

Build this powerful serial-bus analyzer for a fraction of the cost of commercial units—and learn about the ever-popular 68705 microcontroller in the process.



RS-232 TERMINAL/MONITOR

STEVEN AVRITCH

HAVE YOU EVER BEEN FRUSTRATED by a problem with an RS-232 line? Inexpensive breakout boxes with five or six LED's suffice for solving simple problems, but they don't provide enough information to debug the serious kind. On the other hand, full-featured serial bus analyzers (SBA's) give you all the debugging information you need, but can cost close to \$1000.

However, you don't really have to spend that much. Now you can build a powerful RS-232 monitor that does most of what the expensive SBA's do, yet doesn't cost much more than a quality breakout box. In addition, this project can also be used as a portable, battery-operated terminal. Features are summarized in Table 1.

Hardware design

The heart of the project is Motorola's MC68HC705C8 single-IC microcontroller, a 40-pin DIP containing built-in PROM, RAM, serial and parallel I/O ports, timer, and clock.

The monitor requires two serial receivers: one for the TXD line and one for the RXD line. Because the microcontroller has only one

built-in serial port, a second one has been implemented in software. However, the second port still requires a ± 12 -volt RS-232 interface. The schematic is shown in Fig. 1, and the wiring diagram is shown in Fig. 2.

A Maxim MAX232 RS-232 transceiver (IC2) provides the transmission portion of the interface; it converts TTL signals from the microcontroller to RS-232 levels. Unlike most RS-232 transceivers, which require separate +12- and -12-volt power supplies, the MAX232 has built-in charge pumps that generate the required voltages from a single 5-volt supply. A standard 1489 device (IC4) converts incoming ± 12 -volt signals to TTL levels.

Switch S1 is a DPDT unit that selects terminal or monitor mode. In terminal mode the CPU controls the logic levels of both lines. However, in monitor mode, all lines from the primary port (J1) are directly connected to the secondary port (J2), in which case all signals pass straight through and the device simply monitors TXD and DTR.

Other notable components include XTAL1 and the keyboard. The crystal is a standard, readily available device; it must have a

frequency of exactly 2.4576 MHz in order to generate the correct baud rates. The keyboard is a surplus unit from an old TI-99/4 personal computer; the keyboard is inexpensive and readily available through many suppliers.

A beeper may be connected to pin 19 (port B7) of the microcontroller. The beeper will sound whenever the monitor detects a bell character (ASCII 07 or Cntl-G). The beeper must be TTL-compatible (meaning that it must be powered from a 5-volt supply and draw less than 3 mA).

The low power consumption of the microcontroller allows it to operate over a wide range of input voltages (8–15 volts DC); IC3 provides voltage regulation. The output of IC3 is 6 volts; diode D2 drops voltage even further, to about 5.3 volts. D1 provides reverse-polarity input protection.

Operating modes

The RS-232 Terminal/Monitor has four basic modes of operation, as shown in Table 2. You use the keyboard to select mode, as well as a variety of operational parameters. In use, you must place S1 in the Terminal position when in terminal mode, and in the Monitor position when in any of the three display modes.

The terminal-mode display is similar to a dumb ASCII terminal, except that the display is limited to forty characters by two lines. The CPU converts keyboard characters to ASCII, then transmits them over the serial link. Conversely, received characters are displayed in ASCII on the LCD screen.

The Display Bits mode continuously displays the status of the six primary RS-232 signals, as shown in Fig. 3-a. The Display ASCII mode shows activity on the TXD and RXD lines. TXD data appears in ASCII on line one and RXD data on line two (Fig. 3-b). The Display Hex mode is similar, except that each character appears in hexadecimal format, as shown in Fig. 3-c. (A period "." indicates that the line was inactive when a character was received on the other line.)

