The POLY 88 Microcomputer System

The POLY 88 Microcomputer System brings to the user, in one compact package, the capability of developing programs and hardware as well as enjoying the interaction with computers.

The POLY 88 System uses a video monitor for display, a keyboard for input and cassette tape for storage. The system will also connect to a hard-copy terminal. Poly 88 hardware consists of an 8080 based CPU circuit card with on-board memory and I/O, video display circuit card with keyboard input port and graphics capability, and mini-cards that connect to the CPU board via ribbon cable for cassette or serial interface.

The Firmware Monitor is integral to the POLY 88 System. This 1024 byte program in ROM allows the user to display data on a TV screen, enter data into memory using a keyboard, read and dump data to the cassette interface in Kansas City format, and single step through a program while displaying the contents of each of the 8080's internal registers.

Prices: Basic kit including chassis, CPU and video cards — $595, $795 assembled. Cassette option — $90 kit and $125 assembled. 8K of RAM — $300 in kit form or $385 assembled.

Dealers: This system will sell itself.

All prices and specifications subject to change without notice. Prices are USA only. California residents add 6% sales tax. Prepaid orders shipped postpaid. BankAmericard and Master Charge accepted.

PolyMorphic Systems

737 S. Kellogg Avenue, Goleta, Ca. 93017
(805) 967-2351

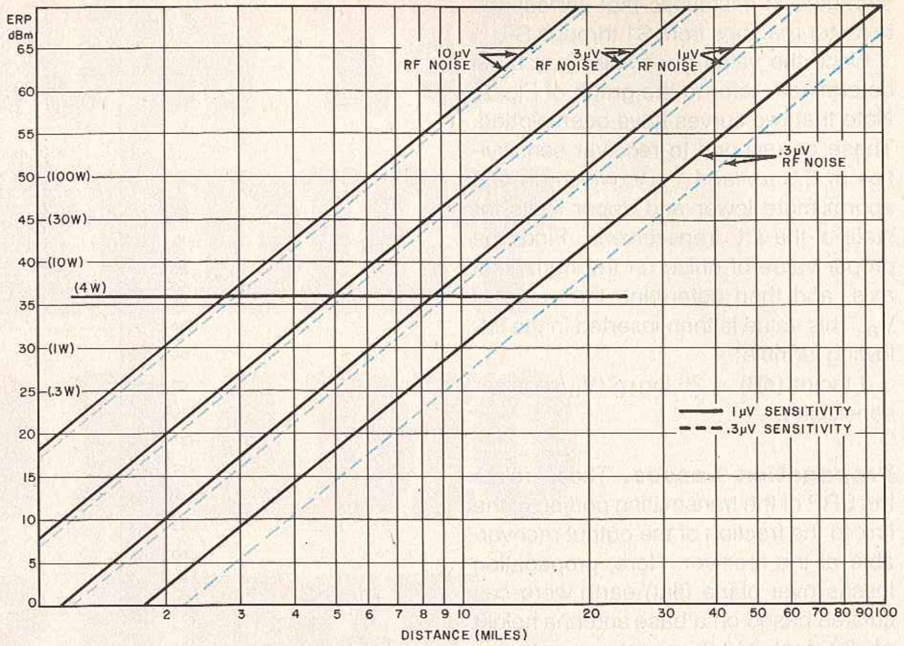


Fig. 3. Effective radiated power versus distance for two sensitivities and various r-f noise levels.

ceiver sensitivity is 1 µV for 10 dB S+N/N. Here's how the range is predicted.

Locate the 36-dBm line and proceed across until the 3-µV curve is reached. Then read the corresponding distance. In this case, it is five miles (8 km). Of course, you can work backwards and determine how much ERP is required for 5-mile coverage. Proceed down the 5-mile line until the 3-µV curve is reached, and then read the corresponding ERP value (36 dBm or four watts).

Further Comments. Mobile antennas are less efficient than base antennas, so it is obvious that mobile range will be more limited. It will typically be three miles (1.8 km). From Figs. 2 and 3, it can be concluded that, for ambient r-f noise levels above 1 µV (which is usually the case on the Citizens Band), an increase in receiver sensitivity of 333⅓%, say, from 1 µV to 0.3 µV for 10 dB S+N/N, reduces the required ERP only 10%. Only in extremely quiet r-f environments (under 0.5 µV), which probably don't exist on the Citizens Band in even the most rural areas, will there be any significant reduction in ERP required for a given distance.

In other words, a sensitivity of 1 µV for 10 dB S+N/N appears to be adequate for most applications. However, whether an S+N/N of 10 dB is sufficient for good intelligibility is altogether another question. If a more sensitive receiver (0.3 µV for 10 dB S+N/N) is used, providing a better ratio, say, 15 dB at 1 µV, the operator has a definite advantage in terms of audio quality or intelligibility.

Another area for consideration is the relative merit of a beam over an omnidirectional antenna. Unquestionably, a beam will allow you to reduce interference (and thereby improve intelligibility) from stations in other directions than the desired station. But let's limit this discussion to the relative merit in terms of range. The maximum permissible height for an omni is 60 feet (18.3 m) above ground, natural formation, or man-made structure. For a beam, the maximum allowable height is 20 feet (6.1 m). It can be shown that if the antenna height is halved, you will require 6 dB more power to reach the same distance.

