BY IRWIN MATH, WA2NDM

MATH'S NOTES

WHAT'S NEW AND HOW TO USE IT

A Primer on Digital Signal Interfaces

With the increasing availability of computer-controllable rigs in the amateur marketplace, the use and understanding of the various digital protocols used in controlling them—in particular RS-232, RS-422, and RS-485—is becoming more and more important. As a result, we thought that it would be a good idea this month to briefly discuss these protocols and see how they are related to each other.

To begin with, we all are familiar with standard "5 volt TTL." Logic 0 is roughly equal to anything from 0 to 0.8 volts, and logic 1 is roughly equal to anything from 2 to 5 volts. These are basic logic levels and are what the internal "works" of any rig using this logic protocol want to see. In addition, the impedance level associated with this logic family is roughly 3K ohms. While all of this is fine, transmitting TTL signals from one piece of equipment to another over any appreciable distance and at any appreciable speed is quite difficult. The worse-case difference between logic 0 and 1 is only 1.2 volts (2 – 0.8), and induced noise pulses greater than this can cause problems. The 3K ohm impedance doesn't help either (although there are 50 ohm TTL systems), since it is quite easy to induce QRM on a long 3K ohm line. This is the main reason for the development of the various specific transmission protocols to transfer data.

The first, and traditional, "grandfather" of all signal-transmission protocols is RS-232, shown graphically in fig. 1. Developed in vacuum-tube days, this protocol attempted to eliminate data errors by transmitting the two logic states as bipolar signals. As stated in the specification, logic 1 would be anything greater in amplitude that —3 volts, while a logic 0 would be anything greater in amplitude than +3 volts. The region between + and –3 volts is a "no-mans land" and is undetermined. In reality, the accepted levels are –5 to –15 volts for logic 1 and +5 to +15 volts for logic 0. The 3K ohm (minimum) line impedance still remains. It is interesting to note that the maximum data rate allowed by the formal specification is 20 kbps.

The intention behind the bipolar transmission was that positive and negative voltage levels should be quite easy to differentiate even in the presence of noise. The 10 volt differential between the logic levels allowed a great deal of noise immunity, and the wide voltage range took care of the losses on long, poorly terminated transmission lines, as well as the uncertainty of the accuracy of the voltage levels at the source. Even the familiar DB-25 connector was specified.

c/o CQ magazine

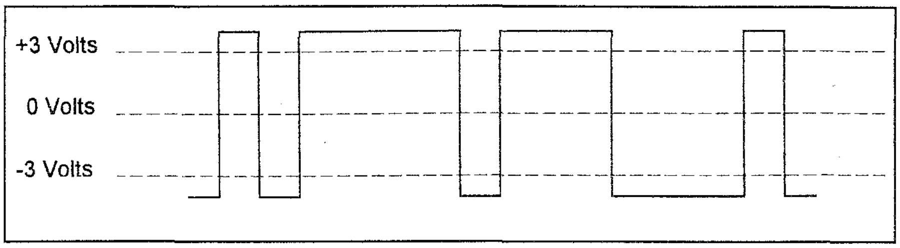

Fig. 1– Typical RS-232 waveshape.

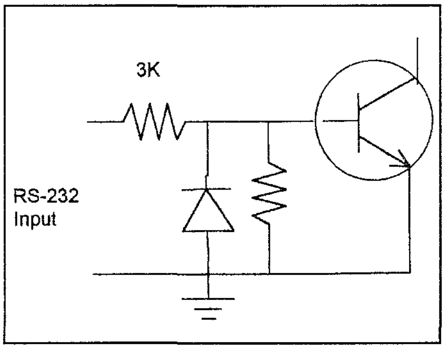

Fig. 2– Input circuit of MC-1489.

RS-232 was so popular that it is still in wide use today—with some minor variations. While the logic levels are still held to the voltages described, in many cases the reason for the negative voltage has been done away with. This can be seen by the input circuit of the popular, widely used MC1489 RS-232 line receiver (to TTL) shown in fig. 2. As you clearly can see, a diode shunts any negative voltage, which keeps Q1 turned off. A positive input turns on Q1 (the diode is reverse biased in this case), and the chip goes on to produce an output. It is interesting to note that in many cases a TTL signal will actually drive this chip as well. Another variation is in the data rate. While the original specification still calls out 20 kbps as the maximum, data rates of over 115 kbps are routinely transmitted and RS-232 chip sets are specified well into the hundreds of kilobits per second.

