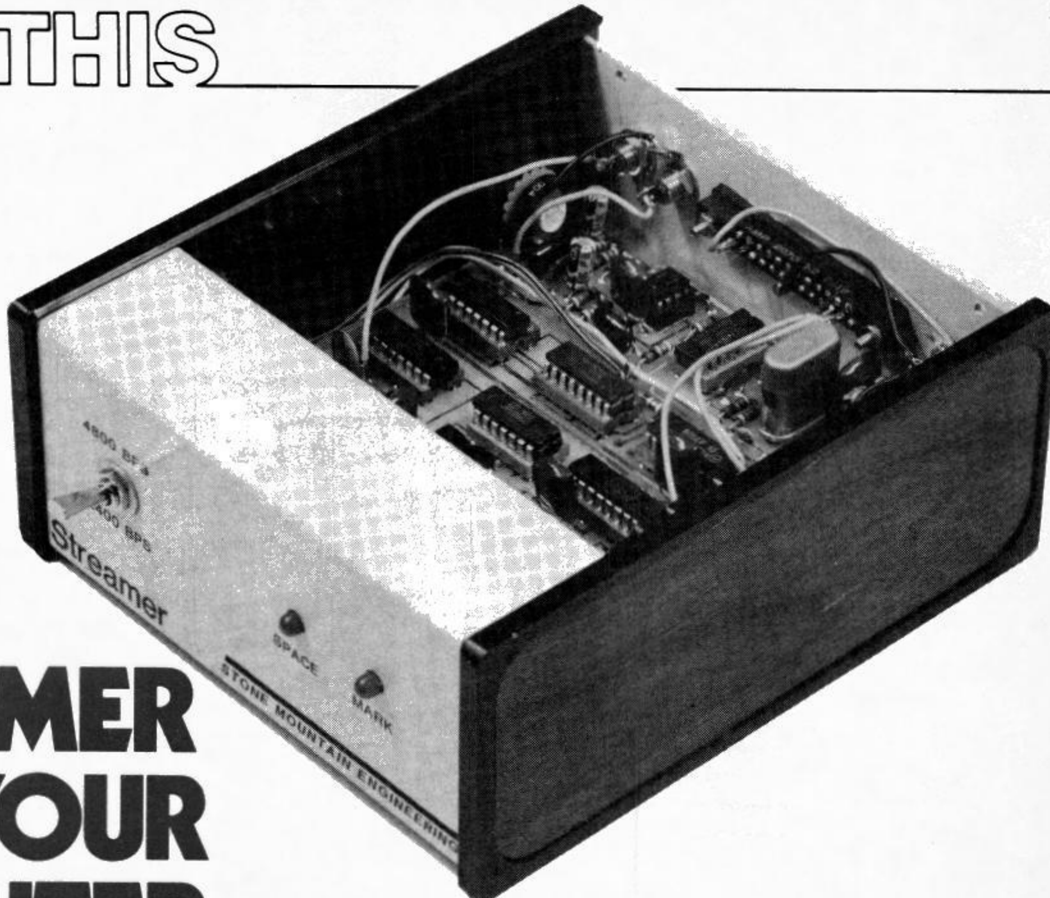

TAPE STREAMER FOR YOUR COMPUTER

MIKE HUDDLESTON

We've looked at the circuit of this universal cassette interface. Now we'll look at its data encoding scheme and then get set to build the device.

Part 2 LAST MONTH, WE DEscribed the circuit of an interface that lets you take data from your RS-232 port and store them on cassette tape. During our description, we made many references to Manchester and NRZ data-encoding techniques. We're going to start off this month with an explanation of those techniques so you can better understand the circuit's operation.

Manchester encoding

Manchester coding is a method of phase-encoding serial data. It was introduced during the early days of data recording as a means of efficiently including clocking information with transmitted data. The technique was invented at Manchester University in England to be used in Ferranti computers and it is in widespread use today in both the computer and the communications industries.

Non-return-to-zero, or NRZ, code is by far the most common means of serial data interchange between computers and their peripherals. Whether represented by TTL levels, RS-232 levels, or current loops, the conventions are the same: An idle line stays at a mark level; a data word is represented by a specific number of bits, mark or space, and each data word is preceded by a start bit (which is a space) and followed by one or more stop bits (which are marks). The word size is not specified, but is usually five to eight bits, and may or may not contain a parity bit for error detection.

