to be a delay between the typing of the space character and the stopping of the 'endless' display of data on the screen. This is not due to lack of response by the ELM624 (it always responds within 20 milliseconds), but is most probably that your display is not fast enough to keep up with the data as it is received, so be patient.

Once interrupted, the ELM624 always completes sending the current line before returning to the ready state, and sending a prompt character ('>'). This occurs very quickly in human terms, but as noted above can take as long as 20 msec, which is very long for computers. If you are operating your LANC device under program/computer control, always wait until the prompt character has been received before sending the next command.

Returning to the status bytes that were received above, let us analyze the response. The first line starts with the byte '02', which shows that the device is currently stopped. The next nibble '1' says that the following nibbles will provide various status information (the exact values depend on the controlled device).

The second line shows that the device is still stopped (02), that the following information will be ss:mm (3), and that recording is still not allowed due to the tab (the 2 in the fourth position). The time code bytes show 11 minutes and 49 seconds in this case.

The third line is similar, but states that hh:dd (4) follow. Combining the two time response lines (the '3' and '4' lines), one can deduce that the counter is currently showing 00 days, 01 hours, 11 minutes and 49 seconds. Some counters are simply a linear reading and do not reflect the actual time duration, so make sure of the interpretation when trying this on your device.

If you look further down the stream of data that you initially received when you said 'play' (1034), you will see that the second hex digit changes after a while. Typically, the received LANC bytes may have

Talking to Your LANC Device (cont'd)

looked something like these:

0 6 1 2 2 4 5 8
0 6 3 2 4 9 1 1
0 6 4 2 0 1 0 0

Comparing this to the other response, you can see that the device is now in play mode (06), with the time/counter still reading 00:01:11:49, as it has just begun to play the tape. The 8 in the last position of the first row indicates that (for this camera), there is a 'memory mark' here to show a significant spot on the tape. This is how you would generally control a device through LANC using computer control - issue a command then monitor the status bytes until the desired change has occurred. This way you can be sure that the command was understood. Of course, if you are manually controlling it from the keyboard, you will know when the device has responded.

It is beyond the scope of this document to detail the Control L standard in any more detail, but hopefully this has been enough to get you started, and has generated some ideas.

Talking to Many ELM624s - Using the Enable

The ELM624 provides an Enable input that can be used to control whether the ELM624 pays attention to the RS232 bus or not. This means that several devices can share one data 'channel', while only those chosen to do so will respond. In this way, controlling many devices is not much different than controlling one, except for the work involved in selecting the appropriate ELM624, and deselecting the others.

RS232 data received at the Rx pin of the ELM624 while the Enable input is high will be used by the IC, and any data that appears at that pin while the Enable is low will be ignored. You can use this in order to selectively send information to several devices that share a common RS232 bus. The messages sent to each device do not have to be sent all at once as a complete string - you can send one byte to device 1, a few bytes to device 2, a byte to device 3, etc., as long as you return back to each device before their internal 20 second timer causes an abort on an incomplete string receive. This is usually quite easy to do.

To ensure that the enable works properly when sending commands to the ELM624, be sure to have it at an active level before the RS232 byte's start bit begins, and maintain it at that active level until at least the end of the stop bit. There is no such restriction for controlling the ELM624's Tx output. The IC will turn the output line off within one bit time of the Enable going inactive, so if you do not want to receive any more from that device, simply bring the Enable line low.

The (continuous) flow of status messages can be monitored periodically while doing other tasks through the Enable control. For example, one could issue a command to rewind the tape (10 36), then periodically check for a 'tape end stopped' status byte (72) while doing other functions with other devices. There is no restriction on how often one uses the Enable input – a typical system with several devices will naturally try to poll each device as often as it can to maintain 'real-time' control. One should keep in mind that commands typically take about 100 msec to complete, however, so attempting to poll a device more often than that may not be of much benefit.

Providing an Enable input simplifies the interface circuitry needed if controlling multiple devices, often reducing your design to basically a channel selector, and one ELM624 per channel. The Example Applications section provides a sample circuit showing one possible way of controlling three devices through one RS232 interface.

