I/O Expansion for the TRS-80

Part 2: Serial Ports

Steve Ciarcia
POB 582
Glastonbury CT 06033

Last month in Part 1, I discussed the attachment of parallel input and output ports to the Radio Shack TRS-80 computer. This was basically a response to the many inquiries I have had on TRS-80 interfacing. As usual, it was a general presentation, intended to first enlighten the reader with interfacing concepts and then tender a few alternative circuits for construction. While TRS-80 owners benefit most directly, many computers have similar bus structure and can just as easily accommodate parallel input/output (I/O) expansion.

The presentation this month of a serial interface for the TRS-80 required a little more thought. Parallel ports are strictly hardware devices which in their simplest form only require execution of a single assembly-language or BASIC instruction to function efficiently. A serial interface, on the other hand, needs a software program to direct its operation. The many registers and buffers involved in the serial communication process must be synchronized by the execution of a *serial-driver* routine stored in memory. Any design for a serial port has to take into account the capabilities and memory location of this routine. Even the most splendid hardware circuit would be a failure if the software driver interfered with other computer functions.

To eliminate any potential problems that might occur, I decided to make my design completely software-compatible with existing TRS-80 serial-driver routines. This does not necessarily minimize circuit complexity by any means, but it greatly enhances potential user acceptance.

I was equally concerned with the power requirements and physical configuration. Radio Shack sells a serial-interface board for the TRS-80, but it cannot be operated independently and requires integral attachment to the expansion interface module. The expansion interface and one serial port add $400 to the cost of the basic computer. Also, with its present hardwired addressing, the TRS-80 can support only one serial port and one parallel printer port.

This RS-232C interface design is compatible with existing TRS-80 serial-interface control software.

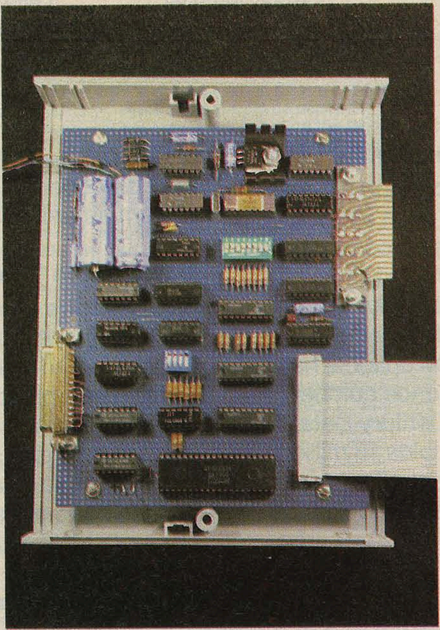


Photo 1: *Prototype of the COMM-80 interface. The ribbon cable at the lower right connects to the expansion-bus port (either the expansion connector on the keyboard/processor unit or connector J2 on the expansion interface). The edge connector at the upper right is for the Centronics-compatible, parallel printer port. The RS-232C DB-25S connector is at the lower left.*

Depending upon the intended application, you may not need the extra functions (eg: disk controller and memory expansion) provided in the expansion interface. The $300 outlay for the expansion interface is an extraordinary expense if you merely intend to attach a modem and use the TRS-80 as a terminal on a timesharing network, such as the Source or MicroNet. Rather than duplicate what I consider to be a restrictive hardware configuration, I have attempted to present a cost-effective communications interface that gives more flexibility in use and has a better price/performance ratio.

The COMM-80 Communications Interface

The approach I decided to take was to combine elements from Part 1 of this article with this one, and produce a stand-alone serial/parallel interface which could plug directly into the expansion-bus connector (the keyboard-unit expansion connector or connector J2 on the expansion interface). Designated the COMM-80, the unit includes a 50 to 19,200 bit per second (bps) RS-232C serial port, a full 8-bit-in/8-bit-out parallel printer port, an auxiliary expansion-port edge connector, and switch-selectable addressing which allows a single TRS-80 to simultaneously connect up to sixteen COMM-80 interfaces. A block diagram of the COMM-80 is presented in figure 1, and a picture of the prototype is in photo 1.

What Is a Serial Port?

Communication between computers, terminals, and other peripheral devices can be in either serial or parallel mode. In parallel mode, the entire information segment (ie: data word) is transmitted or received simultaneously in a single time frame. In serial mode, this same information is divided into its constituent bits and these bits are transmitted individually over a longer period of time. In cases where high-speed data rates are involved, such as in interaction with a floppy-disk drive, the communication is usually in parallel and can involve as many as forty data and control lines. Serial mode is generally used for lower-speed exchanges.

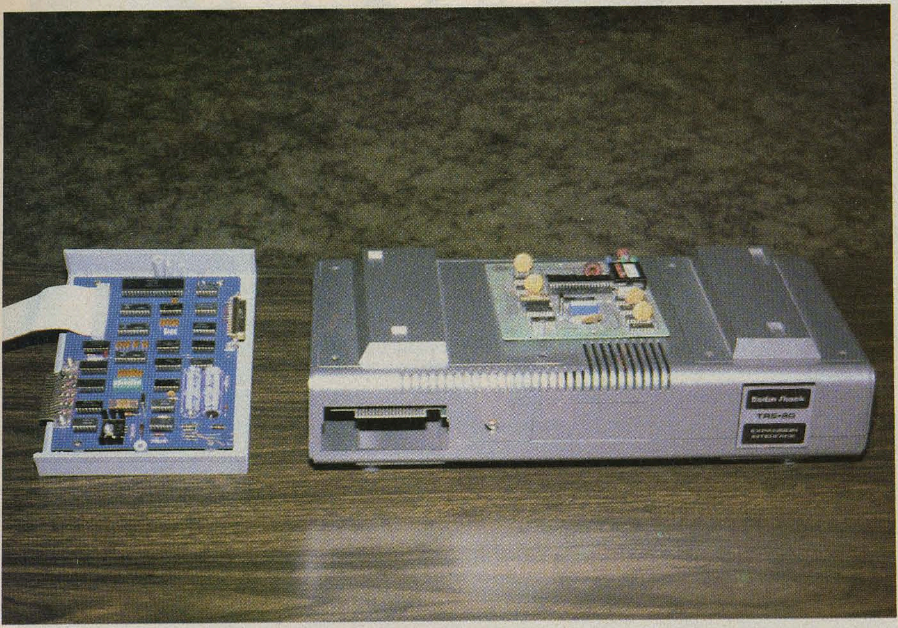


Photo 2: *Here are two ways of adding RS-232 communication capability to the Radio Shack TRS-80. The COMM-80 unit is shown on the left; the combination of the Radio Shack expansion interface and serial-interface board is shown on the right.*

Photo 3: *A TRS-80 equipped with Level II BASIC, the COMM-80 interface, and a Novation CAT modem can be used as a remote terminal for a time-sharing service such as the Source.*

An example a little closer to home is the addition of a video terminal and a printer to a computer system. Both the terminal and printer are designed to accept American Standard Code for Information Interchange (ASCII) coding, which requires only 7 bits to define a character.

