

Networks for Monitoring and Control Using an RS-485 interface

Jan Axelson

August 21, 1995

1 Introduction

This article was taken from the MICROCOMPUTER JOURNAL, the JULY/AUGUST issue, published in 1995 and starting on page 27. It is a good article covering the RS-232, RS-423, RS-423 serial communication standards in general and the RS-485 in detail. It has been converted to Postscript to make it easier to transmit electronically.

2 Background

One computer can certainly do a lot on its own, and two computers linked together can do even more. Even so, some jobs call for a network of three or more computers linked together, issuing commands, sending replies and, in general, exchanging information with each other.

A network is a solution when you need both of the following: computer intelligence to make decisions, gather data or control objects or systems in a variety of locations and the ability for each location to communicate with at least one other location. The computers in a network can be any type. For example, they may all be personal computers, or all microcontrollers or just about any combination of computing devices you can think of. For example, an environmental-control network might have a series of microcontrollers or single-board computers to monitor and control temperature, humidity and other variables, with one personal computer to keep track of all of it and permit users to monitor the system and adjust or modify its operation.

In this article, I begin a two-part series to show you how to build and program a simple network of personal computers, microcontrollers or a combination of the two. The network uses an RS-485 serial link that permits up to 32 devices (or nodes) to connect over a single pair of wires. (By adding repeaters and control logic, you can have more than 32 devices.)

You can use the example network as a building block for a specific application, whether it's gathering data from remote sensors, controlling remote systems or whatever it is you need to accomplish. This type of link is also useful if you have just two devices but need to transmit over distances of up to 4,000 feet.

3 Serial Standards

To begin, let's take a look at RS-485 and how it compares with the more common RS-232 scheme. We'll look briefly at some other serial interface standards as well.

Serial standards describe interfaces for transferring information in serial format, or one bit at time in single file. In contrast, a parallel interface transfers multiple bits simultaneously. All other things being equal, a parallel interface is faster because it carries more information at the same time. However, because of the expense of the extra conductors required to carry all of the bits, parallel links usually aren't practical over long distances.

Table 1 compares four serial interface standards. All are developed and published by the Electronics Industries Association (EIA), with more recent versions developed jointly with the Telecommunications Industries Association (TIA). These are trade associations sponsored by businesses in the electronics and telecommunications industries. To reflect its origin, the official name for what's popularly known as RS-485 is actually EIA-485.

Copies of the standards are available from Global Engineering for 40to55 each. However, you can often find the information you need, including practical tips the standards often leave out, in the application notes from the makers of RS-485 interface chips. Two such sources are National Semiconductor's Interface: Line Drivers and Receivers' and Texas Instruments' Data Transmission Circuits data books.

RS-232 is used for communicating between just two devices, over distances of 50 feet or less, though you can stretch this at lower bit rates. Just

about every IBM-compatible personal computer has an RS-232 port. (See the November/December 1994 and January/February 1995 issues of Micro-Computer Journal for more on RS-232 links.)

RS-485 has several advantages over RS-232. With it, you can have up to 32 driver/receiver units on a link. A link can be as long as 4,000 feet, or 3/4 of a mile. Over shorter distances, transmission speeds can be as great as 10M bits/second. And RS-485 drivers and receivers are inexpensive and require just a 5-volt dc supply in contrast to RS-232's minimum requirement of +5-volt outputs. RS-423 is similar to RS-232, but it permits up to 10 receivers to connect to one driver, or generator. Like RS485, RS-422 can transmit over long distances, but like RS-423, it's intended for use with just one driver and up to 10 receivers.

In this article, I focus on the RS485 interface, since it's the only one of those mentioned that permits multiple drivers and receivers in a link.

4 Inside RS-485

An RS-485 link employs balanced lines, which means that each signal has two conductors, and the signal on the second conductor equals the negative of that on the first. An RS-485 receiver responds to the voltage difference between the two conductors. Another term for this type of measurement is differential measurement.