TABLE 1—FEATURES

- 40-character × 2-line LCD display
- Full keyboard
- Selectable baud rate (300 - 19,200)
- Selectable protocol (number of data and parity bits)
- Programmable scroll rate
- Recall of last two lines displayed
- 8- to 15-volt DC power (9V battery is perfect)
- CMOS design for low current drain and long battery life

TABLE 2—MODES OF OPERATION

Terminal	The unit acts as a simple RS-232 "dumb" terminal. The unit can be connected anywhere a dumb terminal is needed.
Display Bits	Displays status (high or low) of the six primary RS-232 lines (TXD, RXD, RTS, CTS, DSR, DTR).
Display ASCII	Displays TXD (line 1) and RXD (line 2) activity in ASCII format.
Display Hex	Like Display ASCII mode except hexadecimal display.

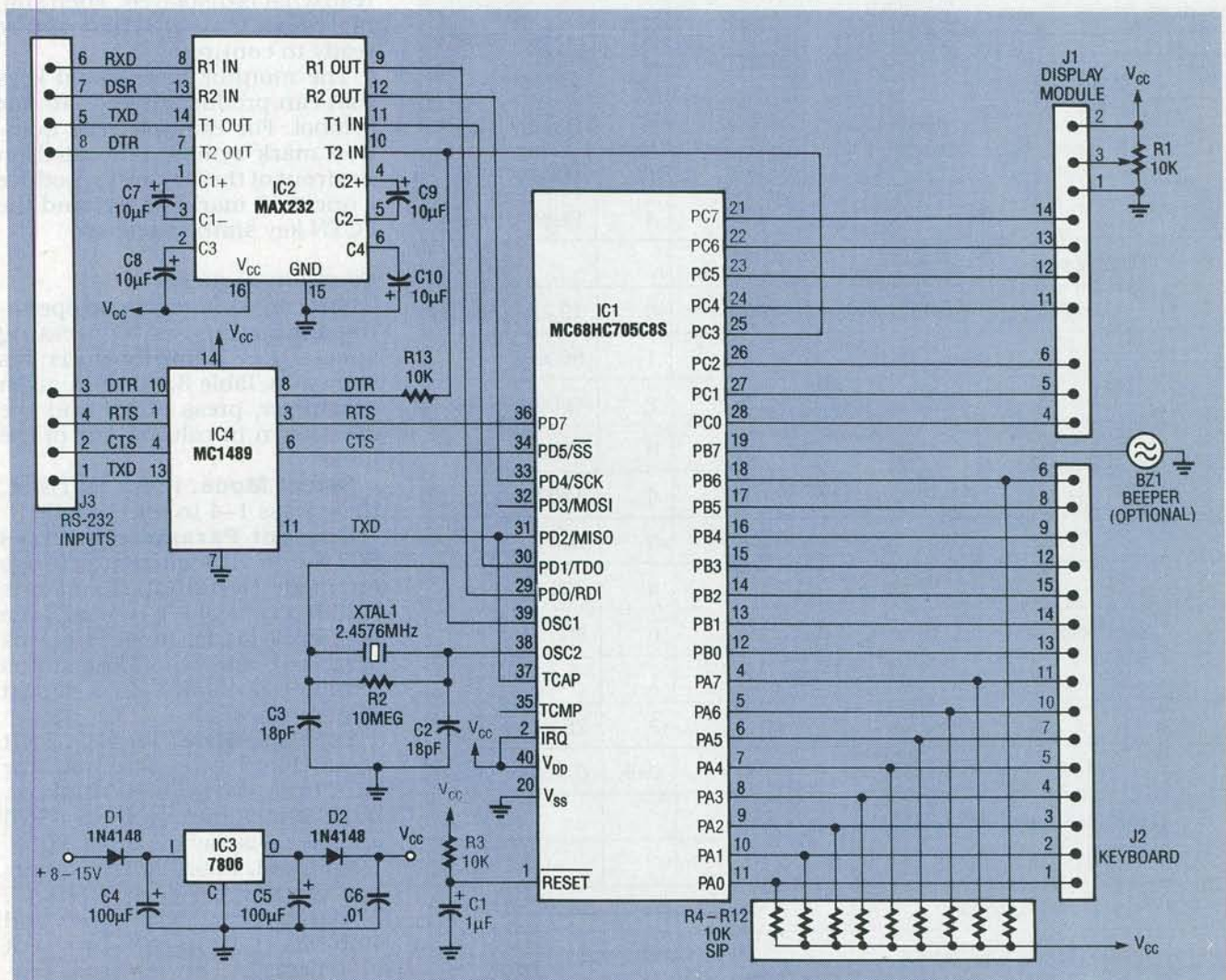


FIG. 1—SCHEMATIC OF THE TERMINAL/MONITOR. The keyboard is a surplus unit from a TI 99/4 computer.