The connections between the computer and the video terminal can be either serial or parallel. The choice in this case is not determined by data rate but by expense. Parallel communication is relatively easy and inexpensive for a computer. Few components are involved, and a 6-foot length of nine-conductor cable (seven lines to carry the 7-bit ASCII data, one line each for data strobe and ground) will not cost too much. Serial interfacing is another matter entirely.

Microprocessors do not naturally communicate in serial format. There are no single machine-language instructions to perform this function. To serialize data we must add a separate hardware device called a universal asynchronous receiver/transmitter (UART). It looks just like a parallel port to the processor, but internally the UART is a very complicated device.

A UART is a special large-scale integration (LSI) circuit that accepts a data byte in parallel form from the processor and converts it into a universally accepted serial format. Any two terminals set at the same data-transmission rate could conceivably be interconnected to communicate, regardless of internal operating-system differences. The expense for this flexibility is in the neighborhood of $200 to $500 per data channel, depending upon the computer bus configuration.

Transmitting Serial Data

Serial data can be transmitted in either *synchronous* or *asynchronous* format. I will address this discussion only to the latter format since asynchronous communication is the technique employed in the COMM-80. The asynchronous format allows unlimited time gaps to occur between transmission of characters.

The internal structure of a UART consists of a separate parallel-to-serial transmitter and a serial-to-parallel receiver joined by common programming pins. The two sections can be used independently provided they adhere to the same bit-format options. Sending a character from the processor is simply a matter of performing a parallel-output operation to the UART. The decoded-output strobe loads the UART with the data and initiates the serialization process.

Figure 2 shows a plot of logic levels versus time during the transmission of a single character. When no data is being sent, the data-transmission line remains in a logic 1 state. A 1-to-0 high-to-low transition on the line signifies that a character is being sent. The first bit is called a *start bit*. The next 5 to 8 bits are data; these are followed by a parity bit. Finally, the end of transmission is defined by the addition of 1 or 2 stop bits at the end of the character. The start, stop, and parity bits are all added as part of the UART's function.

Meanwhile, the receiver section of the UART is continuously monitoring the input line for the start bit of a character. When the start bit comes, the following data bits are placed into a holding register and their parity is checked against the state of the parity bit. Completion is signaled by setting a *data-available* flag. This flag, plus others defining *buffer status, parity*, and *overrun errors*, is read by the processor to determine when input data is ready or when another character can be transmitted. The individual pin functions of a typical UART are described in table 1.

RS-232C Interface Characteristics

So far, I have discussed only serialization of the data. I have said nothing about voltages or logic conventions associated with control of the information transmitted between

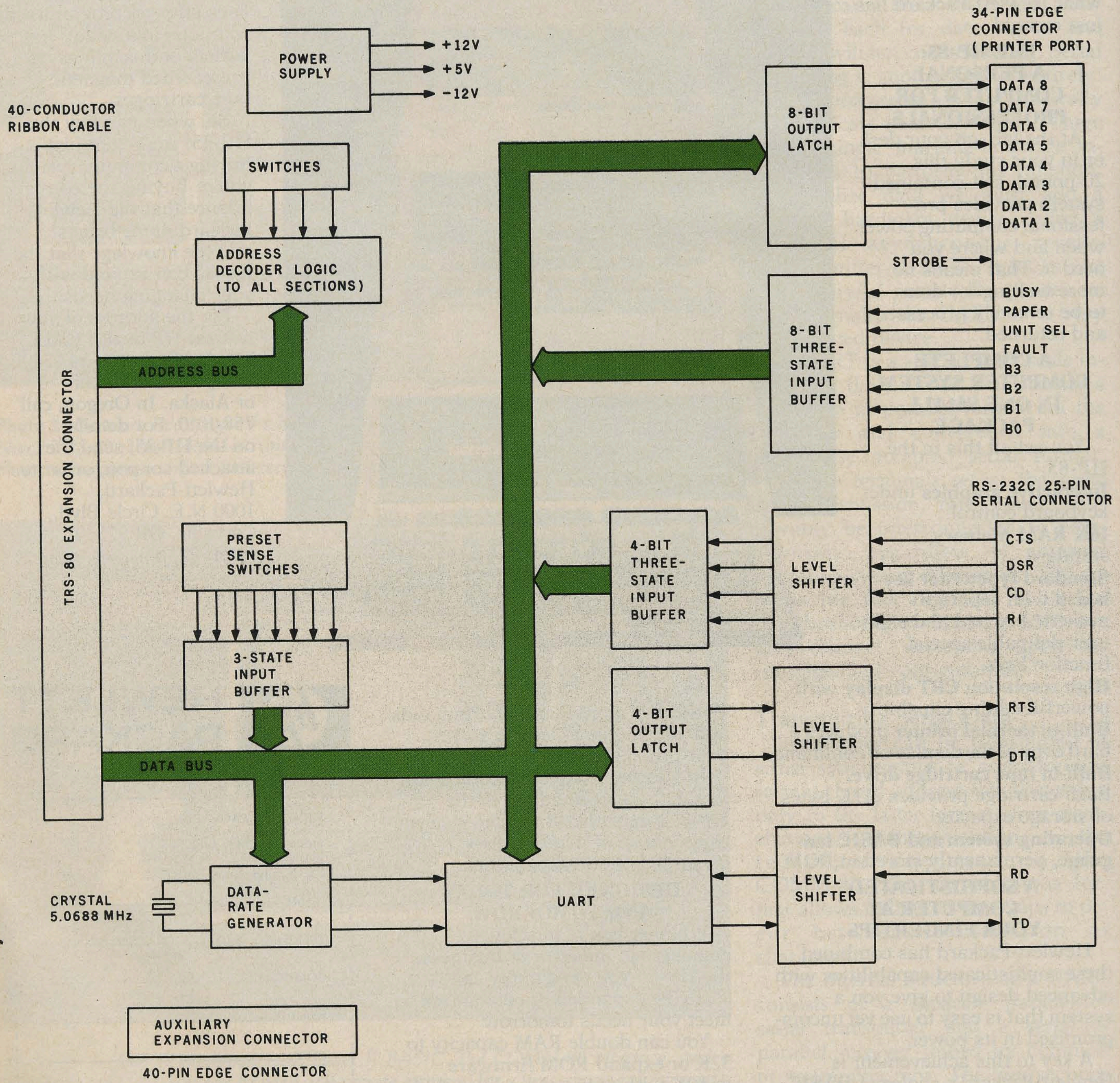


Figure 1: *Block diagram of components and data flow in the COMM-80 serial and parallel interface for the Radio Shack TRS-80.*

equipment. The Electronic Industry Association (EIA) RS-232C electrical specification defines voltage levels and control signals: a logic level 1 is called a "mark" or "off" and is considered to be anything more negative than −3 V. A logic 0 is called a "space" or "on" and is considered to be anything more positive than +3 V. As a rule, designers tend to use +12 V and −12 V for the 0 and 1 logic states.

