$1.25 ■ NOV. 1979

# Radio-Electronics®

THE MAGAZINE FOR NEW IDEAS IN ELECTRONICS

**Computer project**
## TRS-80 BREADBOARD

**Roundup**
## TELEPHONE DIALERS

**How to**
## TUNE UP TAPE BIAS

**All about**
## DESOLDERING TECHNIQUES

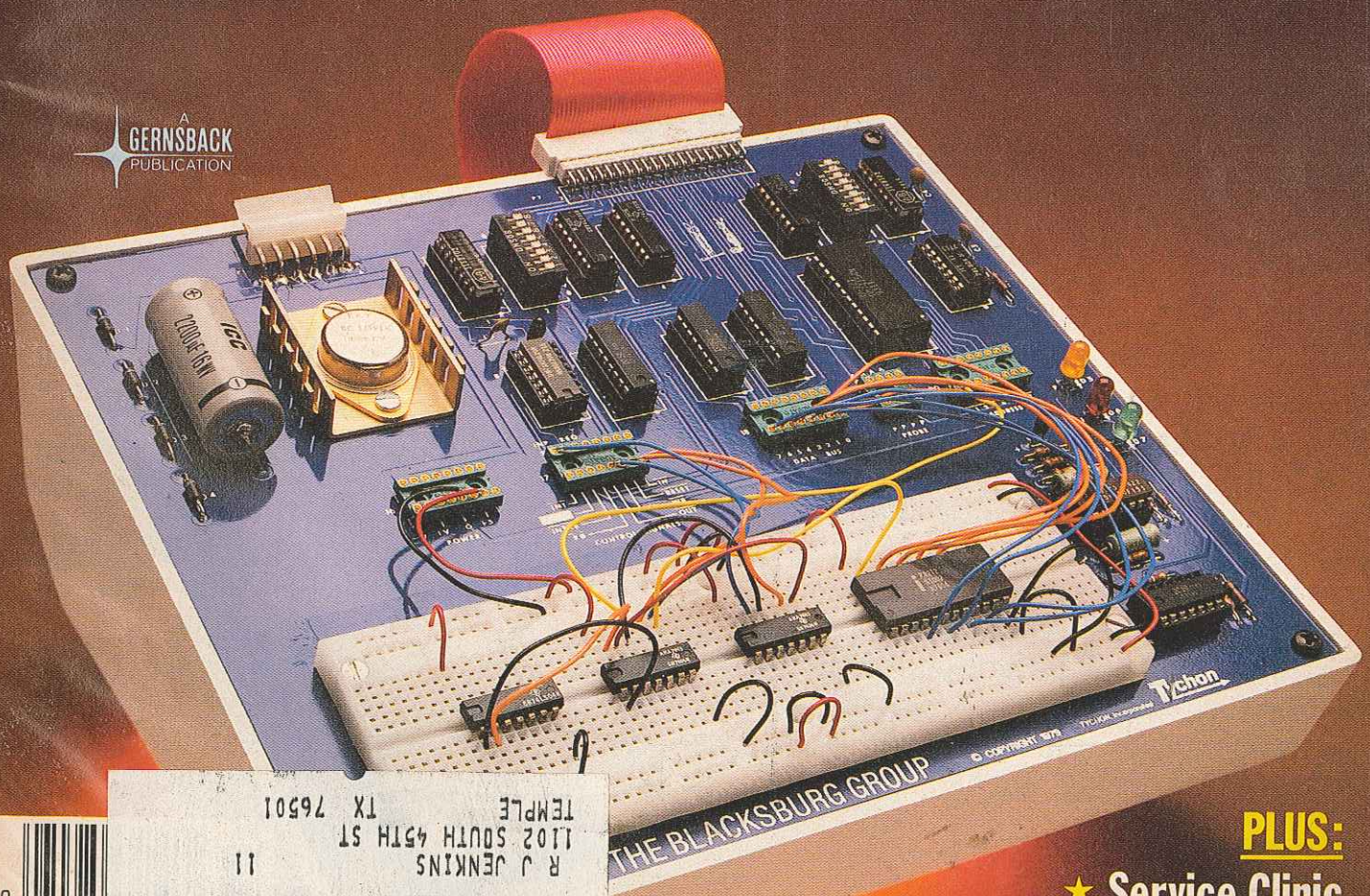
**How to**
## PROTECT OP-AMPS

**Build it for your phone**
## MUSIC ON HOLD

**Digital logic in**
## VCR CIRCUITS

- **EQUIPMENT REPORTS**
- **HOBBY CORNER**

A GERNSBACK PUBLICATION

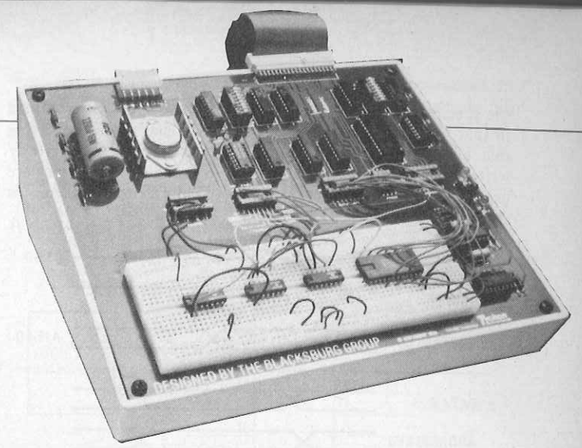





**PLUS:**

★ Service Clinic
★ Communications Corner
★ 2 R.E.A.L. Sound Lab Test Reports

BUILD THIS



# TRS-80 BREADBOARD

*Now you can design, prototype and safely connect various circuits to your Radio Shack TRS-80 computer. This device plugs into the TRS-80 and contains all the buffer circuitry, power supplies and a solderless breadboard to make experimenting and prototyping easier.*

**JON TITUS, CHRIS TITUS, and DAVID LARSEN**

MANY OF THE PRESENT *TRS-80* MICROCOMPUTER USERS ARE INterested in data-processing applications, whether they be for game-playing, business-forecasting, inventory-management, payroll-computation, or educational use. Many users are also interested in having their *TRS-80* microcomputer do something outside of the computer itself, in the so-called "real world." Typical real-world applications for the *TRS-80* involve monitoring or testing external devices and performing some sort of control operation based upon the result of a data processing step. Many articles have been written about programming in BASIC, and many BASIC programs have been developed for the *TRS-80*. Very little has been written, so far, that describes the *TRS-80*'s signals, and how they may be used to interface the computer to the real world so that it can perform useful tasks beyond data processing.