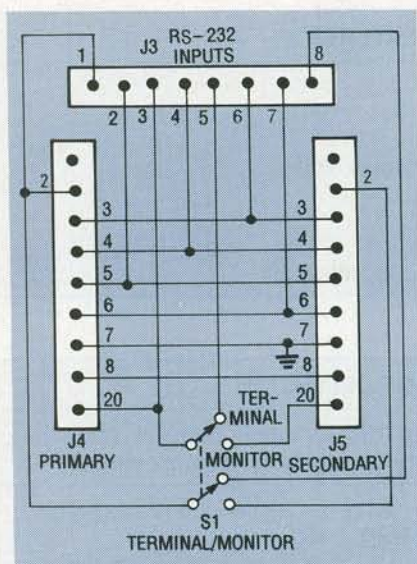
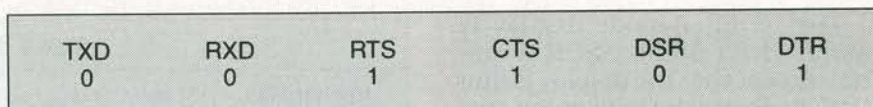
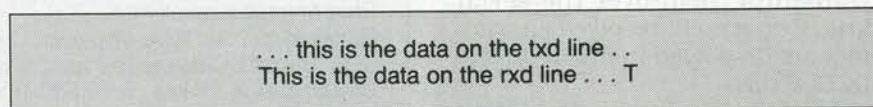


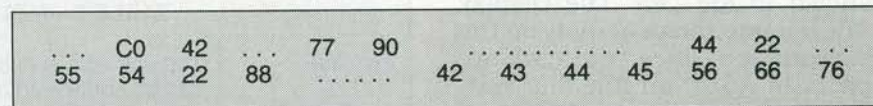
FIG. 2—WIRING DIAGRAM for the terminal/monitor. Connectors J4 and J5 are standard DB-25 males.



a



b



c

FIG. 3—DISPLAY SAMPLES. The Display Bits mode is shown in a; the ASCII mode is shown in b, with TXD on top and RXD on the bottom; and the hex mode is shown in c, again with TXD over RXD.

During operation in ASCII or hex mode, you can press Cntl-S to halt input temporarily and read what is displayed. Then simply press Cntl-Q when you're ready to continue.

The monitor has several keys that can produce more than one symbol. For example, the question mark symbol is located on the front of the "I" key. To produce a question mark, press I and the FCTN key simultaneously.

Function requests

You can change several operating characteristics by pressing special key combinations, as shown in Table 3. To set a given parameter, press FCTN and the key shown in column one of the table.

Select Mode. Press FCTN-M, then press 1-4 to select mode.

Current Parameters. Press FCTN-0 to view current settings for mode (terminal), baud rate (1200), protocol (7E1), scroll rate (0), scroll mode, linefeed status (enabled), and on/off-line status (on-line). (Default values shown in parentheses.)

Set Baud Rate. Press FCTN-1 to set baud rate. The monitor runs from 300 to 19,200 bits/sec, but the maximum rate in ASCII and Hex Display modes is 9600.

Protocol. Press FCTN-2 to define the number of data bits (7, 8), parity (Even, Odd, None), and stop bits (1, 2) in each byte sent or received. The terminal currently supports three popular formats: 8N1, 7O1, and 7E1.