Specification	RS-232	RS-423	RS-422	RS-485
Transmission Mode	Unbalanced	Unbalanced	Balanced	Balanced
Maximum Cable Length*	50 Feet	100 Feet	4,000 Feet	4,000 Feet
Maximum Transmission Speed	20K Bits/Second**	100K Bits/Second	10M Bits/Second	10M Bits/Second
Minimum Driver Output	±5 Volts	+3.6 Volts	+2 Volts	±1.5 Volts
Maximum Driver Output	+15 Volts	+6 Volts	+5 Volts	+6 Volts
Receiver Sensitivity	±3 Volts	+0.2 Volts	+0.2 Volts	+0.2 Volt
Maximum Drivers	1	1	1	32
Maximum Receivers	1	10	10	32
Driver Load (Ohms)	3,000 to 7,000 Ohms	450 Ohms Minimum	100 Ohms Minimum	60 Ohms Minimum
*100K Bits/Second; **Nonstandard* 115K-Bit Chips Available				

Table 1: Comparison of four serial interface standards

In contrast, RS-232 uses unbalanced lines in which each signal has just one conductor and the receiver responds to the voltage difference between this conductor and a common ground conductor used by all signals. Another term for this type of measurement is single-ended.

Balanced lines have a couple of advantages over unbalanced ones. One is that noise (voltage spikes, oscillations or interference in general) tends to cancel out.

In an unbalanced line, a signal's return current in the ground conductor can generate noise at the receiver. In a balanced line, the differential signals generate two equal but opposite return currents. Since the currents add up to zero, for all practical purposes, there's no return current to cause problems.

Another advantage to balanced lines is that they're immune (within limits) to differences in ground potential between nodes. In a network that has long lines, the ground potential may vary as much as several volts from one node to another. But a differential measurement doesn't care about this difference, since it measures only the voltage between the two signal conductors.

Use of balanced lines is the main reason why RS-485 can transmit so much farther than RS-232.

Many RS-485 links use inexpensive twisted-pair cable, which consists of two insulated conductors that spiral around each other every inch or so. Radio Shack sells AWG 22 stranded twisted-pair cable as "alarm conductor." The simple act of twisting together the conductors tends to cancel any noise the lines pick up due to electromagnetic interference.

Although RS-485 interfaces normally use a 5-volt power supply, the logic levels at the drivers and receivers aren't standard 5-volt TTL or CMOS voltages. The two signals are designated A and B. For a valid output, the voltage difference between A and B can be as small as 1.5 volts, although a 5-volt difference is typical. If terminal A is at least 1.5 volts more-positive than terminal B, the output is a logic 0. And if terminal A is at least 1.5 volts more-negative than terminal B, the output is a logic 1. If the difference is less than 1.5 volts, the output is undefined.

At the RS485 receiver, the difference between the A and B inputs needs to be just 0.2 volt for a valid logic level. If A is at least 0.2 volt more-positive than B, the receiver sees a logic "0", and if B is at least 0.2 volt more-positive than A, the receiver sees a logic "1". If the difference between A and B is less than 0.2 volt, the logic level is undefined.

The difference between the voltage requirements at the driver and receiver means that the signal can attenuate as much as 1.3 volts along the net-

work conductors for the receiver to still recognize it properly. If the driver generates a 5-volt difference, the margin is even greater, at 4.8 volts.