### The TRS-80

The *TRS-80* computer is available in a number of configurations. Since the *Level-I* BASIC does not include any general-purpose input/output (I/O) commands, it is useless for the control of I/O devices. The *Level-II* BASIC interpreter does recognize four general-purpose I/O commands, so it will be the basis for our discussion. We will discuss these commands shortly, but first, we need to take a look at the *TRS-80*'s signals that are provided for the control of external devices.

If you haven't done so already, you may wish to remove the plastic hatch cover at the left-rear corner of the keyboard's housing. This cover provides access to the RESET pushbutton, and to a double-sided, 40-pin male edge connector. The edge connector provides the means for connecting external devices to the *TRS-80*'s bus. The signals that are available along with their abbreviations and functions, are listed in Table 1. You will note that some of the signal abbreviations are followed by an asterisk. This designates that the signal is normally a logic one, and that the action described takes place when the signal is in the logic zero state. We will use the more familiar "bar" notation throughout this article, since it is a standard. Thus, the *TRS-80* bus signal, IN*, will be noted as $\overline{IN}$.

At this point, the four important bus signals for interfacing are $\overline{IN}$, $\overline{OUT}$, $\overline{WR}$ and $\overline{RD}$, along with the 16-bit address bus (A15–A0), and the eight-bit data bus (D7–D0). Some readers that are familiar with the S-100 bus will wonder about the use of a single data bus, instead of the "split" buses found in some early computer systems. The *TRS-80* uses a single eight-bit bi-directional data bus to communicate between I/O devices, memories and the Z80 microprocessor IC.