Power Control

Beginning with v3.0 of the ELM624, an output pulse can be generated on the LANC output, simply by issuing an AT SP command. The 'pulse' is actually an active (low) signal that lasts for 150 milliseconds.

Many LANC devices respond to this pulse, some by toggling the power on and off each time it is issued, but most by waking up when they receive it. In order to use this command, you would typically start the device you wish to control, and either let it go to sleep by itself, or issue a power off command (105E, 102A, or something similar). When the device is in this 'sleep state', it can often be 'woken' simply by issuing a Send Pulse command (AT SP). Note that not all devices can be turned off by software commands, and not all will wake up to this signal, but you may be fortunate and yours may.

Monitoring from Powerup

The ELM624 IC can be put into a special continuous monitoring mode at powerup, or during a software reset (AT Z) without any input being required from the user. This allows the IC to be used in translator type projects such as 'LANC logic probes', or simply inputs to computer based controllers.

If the RS232 Rx pin is found to be at an active (high) level throughout the entire powerup sequence, the IC will print the ID string (ELM624 v3.0), will set the output mode to 'Raw Data', and will then immediately go perform a monitor all command. This can be very useful if all you need to do is to monitor the signals being sent between other LANC devices. Note that in the raw data mode, the ELM624 performs no translation of the received LANC data. It simply leaves each byte as the raw value which was received, and resends them to the connected PC along with a single terminating carriage return character. No linefeed is sent after the carriage return, regardless of the AT L0/L1 setting.

This monitoring mode will remain in effect until the Rx input (pin 5) returns to a low level, no matter whether there are sync signals or not, or even if there are only sporadic sync signals. Once the Rx does go low, the ELM624 will perform a 'soft reset', restarting itself into the normal mode of operation. A software and will be ready for your commands. The Example Applications section shows an example of how you might wire a 'LANC logic probe'.

Error Messages

There are actually very few errors that the ELM624 can report. There can be errors in the user input, or there can be problems with the LANC signals:

NO SYNC

There is no recognizable synchronizing signal at the ELM624's LANC input. The IC has searched for some time, attempting to detect a signal that it could synchronize to, and failed. Check your connections, and the power to the LANC device, then try again.

SYNC ERROR

A problem has occured while receiving one of the eight LANC bytes. A synchronizing signal had been detected, but either that was a false detection, or something has now happened to the signal. As with the NO SYNC condition, check your connections, and the power to the LANC device, then try again.

?

This is the standard response for a misuderstood command received on the RS232 bus. Usually it is due to a typing mistake, but occasionally it can be from problems with the connection to the computer.

Design Considerations

The ELM624 is an experimenter's integrated circuit, that may have a few peculiarities that must be considered in any design.

Foremost, one should note that the LANC pin is connected directly to the LANC or Control L bus. This presents two areas of concern. One, if the controlled device is powered before the ELM624, there may be a backfeed through the (inherent) LANC pin protection diode into the ELM624 circuitry, and if loading is not very significant, there could be enough voltage developed to partially power the ELM624 (even if somewhat erratically). To avoid this, it is preferrable to power the ELM624 circuitry before the controlled device. This is likely required anyway, as many LANC devices use the presence of a voltage to determine that a controller has been connected to its data port. Also, adding a load such as a power monitor led (at the ELM624 circuit) can help to reduce the level of backfed voltage, if it is a problem.

The other concern is related to the LANC pin itself. The ELM624 is a CMOS integrated circuit, and so is succeptible to a 'latch-up' phenomena that can occur if large currents are allowed to flow from external sources into the pin in an uncontrolled way. To reduce the possibility of latchup, try to reduce exposure by keeping connecting cables as short as possible, and consider placing a small value resistor (100Ω to 220Ω) in series with this pin. The Example Applications section shows such a resistor connected to the LANC pin.

On the PC side of things, the main consideration with the RS232 interface is the fact that the receive signal is inverted from what might be expected. This simplifies the RS232 interface circuitry, but may cause some confusion. Precautions should also be taken in the circuit design, to allow for the possibility that this input may be left floating due to a disconnected serial cable. Typically, this only requires a large-valued resistor between the RS232 TxD pin and Vss, as shown in the Example Applications section. This is the same interface that is recommended for our popular OBDII interface ICs, so the information provided on our web site under 'Auto Info' would likely be useful, if you are having trouble.