Therefore, if you now have or are planning an omnidirectional antenna with unity gain mounted at 60 feet (18.3 m), and want to weigh the advantages of installing a high-gain beam and rotor system, consider this. You must subtract 6 dB from the beam's gain because of its lower height. In terms of the graph of Fig. 3, start at 36 dBm ERP, add the beam's gain, subtract 6 dB, and proceed across the graph until the appropriate r-f noise curve is reached. Then note the predicted range. If the gain of the beam is 6 dB, the range will be the same for the omni and the beam. If gains of 9 dB or more are not available, or if the beam will not be mounted considerably higher than 30 feet (9.15 m) above average ground level, the omnidirectional antenna at 60 feet (18.3 m) above ground level is the better choice. ◇

Popular Electronics®

Special Focus on Digital Electronics

Digital techniques play a leading role in electronics today. Switching, coding and logic function methods are not difficult to grasp. And equally important, they make it possible to create many interesting electronics circuit designs. In this special supplement, many of these concepts are illustrated, including projects for a digital auto fuel gauge and a shirt-pocket digital stopwatch. In addition, the first article gives many helpful hints on how the electronics hobbyist should go about choosing a microcomputer from the many available today.

How to Select a Hobbyist Microcomputer

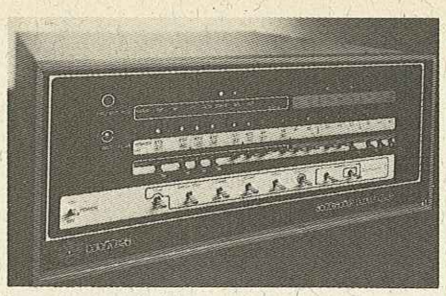
BY STEPHEN B. GRAY
Senior Editor

DECIDING which microcomputer to buy is quite a challenge. Not only are there several dozen on the market, but they're available in a wide range of prices, with a variety of features and peripherals, and with several different MPU's (microprocessor units), such as the 8080, 6800, 6502, F8 and 6100, among others.

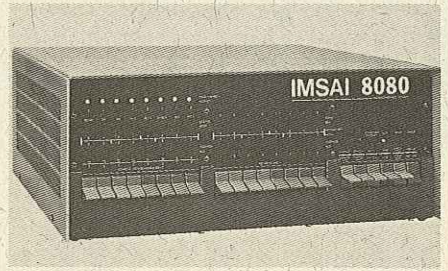
One of the easiest ways to narrow down your choice of a microcomputer is to decide which basic type is best for your own use. To do this requires a breakdown of microcomputer types, as in the following paragraphs.

1. Box With. The best-known type of microcomputer looks very much like a minicomputer: a box with a bunch of switches and lights on the front panel. Two hobby computers of this type are the MITS Altair 8800b and the Imsai 8080. This microcomputer type is the most widely used among hobbyists, with the widest choice of peripherals and memory expansion boards.

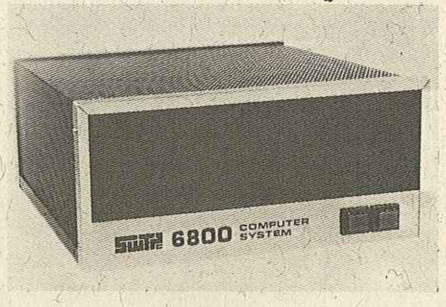
2. Box Without. The second type of computer is also a box, but with a bare



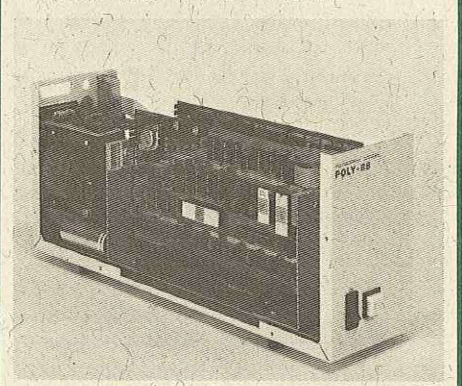
Type 1:
MITS Altair 8800b



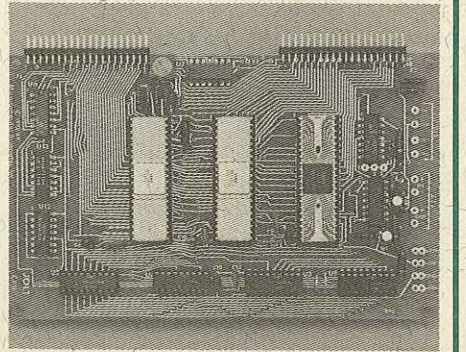
Type 1:
Imsai 8080



Type 2:
Southwest Technical 6800



Type 2:
PolyMorphic Poly-88



Type 3:
Microcomputer Assoc. JOLT

minimum of switches and lights. An example is Southwest Technical Products' 6800, which has only two switches, for power and reset. Only the power switch has a light.

There are two main differences between Type 1 and Type 2 computers. First, with a Type 1, you can load short programs and operate the computer manually, reading the results on the LED display. Obviously, you can't do this with a Type 2 machine. The second difference is in loaders. When you turn on any microcomputer, you can't put a program into memory until a bootstrap loader is inserted first. This acts as a set of signs to guide the program to the right places in memory. With most of the Type 1 computers, you have to load the bootstrap instructions by hand, using the front-panel switches. With most of the Type 2 computers, which have stored loaders, all you do is press RESET and the loader is inserted automatically.

However, just because a microcomputer has a full set of switches and lights isn't always a sign that the loader must be inserted by hand. The MITS 680b is a good example of a micro with switches, lights, *and* a bootstrap loader in permanent memory which doesn't "drop out" when the power is turned off.