There are several good reasons for the proliferation of NRZ code. First, it's easy to understand. (If you take a look at Fig. 4, you'll probably be able to immediately see what's going on with the NRZ code before we even discuss it.) Second, NRZ is supported by numerous LSI communications controllers (UART's, USART's, etc.). Third, almost every peripheral available uses it.

One characteristic of NRZ code is that it must be capable of preserving very long periods of idleness or marking. That implies that the link must have a low-end frequency response reaching down to DC. In the typical data-equipment environment, that requirement is met by hardwired connections. But when connections without DC continuity are used for data, NRZ code cannot be used. Telephone lines and audio tape, for example, where frequency response drops off below about 30 Hz, are two applications for which raw NRZ code is unsuitable. Due to the very nature of NRZ coding, there is only one place in an entire transmitted word where bit timing may be recovered during reception. That is the initial mark-to-space transition at the beginning of the start bit. Since all other bits in the word are undefined (and indeed, may be all spaces or all marks), it is easy to see that there are simply no other places in the word that can be predicted. That technique is known as *word synchronization*, as all the bits of the word are recovered by timing from that one known point.

Word synchronization implies that timing errors are cumulative: The longer the data word, the more likely the chances for recovery errors. Recognizing that, devices using NRZ coding generally are designed with crystal-controlled clocks at each end of the link, and word length is kept under ten or so bits in order to avoid timing errors.

From the above, we can see that NRZ coding is unacceptable for audio magnetic media: Audio tape devices are not responsive to DC levels, and the lack of stability

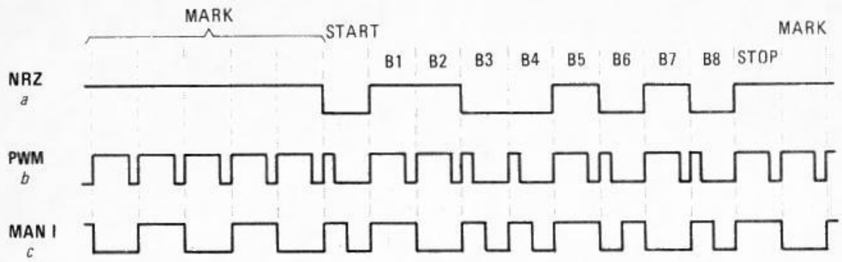

FIG. 4—THREE WAYS TO REPRESENT the serial data stream 11001010. Shown in *a* is NRZ or non-return-to-zero encoding. In *b* is pulse-width modulated encoding, and in *c* is Manchester IA.