In addition to standardizing the serial format, the EIA also specifies that the connector for RS-232C be a 25-pin, D subminiature type (called a *DB-25*). The pin assignments and functions are shown in table 2.

The COMM-80 Hardware

The COMM-80 is driven only by signals present on the buses of the computer. All sections communicate with the processor as memory-mapped or directly addressed input/output ports. Figure 3 illustrates the complete schematic diagram of the COMM-80 interface in three sections.

There are two major sections: parallel printer port and serial port. They are joined together by a common address-decoding circuit and power supply.

Address Decoding

A standard TRS-80 expansion interface has an edge connector commonly called the Centronics printer port. It actually combines an 8-bit parallel output port and a 4-bit parallel input port. The addressing for this section is hardwired for hexadecimal memory location 37E8. Part of this same address decoder is used for the Radio Shack serial-interface board. Coincidentally, the Radio Shack serial interface is decoded to use I/O port addresses E8 thru EB for data-transfer and control functions.

The address-decoding section of the COMM-80, consisting of IC1 thru IC7, is designed to decode this set of addresses as well as a range of other addresses. The range for the printer port is hexadecimal memory addresses 3708 to 37F8, and the serial range is hexadecimal I/O addresses 08 to F8. Figure 4 illustrates the switch settings for the different ranges.

There is a particular rationale for setting up the addresses this way. A user attaching a COMM-80 to his system would naturally set the switches for the range E8 thru EB, and the interface would then be completely compatible with standard TRS-80 software. Should an expansion-

Pin Number	Name	Symbol	Function
1	V_{CC} Power Supply	V_{CC}	+5 V Supply
2	V_{GG} Power Supply	V_{GG}	−12 V Supply (Not connected on AY-5-1015
3	Ground	V_{GL}	Ground
4	$\overline{\text{Received Data Enable}}$	$\overline{RDE}$	A logic 0 on the receiver-enable line places the received data onto the output lines.
5	Received Data Bits	RD8	These are the eight data output lines. Received characters are right justified; the least significant bit (LSB) always appears on RD1. These lines have three-state outputs.
6		RD7	
7		RD6	
8		RD5	
9		RD4	
10		RD3	
11		RD2	
12		RD1	
13	Parity Error	PE	This three-state line goes to a logic 1 if the received-character parity does not agree with the selected parity.
14	Framing Error	FE	This three-state line goes to a logic 1 if the received character has no valid stop bit.
15	Over-Run	OR	This three-state line goes to a logic 1 if the previously received character is not read (DAV line not reset) before the present character is transferred to the receiver-holding register.
16	$\overline{\text{Status Word Enable}}$	$\overline{SWE}$	A logic 0 on this three-state line places the status word bits (PE, FE, OP, DAV, TBMT) onto the output lines.
17	Receiver Clock	RCP	This line will contain a clock whose frequency is sixteen times the desired receiver data rate.
18			
18	$\overline{\text{Reset Data Available}}$	$\overline{RDAV}$	A logic 0 will reset the DAV line.
19	Data Available	DAV	This three-state line goes to a logic 1 when an entire character has been received and transferred to the receiver holding register.
20	Serial Input	SI	This line accepts the serial bit input stream. A marking (logic 1) to spacing (logic 0) transition is required for initiation of data reception.
21	External Reset	XR	Resets shift registers. Sets SO, EOC, and TBMT to a logic 1. Resets DAV, and error flags to 0. Clears input data buffer. Must be tied to logic 0 when not in use.
22	Transmitter Buffer Empty	TBMT	The three-state transmitter buffer-empty flag goes to a logic 1 when the data bits holding register may be loaded with another character.

Table 1: *Pin functions for the AY-5-1013, AY-5-1015, or COM2017 UARTs.*

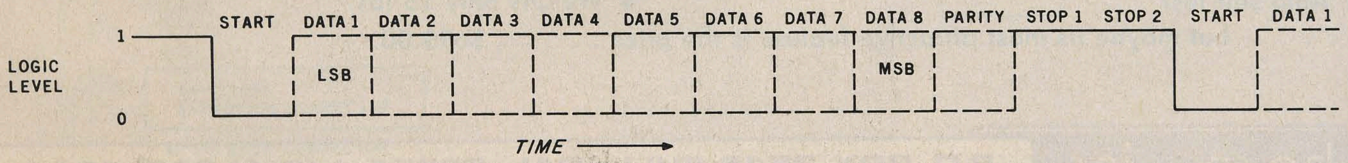


Figure 2: *Logic levels plotted against time during the transmission of an 8-bit data word in asynchronous serial format.*

Pin Number	Name	Symbol	Function
23	$\overline{\text{Data Strobe}}$	$\overline{DS}$	A strobe on this line will enter the data bits into the data-bits-holding register. Initial data transmission is initiated by the rising edge of $\overline{DS}$. Data must be stable during entire strobe.
24	End of Character	EOC	This line goes to a logic 1 each time a full character has been transmitted. It remains at this level until the start of transmission of the next character.
25	Serial Output	SO	The entire character is transmitted bit by bit (that is, serially) over this line. It will remain at logic 1 when no data is being transmitted.
26–33	Data Bit Inputs	TD1 TD2 TD3 TD4 TD5 TD6 TD7 TD8	There are up to 8 data-bit-input lines available.
34	Control Strobe	CS	A logic 1 on this lead will enter the control bits (EPS, NB1, NB2, TSB, NP) into the control-bits-holding register. This line can be strobed or hardwired to a logic 1 level.
35	No Parity	NP	A logic 1 on this lead will eliminate the parity bit from the transmitted and received character (no PE indication). The stop bit(s) will immediately follow the last data bit. If not used, this lead must be tied to a logic 0.
36	Number of Stop Bits	TSB	This lead will select the number of stop bits (1 or 2) to be appended immediately after the parity bit. A logic 0 will insert 2 stop bits.
37, 38	Number of Bits Per Character	NB2, NB1	These two leads will be internally decoded to select either 5, 6, 7, or 8 data bits per character. *NB2 NB1 bits/character*: 0 0 5; 0 1 6; 1 0 7; 1 1 8
39	Odd/Even Parity Select	EPS	The logic level on this pin selects the type of parity which will be appended immediately after the data bits. It also determines the parity that will be checked by the receiver. A logic 0 will insert odd parity, and a logic 1 will insert even parity.
40	Transmitter Clock	TCP	This line will contain a clock whose frequency is sixteen times the desired transmitter data rate.

interface module be added to the system later, the user would merely flip a switch specified by table 3 to change the port address (the expansion interface is set only for 37E8). The switch circuit is shown in figure 4. The system could then accommodate two printers. As table 3 shows, there are sixteen possibilities, so there could be sixteen printers and sixteen serial ports. From this point on, however, I will refer only to the addressing range of E8 thru EB.