The flow of data on the buses is carefully synchronized by the Z80 through the use of the $\overline{IN}$, $\overline{OUT}$, $\overline{WR}$ and $\overline{RD}$ control signals. Individual memory locations and I/O devices are specified by the 16-bit address bus signals A15–A0. The *TRS-80*'s address bus and its control signals are all uni-directional; that is, the signals are all generated by the Z80 microprocessor, and transmitted to external devices.

### I/O devices and memory

At this point, we need to distinguish between the I/O devices, and the memory locations that may be addressed by the *TRS-80*. In each case, special signals are generated to control and synchronize the flow of information between the memory IC's, or I/O devices, and the *TRS-80*. An understanding of these signals, and how they affect external devices is very important. In fact, interfacing the *TRS-80* would be impossible if we did not know how to use these signals.

The *TRS-80* addresses a specific memory location through the use of a 16-bit address on the address bus. This gives the *TRS-80* the ability to directly address up to $2^{16}$ or 65,536 different memory locations. Once the 16-bit address is specified either the $\overline{WR}$ (write) or the $\overline{RD}$ (read) control signal goes low (logic zero), indicating to the memory IC that it is to either "write" the eight-bit value currently present on the data bus into

the specified location, or that it is to transfer the byte contained in that location onto the data bus for a "read" operation. Note that even though your computer may only have 4K of read/write memory associated with it, all 16 of the address lines are used. A timing diagram for the read and write operations is shown in Fig. 1.

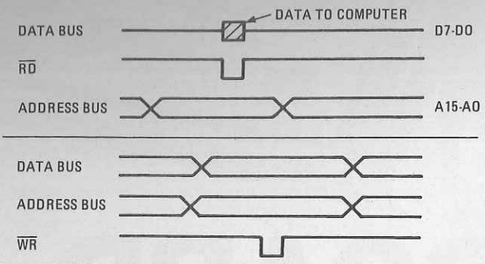



**FIG. 1—TIMING RELATIONSHIPS for the memory-read (RD) and memory-write (WR) operations.**

The *TRS-80* controls external I/O devices in a similar manner, using the address bus, the data bus, and two control signals. Instead of using the read and write signals, two separate signals are used just for the control and synchronization of I/O devices. These are the IN and OUT signals. The IN signal synchronizes the flow of information *to* the *TRS-80*, while the OUT signal synchronizes the flow of information *from* the *TRS-80* to external devices.

While the address bus is used to address either a memory location or an I/O device, the use of the bus differs in each case. While 65,536 memory locations may be addressed, the *TRS-80* can only address 256 I/O devices, since only address lines A7 through A0 are available for device addressing. This isn't much of a limitation, however, since few users will have more than a few devices connected to their computer system. During I/O device addressing, the remaining address lines, A15 through A8, are used by the Z80 IC to transfer other information, but it is unimportant for I/O device addressing and for normal I/O device interfacing. A typical timing diagram for I/O device addressing and synchronizing is shown in Fig. 2.

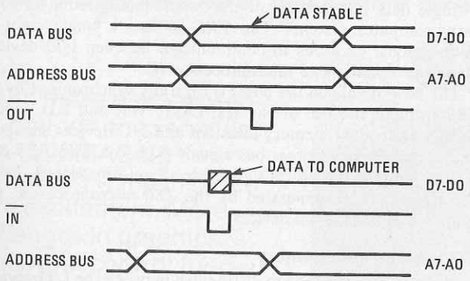



**FIG. 2—TIMING RELATIONSHIPS for I/O device data transfers. The IN and OUT signals are about 1.3 μs.**

At this point, there are four areas that we must cover before we can interface the *TRS-80*. These are: device address decoding, device selecting, I/O ports, and software (programming).

### Address decoding and device selection

These two topics are covered together, since it is difficult to separate them. To address and select an I/O device, the information on address lines A7 through A0, must be decoded so that the addressed device is only selected when the proper address is present on these lines. Since the address bus lines serve two purposes—the addressing of memory locations, and the addressing of I/O devices—some additional information is necessary so that external devices can distinguish between memory addresses and I/O device addresses. The IN or the OUT pulse can be gated with a decoded address signal to provide this distinction.