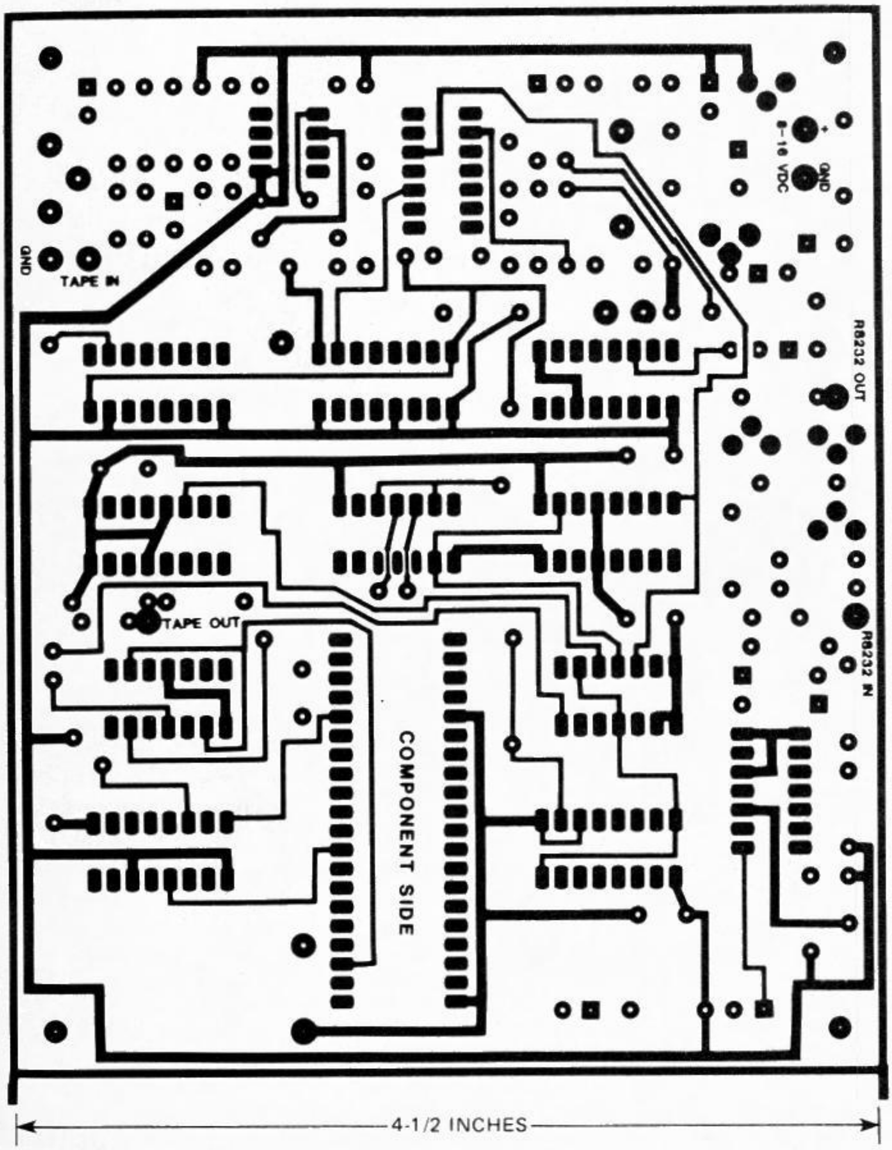

FIG. 5—THE COMPONENT SIDE of the double-sided streamer circuit board. Note that the square pads represent the positive end of electrolytic capacitors or the banded (cathode) end of diodes.

of their motor-driven capstans precludes precision clocking. An audio data interface, then, must convert NRZ marks and spaces to signals that can be successfully recorded and recovered. In addition, the interface must compensate for the tape device's inherent timing instability.

Historically, the most common method of audio data recording has been to represent NRZ data with two audio tones, one for mark, and another for space. (That modulation technique is known as FSK or *F*requency *S*hift *K*eying.) Those two frequencies are then detected during playback with either high-Q audio filters or phase-locked loops. With either technique, some individual tuning is required, and the higher the data rate, the more critical the tuning becomes.

However, if you're not interested in maximum performance, then it's possible to design quite simple interfaces. The combination of modulating fairly high audio tones with low data rates allows greatly simplified (and thus lower-cost) decoder hardware to be used. The old Kansas City Standard—300 baud 1200- and 2400-Hz audio tones—is a prime example.

The maximum frequency of the audio tones is limited by the available bandwidth of the recording medium. In order to maximize data rate, the available bandwidth should be made use of and the recording format should be bandwidth-efficient. (In other words, the ratio of the higher audio-tone frequency to the data rate should be low.) The 300-baud format discussed above uses a 2400 Hz maximum frequency, indicating a frequency to data ratio of 8 to 1. The Manchester code used by the Streamer sports a ratio of 1 to 1—an eight-fold increase in efficiency. By doubling the modulation frequency, a 16× speed advantage is attained!

Building the Streamer

Now that we understand the theory of Manchester encoding and of the Streamer's circuit, we can get on to building it. Because of the large number of discrete components, it is highly recommended that the Streamer be built on a printed circuit board. Full scale artwork for the component and foil sides of a suitable circuit board is shown in Fig. 5 and 6. If you can't make your own board, you can buy a pre-etched, drilled, silk-screened, and solder-masked board from the source listed in the parts list.

The parts-placement diagram for the Streamer is shown in Fig. 7. When you install the parts, use a clean, low-power soldering iron. The finer the tip, the better. If you purchase a PC board, you'll note that it has a solder mask, so the chances of the solder inadvertently bridging is greatly reduced. Even so, it pays to be careful. If you make your own board, take particular care to avoid solder bridges. They may be hard to find and will definitely keep the unit from operating.

There is nothing critical about the components. Everything is available through vendors that regularly advertise in **Radio-Electronics**. Normal precautions should be taken in the handling of the CMOS IC's as they can be destroyed by static charges.