The Printer Port Is a Full 8 Bits

Since I explained parallel ports in detail last month, I will discuss the printer port briefly. Initially my intention was to provide a general-purpose I/O port so that the user could connect some of my other projects and interface designs. As it worked out, however, I decided to combine efforts and configure the parallel port to serve as the printer port as well. The major difference is that the COMM-80 incorporates a full 8-bit input and a full 8-bit output port. Its address is nominally hexadecimal 37E8 in memory-address space. Writing to memory location 37E8 latches data onto IC14 and IC15 (both 74LS75 devices), and reading memory location 37E8 gates the printer status signals through the three-state buffer IC19 (a 74LS244 device).

Once you have installed an RS-232 port, a whole new world of peripherals opens up.

Serial Port

The serial-port section requires four input and four output strobes to operate. As previously mentioned, the serial-port control addresses are nominally set for hexadecimal E8 thru EB. Figure 5 more explicitly illustrates the hardware derivation of these signals and lists their functions. These strobe signals coordinate the RS-232C handshaking, the sense switches, the data-rate generator, and the UART. All four subsections can be independently controlled in software by reading and writing to the appropriate port address.

The sense switches, for instance, are merely a convenience. It is a way for the user to present a frequently used combination of options. These switches, outlined in figure 6, allow selection of data rate, word length, parity condition, and number of stop bits. There is, however, no physical connection between these switches and the other sections. The software-driver routine coordinates the option selection.

First the routine determines the state of the switches by reading input port E9. It determines from the setting of switches SW6 thru SW8 what data rate the user wants. The particular code for that rate, selected from table 4, is written to output port E9. The remaining switch settings are written into the UART control register EA. Three bits of this output (b_0 thru b_2) and input port E8 are used for the RS-232C handshaking. The data-rate generator is presented in figure 7.

The sense switches are not absolutely necessary for operation of the serial interface. Most software drivers, such as the ST80 program written by Lance Micklus, offer a selection of the options through the keyboard. Separate data rates for the

Text continued on page 54

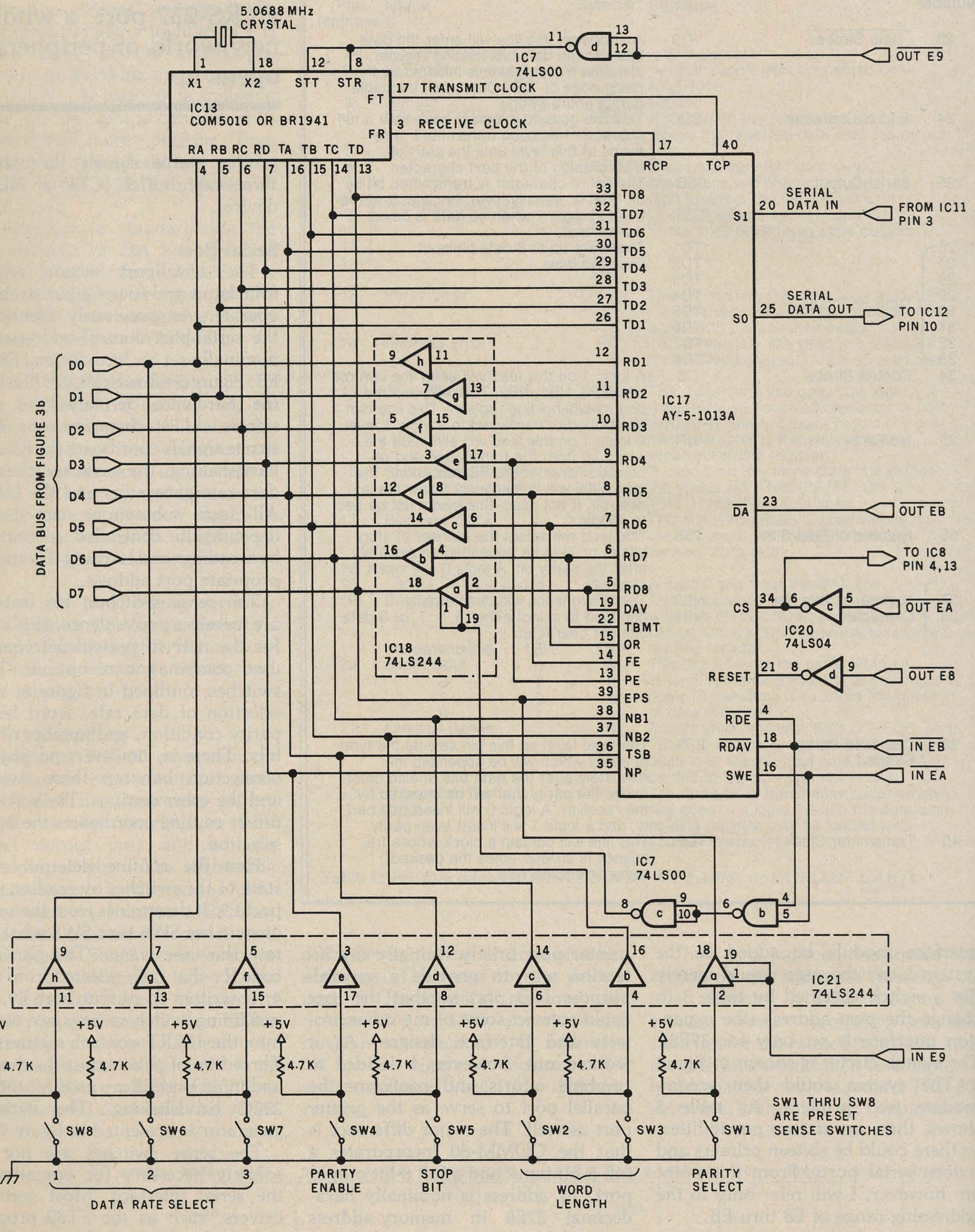


Figure 3a: *Section of schematic diagram of COMM-80 interface circuit. Shown here are the data-rate selector, the UART, and the option-selecting switches. The data-rate selector can be either a COM5016 or a BR1941. Various UARTs can be used instead of the AY-5-1013A, including the TR1602, COM2017, S1883, and TMS6011. A UART that uses a single +5 V power supply, such as the AY-3-1015, may also be substituted.*