5 RS-485 Chips

An easy way to create an RS-485 interface is to use one or more of the available RS-485 chips. One series, the 7517x, is available from

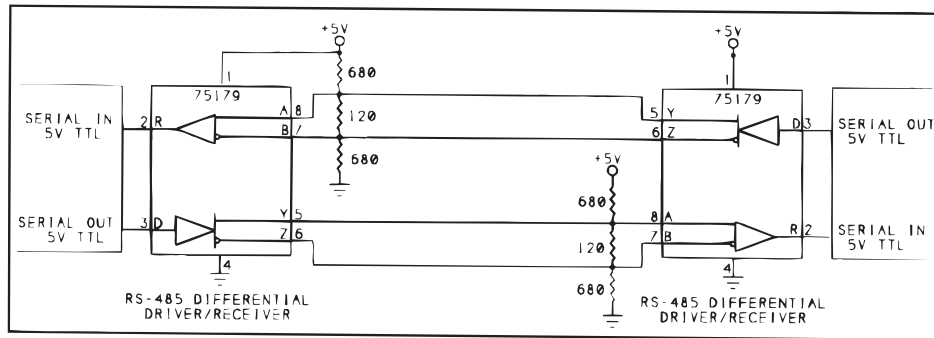


Figure 1: A full-duplex RS-485 link requires a pair of conductors in each direction.

Texas Instruments, National Semiconductor and Motorola. The 9517x series is similar. Maxim Semiconductor has come out with its MAX48x series. The chips are available from many parts sources. Parts catalogs may list them under Linear or Interface chips.

The 75179 contains one driver and one receiver in a single package. The driver translates 5-volt TTL signals to RS-485, and the receiver translates RS-485 signals back to 5-volt TTL. With a pair of driver/receiver chips, you can create a full-duplex link that can carry data in both directions simultaneously, as Fig. 1 illustrates.

Many RS-485 links are half-duplex, wherein each node can both send and receive data but can't do both simultaneously. Although RS-232 links are typically full-duplex, there are some good reasons why you might use half duplex operation for RS-485.

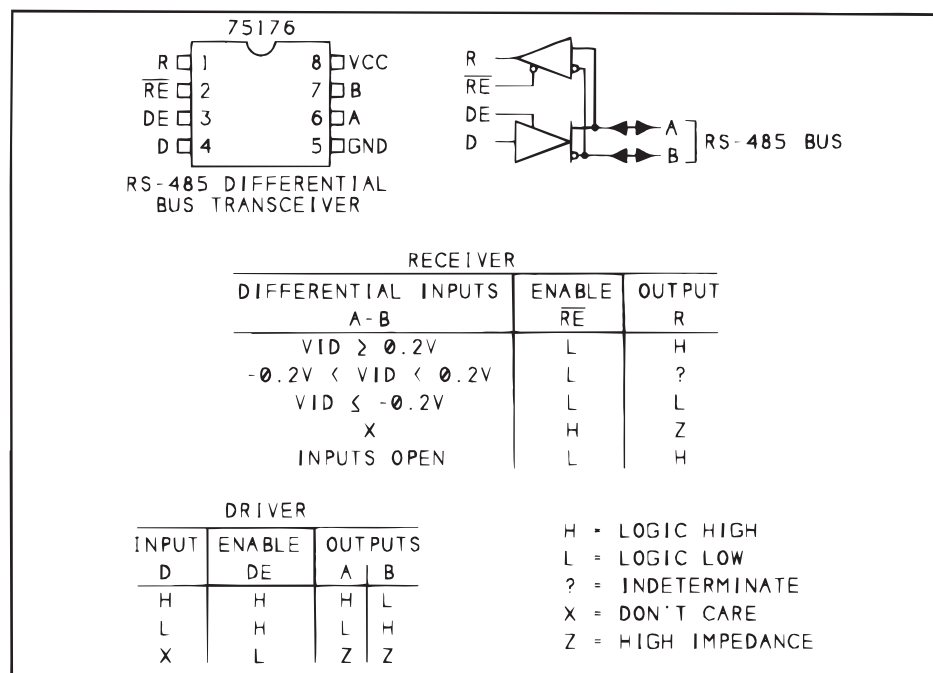


Figure 2: The 75176 chip contains an RS-485 xmitter, receiver, direction control and two-pin RS-485 interface.