Note too, that not all micros have the same internal expansion capability. This sometimes accounts for differences in size and, naturally, influences price too.

3. PC Board. The third main type of microcomputer consists of a printed-circuit board without input or output. These were first introduced for use in commercial products, or for engineering evaluation, and many are still sold for such purposes. Several recent ones are being sold mainly to hobbyists. The best-known of Type 3 are the Jolt and SC/MP.

All the computers described up to this point have neither separate input nor output. So unless you enjoy loading programs via front-panel switches (if your microcomputer has them) and reading out the program results from the front-panel lights, you'll need some more hardware. This means a keyboard for putting data into the computer, and a more sophisticated readout for checking that the program is correct and for reading the results. There is already a wide variety of keyboard terminals available, and the most common readout today is a TV screen.

4. All-On-One-Board. For those who want a complete computer with less sophisticated inputs and outputs than teletypewriter and video monitor, there are many everything-on-one-board microcomputers. This type includes a small keyboard and some form of readout. The readout is sometimes individual LED's, but is usually segmented alphanumeric display. The KIM-1 is the best known of these, although several others are coming up fast. Two units come with a case, the Infinite UC 1800 and the Hamilton/Avnet Pacer. They have built-in power supplies, whereas most of the others don't.

Just about the least expensive Type 4 microcomputer for the hobbyist who wants to learn the basics is the Elf, featured as a construction project in the August and September 1976 issues of POPULAR ELECTRONICS. This hardware and software trainer, with RCA COSMAC MPU, toggle-switch input, hex LED display, 256 bytes of RAM, four input lines, and a latched output line, costs about $80 to build. Memory is expandable at minimum cost.

Nearly all the computers of this type are on a single pc board; two exceptions are the Mike 3 and Mike 8, from Martin Research. Each is a stack of several boards, separated by spacers, with the keyboard and display on the console board at the top, CPU on a second board, memory on a third, etc. This modular approach permits using different CPU boards, either for the 8080A MPU, Z-80, or 8008.

The keyboard almost always has 16 hex keys for entering programs in machine language plus various control keys. These boards are popular among people who want to learn what computers are all about, at minimum cost. For those who want to go further, more memory can be added, as can be peripherals such as a full keyboard and/or a printer, to start with.

5. All-In-One Box. Another type of computer that doesn't require buying a keyboard or TV set has a built-in keyboard and CRT, such as the various models of the Sphere. Although this type of computer is expensive, it does have everything you'd need for almost any type of programming. However, you are locked into the integrated input/output system much as you are for an FM tuner when it's built into an FM receiver. A printed output can be added on, as it can to almost any hobby computer. The cost of a simple printer has decreased substantially. For example, Southwest Technical offers one in kit form for $250, and Electronic Products Associates has an assembled printer for $450.

Among the computers of this type, the Intecolor 8001, with an 8-color CRT, is unique. This adds an extra dimension to graphics and to just about anything you want to put on the screen.

Intelligent Terminals. A step up from most hobby terminals, which can be used only as input/output devices, is the intelligent terminal. With one of these, you can write, edit and store programs for transmission to a larger computer directly, or to a time-sharing computer over a telephone line (using a modem device).

Any hobby computer with a keyboard, RS-232 or 20-mA current-loop interface, and enough memory can be used as an intelligent terminal, of course. All you need are the right programs. The SOL terminal from Processor Technology provides these programs in the form of pre-programmed PROM's, called "Personality Modules," at three levels.

One module allows simple terminal operations. A second-level module makes SOL an editing terminal. The top-level module transforms it into an intelligent terminal as well as a stand-alone computer.

Programming. An important factor in choosing a hobby micro is to decide at which level you want to program. How much memory your computer has will determine its price and also what kind of programming language you can use.

With only a few hundred bytes of memory, you'll usually be restricted to programming in machine language, or to short programs in assembly language. Some people enjoy working in machine language, down at the bit level, using instructions such as 00111010, which is the 8080 code for "load the accumulator with the contents of the specified memory address."

But working with machine language may be boring to all but real "computer freaks." Also, you can easily make mistakes that aren't at all quickly apparent when working with only zeroes and ones. With a little more memory, though, you can program in assembly language. In order to do this, you must load an assembler into your computer's memory. This is a program that translates the assembly-language instructions, such as LDA, into machine language; in this case, 00111010.

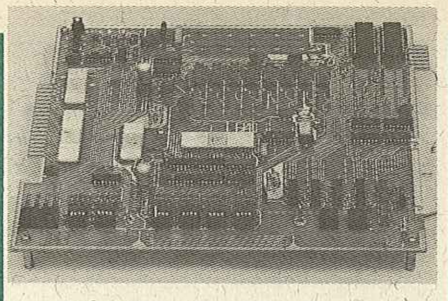
In assembly language, you use *mnemonic* names for program instructions; these are easy-to-remember abbreviations, such as LDA for "load accumulator" and MOV for "move the contents of the accumulator to register B." To add one number to another in 8080 assembly language takes eleven steps, including five mnemonics and three pairs of address codes. Address codes are in pairs because addresses take up two *bytes*; that is, groups of 8 bits. (An 8080 machine can address 2^{16} memory locations.)

The program for adding two numbers consists of these steps: load the accumulator with the number to be found at, say, memory address 128. Then take what's in the accumulator and move it to register B. Next, load the accumulator with the number to be found at address 129, and add the contents of register B to what's in the accumulator. Take the sum that's now in the accumulator, and store it at address 130. If you've previously stored numbers at addresses 128 and 129, this program will add them together and put the sum in 130. Actually, you can use any memory addresses you want, instead of 128, 129 and 130, as long as you don't select an address that's higher than the maximum address in your system.