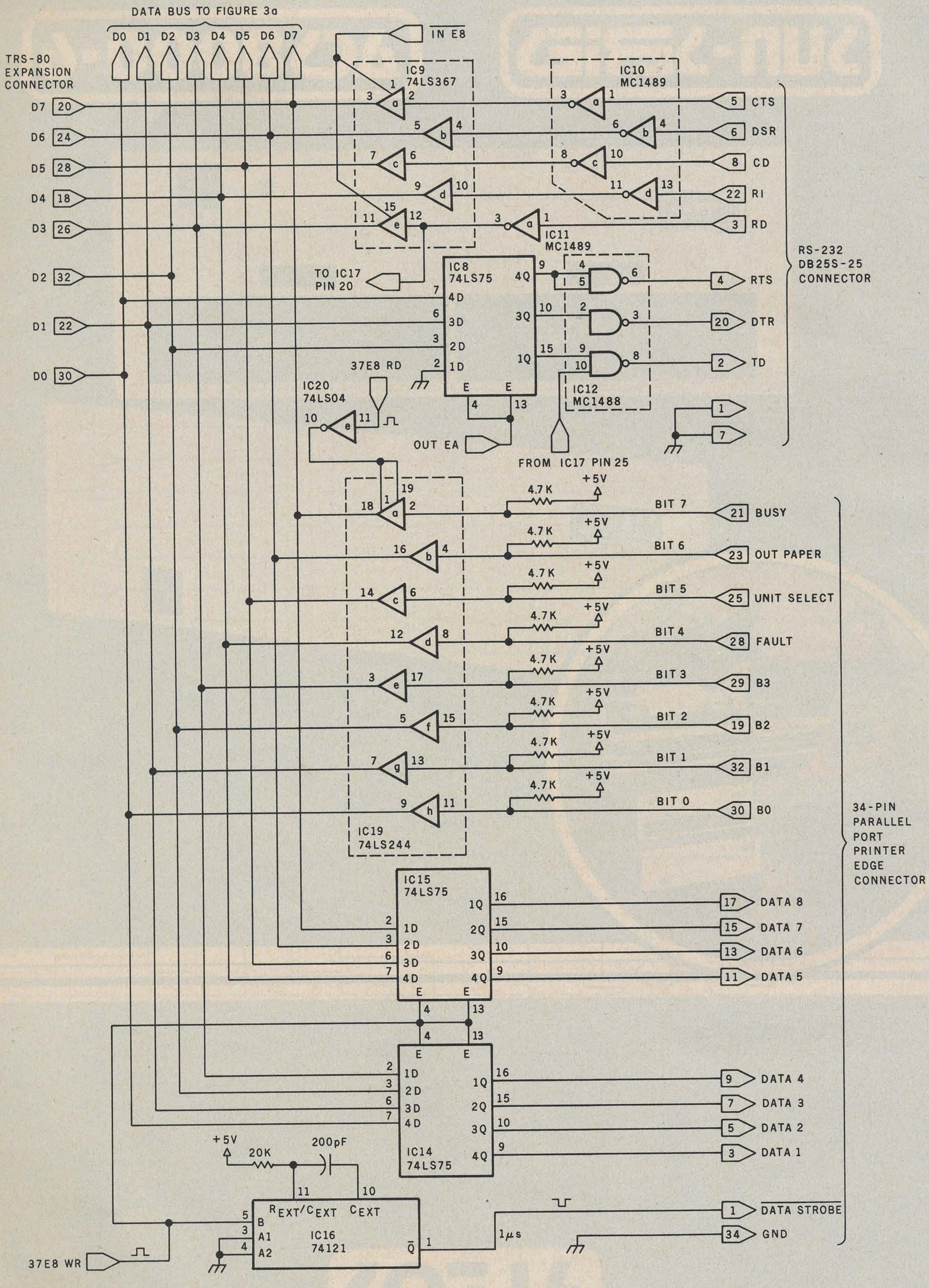


Figure 3b: *Section of schematic diagram of COMM-80 interface. Connections to data buses and peripheral connectors are presented here. Some care must be exercised in connecting the COMM-80 to the expansion bus. It is best to use shielded ribbon cable. The production version of the COMM-80 includes two auxiliary expansion-bus edge connectors, which are like the one on the back of the keyboard/processor unit.*