TABLE 3—FUNCTION REQUESTS

1st	Description	2nd	Description
M	Select mode	1	Terminal
		2	Display Bits
		3	Display ASCII
		4	Display Hex
0	Display current parameters		
1	Set baud rate	0	19,200
		1	9600
		2	4800
		3	2400
		4	1200
		5	600
		6	300
2	Set Protocol	0	8N1
		1	7N1
		2	7E1
3	Set scroll rate delay	0-9	0=off, 9=max
4	Toggle scroll/two-line mode		
5	Enable/disable linefeed		
6	Recall last two lines		
L	Toggle local/online modes		
H	Display help information		

PARTS LIST

All resistors are 1/4-watt, 5%

R1—10,000 ohms, PC-mount potentiometer

R2—10 megohms

R3, R13—10,000 ohms

R4-R12—10,000 ohms, SIP

Capacitors

C1—1 μ F, 16 volts, radial electrolytic

C2, C3—18 pF, ceramic disk

C4, C5—100 μ F, 25 volts, axial electrolytic

C6—0.01 μ F, ceramic disk

C7-C10—10 μ F, 16 volts, radial electrolytic

Semiconductors

IC1—MC68HC705C8 CMOS microcontroller

IC2—MAX232 5-volt RS-232 transceiver

IC3—MC7806CT 6-volt regulator, TO-220 case

IC4—MC1489A RS-232 receiver

D1, D2—1N4148 switching diode

Other Components

XTAL1—2.4576 MHz

J1—16-pin 0.1" dual-row header

J2—30-pin 0.1" dual-row header

J3—8-pin 0.1" dual-row header

J4, J5—DB25 connector, male

S1—DPDT switch

BZ1—TTL-compatible beeper (Radio Shack #273-65 or equivalent)

Display—40 \times 2 line LCD display module (Hitachi LM018L or equivalent)

Keyboard—Surplus TI-99/4 (48-key, 15-pin connector)

Note: The following are available from Simple Design Implementations, P.O. Box 9303, Forestville, CT 06010. (203) 582-8526:

- Complete kit including everything in parts list, IC sockets, PC board, instructions, and schematic—\$75 + \$5 S/H
- Preprogrammed MC68HC705C8, instructions, and schematic—\$28.00 + 2.50 S/H

- Software on 5 1/4-inch IBM-compatible floppy disk—\$15.00 + \$2.50 S/H
- PC board only—\$13 + \$2 S/H.

CT residents please add 8% sales tax.

Scroll Rate Delay. Press FCTN-3 to vary scroll rate. Data may scroll by faster than you can read it; variable scroll rate allows you to reduce scrolling speed by disabling the DTR line for a while after receiving each carriage return. Scroll-rate delay may vary

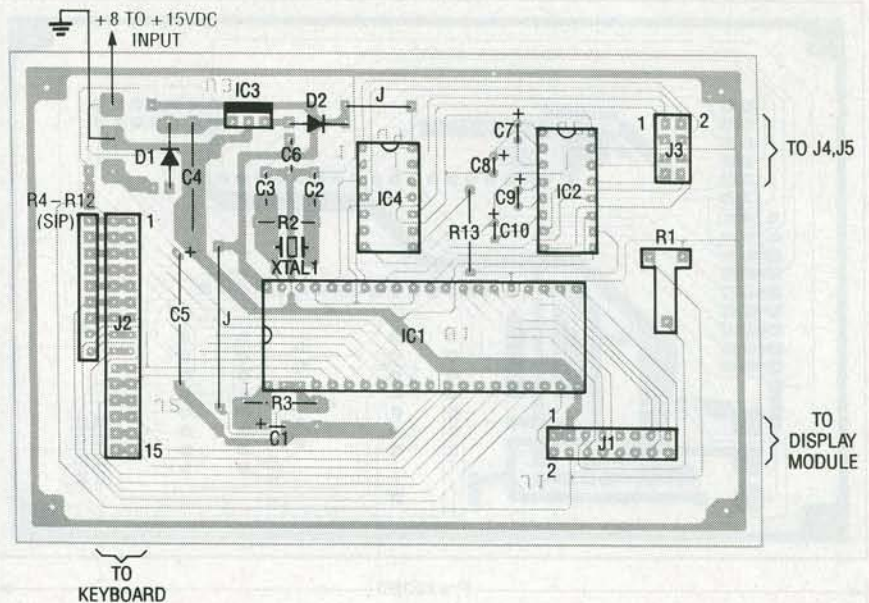


FIG. 4—PARTS PLACEMENT DIAGRAM. Mount all parts on the board as shown here.

from 0 (no delay) to 9 (maximum delay—about three seconds). The device connected to the monitor must recognize DTR and stop sending data when it is low. This works only in terminal mode.