None of the capacitors are used for timing, so they may have tolerances as low as 20% without ill effects. The power-supply filter capacitors, C14 and C15, can be as large as you want (as long as they fit on the board!). If the DC supply isn't filtered, however, C14 must be at least 220 µF to smooth out the ripples.

The Streamer is overdesigned with power-supply bypass capacitors. While it never hurts to include them, feel free to eliminate three or four if you want—it won't impair the circuit's operation. Note that the bypass capacitors—although listed in the Parts List—were not shown in the schematic of Fig. 1. They are, however, shown in the parts-placement diagram.

The resistor values also are not critical. All resistors may have 10% tolerance, and you may even go to either the next higher or next lower standard value, if it's more convenient. The PC-mounted potentiometer may be replaced, if desired, with a 1000-ohm resistor, as long as a jumper is added between the audio input terminal and the negative side of C1. That potentiometer is used only when extremely poor quality tape information requires an additional "tweak." In normal use, it will never be touched.

The use of IC sockets always seems to be a controversy. On the one hand, including them adds a potential long-term reliability problem; on the other, trouble shooting soldered-in IC's is a nightmare. Ultimately, it is the decision of the builder; but we recommend their use, as long as high quality sockets are used. (Cheap ones may cause more problems than they solve.)

Be careful to observe polarity on the diodes and electrolytic capacitors. The proper polarity is shown in Fig. 7. If you study the foil patterns, you'll notice that as an extra precaution, the PC board uses square pads to denote the positive end for the capacitors, and the cathode (banded) end of the diodes. The parts-placement pattern shows the transistor orientation for TO-92 packages. If you use substitute transistors, be careful that the right wires go into the right holes. The LED's must have long enough leads to reach the front panel. If they don't, simply solder enough additional wire so they do. It won't matter if they are a little longer than necessary.

The Streamer PC board may be mounted in its own enclosure (the one you see in the photos is available from the source mentioned in the Parts List). Alternatively, it can be mounted inside your computer, or even inside the tape deck. Mounting it inside the tape deck is a good idea if the tape deck supplies the Streamer power (as long as the deck won't be used for other recording!). Baud-rate switch S1 can be eliminated, since only one data rate will be used, and the DB-25 connector can be located on the rear of the