As a final reminder, one should recognize that the ELM624's RS232 interface does not employ any hand-shaking signals, so the controlling computer must be careful to wait for a complete response before issuing the next command, or else characters could be lost. This isn't too difficult to do if one monitors the data stream for the prompt ('>') character before issuing the next command (and making sure you also allow time for the stop bit to finish).

Example Applications

Our first example, Figure 3, shows how the ELM624 might typically be used in a circuit. A very basic RS232 interface has been provided, and the LANC connection is direct, but the circuit is quite functional as shown.

There are two types of Control L/LANC connectors that are commonly used in the industry. One is the mini-DIN type (similar to the S-Video connectors), and the other is the 2.5mm stereo plug type (like a headphone plug, but slightly smaller). Both are shown in Figure 3, for reference, but you will have to choose the one that is appropriate for your application.

Connected between the LANC interface connector, and the LANC input (pin 7) of the ELM624 is a 100Ω resistor, used to help prevent any latch-up, and to some extent for protection of the ELM624 IC from direct shorts to the supply. This is recommended as most Control L connectors provide DC on one of the pins, which could be a source of damaging energy. If a 2.5mm stereo plug is used, the tip is generally the LANC signal, the small ring is often a DC supply, and the sleeve is the circuit common or V_{SS}.

In this circuit, a pullup resistor (4.7KΩ) has also been added from the LANC pin to V_{DD}, in order to improve the rise time of the LANC signal. Depending on cable capacitance, etc. you may find that it is not required in your application, and the ELM624's internal resistance is sufficient. A good rule of thumb for any CMOS circuit is to try to maintain the time constant at 1 µsec or less, if possible.

A very basic RS232 interface is shown connected to pins 5 and 6 of the ELM624. This circuit 'steals' power from the host computer in order to provide a full swing of the RS232 voltages without the need for a negative supply. The RS232 pin connections shown are for a 9 pin connector. If you are using a 25 pin, the connections would be 3(RxD), 7(SG) and 2(TxD).

RS232 data from the computer is directly connected to pin 5 of the IC through only a 47KΩ current limiting resistor. This resistor allows for voltage

Example Applications (continued)