In a link between just two devices, full-duplex is convenient because each device can transmit at any time without concern for which device its turn it is. But with many nodes in the link, there's no guarantee that a signal path will be free, and figuring out when it's okay to use each of two signal paths just adds more complications. So if you have three or more nodes, it's simpler to have just one signal path and permit only one node at a time to transmit.

Also, a half-duplex link is less expensive to implement because you need just two signal conductors to carry data in both directions, rather than a pair of conductors for each direction. Over short distances, a couple of extra conductors is no big deal, but if you're going 100 or 1,000 feet, costs begin to add up.

For a half-duplex link, you can use a transceiver chip like the 75176, the pinout for which is given in Fig. 2. This chip includes one RS-485 driver, one receiver and an enable input for each. Since the driver is enabled by a logic-high signal and the receiver is enabled by a logic-low signal, you can use the same signal to control both, and only one will be on at a time.

Alternatives to the 75176 are the MAX481 and MAX483. The 75176 can operate as fast as 10M bits/second, while the MAX481 is guaranteed only to 2.5 M bits/second, and the MAX483 to 0.25 M bits/second. National also makes a DS3695 transceiver chip that's similar to the 75176. All of these chips have the same pinout.

Many RS-485 links operate at 9,600 bits per second at most. For these, you can use any of the above chips. Slower devices reduce the likelihood of transmission errors caused by signal reflections on the cable. If you don't need high-speed performance, slower chips will do the job and may help to avoid trouble.

All of the transceivers listed protect themselves with current limiting and thermal shutdown. Without this protection, if two or more drivers in the network turn on at once, component damaging currents could result.

Current limiting restricts the output of the RS-485 drivers to 250 mA. If an output continues to source or sink high current, the chip will heat up and the thermal shutdown circuits will eventually switch the output to a high impedance state. Of course, this makes the output unusable until it cools down, but at least the components will survive.