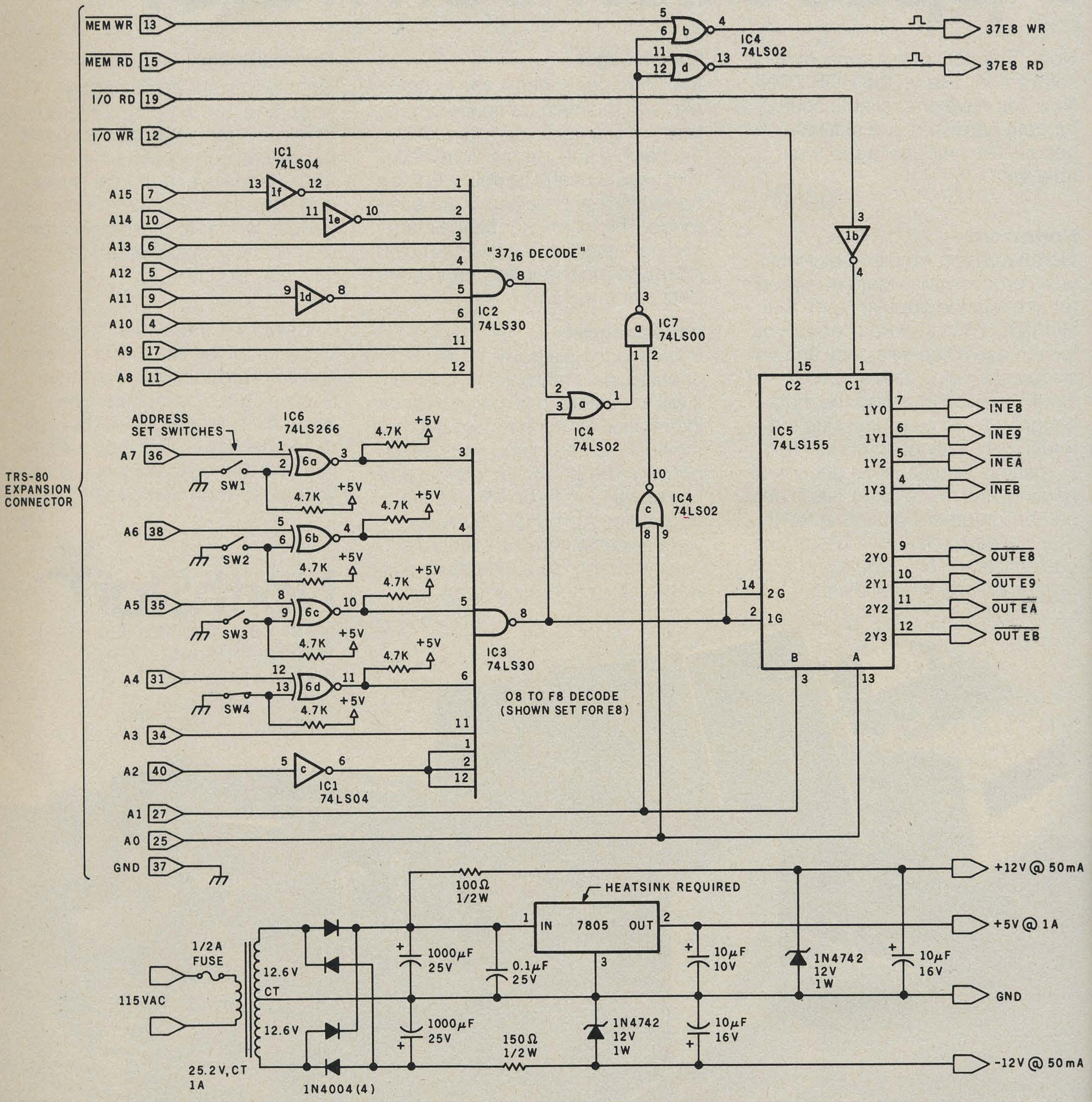


Figure 3c: *Section of COMM-80 interface circuit, including power supply and address-selection circuitry. Power to the interface should not be cut off while the TRS-80 is in operation, lest programs be lost. Both units should be powered up and down simultaneously.*

Text continued from page 49:
transmitter and receiver can also be established. This is easily accomplished by a direct output command to the data-rate generator using the codes from figure 6.

From this point on, serial communication proceeds by simply loading the UART with the data to be transmitted (using the Z80 instruction OUT EB) and reading the UART status register to see if the byte has been completely sent or if there is a received data word available (with the IN EA instruction).

The software driver needed for this interface is too long to discuss in this

Text continued on page 58

Pin 1	PGND — Protective Ground This is chassis or equipment ground. It may also be tied to signal ground.
Pin 2	TD — Transmit Data This is the serial data from the terminal to the remote receiving equipment. When no data is being sent it is in a marking (1) condition.
Pin 3	RD — Receive Data This is the serial data from the remote equipment which is transmitted to the terminal.
Pin 4	RTS — Request to Send Controls the direction of data transmission. In full-duplex operation an "on" sets transmit mode and an "off" sets non-transmit mode. In half-duplex operation an "on" inhibits the receive mode and an "off" enables it.
Pin 5	CTS — Clear to Send Signal from the modem to the terminal indicating ability to transmit data. An "on" is "Ready" and an "off" is "not ready."
Pin 6	DSR — Data Set Ready Signal from the modem to the terminal. An "on" condition indicates that the modem is ready.
Pin 7	SGND — Signal Ground
Pin 8	CD — Carrier Detect An "on" indicates reception of a carrier from the remote data set; "off" indicates no carrier is being received.
Pin 20	DTR — Data Terminal Ready: "on" connects the communication equipment to the communications channel; "off" disconnects the communications equipment from the communications channel.
Pin 22	RI — Ring Indicator An "on" indicates that a ringing signal is being received on the communications channel.

Table 2: *Designations of pins on the DB-25 connector when used for communication with an RS-232C interface system and description of corresponding signals.*

Address Range	SW1	SW2	SW3	SW4
08 thru 0B	Closed	Closed	Closed	Closed
18 thru 1B	Closed	Closed	Closed	Open
28 thru 2B	Closed	Closed	Open	Closed
38 thru 3B	Closed	Closed	Open	Open
48 thru 4B	Closed	Open	Closed	Closed
58 thru 5B	Closed	Open	Closed	Open
68 thru 6B	Closed	Open	Open	Closed
78 thru 7B	Closed	Open	Open	Open
88 thru 8B	Open	Closed	Closed	Closed
98 thru 9B	Open	Closed	Closed	Open
A8 thru AB	Open	Closed	Open	Closed
B8 thru BB	Open	Closed	Open	Open
C8 thru CB	Open	Open	Closed	Closed
D8 thru DB	Open	Open	Closed	Open
E8 thru EB	Open	Open	Open	Closed
F8 thru FB	Open	Open	Open	Open

Table 3: *Use of the switch-selectable address decoder allows the I/O address range to be varied over the range shown here according to the switch positions specified. (See figure 4.) Radio Shack software uses the address range hexadecimal E8 thru EB.*

Listing 1: *Part of the output generated during a timesharing session on the Source, in which the TRS-80 equipped with the COMM-80 and a modem was used as a terminal. The Source is a service of the Source Telecomputing Corporation of McLean, Virginia. The hard copy was produced by an LA36 DECwriter connected to the TRS-80 through the COMM-80.*

```
>DATA SYSCOM

************************* SYSTEM COMMANDS ***********************

COMMAND                    DESCRIPTION
-------                    -----------

BASIC           PROGRAM IN THE BASIC LANGUAGE.
CHAT            TALK TO ANOTHER USER ON THE SYSTEM.
CRTLST          DISPLAYS THE CONTENTS OF A FILE, STOPPING EVERY 24
                LINES TO GIVE YOU TIME TO CATCH UP. (TYPING A RETURN
                RESTARTS THE DISPLAY.)
```

Listing 1 continued on page 58

Listing 1 continued:

```
DATA          DISPLAYS CERTAIN TCA LIBRARY PROGRAMS AND DATA BASES.
DATE          GIVES TIME AND DATE.
DEL           DELETES A FILE.
DELAY         AUTOMATICALLY DELAYS OUTPUT TO PRINTING TERMINALS
              WITH SLOWLY RETURNING CARRIAGES.
ED            TEXT EDITOR.
ENTER         TYPE IN A FILE.
FILES         PRINTS THE NAME OF ALL YOUR FILES.
FORTRN        COMPILES A FORTRAN PROGRAM.
ID            SYSTEM SIGN-ON COMMAND.
INFO          DISPLAYS CERTAIN OTHER LIBRARY PROGRAMS AND DATA BASES.
LOAD          LOADS A FORTRAN PROGRAM.

MAIL          INVOKES THE ELECTRONIC MAIL PROGRAM.
NSORT         SORTS A FILE.
OFF           SIGNS A USER OFF THE SYSTEM.
PLAY          PLAYS COMPUTER GAMES.
POST          INVOKES THE CLASSIFIED AD/BULLETIN BOARD PROGRAM.
R             RUNS A LIBRARY PROGRAM.
TIME          DISPLAYS THE TIME USED FOR THE CURRENT SESSION.
RUN           RUNS A LOADED FORTRAN PROGRAM.
TY            LIKE CRTLST, BUT DOES NOT STOP AFTER 24 LINES.
USAGE         SUMMARY OF YOUR SYSTEM USAGE THIS MONTH.

NOTE: A COMPLETE LIST OF SYSTEM DOCUMENTATION AND PROGRAMMING
MANUALS MAY BE VIEWED BY TYPING DATA SYSDOC.

>ONLINE
CL0158      CL0619      TCA056      TCA088      TCA088      TCA290
TCA422      TCA434      TCA516      TCA569      TCA575      TCA612
TCA743      TCA766      TCA830      TCA914      TCB419      TCD011
TCD106      TCD140      TCD202      TCD248      TCD390      TCD419
TCD419      TCD437      TCD444      TCD459      TCD460      TCE052
TCE129      TCE201      TCE217      TCE274      TCE317
>DATA UPI
****************** UNITED PRESS INTERNATIONAL ******************

1) TO ACCESS THE UPI DATANEWS SYSTEM, SIMPLY TYPE "UPI" AND
PRESS "RETURN".
2) THEN SELECT "NATIONAL", "REGIONAL" OR "STATE" NEWS OR "FEATURES".
"FEATURES" INCLUDES MOST MAJOR NEWS SYNDICATES (NEW YORK TIMES,
UNITED FEATURES, ETC.) AS WELL AS SYNDICATED COLUMNISTS. FOR A
COMPLETE LIST OF FEATURES, INDEXED BY LOGICAL CONTENT, RETURN TO
THE "COMMAND" LEVEL, AND TYPE...........................UPI
3) SELECT FROM THE "GENERAL", "BUSINESS" OR "SPORTS" CATEGORIES;
 THE SYSTEM WILL THEN ASK YOU FOR ONE OR MORE "KEYWORDS".
```