**TABLE 1—TRS-80 EXPANSION CONNECTOR SIGNALS**

| PIN | SIGNAL | DESCRIPTION |
|---|---|---|
| 1 | RAS* | Row address strobe for dynamic memories † |
| 2 | SYSRES* | RESET signal for resetting I/O devices |
| 3 | CAS* | Column address strobe for dynamic memories † |
| 4 | A10 | Address bus bit |
| 5 | A12 | Address bus bit |
| 6 | A13 | Address bus bit |
| 7 | A15 | Address bus bit (MSB) |
| 8 | GND | Ground |
| 9 | A11 | Address bus bit |
| 10 | A14 | Address bus bit |
| 11 | A8 | Address bus bit |
| 12 | OUT* | OUT signal for the control of output ports |
| 13 | WR* | WR write signal for the control of memory-writing |
| 14 | INTAK* | INTAK interrupt acknowledge signal |
| 15 | RD* | RD read signal for the control of memory-reading |
| 16 | MUX | Dynamic memory multiplexer control † |
| 17 | A9 | Address bus bit |
| 18 | D4 | Data bus bit |
| 19 | IN* | IN signal for the control of input ports |
| 20 | D7 | Data bus bit (MSB) |
| 21 | INT* | INT interrupt signal to Z-80 chip |
| 22 | D1 | Data bus bit |
| 23 | TEST* | Test input † |
| 24 | D6 | Data bus bit |
| 25 | A0 | Address bus bit (LSB) |
| 26 | D3 | Data bus bit |
| 27 | A1 | Address bus bit |
| 28 | D5 | Data bus bit |
| 29 | GND | Ground |
| 30 | D0 | Data bus bit (LSB) |
| 31 | A4 | Address bus bit |
| 32 | D2 | Data bus bit |
| 33 | WAIT* | WAIT generates a processor wait state † |
| 34 | A3 | Address bus bit |
| 35 | A5 | Address bus bit |
| 36 | A7 | Address bus bit |
| 37 | GND | Ground |
| 38 | A6 | Address bus bit |
| 39 | +5V | +5 volts (DO NOT USE) |
| 40 | A2 | Address bus bit |

NOTES:
† These signals are not used for interfacing, and no connections should be made to these pins.
* Signals active in the logic-zero state
Viewed from the rear of the keyboard housing, pin 1 is in the upper left-hand corner, with odd-numbered pins across the top, while pin 2 is in the lower left-hand corner, with even-numbered pins across the bottom.

A typical gating scheme is shown in the schematic diagram in Fig. 3. In this simple example, an eight-input gate has been used to detect the proper combination of ones and zeros on the address bus corresponding to the decimal address of 24. While the address output from the NAND gate has been provided, this is not very useful, by itself. The useful signals are those that result from gating the device address output of the NAND gate with function pulses IN and OUT as shown in Fig. 3. Since these are the pulses that are used to select and control external devices, they are called *device select pulses*. In all cases, external devices are controlled through the combination of an address and a function pulse. These are generally gated together to generate a device select pulse.

You should not be surprised to see that the device address that was generated in Fig. 3 has been used with both an input device and an output device. Since the IN and the OUT pulse are never generated simultaneously, each device address may be



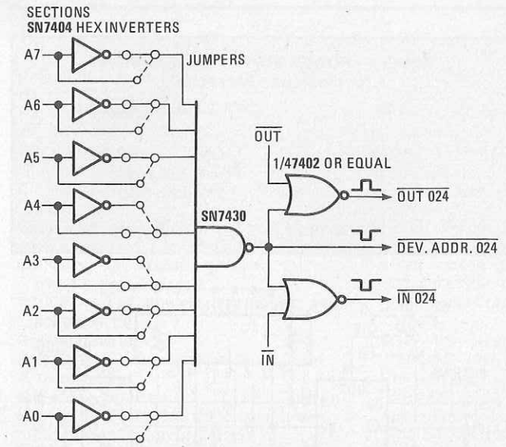



**FIG. 3—A SIMPLE gate-based device address decoder and device selector circuit.**

used for both an input device, and an output device. In many cases, the two devices with the same address may not be related in function. In general, though, two input devices are *never* assigned the same device address, and the same holds true for output devices. This avoids bus conflicts, much as having people talk in turn avoids the problems that would occur if they all talked at the same time.