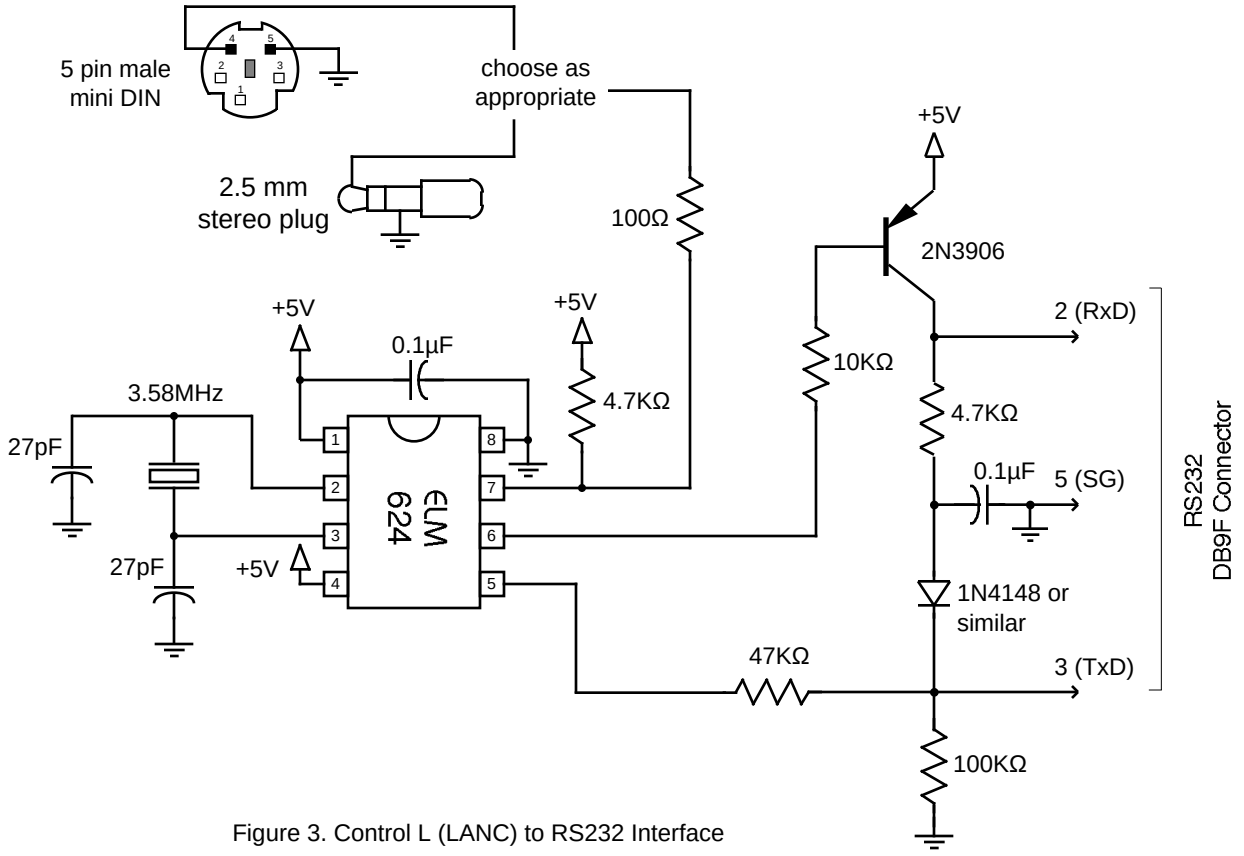


Figure 3. Control L (LANC) to RS232 Interface

swings in excess of the supply levels while preventing damage to the ELM624. A single 100KΩ resistor is also shown in this circuit so that pin 5 is not left floating if the computer is disconnected.

Transmission of RS232 data is via the single PNP transistor connected to pin 6. This transistor allows the output voltage to swing between +5V and the negative voltage stored on the 0.1µF capacitor (which is charged by the computer's TxD line). Although it is a simple connection, it is quite effective for this type of application.

Finally, the crystal shown connected between pins 2 and 3 is a common TV type that can be easily and inexpensively obtained. The 27pF crystal loading capacitors shown are only typical, so you may have to select other values depending on what is specified for the crystal you obtain. This crystal frequency works for both 50Hz and 60Hz systems, and should not be changed.

The circuit of Figure 4 shows the IC connected to make use of the new 'powerup monitor mode'. In this case, the RS232 Rx line has been hard-wired to a high logic level (V_{DD}), so the ELM624 immediately executes a monitor all command on power up. The basic RS232 transmit circuit is again used to send the data to a connected PC, but has had a few of the components removed since there is no need to receive data. If connecting the ELM624's pin 6 directly to another logic circuit, you can eliminate the RS232 interface entirely. Your circuit need only be compatible with the CMOS logic levels, and should expect that pin 6 will be at a high level when idle.

Our final example, Figure 5, shows how one might control multiple devices using one RS232 interface. We show three ELM624s being controlled, but there's no reason why you can't control more.

In this circuit, he ELM624s have been connected with their RS232 Rx pins in parallel, and the Tx lines OR-tied together through the 10KΩ resistors. A single ELM621 has been used as the device to selectively

Example Applications (continued)