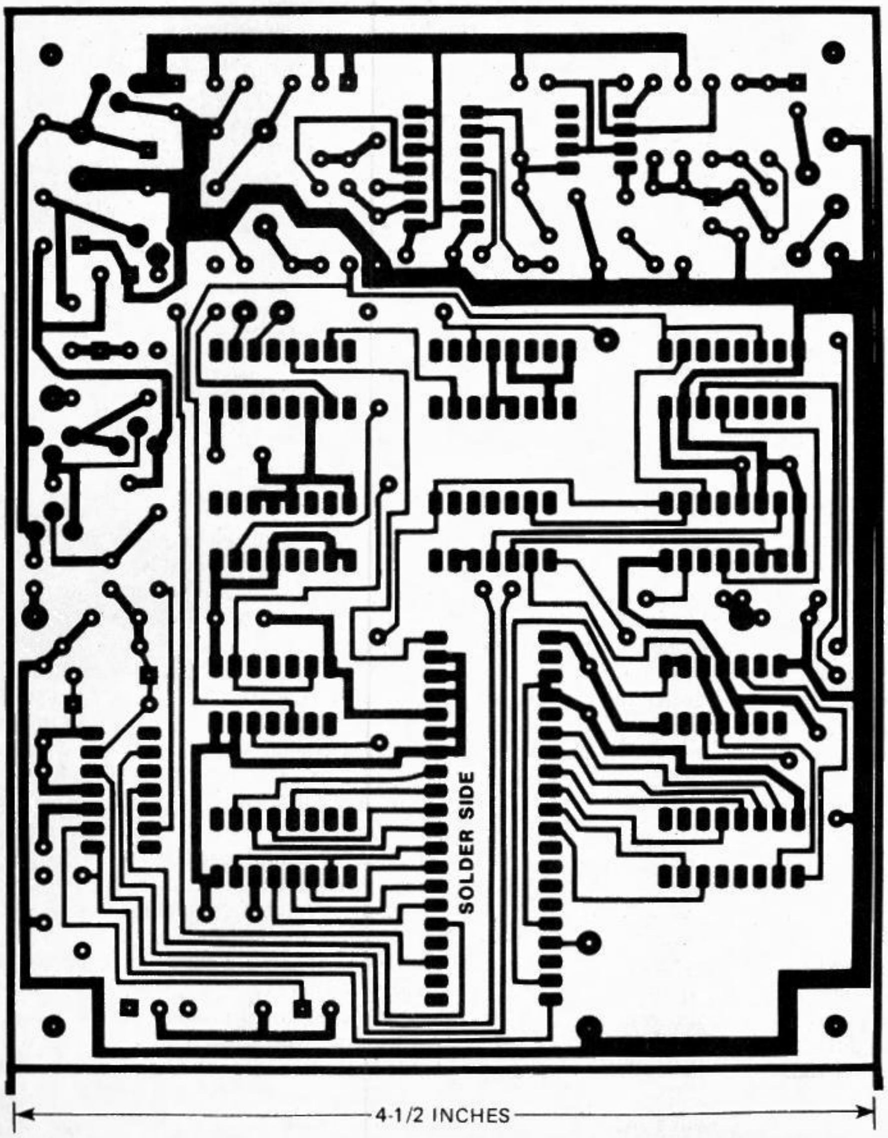

FIG. 6—THE SOLDER SIDE of the streamer circuit board. Note that square pads are used here for the same reason as on the component side.

PARTS LIST

All resistors ¼ watt, 10% unless otherwise noted

R1—1000 ohms, PC-mount, trimmer potentiometer
R2, R5, R6, R11, R16, R17, R20, R28, R29—1000 ohms
R3, R4, R7, R8, R13, R14, R18, R19, R22, R23, R25, R26, R30—10, 000 ohms
R9—1 Megohm
R10—100,000 ohms
R12, R21, R27—47,000 ohms
R15—10 Megohms
R24—2200 ohms
R31—330 ohms

Capacitors

C1, C4, C13—10 µF, 25 volts, electrolytic
C2, C21—0.001 µF, ceramic disc
C3, C12, C16—0.1 µF ceramic disc
C5, C7, C17, C18, C20, C22, C23, C24—0.01 or .1 µF bypass capacitors (not shown in schematic)
C6, C10—20 pF, ceramic disc
C8, C9—250 pF ceramic disc
C11—5 pF, ceramic disc
C14—100–330 µF, 25 volts, electrolytic
C15—47–220 µF, 25 volts, electrolytic
C19—0.01 µF, ceramic disc

Semiconductors

IC1—LM392 or LM2924 op-amp/comparator
IC2—4070 or 74C86 quad XOR gate
IC3—4040 12-stage binary ripple counter
IC4, IC6—4029 presettable up/down counter
IC5—4520 dual 4-bit synchronous counter
IC7—4011 quad 2-input NAND gate
IC8—4027 dual J-K flip-flop
IC9, IC12—74C74 dual D-type flip-flop
IC10—4015 dual 4-bit static shift register
IC11—6402 CMOS UART (Intersil)
IC13—4021 8-stage static shift register
IC14—LM339 quad comparator
IC15—78L05 low power 5-volt regulator
D1–D5—1N914 or similar
D6, D7—standard red LED
Q1, Q3—2N3904
Q2, Q4—2N3906
XTAL1—2.4576 MHz crystal

MISCELLANEOUS: PC board, enclosure, DPDT switch, DB25 connector, phono jacks for tape deck connectors, hardware, solder, etc.

The following are available from Stone Mountain Engineering Co., PO Box 1573, Stone Mountain, GA, 30086: Printed circuit board, double-sided with plated-through holes, solder masked and silkscreened, for $28; Enclosure, with all holes punched and legends silkscreened, $16; Both PC board and enclosure for $40. All orders must include $1.50 shipping and handling, and Georgia residents please enclose 3% sales tax.