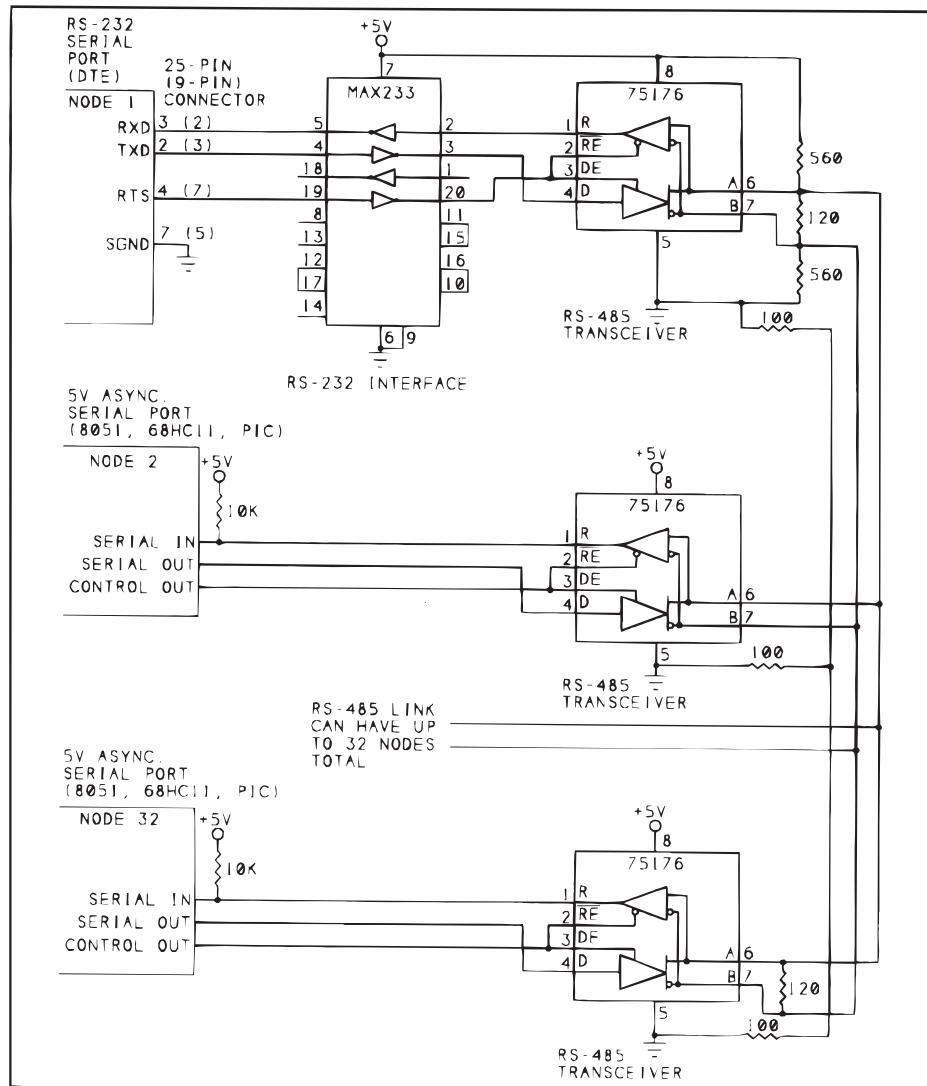


Figure 3: A RS-485 link can connect up to 32 nodes along the same pair of conductors.

6 Network Circuits

Shown in Fig. 3 are circuits for an RS-485 network that can connect up to 32 personal computers and microcontrollers in any combination. A third conductor provides a ground reference. Each node can be a personal computer, microcontroller or any device that has a serial port.

On a personal computer, you have two options for creating an RS-485 port. You can buy an expansion card with an RS-485 port on it, or you can add an external device that converts one of the RS-232 ports to RS-485. If you decide to convert from RS-232, there are two options in this case as well. One is to buy a converter that plugs into the RS-232 connector. The other is to make your own. The Sources box in this article lists suppliers for RS-485 cards and converters.

To make your own converter, you must first translate between the RS232 voltages and 5-volt TTL levels. You can use any of the many converter chips now available for this. Figure 3 uses a MAX233, which also re-inverts the signals inverted by the personal computer's RS-232 interface.

The port interface uses three RS-232 lines. TXD transmits data, RXD receives data and RTS controls the direction of data in the '176.

The TTL side of the 75LS 176 connects to the MAX233. When RTS is low, pins 2 and 3 of the '176 are high and TXD can transmit data to the RS485 link. When RTS is high, pins 2 and 3 of the '176 are low and RXD can receive data from the RS-485 link.

For a microcontroller interface, the circuits are even simpler. Most microcontrollers—including the 8051, 68HC11 and some PICs—have built-in asynchronous serial ports, with programming instructions for reading from and writing to the port. The ports use 5-volt logic, not RS-232 voltages. For an RS-485 interface, this saves you the trouble of having to convert back to 5-volt levels. Some microcontroller boards have an RS485 interface built into them.

In Fig. 3, the transmit, receive and control pins connect directly to the '176. Microcontrollers don't normally have a dedicated RTS pin, but you can use any port output and control it directly in your program.

A pull-up resistor at the microcontroller's serial input holds the input high when the node is transmitting and the receiver's output is off, or in high-impedance state. You don't need the pull-up resistor if your microcon

troller's port has an internal pull-up (as on the 805 1) or if the receiver drives an input to a MAX233 or other RS-232 interface chip.

Proper control of RTS can be one of the trickiest parts of getting a half duplex network up and running. In short, when a driver is transmitting, RTS must remain low until the driver is finished sending, and RTS must then switch high before the answering node begins to send a response. You can control RTS in software or add circuits to automatically switch the control line when a node is transmitting.

7 Wiring tips

Unlike the case with the RS-232 standard, RS-485 doesn't specify an interface connector, signal functions or pin assignments. These are left for you to designate. Keep the two signal conductors (A and B) adjacent to each other on any connector you use. In twisted-pair cable, the A and B conductors for a signal should, of course, be in the same pair.