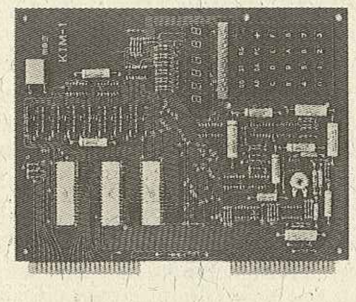
If you'd rather write programs with mnemonics such as LDA, MOV and STA, then you need, as previously noted, an assembler program, which is also stored in memory along with your own program. For example, the MITS Altair 8800b assembler takes up 5500 bytes of memory, so if you're going to be writing programs of any real length, you'll need at least 8k bytes of memory.

But suppose you're not really interested in programming for programming's sake, but rather in what the program will do for you. If so, then you might prefer BASIC, a high-level language that will do in a single instruction, LET C = A +B, what requires eleven assembly-language instructions to do. That single BASIC instruction will store the sum of A and B in memory location C, which is determined by the BASIC interpreter all by itself, thus taking care of much housekeeping. Should you want to show the answer on your TV screen, or print it out, simply write PRINT C. Or you can combine both steps by writing a single BASIC instruction, PRINT A+B.

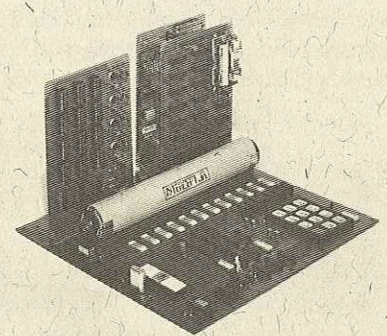
Just about all high-level-language programs written for hobby computers are in BASIC (there are several varieties of BASIC, each with minor differences). You can get several BASIC interpreters for some computers, requiring 4k, 8k or 12k bytes of memory. The 8k and 12k versions offer more features than the 4k BASIC. The 8k BASIC interpreter, which turns LET C = A+B into machine language, takes up 5.7k bytes of memory in



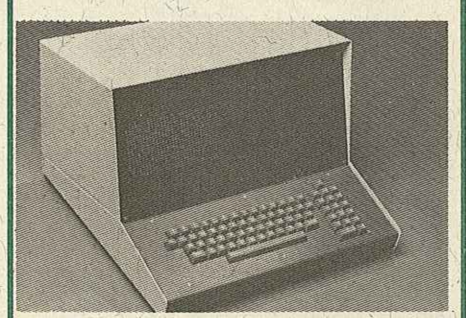
Type 3:
HAL MCEM-8080



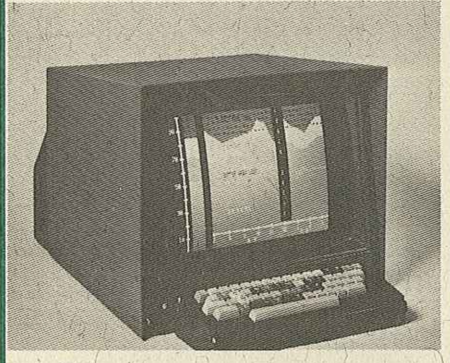
Type 4:
MOS Technology KIM-1



Type 4:
Intersil Intercept Jr.



Type 5:
Sphere 310



Type 5:
Intelligent Sys. Intecolor 8001

the Altair 8800b, for example. MITS specifies it as requiring 8k bytes of memory so that you'll have 2.3k bytes for your own use in writing programs. Incidentally, although many serious computer hobbyists will be satisfied with 8k or 16k of memory, many hobby computers can be expanded to 65k.

Hobbyist Bus. The MITS Altair 8800 microcomputer was the first to be sold in large volume, and set a bus standard that some other micro manufacturers have followed. This standard is based on the 100-pin bus, to which all the Altair 8800 boards are connected in common. Consequently, many other manufacturers of CPU boards, memory boards, and peripheral boards have tailored their designs so they will plug into the Altair 8800, and also into the busses of several other computers that use the Altair bus structure, including the Imsai 8080, the PolyMorphic Poly-88 and Processor Technology's SOL. As a result, there are more boards for CPU and memory, and for peripherals such as printers, disk drives, graphics devices, cassette memory, etc., available to owners of computers using this bus.

There are other bus lines, of course. For example, the Southwest Technical 6800 computer utilizes a different bus, with a growing number of boards for it.

Price. Of the five basic types of hobby computers, the cheapest is the pc-board-only, with which you need a power supply, an input, and an output. The complete-computer-on-a-board follows, and usually requires the addition of only a power supply. Next is a box-type computer with which you'll need input and output peripherals. With a box-with-CRT type, which gives you the most equipment at a minimum cost, on a one-shot basis, you already have the peripherals, unless you also want a printed output.

Let's look at what it costs to buy a microcomputer with enough memory for the various levels of programming, taking into consideration several of the best-selling micros. The basic computer, without memory can range from $212 to $840 in kit form, $500 to $1100 assembled. These wide ranges are due to some of the computers being full-featured models, others being "barebones" types. Not many 1k memory boards are being offered any more; they used to be about $120 kit, $160 assembled. Using just a basic 8-bit computer and 1k of memory, you could write programs containing up to about 500 instructions, if you don't mind flipping switches for hours and hours.