### I/O ports

Input/output ports, or I/O ports, are easily constructed. In most cases, output ports are simply latches that have their inputs connected to the data bus, and their outputs connected to the device that is to receive the transmitted data. The latches are triggered by an output device select pulse. A typical output port is shown in Fig. 4. Latches are used as output ports, since they

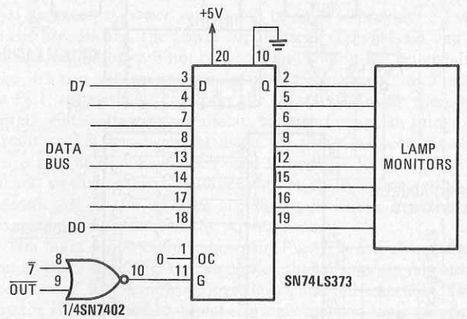



**FIG. 4—LATCHED OUTPUT PORT used to drive lamp monitors or other indicators. (0 = ground or logic zero.)**

can be triggered with short pulses that transfer information from their inputs to their outputs. When the pulse has been removed, the transferred information remains at the outputs until it is updated with new information or until power is removed from the system. In this way, the information is available to the output port for a long time, in fact, as long as the output device needs it, it is there. If latches were not used, the information would only be present for a short period (less than 2 microseconds in the *TRS-80*), hardly enough time for an external device to print a character, close a relay, turn on a heater, open a valve, or take any meaningful action.

Input ports are generally three-state buffers such as the DM8095, or SN74365 devices. These three-state buffers have a third state that allows them to appear electrically disconnected from the device to which their outputs have been connected. In this unselected, or high-impedance mode, these devices do not present any outputs to the lines to which they are connected, making them ideal for use on the *TRS-80*'s data bus. Since they are disconnected most of the time (when not transferring data), they do not interfere with the normal operation of the other input ports on the bus, or with output ports and memories. A typical input port is shown in Fig. 5.

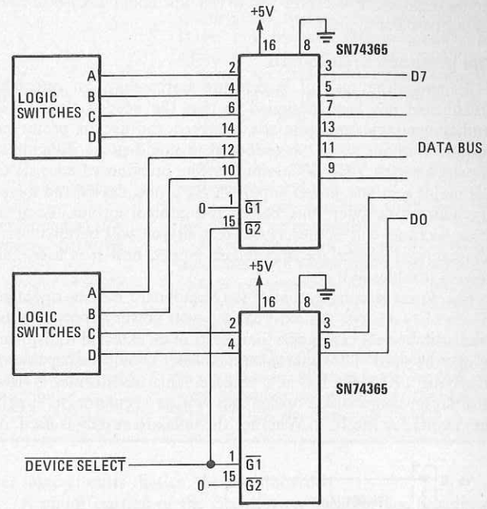



**FIG. 5—THREE-STATE input port using 74365 IC's.**

A device select pulse is used to turn on the three-state buffers so that they can transfer the information that is present at their inputs, through to their outputs, and onto the *TRS-80*'s data bus. The actual transfer takes place when the IN pulse is in the logic zero state. Input ports are activated by a device select pulse that is a combination of a device address, and the IN function pulse. Now you should see why input devices are not assigned the same address. They would both try and use the bus at the same time, and the computer could not distinguish between either of the devices.

### Software

The transfer of data to and from I/O devices and the *TRS-80* is controlled through the use of the BASIC commands, INP and OUT. In each case, a device address must be specified as a part of the overall command. Thus, the command OUT 6,120 would transfer the decimal value 120 to the output port that has been assigned the decimal address 6. The command, A = INP(12) would set variable A, equal to the value that was input from input port 12. Since we do not know what value is to be input, a value is never incorporated within an INP command.

The device address associated with the INP and OUT commands is always expressed in decimal format, so you must convert these values to binary values to find out what bit patterns to expect on the address bus outputs. The value that is to be transferred by an output operation is also expressed as a decimal number. Since eight data bits and eight address bits are used in I/O device data transfers and in device decoding, the values for the data, and for the addresses, must always be within the range of zero to 255, inclusive. Other values will result in an error condition.