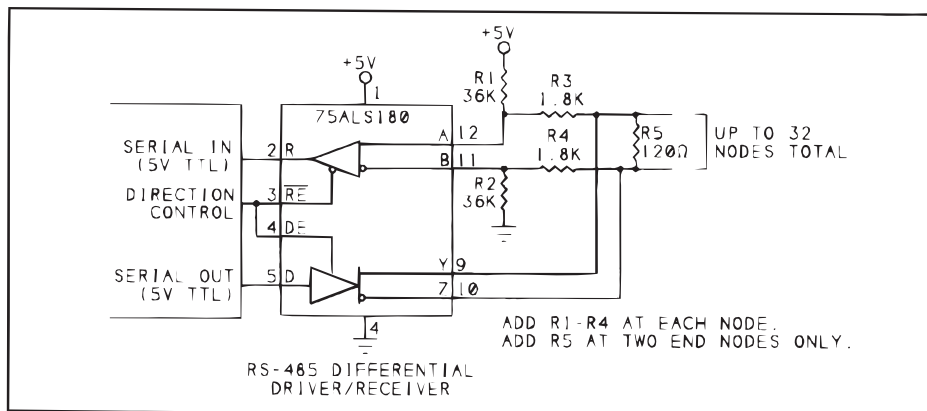


Figure 4: The 75ALS180 driver/receiver with four-conductors.

The 75ALS180 driver/receiver has a four-conductor RS-485 interface that enables you to add short-circuit protection to the receiver with resistors R3 and R4. Be careful not to transpose the network conductors. Connect all of the A pins to one conductor and all of the B pins to the other conductor. Connect each node as close as possible to the network conductors. If you use

a stub—a separate pair of conductors that connects a node to the network—keep its conductors as short as possible.

On all but the shortest links, you should treat the RS-485 cable as a transmission line. When the conductors are short and frequencies are low, you can consider a conductor to be a perfect R0S-ohm connection. But a transmission line behaves differently from an ordinary short conductor. A transmission line has a characteristic impedance, which is a resistive value that varies with the conductors' diameter, spacing and type of insulation, but it doesn't vary with conductor length.

The theory and math behind transmission-line behavior can be a little forbidding, but the main thing to remember is that a transmission line should always terminate in a load equal to its characteristic impedance. You do this by connecting a resistor equal to the characteristic impedance across the inputs at the receiver at each end of the link. No matter how many nodes are in the network, you should have just two termination resistors, one at the receiver at each end of the network.

With proper termination, the entire signal at the receiver is applied across the termination resistor. A slight impedance mismatch won't hurt anything. However, with a gross mismatch, a large part of the signal will reflect back to the driver and then back to the receiver, which may then interpret the reflection as valid signal on the line.

Line reflections are more of a problem when the edge speed, or rise and fall times, of the driver's output are very short. Edge speed isn't related to bit rate, or number of bits transmitted per second. Instead, the critical factor is how fast the voltage changes when the logic levels switch.

A 300-bps link that uses a highspeed driver will be more likely to have reflections than a 19,200-bps link that uses a slower driver. (But a link that uses a slower bit rate will be more likely to ignore reflections, because the reflections occur after a voltage transition and the serial port reads the logic levels in the middle of the bit. With a slower bit rate, the bits are wider.)

The slew rate given in the driver's data sheet specifies the rate of voltage change at the output. The lower the slew rate, the slower the edge speed.

For AWG 24 stranded twisted-pair cable, the characteristic impedance is around 120 ohms. Therefore, a proper termination would be a 120-ohm resistor across terminals A and B at each end of the link. RS-485 drivers must be able to drive a total termination resistance of 60 ohms. On a

halfduplex line, the two 120-ohm resistors in parallel are equivalent to 60 ohms.. Consequently, it's best not to use anything smaller than 120 ohms to terminate a half-duplex line.