Stepping up to assembly language, you'll need two 4k memory boards, each of which run from $125 to $167 kit, $279 to $325 assembled. If you buy the Altair 8800b and the two 4k boards at the same time, you get a "software package" for $75, which includes the assembler and several other programs. The Southwest Technical 6800 editor/assembler package is $14.95.

You'll need some sort of input/output, of course. To connect your computer to your TV set requires an interface that can cost from $40 to $148 kit, $60 to $180 assembled. For keyboard input, you may be able to use the same interface if it can handle two serial I/O devices. You'll also need a keyboard terminal, such as a Model 33 Teletype. This, however, is expensive, costing between $769 and $1500 new, depending on what features you select. With an ASR33, you can enter a program from either the keyboard or punched paper tape. Or you could get a hobby unit, such as Southwest Technical's CT-1024 terminal. With this, the program is entered via the keyboard. The CT-1024 kit, less cabinet and power supply, is $175; there are various options available.

Programs can be entered into the computer much faster by using a cassette. To enter the 8k BASIC interpreter into the Altair 8800b takes 12 minutes from paper tape, 4 minutes from cassette. Typical cassette interfaces range from $35 to $138 kit, $65 to $195 wired. You can buy a 4k BASIC interpreter for $4 to $60 depending on manufacturer. The 8k BASIC interpreter ranges in price from $8 to $75.

Which to Pick? A major question to answer is: will you be satisfied with programming in assembly language, or do you want to program in BASIC? If you're sure you'll be happy with assembly language, you have two types of computers to choose from. The least expensive is the all-on-one-board computer, Type 4, such as the KIM-1 or 6502 Familiarizor, where the only extra to buy is a power supply except for a couple that have it built in. The other choice is Type 3, the pc board with no I/O, such as the SC/MP. To use one of these, you'll need a power supply, keyboard, and some sort of output, either a printer or a TV receiver or video monitor.

If you're more interested in programs than in computers, and want BASIC, you have three choices. The Type 1 computers, including the Altair 8800b and Imsai 8080, require interfaces and peripherals for input and output, as do the Type 2 machines, such as Southwest 6800 and Poly-88. You can add these at any time. You'll need no additional hardware if you buy a Type 5 computer, which has both CRT and keyboard.

Summarizing, you must decide what you plan to do with the microcomputer now and in the future, as well as what your bankbook can tolerate.

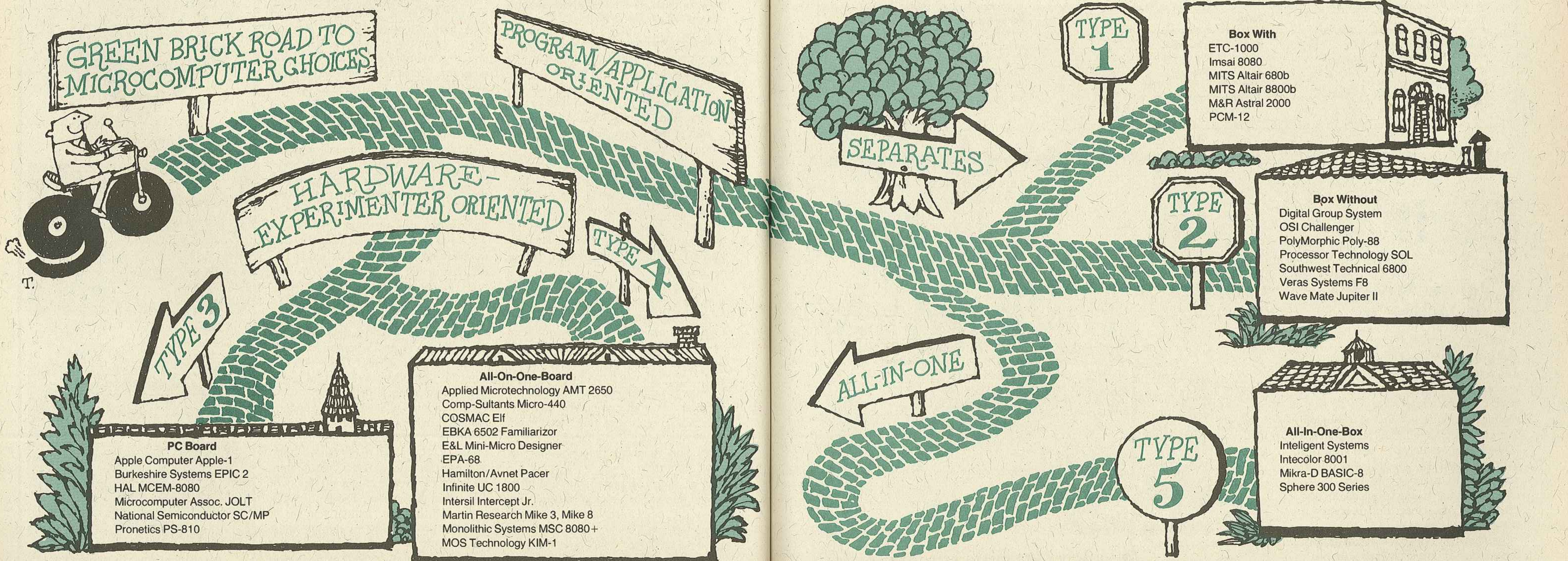
Are you determined to be an experimenter, more interested in hardware and/or learning the fundamentals of computers? If so, a Type 3 or 4 unit might be your best bet.

If you're more interested in "talking" to your computer and getting results easier and faster, but wish to add peripherals of your own choice at some future time, a Type 1 or 2 could be the way to go.