Variables may be specified within either the INP, or the OUT instructions, provided that they have been preset to a valid value prior to the use of the variable in an instruction. Thus, OUT X,Y, OUT 7,Z, OUT Q,10, M=INP(10) and L=INP(A) are all valid commands that will be correctly interpreted by the *Level II* BASIC. If fractions are specified in these instructions,

they are ignored. For example, if you attempt to output the value 6.125 to output port 7 with an OUT 7,6.125 instruction, the value 6 is transferred, with the fractional portion of the value being ignored.

This introductory information should serve to help you understand how the *TRS-80* may be easily interfaced to external devices. Since it is impossible to cover all of the basic interfacing techniques, we refer you to the new book, *TRS-80 Interfacing* (see Parts List).

### The interface breadboard

To make the task of interfacing fairly easy, an interface breadboard has been designed so that the needed signals are readily available and properly buffered for use in prototype design. Without such a breadboard, it could prove difficult to interface to the *TRS-80* computer. The breadboard consists of five major sections; power supply, logic probe, device and memory address decoder, bus buffer and control circuit. Each of these sections will be described, so that you will better understand how the interface breadboard works, how it is used and how to troubleshoot it.

The power supply section of the breadboard may be operated in one of two ways. An external +5-volt power supply may be used, as long as it can supply 1 ampere, or an external transformer may be used. The external transformer should be capable of supplying 12.6 volts AC at 1 ampere. This transformer is used with an on-board diode bridge and voltage regulator, to supply the 5 volts for the IC's. Whether the onboard supply is used, or

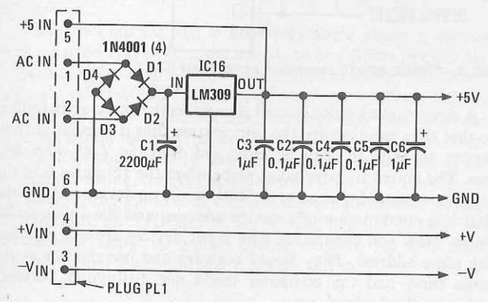



**FIG. 6—SCHEMATIC DIAGRAM of the breadboard's power supply.**

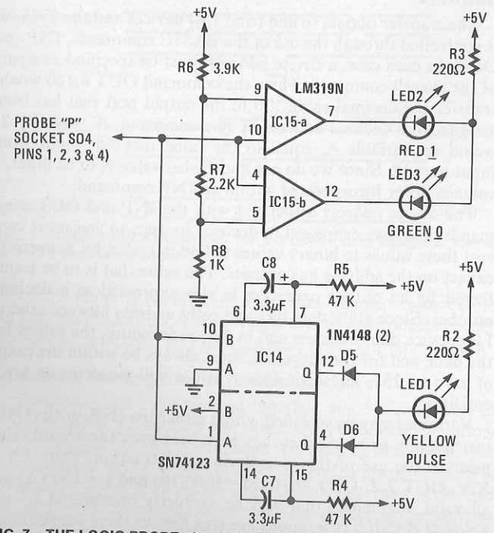



**FIG. 7—THE LOGIC PROBE circuit. A 74LS123 may be substituted for the 74123.**

**TABLE 2—POWER SUPPLY CONNECTIONS for the Power Socket, SO1.**

| PIN | VOLTAGE AVAILABLE |
|---|---|
| 7, 10 | +5 |
| 5, 12 | GND |
| 3, 14 | +V (External) |
| 1, 16 | −V (External) |