All conductors also have a set resistance per foot, which becomes significant only if the conductors are very long. AWG 24 stranded wire has a resistance of about 25 ohms per 1,000 feet, which is separate from the conductor's characteristic impedance.

Another concern in an RS-485 network is ensuring that all receivers see a valid logic 0 when no drivers are active in the link. When this occurs, the signal level at a receiver's inputs is undefined. This could cause transmission errors as the receiver detects unintended signal levels or transitions.

Most RS-485 chips include a failsafe feature that's supposed to hold input A more-positive than input B when no signal is applied to the receiver. The problem is that fail-safe circuits don't work when you add termination resistors.

Fail-safe circuits typically consist of an internal 100,000-ohm pull-up resistance at input A and a matching pull-down resistance at input B. When you add the termination resistors, the resulting voltage divider brings the voltages at the two inputs very close, with less than the required 0.2-volt difference between them.

Fortunately, there's a solution. Figure 3 adds two additional resistors: a 560-ohm resistor from input A to +5 volts and a matching resistor from input B to ground. This arrangement holds terminal A about 0.25 volt more-positive than terminal B. In turn, this causes pin 1 to be high when no drivers in the link are active and still allows any driver to pull low terminal A.

One set of resistors at one end of the link biases the entire network. You don't need biasing resistors if your network always has an active driver.

Although RS-485 signals themselves don't use a ground reference, all nodes should have a common ground connection, unless the network conductors are optically isolated. There are two ways to achieve the ground connection. One is to connect the transceivers signal-ground pin to an earth ground at each node. The other way is to add a third conductor in the network cable. At each node, this conductor connects through a

100-ohm resistor to the transceiver's ground pin. The resistor protects the circuits by limiting currents that may develop in the ground conductor.

Alternate Interfaces If your network travels through an electrically noisy environment, optical isolation might be worth the trouble and expense it takes to implement it. With optical isolation, the network conductors connect optically, not electrically, to each node, thus eliminating the need for a common ground conductor. An easy way to isolate a network is with Maxim's MAX1480. This 28-pin chip actually contains its own transformer for power-supply isolation, plus optical isolators for the signal lines. Although the eight-pin transceiver chips like the 75176 are easy to use and inexpensive to purchase, Fig. 4 illustrates an alternate interface, which I've adapted from a circuit in Texas Instruments' data book. This circuit adds resistors that protect the network from damage if the network conductors should short circuit, or accidentally connect directly to each other. Like the 75176, the 75ALS180 contains a driver/receiver pair and an enable line for each. But the RS-485 side of the chip has four conductors instead of two. You can use the '176 in a full-duplex link, but the circuit illustrated in Fig. 4 is a half-duplex node, with the two-conductor interface created by tying together the driver and receiver pairs. In Fig. 4, only the two end nodes should have termination resistor R5, but all nodes have RI through R4. Resistors RI and R2 bias the line to a logic 0 if no driver is active, and resistors R3 and R4 protect the receiver and ensure that its inputs remain biased even if the line is shorted. If the network has fewer than 32 nodes, you can reduce the values of RI through R4 by multiplying each by half the total number of nodes in the network. For example, with just two nodes, the value of RI and R2 would be 2,200 ohms the value of and R3 and R4 would be 110 ohms. Since some signal voltage does drop across R3 and R4, reducing the values of these resistors means that the receiver's inputs will see a larger signal. The RS-485 standard specifies that you can have up to 32 unit loads, or driver/receiver pairs. But what can you do if you need more than 32 nodes? The answer is to use a repeater circuit, which regenerates the RS-485 signals and allows you to add up to 32 more nodes. You can also use a repeater to extend the length of a network to beyond 4,000 leer.