Should you want an all-in-one type of micro, with peripherals already incorporated, then perhaps a Type 5 should be considered.

There are other factors to weigh, of course, including language availability (do they have assembler or BASIC?), reputation of the computer manufacturer (how good are their computers and how long will they stay in business?), whether or not you plan to join a computer club for sharing ideas and trading information on software, and so on.

More Help. In addition to asking a manufacturer to send information on his microcomputer for performance details (see address listing), there are many



other ways to help you decide which to buy. There are over 90 computer clubs, many with membership in the hundreds, where you can talk with people who are using various hobby computers. Dozens of computer stores around the country will show you how their products work, and answer your questions in detail. Magazines and club newsletters devoted to the computer hobbyist are also excellent sources of information. And if you get to a hobby-computer convention, such as the ones that were held in New Jersey (Trenton and Atlantic City), you can check out dozens of computers and peripherals in a single day, as well as listen to talks about hardware, software and applications.

Whatever choice you make, you'll find yourself in a new, exciting field that will add to your knowledge and fun. ◇

DIRECTORY OF MICROCOMPUTER AND PERIPHERAL MANUFACTURERS

Apple Computer Co.
770 Welch Rd., Palo Alto, CA 94304

Applied Microtechnology
100 N. Winchester Blvd., Santa Clara, CA 95050

Burkeshire Systems
P.O. Box 512, Mountain View, CA 94040

Comp-Sultants
P.O. Box 1016, Huntsville, AL 35800

Cromemco
One First St., Los Altos, CA 94022

Digital Group, The
Box 6528, Denver CO 80206

Dutronics
P.O. Box 9160, Stockton, CA 95208

E&L Instruments, Inc.
61 First St., Derby, CN 06417

EBKA Industries, Inc.
6920 Melrose Lane, Oklahoma City, OK 73127

Electronic Control Technology
P.O. Box 6, Union, NJ 07083

Electronics Products Associates, Inc.
1157 Vega St., San Diego, CA 92110

Electronic Tool Co.
4736 El Segundo Blvd., Hawthorne, CA 90250

Gnat Computers
8869 Balboa Ave., Unit C, San Diego, CA 92123

Godbout Electronics
Box 2355, Oakland Airport, CA 94614

Hal Communications Corp.
P.O. Box 365, Urbana, IL 61801

Hamilton/Avnet Electronics
10950 West Washington Blvd., Culver City, CA 90230

IMS Associates, Inc.
14860 Wicks Blvd., San Leandro, CA 94577

Infinite Inc.
P.O. Box 906, Cape Canaveral, FL 32920

Intelligent Systems Corp.
4376 Ridgegate Dr., Duluth, GA 30136

Intersil, Inc.
10900 N. Tantau Ave., Cupertino, CA 95014

Lear Siegler Inc.
714 N. Brookhurst St., Anaheim, CA 92803

M&R Enterprises
P.O. Box 1011, Sunnyvale, CA 94088

Martin Research
3336 Commercial Ave., Northbrook, IL 60062

Microcomputer Associates
2589 Scott Blvd., Santa Clara, CA 95050

Micro Peripherals, Inc.
P.O. Box 22101, Salt Lake City, UT 84122

Mikra-D, Inc.
30 Main St., Ashland, MA 01721

MITS
2450 Alamo SE, Albuquerque, NM 87106

Monolithic Systems Corp.
14 Inverness Dr. East, Englewood, CO 80110

Mos Technology, Inc.
950 Rittenhouse Rd., Norristown, PA 19401

National Multiplex Corp.
3474 Rand Ave., So. Plainfield, NJ 07080

National Semiconductor Corp.
2900 Semiconductor Dr., Santa Clara, CA 95051

Ohio Scientific Instruments
11679 Hayden St., Hiram, OH 44234

PCM Company
Box 215, San Ramon, CA 94583

PolyMorphic Systems
737 S. Kellogg, Goleta, CA 93017

Processor Technology
6200 Hollis St., Emeryville, CA 94608

Pronetics Corp.
P.O. Box 28582, Dallas, TX 75228

RCA Solid State Division
Box 3200, Somerville, NJ 08876

Scientific Research Instruments Co.
P.O. Drawer 2096, Ashland, VA 23005

Southwest Technical Products Corp.
Box 32040, San Antonio, TX 78284

Sphere Corp.
P.O. Box 213, Bountiful, UT 84010

Tarbell Electronics
144 Miraleste Dr., #106, Miraleste, CA 90732

Vector Electronics Co., Inc.
12460 Gladstone Ave., Sylmar, CA 91342

Veras Systems
P.O. Box 74, Somerville, MA 02143

Wave Mate
1015 W. 190th St., Gardena, CA 90248

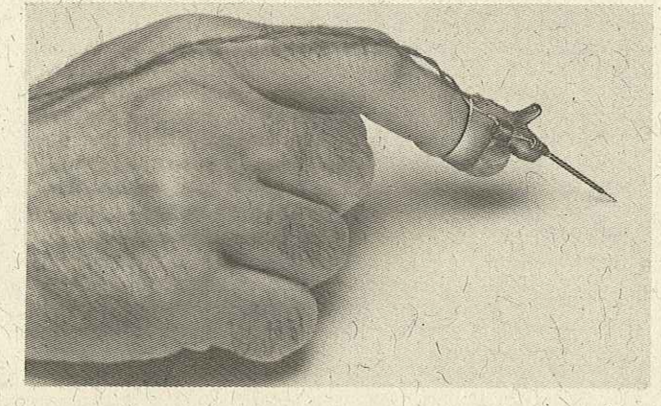
Wintek Corp.
902 N. 9th St., Lafayette, IN 47904

BY LESLIE SOLOMON
Technical Editor

Digit Probe

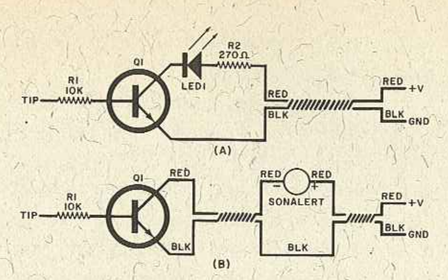
Compact, easy-to-use logic probe fits your finger.