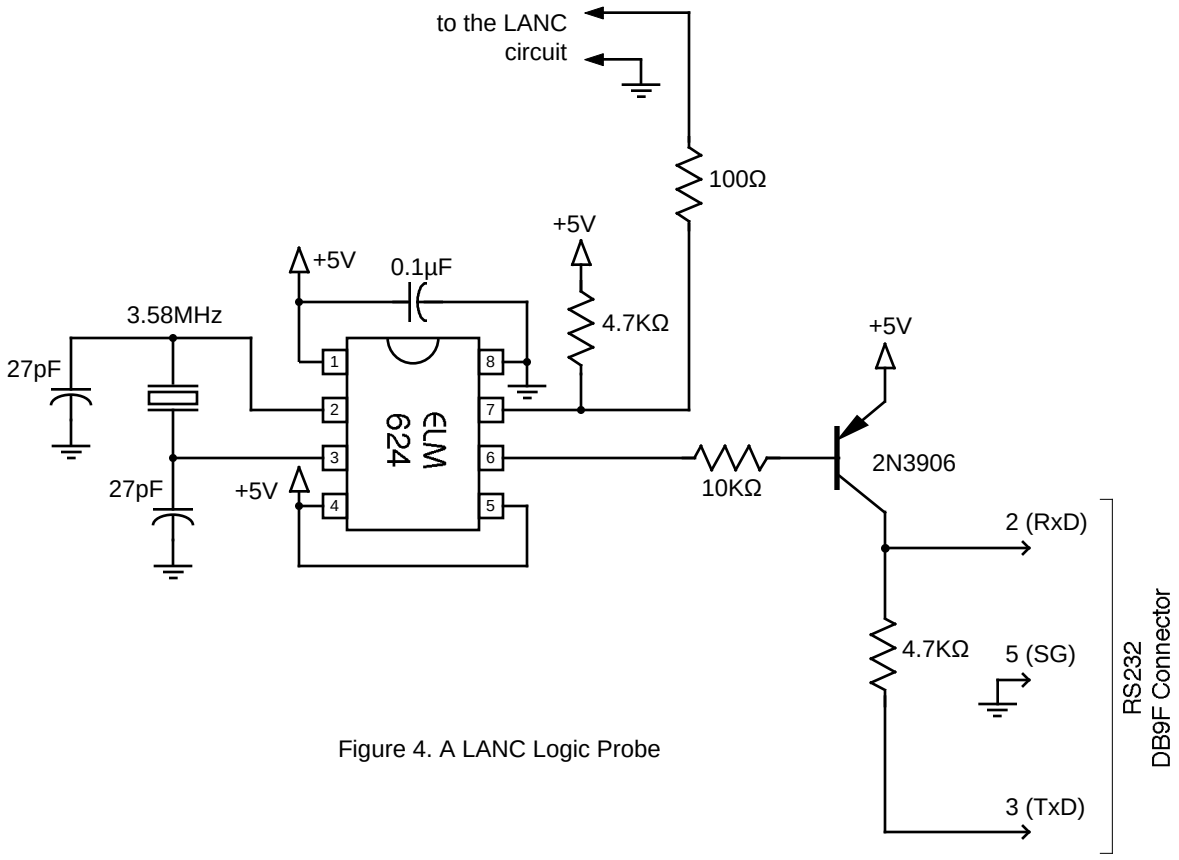


Figure 4. A LANC Logic Probe

enable each ELM624. The ELM621's ability to respond to ASCII commands to control its port pins works well in this application.

After power on (or an AT Z reset), the ELM621 sets all three of its port pins to be inputs. To avoid allowing this condition to enable all the ELM624s at the same time, 4.7KΩ resistors have been connected from these pins to V_{SS}, pulling the voltages on the Enable inputs to something very close to 0V, so all three ELM624s will be disabled. Note that the ELM621 Tx line is not used, so one will not see the 'ELM621...' message that it sends at powerup.

To begin talking to the ELM624s, one must first issue the ELM621 command AT CA OA (Clear All, Outputs All). The ELM621 will act as commanded, clearing all three pins and configuring them all as outputs. After that, the ELM624s can be selectively enabled by setting individual ELM621 outputs high (sending AT S1, AT S2, or AT S3 as appropriate). Similarly, the ELM624s can be disabled with an AT CA command. For more information on these ELM621 commands, see its latest data sheet.

Once selected, communications with individual ELM624s will occur as normal (as if no ELM621 were present). The ELM621 may not understand some of your ELM624 commands, but since its transmit line is not used, you will not 'hear' the complaints. To control several devices, one simply sends the commands to each one sequentially, while enabling and disabling the ELM624s as appropriate.