RS-422 is an attempt at transmitting higher data rates than the 20 kbps "limit" of RS-232. This protocol addresses the noise issue by providing a balanced, terminated twisted-pair line between the transmitter and receiver, as shown in fig. 3. Signals are sent differentially, meaning that for logic 1, conductor A is at +5 volts with regard to conductor B, and for logic 0, conductor B becomes +5 volts with regard to conductor A. Any noise induced on the line presents the same polarity on both conductors and is ignored. Because of the use of a twisted pair of wires, nominal line impedance of an RS-422 link is only 110 ohms, which by itself goes a long way toward increasing noise immunity. In addition, the data rate for RS-422 is specified at up to 10 megabits. As in the case of RS-232, RS-422 is very much with us, and chips such as the National Semiconductor DS8921 (shown in fig. 4) commonly exist to convert this protocol to and from TTL.

Since RS-232 and RS-422 both have discrete 1's and 0's and separate transmit and receive paths, converting from one to the other is done easily. Fig. 5 shows a simple circuit that will do the job quite well. The *only* consideration is to be sure that the data rates are compatible.

Of course, the data communications industry was not happy with the need to run three or four wires for a full-duplex link as with the above two protocols, so RS-485 was developed.

RS-485 is similar to RS-422 in that the signal is also differential. The main difference is that only two wires are used, and as a result, only half-duplex (one way) operation is permitted. This allows the two wires to act either as a transmit line or receive line. Obviously, this entails a mode-select (push-to-talk, if you will) feature. The best way to illustrate this is to refer to fig. 6, a schematic of a DS3696,

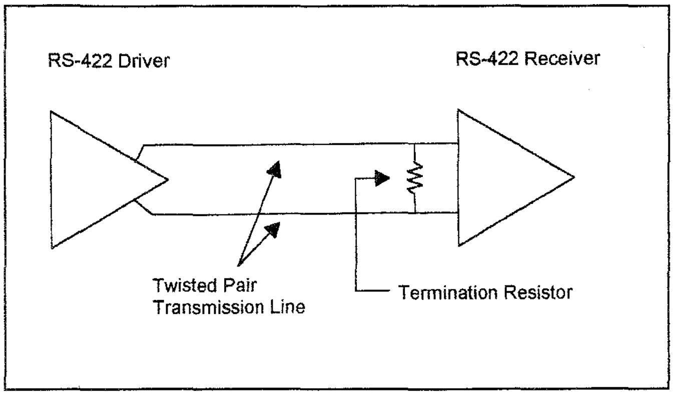

Fig. 3– Typical RS-422 transmission system.

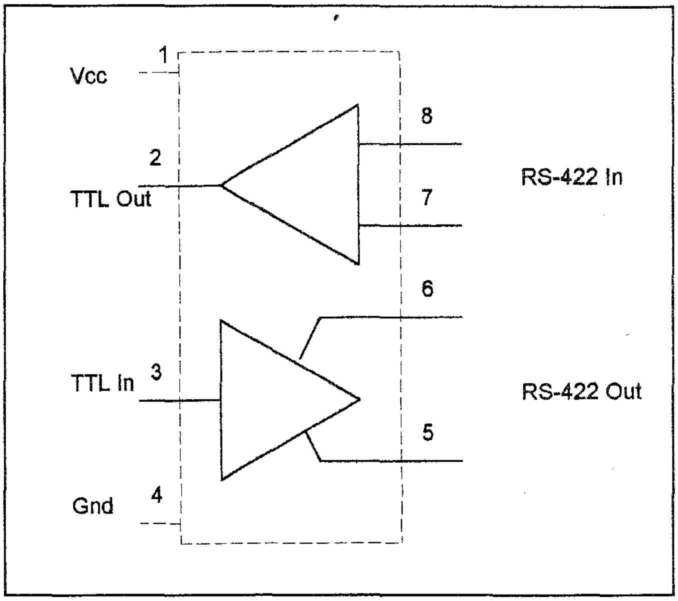

Fig. 4– Block diagram of DS-8921.

a typical TTL to RS-485 and RS-485 to TTL converter chip. As you can see, the transmitter and receiver sections of the chip are connected together. The mode select is then used to determine which portion is operational. When the transmit portion is selected, the receiver output is blocked. When the receive portion is selected, the transmit port is pulled to a high impedance so that it does not load the receiver. This high-impedance state is called the "tri-state" condition. Incidentally, the receiver section's input always presents a high impedance; it is its output that is shut off by the mode control.

The fact that the transmitter can be placed in a tri-state condition allows a data bus to be implemented fairly easily. Fig. 7 shows how this usually is accomplished. As you can see, any one station (at a time) can transmit while all other stations receive. Since all non-transmitting stations present a high-impedance load (the transmitters are in tri-state), the system works.