EVER SEE a logic probe that was so compact that it could fit over the tip of your finger? Although there is such a device (see photo), don't look for it commercially—you have to make it yourself. We call this ultra-compact little gem the "Digit Probe," mainly because in use it's like an extension of your index finger. Used in this manner, the Digit Probe makes it easy to trace pulses around crowded IC assemblies and pc board foil traces that all look alike.



The Digit Probe fits on finger.

As shown in the schematic diagrams, the circuits for the Digit Probe are basic HI/LO indicators. Circuit A provides a visual indication of conditions existing in the circuit under test via light-emitting diode *LED1*. Circuit B provides an audible indication via the Mallory Sonalert®. Circuit A is convenient for tracing pulses in an operating system, while circuit B is more convenient when you have to look away from the system under test to make equipment adjustments and can't monitor a LED.



Use either a LED readout or audible signal.

All components (except the Sonalert) in both circuits should be kept as small as possible so that the assembled circuit can be mounted on an ordinary plastic guitar/banjo pick. Use a miniature general-purpose npn transistor for *Q1* and ⅛-watt resistors for *R1* and *R2*. Any size of color discrete light emitting diode can be used for *LED1*.

Construction. The Digit Probe circuit mounts directly on the outer surface of the guitar/banjo pick and is held in place with quick-setting clear epoxy cement. Assembly is very easy and non-critical, but you will have to take care to keep the physical layout as compact as possible.

Start construction by trimming both leads of the two resistors to ¼" (6.4 mm) and bending the lead stubs into hooks. Pre-tin the head of a straight pin with solder and solder the head of the pin to one lead of *R1*. Solder the other lead of *R1* to the base lead of *Q1*. Solder the cathode lead of *LED1* to the collector lead of *Q1* and the anode lead to one end of *R2*. Solder separate 36" (about 1-m) lengths of small-diameter stranded hookup wire to the free end of *R2* and the emitter lead of *Q1*, using red and black insulation, respectively. Terminate the free ends of the hookup wire with miniature alligator clips. Finally, loosely twist together the hookup wires.

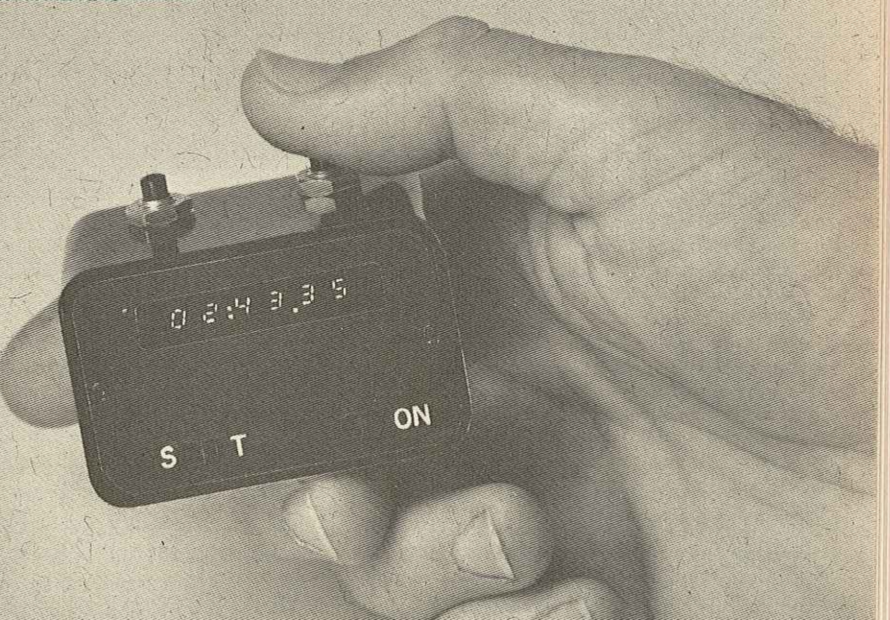
If you're planning to make the audible version of the Digit Probe, eliminate *R2* and *LED1*. Wire the circuit as described above, locating the Sonalert about 10" (25.4 cm) from the alligator clip end of the twisted-pair power cable.

Liberally coat the area of the guitar/banjo pick on which the Digit Probe circuit is to mount with epoxy cement. Press the circuit into the cement, orienting it as shown in the photo. Slip over the projecting straight pin a length of plastic sleeving, leaving about ⅛" to 3/16" (3.2 to 4.8 mm) near the point of the pin exposed. Coat the circuit with more epoxy cement to assure a firm mechanical anchor. Then allow the cement to cure for at least 24 hours before using the probe.