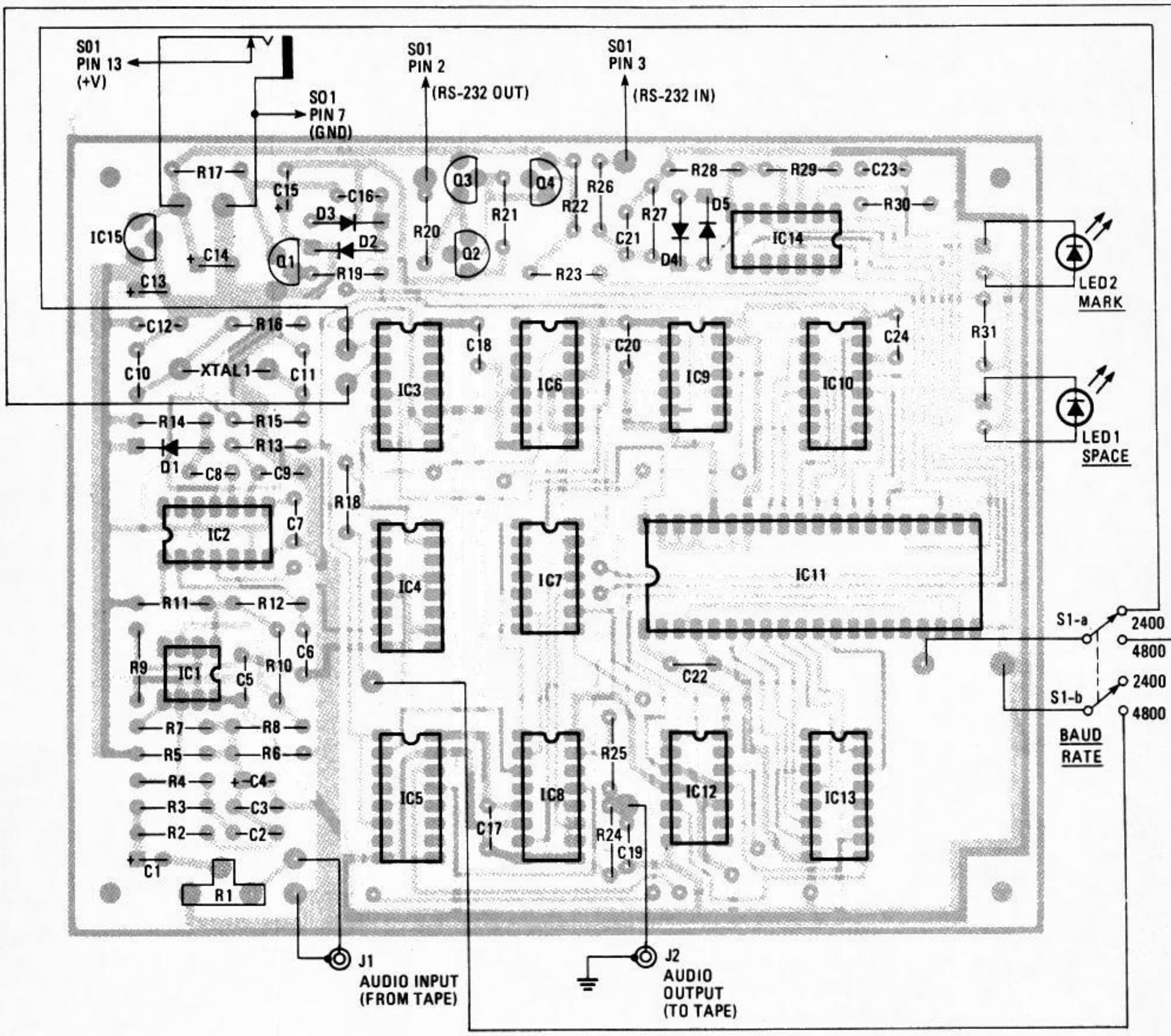

FIG. 7—PARTS PLACEMENT DIAGRAM for the Streamer shows both the on-board and off-board components.

deck. Locating the LED's may be a problem, though, depending on the configuration of your tape deck.

Installing the Streamer inside a computer is the least attractive option, as this precludes its use with other computers. One of the most important uses for the Streamer is to transfer files from one computer to another, and that cannot be accomplished if it is dedicated to one unit.

If the Streamer is to be mounted in its own enclosure, the procedure should be to: **1.** Assemble the PC card. **2.** Wire the DPDT switch (S1) to the card. **3.** Wire the connectors on the rear panel. **4.** Install the card and attach the rear panel wires. Follow the schematic for the proper connections to the DB-25 and the closed-circuit jack carefully. The power connection to DB-25 pin 25 is optional, and may be omitted if an external plug-in supply is to be used.

Troubleshooting

Initial trouble-shooting can be accomplished with an ordinary 20,000 ohms/volt (or better) volt-ohm-milliameter. Connect the VOM, on its highest current scale, between the Streamer and its power supply. The streamer should draw in the range of 10–30 mA, and no LED's should be on. A very high current would indicate a short or a component in backwards, while a low current would indicate an open in either the power supply lines or the ground return.