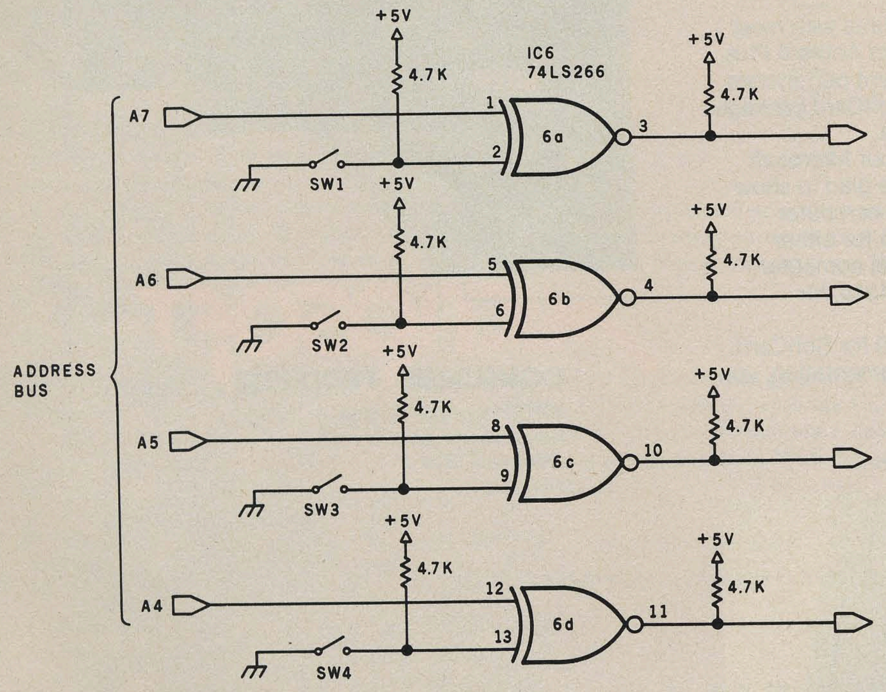


Figure 4: *By closing the proper switches, one of sixteen possible address ranges in the I/O-address space can easily be selected. The switches are optional; the desired address range may be hardwired. For complete compatibility with standard TRS-80 software, the hexadecimal address range E8 thru EB should be chosen.*

T_A / R_A	T_B / R_B	T_C or R_C	T_D / R_D	Data Rate	Clock Frequency
0	0	0	0	50	800 Hz
1	0	0	0	75	1200 Hz
0	1	0	0	110	1760 Hz
1	1	0	0	134.5	2152 Hz
0	0	1	0	150	2400 Hz
1	0	1	0	300	4800 Hz
0	1	1	0	600	9600 Hz
1	1	1	0	1200	19.2 kHz
0	0	0	1	1800	28.8 kHz
1	0	0	1	200	32.08 kHz
0	1	0	1	2400	38.4 kHz
1	1	0	1	3600	57.6 kHz
0	0	1	1	4800	76.8 kHz
1	0	1	1	7200	115.2 kHz
0	1	1	1	9600	153.6 kHz
1	1	1	1	19200	316.8 kHz

Table 4: *Chart to select data rates for the COM5016 data-rate generator. Transmission and reception rates may be set independently, according to the parameters specified here.*

Text continued from page 54:
article. Also, since this interface is software-compatible with existing TRS-80 hardware, there is no need to write your own driver routine. There are many sources, including the one listed with this article.

Using the COMM-80

Once you have an RS-232C port installed in your computer, a whole new world of peripherals opens up. The electronics industry has been turning out thousands of printers each year which use the RS-232C interface. For example, if you are interested in word processing, then you can attach a high-quality daisy-wheel printer to your TRS-80. Certain peripherals require a 20 mA current-loop interface; the required circuit is demonstrated in figure 8.

The most obvious application for the COMM-80 is to transform the TRS-80 from a mild-mannered personal computer into a full-fledged computer terminal. Photo 3 shows the system connected to a modem in actual use on the Source timesharing system. Listing 1 is a printout (from an LA36 DECwriter II also connected to the same serial interface) of typical user interaction on this national computer timesharing network. A look at

Text continued on page 62

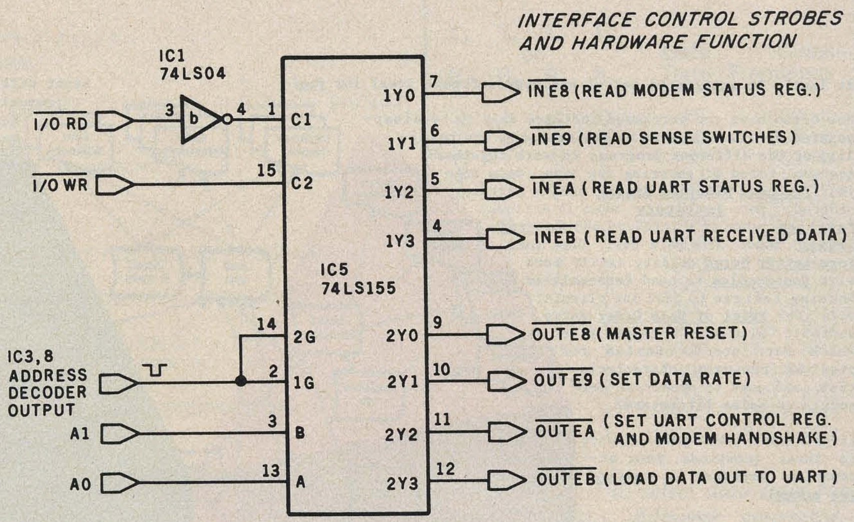
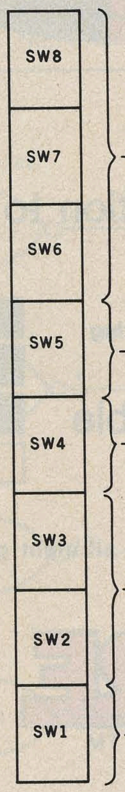