Two-line/Scroll Mode. Press FCTN-4 to switch between the continuous scrolling and two-line modes. In scroll mode the terminal scrolls incoming data at the scroll rate set by FCTN-3. In two-line mode the terminal stops the display every time both display lines are filled. You must then press the space bar to continue. This feature works only in terminal mode.

Enable/Disable LF. Press FCTN-5 to switch between responding to and ignoring incoming linefeed characters. Some terminals, modems, and host systems issue a linefeed (LF) in addition to a carriage return (CR) at the end of every line. Extra linefeeds cause a blank line to be displayed on the screen, hence make it difficult to read. Tell the monitor to ignore extra linefeeds by selecting the LF Disabled mode. This feature works only in terminal mode.

Recall Last Two Lines. Press FCTN-6 to review the last two lines that scrolled off the screen. Press any key to return to the current two lines. Note that communication is disabled via the DTR line when displaying the last two lines. This feature works only in terminal mode.

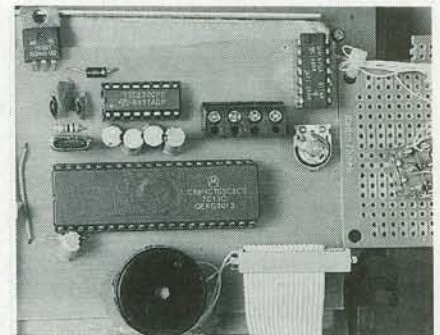


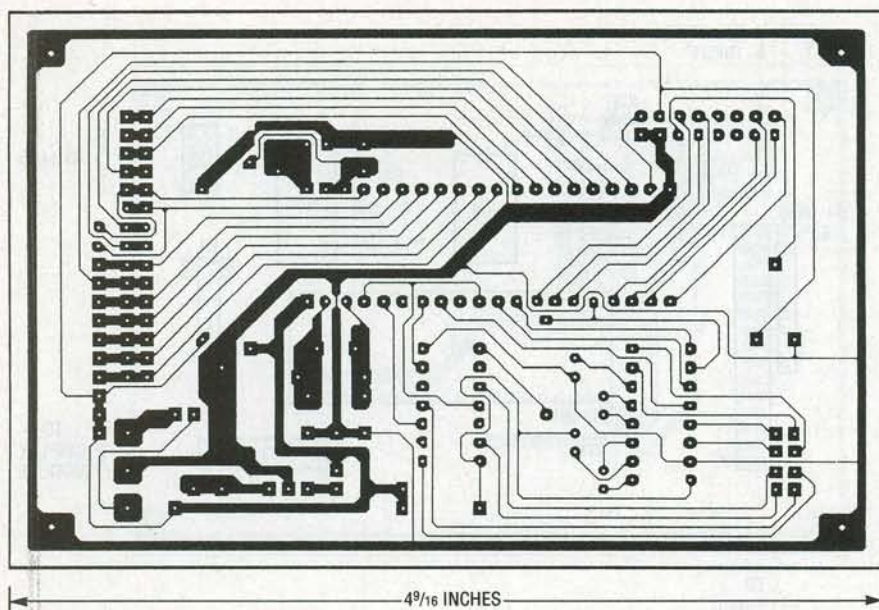
FIG. 5—THE AUTHOR'S PROTOTYPE. After applying power to the unit a brief sign-on message should appear. The display contrast can be adjusted via potentiometer R1.

Help. Press FCTN-H to view a brief summary of the function associated with each. You are able to view this screen from any operating mode.