If the above test is successful, remove the meter and connect the Streamer directly to the power supply. Measure the +5 volt supply at any convenient place. If it is above 5.25, or below 4.75, IC15 may be defective. Now measure the voltage at the end of R21 closest to Q2. There should be a negative voltage with a magnitude slightly less that that of the positive supply. If there is, that indicates the clock, the negative supply, and IC3 are functioning. Measure the the voltage on IC8, pin 15. If it reads about 2.5 volts, then the Manchester encoder is working.

Connect the encoder's AUDIO OUTPUT to the AUDIO INPUT. Adjust R1 to the normal full-on position. (If you are using the PC board, this is the full-clockwise position, viewed from the board's edge.) If the MARK LED comes on, both the encoder and decoder are working. That is about all the testing that can be done with a VOM. If everything looks good, it should work properly the first time.

If an oscilloscope is available for testing, much more extensive trouble shooting may be accomplished by referring to the schematic and the theory of operation. A word of caution, however; the bit sequence that appears at the output of IC12-b will be in a different order than that received. That "bit shuffling" was done to simplify the board layout. The received bits, then, will also be in a scrambled order until IC10 corrects them.

Using the Streamer

The Streamer is one of the simplest add-ons to any computer system. As long as your computer has an RS-232 port capable of a transmission rate of 4800 baud and a reception rate of 9600 baud, and as long as you have some sort of software to support storing and loading, you should have no problems. The tape machine can be just about anything—even a cheap portable (although, as we'll mention shortly, you'll sacrifice some performance). If you have a tape deck as part of your stereo system, it's probably ideal.

The supporting software may be the SAVE and LOAD commands with a BASIC interpreter, as long as they can be routed to an RS-232 port. BASIC programs can also be conveniently saved by LISTing them out to the Streamer, and read back in through an RS-232 port assigned as the console. The latter method has the advantage of allowing BASIC programs from different machines to be loaded, as the ASCII listing is, in effect, the same information that would be entered through the keyboard. The same can be done with source files, or, for that matter, any ASCII file.

Machine-language program storage and retrieval can be handled by any of a great many approaches, at least one of which is probably resident in the computer you now use. The routine the author uses on his system, like many others, transmits in sequence a delimiter stream, a load address, 255 bytes of data, and then a checksum. That is followed immediately by the next load address, data, checksum, and so forth, until all the data is done. When the tape is played back to the computer, each checksum is compared with a calculated checksum, and any error causes the routine to halt.

If you ever run across data errors when loading programs back into your computer, the cause is probably dirty tape heads. Simply cleaning the heads should eliminate the problem.

When configuring your RS-232 port, remember that the Streamer works with 8-bit data words. Those eight bits can be all data, seven data and one parity bit, or seven bits, no parity, and at least two stop bits. The only real requirement is that start bits must be at least nine bit-times apart, such as with eight data bits and one stop bit. In storing 7-bit ASCII files, it is normal to follow with a parity bit. The Streamer will treat the eighth bit as part of the data, faithfully recording it and playing it back. The Streamer itself does no parity checking; it simply records the data and returns what is presented to it.

The audio output of the Streamer is designed to present a signal compatible with the audio input of a hi-fi type tape deck. Since modern decks have input-level controls, the control should be adjusted for best performance. Unlike conventional cassette interfaces, that adjustment is not critical at all. To determine your optimum adjustment, use the tape counter to record segments at various settings, then play them back, noting any recovery errors. The errors should occur at the extremes of the level control settings. Simply set the control approximately half way between where errors occurred, and you're under way.

The Streamer can be used with low-cost portable tape recorders, with some loss of performance: Because of their lower bandwidth capability, the Streamer must be operated at 2400 baud instead of 4800 (which is possible with hi-fi type decks). The audio signal out of the streamer is about 0.9 Volt peak to peak, which suits most decks just fine. But you'll probably have to reduce that level if you want to apply it to a portable recorder. You can do that with either a resistive voltage divider at the recorder's input, or by simply reducing the value of R24: Try values in the 100- to 1000-ohm range.

Whether you use the Streamer for primary data and program storage, disk backup, or to exchange programs with other computers, it will undoubtedly be a welcome addition to your computer system. **R-E**