Figure 5: *Detail figure demonstrating interface-control strobes. The address decoder (made up of IC3 and IC6) can be set within the range of hexadecimal 08 to F8. TRS-80 compatibility requires a low address of E8. The output-strobe address notations presented refer only to this setting. Switch settings for other addresses are given in table 3.*

SENSE SWITCHES



DATA RATE	SW8	SW7	SW6
110	CLOSED	CLOSED	CLOSED
150	OPEN	CLOSED	CLOSED
300	CLOSED	CLOSED	OPEN
600	OPEN	CLOSED	OPEN
1200	CLOSED	OPEN	CLOSED
2400	OPEN	OPEN	CLOSED
4800	CLOSED	OPEN	OPEN
9600	OPEN	OPEN	OPEN

STOP BITS	SW5
1	CLOSED
2	OPEN

PARITY ENABLE	SW4
ENABLED	CLOSED
DISABLED	OPEN

WORD LENGTH	SW3	SW2
5 BITS	CLOSED	CLOSED
6 BITS	OPEN	CLOSED
7 BITS	CLOSED	OPEN
8 BITS	OPEN	OPEN

PARITY SELECT	SW1
ODD	CLOSED
EVEN	OPEN

Figure 6: *Programmable sense switches are read by the processor to allow preselection of UART options under program control. The correspondence of options and switches is illustrated here.*

← Circle 35 on inquiry card.

Circle 36 on inquiry card. →

Number	Type	+5 V	GND	−12 V	+12 V
IC1	74LS04	14	7		
IC2	74LS30	14	7		
IC3	74LS30	14	7		
IC4	74LS02	14	7		
IC5	74LS155	16	8		
IC6	74LS266	14	7		
IC7	74LS00	14	7		
IC8	74LS75	5	12		
IC9	74LS367	16	8		
IC10	MC1489	14	7		
IC11	MC1489	14	7		
IC12	MC1488		7	1	14
IC13	COM5016	2	11		9
IC14	74LS75	5	12		
IC15	74LS75	5	12		
IC16	74121	14	7		
IC17	AY-5-1013A	1	3	2	
IC18	74LS244	20	10		
IC19	74LS244	20	10		
IC20	74LS04	14	7		
IC21	74LS244	20	10		

Table 5: *Power supplies needed by the integrated circuits in the COMM-80.*

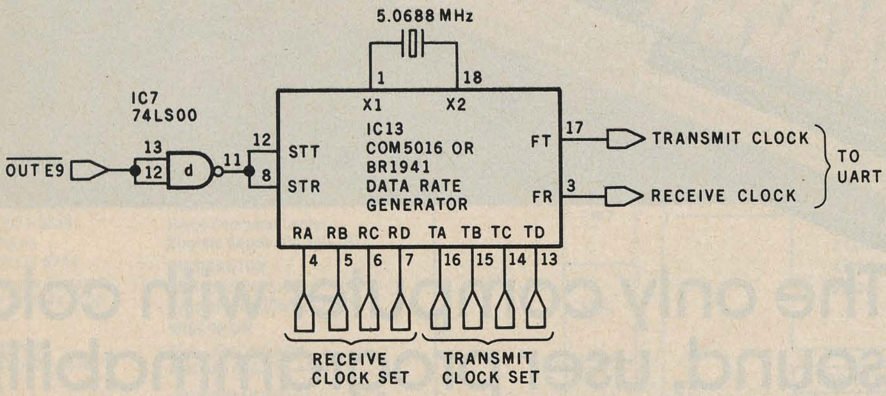


Figure 7: *The data-rate generator determines how fast data is sent and received. Transmission and reception rates can be set independently. The specifications for setting up the various possible data rates on the COM5016 are presented in table 4.*

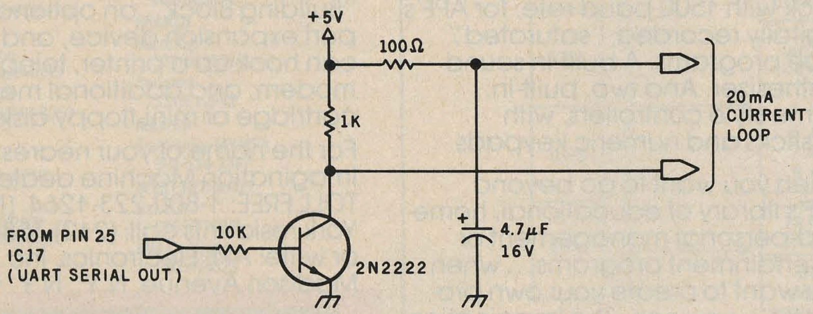
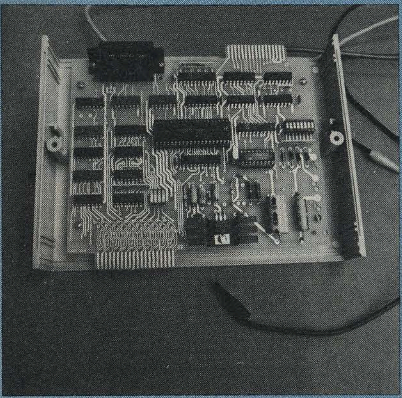


Figure 8: *Some peripheral devices (ie: a Teletype ASR33) must be connected by means of a 20 mA current-loop circuit; such a circuit that can be attached to the COMM-80 is shown here.*

Text continued from page 58:

some of the capabilities available through these networks might convince some people to use the network's facilities rather than spend thousands of dollars to build up an independent single-user system. At $2.75 per hour of connect time, it seems a reasonable alternative. For those of you wishing to contact me via the Source, my electronic-mail identification is TCE317. I welcome questions on this or any other topics that I might possibly be able to answer. ■



The COMM-80 is available assembled and tested in an attractive 20.95 by 15.24 by 6.35 cm (8.25 by 6 by 2.5 inch) enclosure, including expansion-bus connector and cable, an auxiliary 40-pin expansion-port edge connector, a 34-pin Centronics parallel printer port, RS-232C serial port with DB-25S connector, user's manual, power supply, and terminal software.

Price (including shipping):
$179.95 (New York residents please add appropriate sales tax).

Order from:
The MicroMint Inc
917 Midway
Woodmere NY 11595
(516) 374-6793

Next Month

Handheld remote controls for the home-control system as described in the January 1980 Circuit Cellar.