This opens up even more doors for the creative experimenter, and once again, is only limited by your imagination…

Example Applications (continued)

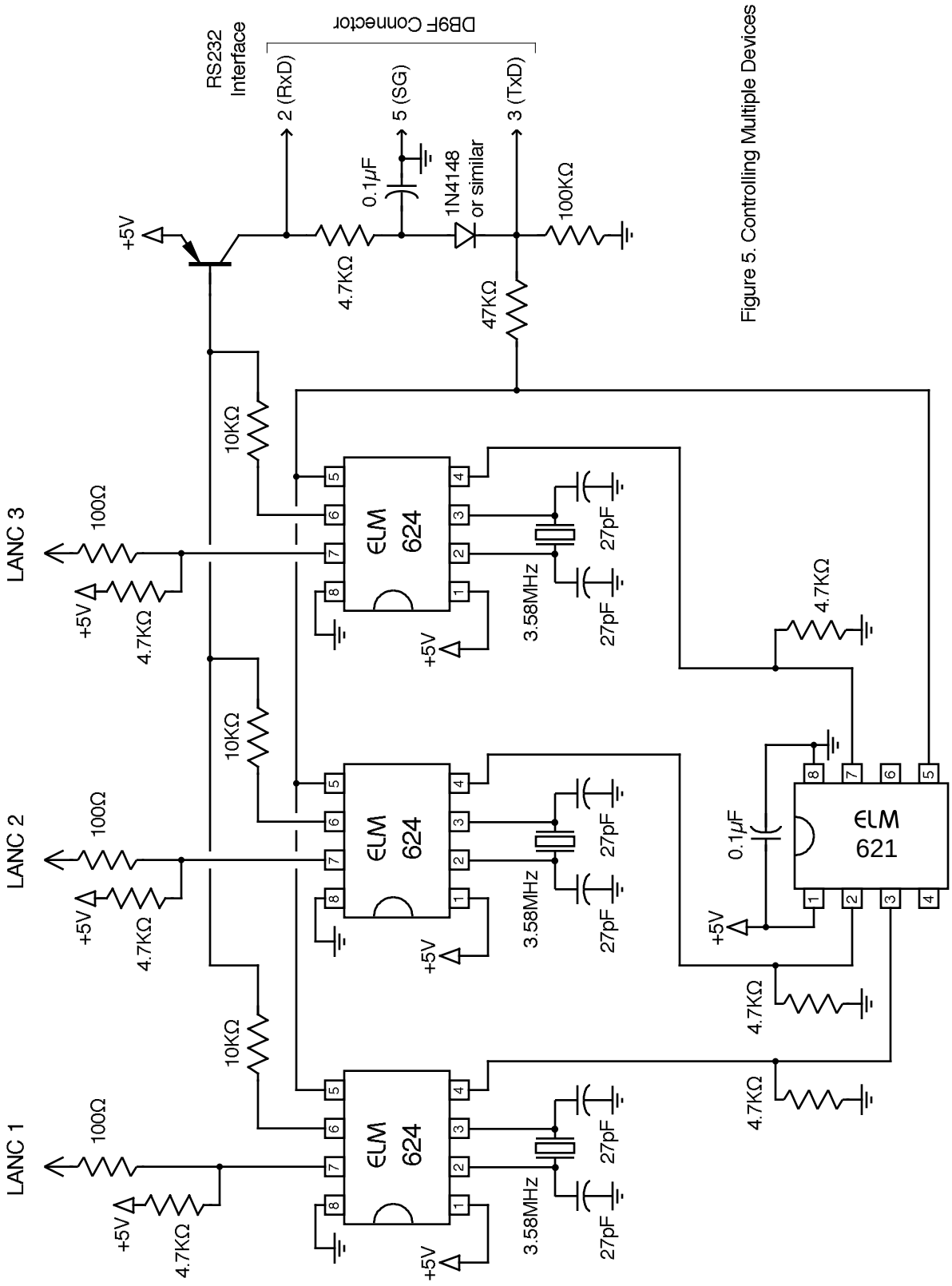


Figure 5. Controlling Multiple Devices