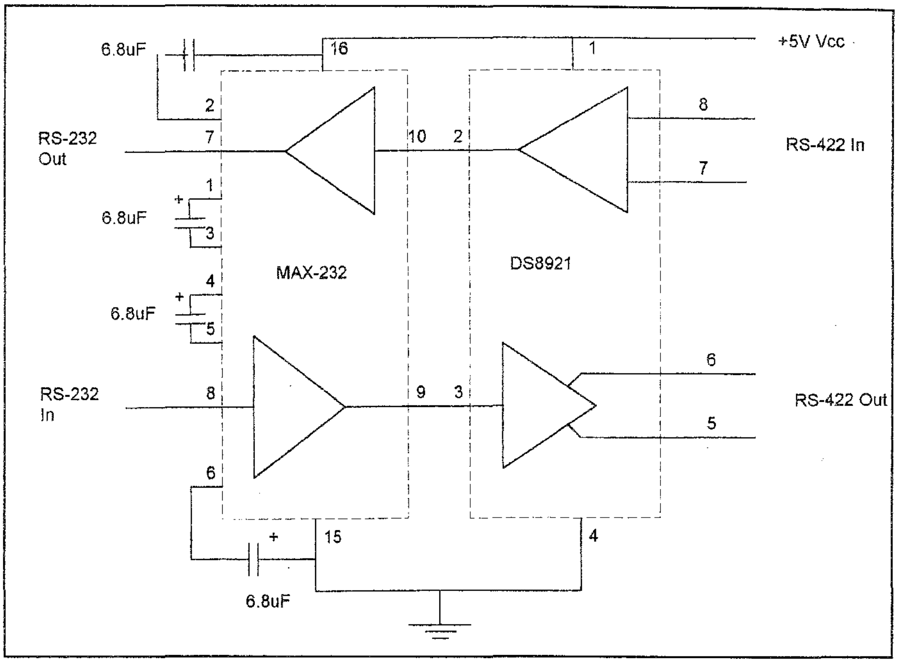

Fig. 5– RS-232 to RS-422 converter.

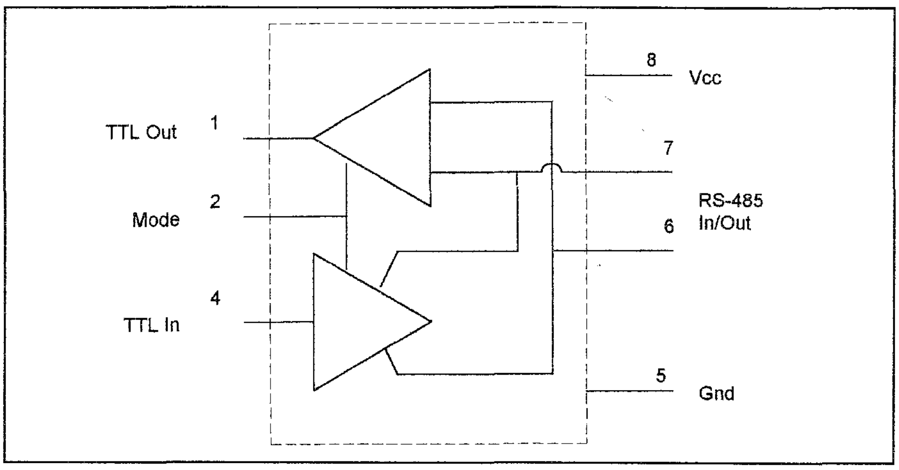

Fig. 6– Block diagram of DS-3696.

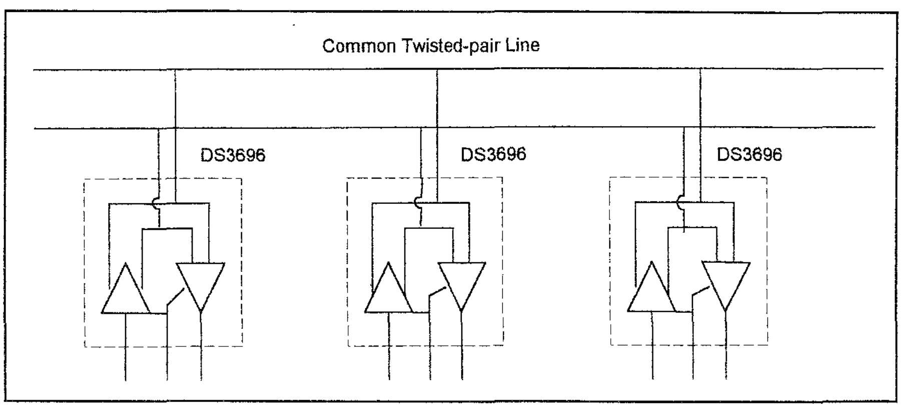

Fig. 7– RS-485 data bus transmission system.

The only "extra" information other than data that is needed is the determination when to switch a station from transmit to receive (via the mode control), and that is done with the system software.

I hope the above is useful to you, and that the various protocols you encounter in your investigations are a bit less confusing.

73, Irwin, WA2NDM