All other pins are unconnected

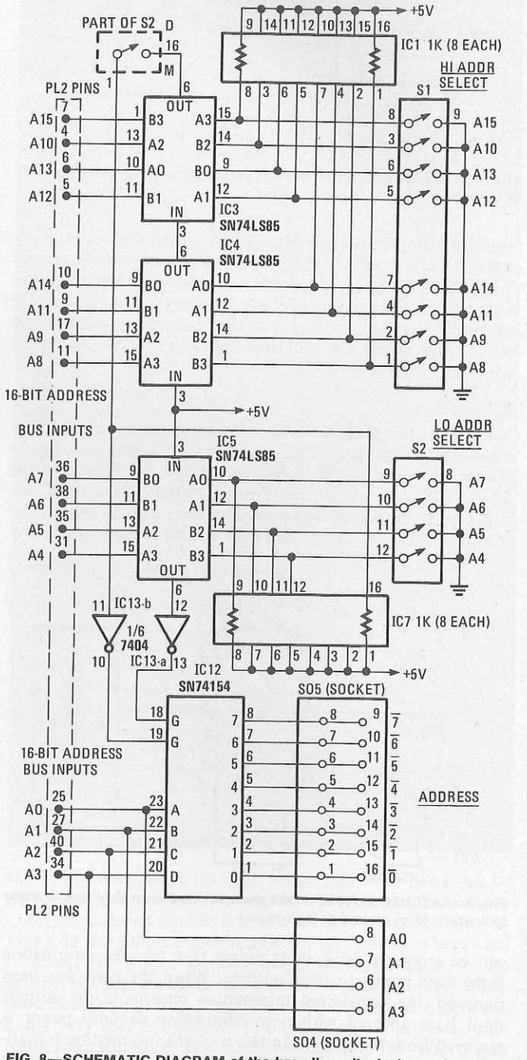



**FIG. 8—SCHEMATIC DIAGRAM of the breadboard's device and memory address decoding circuits.**

the external supply is used, the power supply for the breadboard is separate from the five-volt power supply that is used to power the *TRS-80*. The internal computer power supply just doesn't have the necessary power to drive the breadboard. A schematic of the power supply circuit is shown in Fig. 6.

If the on-board power supply is used, the 12.6-volt transformer is connected to pins 1 and 2 on plug PL1, and rectifier diodes D1–D4, filter capacitor C1, and the voltage regulator are all



## PARTS LIST

**Resistors ¼ watt, 5%**
R1, R8—1000 ohms
R2, R3—220 ohms
R4, R5—47,000 ohms
R6—3900 ohms
R7—2200 ohms
C1—2200 µF, 16 volts, electrolytic, axial leads
C2, C4, C5—0.1 µF, 50 volts, disc ceramic
C3, C6—1 µF, 35 volts, tantalum electrolytic
C7, C8—3.3 µF, 50 volts, electrolytic, axial leads
**Semiconductors**
IC1, IC7—16-pin resistor network (eight 1K resistors)
IC2, IC6—Not used
IC3—IC5—SN74LS85 quad comparator (do not substitute SN74L85)
IC8—SN74LS20 dual 4-input NAND gate
IC9—SN74365 or DM8095 three-state buffer
IC10, IC11—8216 non-inverting bus buffer (Intel or equal)
IC12—SN74154 4-line to 16-line decoder
IC13—SN7404 hex inverter
IC14—SN74123 or SN74LS123—dual retriggerable one-shot
IC15—LM319N dual comparator (14-pin package)
IC16—LM309K, voltage regulator, 5 volts, 1 amp.
D1–D4—1N4001 or equal, 50 PIV, 1-amp, diode
D5, D6—1N4148 or 1N4154 small-signal diode
LED1—yellow LED
LED2—red LED
LED3—green LED
S01, S02, S03, S05—High-quality 16-pin DIP socket (Augat 516-AG-10D or equal)
S04—high-quality 8-pin DIP socket (Augat 508-AG-10D or equal)
PL1—Molex right-angle 6-pin connector (PN 09-75-1061) optional. Requires 1 mating female housing (PN 09-50-7061) and 6 connector pins (PN 08-50-0106 or 08-50-0108)
PL2—40-pin right-angle jumper header, AP Products 923875R or equal
T1—transformer, 12.6 volts, 1 amp
**Miscellaneous**
Solderless breadboard socket. E&L Instruments model SK-10, AP Products model Superstrip II, Continental Specialties model EXP-300 or equal.
Cable assembly, 40-pin header on one end and 40-pin card-edge connector on the other—facing the same direction.