Figure 5 shows a network with a repeater circuit that contains one 75177 and one 75178 repeater chip. Like the transceivers, the repeaters each have a control input that determines the direction of signal flow through the chip. The '177's input is active-high, and the '178's is active-low. In a half-duplex link, you have to control the repeater's direction. For example, you might want the repeater to follow the direction of a master, or controlling, node in the network. To do so, you can use the same signal to control both

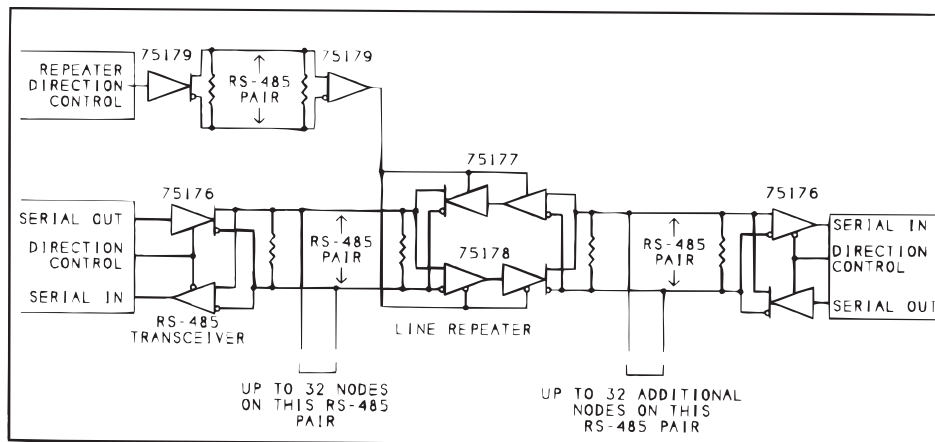


Figure 5: Using a line repeater.

the master node and the repeater. If the repeater is far from the master node, however, you may have to wire the direction-control signal as another RS-485 signal, as Fig. 5 illustrates.

8 Information & Part sources:

RS-485 Adapters B&B Electronics PO Box 1040 Ottawa, IL 61350 Tel.: 815-434-0846; fax: 815-434-7094

RS-485 Boards ComputerBoards 125 High St. Mansfield, MA 02048 Tel.: 508-261-1123; fax: 508-261-1094

RS-485 and Other EIA Standards Global Engineering Documents 15 Inverness Way E. Englewood, CO 80112-5704 Tel.: 800-624-3974 or 303-792-2181; fax: 303-790-0730

RS-485 Boards Keithley Metrabyte 440 Myles Standish Blvd. Taunton, MA 02780 Tel.: 508-880-3000; fax: 508-880-0179

RS-485 Chips Maxim Integrated Products 120 San Gabriel Dr. Sunnyvale, CA 94086 Tel.: 1-800-998-8800 or 408-737-7600

RS-485 Chips x8640 Motorola Semiconductor Products PO Box 20912 Phoenix, AZ 85036 Tel.: 1-800-521-6274

RS-485 Chips National Semiconductor Corp. 2900 Semiconductor Dr. PO Box 58090 Santa Clara, CA 95052-8090 Tel.: 800-272-9959 or 408-721-5000

RS-485 Adapters Personal Computing Tools 90 Industrial Park Rd. Hingham, MA 02043 Tel.: 7-6728 or 617-740-0120; fax: 617740-2728

RS-485 Adapters and Software RE Smith 4311 Tylersville Rd. Hamilton, OH 45011 Tel.: 513-874-4796; fax: 513-874-1236

RS-485 Chips Texas Instruments Literature Response Ctr. PO Box 809066 Dallas, TX 75380-9066 Tel.: 800-477-8924

9 Moving On:

Next time around, I'll have a special article on computer-based test equipment for electronics. However, after that, I'll return to RS-485 with an exploration of network programming.

Send your comments, suggestions, etc. to me by e-mail at janaxel@aol.com, or by mail at Lakeview Research, 2209 Winnebago St., Madison, WI 53704.

Jan Axelson (last page 36 / MICROCOMPUTER JOURNAL / July/August 1995