Local/On-line Mode. Press FCTN-L to switch between local and on-line modes. In local mode, RS-232 communications are disabled; characters typed on the keyboard appear on the LCD display immediately. In on-line mode, RS-232 communications are enabled. Characters typed on the keyboard are transmitted over the serial link and are not displayed until they are echoed by the host system. This feature works only when used in the terminal mode.

Interfacing

To use the device as a dumb



USE THIS FOIL PATTERN if you would like to etch your own board.

TABLE 4—TERMINAL VOLTAGES

Pin	Description
2	-7 to -12 VDC
3	0 VDC
7	Check for continuity to ground
8	0 VDC
20	Press Cntl-S and Cntl-Q to toggle between +7 to +12 and -7 to -12 VDC

ASCII terminal, be sure to place switch S1 in Terminal position, then make sure that the unit is operating in terminal mode, and then enter the correct operating characteristics (baud rate, protocol, etc.). Last, connect the unit to the host system via a standard 25-conductor cable. If the unit doesn't seem to work, you may need to reverse pins 2 and 3 using a null-modem cable or adapter.

To use the device as a monitor, connect it in series with the two devices (host and terminal, host and modem, etc.), place S1 in Monitor position, and choose one of the terminal modes. Then all you have to do is set protocol, baud rate, etc.

Software

The MC68HC705C8 software consists of an assembly-language

program; unfortunately the listing is too long for publication. The program must be assembled and the resultant object code must be burned into the microcontroller's built-in EPROM. If you don't have facilities for assembling the program and burning it in, don't worry—you can purchase a pre-programmed MC68HC705C8 from the author; take a look in the parts list for details.

If you wish to modify the source code (or maybe just look it over), you can download it from the RE-BBS (516-293-2283, 1200/2400, 8N1) as a file called SMART232.SRC, or you can otherwise order it directly from the author.

Construction

You can build this project using wire-wrap or a PC board. A foil pattern is provided if you'd like to make your own board; you can also purchase a ready-to-use board from the source mentioned in the parts list if you don't want to make your own.

Using Fig. 4 as a guide, assemble the PC board starting with the passive components. Then mount sockets for the IC's, connectors J1-J3, and don't forget the two jumpers. Check your work carefully, correct any mistakes, and then insert the IC's into the sockets. Connect the cables from the keyboard and LCD module. Then make the connec-

tions between J1, J2, and the PC board. A photograph of the author's prototype unit is shown in Fig. 5.

After verifying all wiring, apply power to the unit; a brief sign-on message should appear on the display. Adjust Contrast potentiometer R1 for best effect. Now set switch S1 to the appropriate position, and set the desired mode and operating characteristics using the Function keys. That's all there is to it!

Troubleshooting hints

If the sign-on message does not appear on the display on power-up, the most likely cause of the problem is a wiring error. The error is probably in the power or ground lines (IC1, pins 20 and 40), the oscillator lines (IC1, pins 38 and 39), the reset line (IC1, pin 1) or the display interface lines (pins 1-6 and 11-14 of the LCD display).

If the terminal comes up with the initialization message but does not seem to communicate, check the keyboard and RS-232 driver. Check the keyboard by pressing FCTN-0 to display system parameters. If they don't appear, check the keyboard wiring. If the terminal does display system parameters, press FCTN-L to put the terminal in local mode and then test the keyboard by pressing every key and checking for the correct letter on the display. If the keyboard checks out, the problem is probably located somewhere in the RS-232 driver, in the cabling, or in the RS-232 line itself.

Remove the connector from the terminal's primary port, set the DPDT switch to terminal mode, and check the voltages on the terminal's primary port. They should be close to the values shown in Table 4. If any of the values do not agree with Table 4, check the wiring of the RS-232 driver (IC2).

Wrapping up

The author housed his prototype in a custom-built enclosure that resembles a miniature RS-232 terminal. You could use an off-the-shelf metal or plastic case just as well. Just keep in mind that, however you package it, this terminal/monitor is a useful and fun project. R-E