**The following parts are available from E & L Instruments, Inc., 61 First St., Derby, CT 06418.**

**Order No. 355-6125—Complete kit including PC board, case and all parts. Does not include interconnect cable. Specify 117V or 230V version. $139.00.**

**Order No. 355-6175—Interconnect cable assembly (connects breadboard to TRS-80 computer). $25.00.**

**Order No. 355-6100—Assembled 117-volt version. $185.00.**

**Order No. 355-6150—Assembled 230-volt version. $185.00.**

**Connecticut residents add state and local taxes as applicable.**

**A pre-drilled and etched PC board is available from Techniques, Inc., 235 Jackson St., Englewood, NJ 07631, for $24.50 postpaid. New Jersey residents add 5% sales tax.**

**Copies of the book *TRS-80 Interfacing* (published by Howard W. Sams and Co.) is available for $7.95 plus 79¢ for shipping and handling from Group Technology, Ltd., PO Box 87, Check, VA 24072**

installed. We suggest the use of a small heat sink with the voltage regulator. Be sure that it is tightly fastened to the voltage regulator and to the PC board. When the breadboard is used in this way, +5 volts are available at pin 5, and ground is available at pin 6. These connections may be used for external interfacing, if required. The actual use of a connector for PL1 is optional. You may wish to connect the power transformer or external power supplied directly to the interface breadboard without the connector.

If an external power supply will be used to provide +5 volts to the breadboard, the power supply parts (D1–D4, C1 and the voltage regulator) are not required and should not be installed. The +5 volt and ground connections are made to pins 5 and 6, on PL1, respectively, to power the system. To make the power supply voltages readily available for interfacing, an integrated circuit socket has been set aside for these connections. Two spare pins at PL1 have been used to connect to the POWER IC socket, so that external voltages may be easily connected to the system. All of the voltages are shown in Table 2, with their respective connections at the POWER socket.

The logic probe circuit shown in Fig. 7 is useful in helping you to determine the logics state of the various signals on the breadboard. It will indicate logic levels and pulse activity. Comparator IC15 is used to detect the logic one and logic zero logic levels, while dual monostable IC14 is used to detect and stretch pulses so that they may be easily observed. A green (logic ZERO), a red (logic ONE) and a yellow (PULSE) LED are used as indicators. The input to the probe is available at pins 1–4 on the socket at SOF. These inputs are all marked with a P. All of these four inputs are in parallel, and any one may be used. Since an SN74LS123 monostable is used as the pulse-stretcher, the input to the logic probe should be thought of as two low-power Schottky loads. You may wish to substitute an SN74123 monostable multivibrator, which will increase the input load to two TTL loads.

If you have an external logic probe, or an oscilloscope, you may not want to build the logic probe portion of the breadboard circuit. Since the remainder of the breadboard circuitry is independent of this section, it can be left out. We found that the logic probe portion of the breadboard is quite useful for testing and troubleshooting interface circuits as well as the various other logic circuits that can be breadboarded.

A major portion of the circuitry on the interface breadboard is devoted to device and memory address decoding, as shown in Fig. 8. The decoders can be operated as either memory address decoders, or as device address decoders, depending upon whether device or memory-mapped I/O will be used in your interface. In the device addressing mode, only address bits A7 through A0 are decoded. In the memory-mapped mode, all 16 address lines are decoded. In each case, addresses are absolutely decoded, meaning that all of the address bits in the respective group have been decoded. The decoding scheme used on the breadboard includes the necessary comparators and a decoder for both the device and memory mapped I/O modes.

In the device addressing mode, four-bit comparator IC5 is used to compare four preset address bits with four of the address bus lines, A7 through A4. The address bits are preset using a dual in-line switch package, S2. The positions are clearly marked, "4," "5," "6," and "7." You must be sure that the open or OFF position of the switch is in the logic one position (right-hand side). Pull-up resistors (in IC7) have been provided so that the open switch position provides a logic one to the comparators. Although a dual in-line resistor network has been specified, individual 1,000-ohm resistors may be used in its place. Use resistors with a 5% tolerance.

When an address match takes place between the preset address bits, and the address information on address bus, decoder IC12 is enabled and decodes the remaining four address bus bits, A3 through A0. Although the decoder IC12 can decode sixteen addresses, only eight have been implemented on the breadboard. The decoded address appears as a logic zero at its respective output, while all of the other outputs remain in the logic-one state.

Next month we will continue the discussion of the operation and applications of the TRS-80 interface breadboard and will provide illustrations showing the various types of buffering and control circuits. Following will be construction details including PC board foil patterns and a component layout. Also included will be schemes for testing the interface breadboard and various circuits that you can prototype such as a digital-to-analog converter circuit and a traffic-light simulator that is software controlled.

**R-E**