In Use. Slip the Digit Probe over the index finger of the hand you would normally use to hold a probe during tests. Clip the alligator clips on the black and red twisted-pair hookup wire to the − and + supply lines of the circuit under test. Then, using the probe is as simple as pointing your finger. ◇

BY WAYNE KASHINSKY

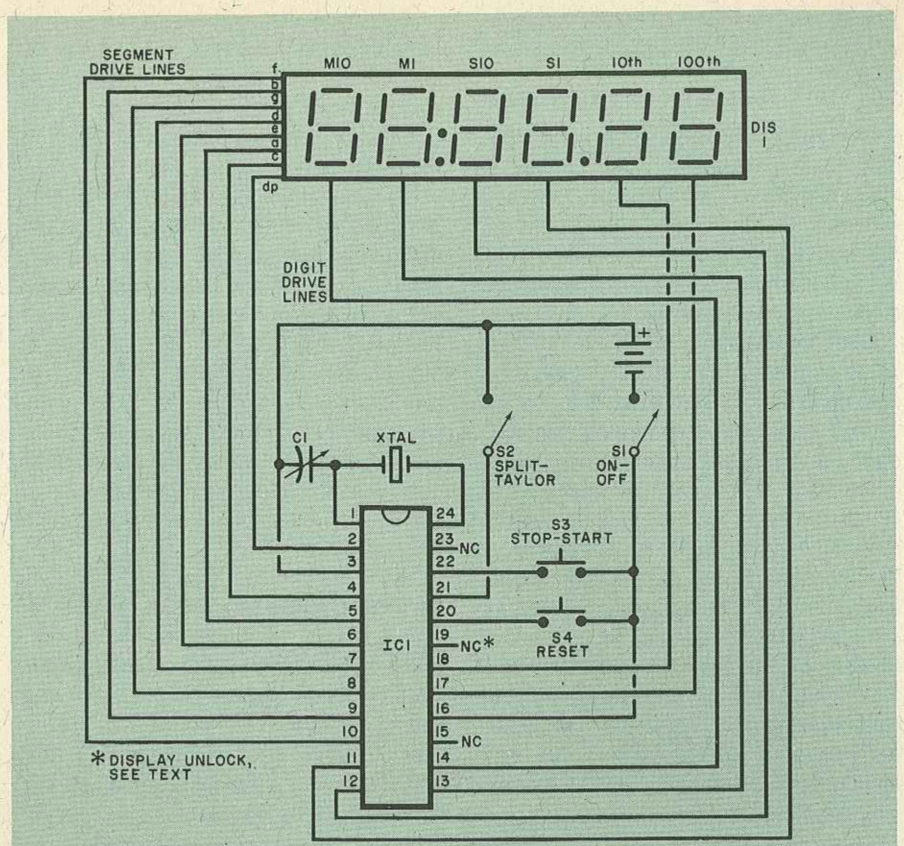
Build a Miniature Digital Stopwatch



Times from 1/100 s to 59 min, 59.99 s in split or Taylor modes.

THERE have been many sports timer construction articles, but never one for a project as small as this. Although this six-digit LED readout timer can measure time intervals from one 1/100 of a second to 59 minutes, 59.99 seconds, it can be held easily in the palm of the hand and stored in a shirt or jacket pocket. Timing can be in either the split-cumulative mode (display frozen when START-STOP pushbutton is depressed and total elapsed time with each successive switch operation) or the Taylor-sequential mode (time interval displayed between successive switch operations).

The single IC used in this timer has a built-in crystal-controlled oscillator, a low-battery indicator (decimal points come on), and internal digit and segment drivers. The output transistors can handle up to 20 mA per segment and do



PARTS LIST

B1—Three AAA cells in series
C1–8-40-pF subminiature trimmer capacitor (optional, see text)
DIS1—Common-cathode six-digit calculator readout stick on 2-in. board
IC1—7205 timer (Intersil)
S1, S2—Spdt subminiature slide switch
S3, S4—Spst miniature pushbutton switch, normally open
XTAL—3.2768-MHz crystal
Misc.—Plastic case and cover (Pomona 2104 or similar), 24-pin IC socket (optional), Molex pins (optional), press- on type or fine brush and white paint, hook-up wire.
Note—The following are available from AD-AGE, Box 1004, New Brunswick, NJ 08903: kit K1, consisting of IC1 and XTAL, for $21.95; kit K2, consisting of all parts, including drilled case, except batteries, for $39.95. Include $1.50 for shipping. New Jersey residents, add 5% sales tax.

Fig. 1. Complete timing circuit is contained in 7205 IC.

not require external current-limiting resistors. The total average current demand is less than 40 mA so that three AAA cells or 3-N rechargeables can be used as the power source for up to 12 hours. When the battery voltage drops below 2.6, the indicator comes on. Generally, the timer can still be used for about 15 minutes after this occurs. The simplicity of the circuit can be seen in Fig. 1. A complete description of the circuit's operation can be found in the August, 1976, issue of POPULAR ELECTRONICS, p. 73.

Construction. Although any type of construction can be used, to make the sports timer as small as possible, the foil pattern shown in Fig. 2 should be used. Note that components are mounted on both sides of the board as shown in Fig. 3. The IC is mounted on the blank side of the board, preferably using a socket or Molex pins, or it can be soldered in place, depending on the cells used.

The LED display is a conventional 2-inch calculator 6-to-9-digit stick with flat red lens. In the prototype, an NSN-66A (National Semiconductor) was used, but a brighter readout can be obtained with an NSA-1188. Other types of readouts can be used if the appropriate jumpers are made from the display pads on the board. The NSN-66A can be mounted on the foil side of the board as shown in Figs. 2 and 3. If the bottom surface of the display stick has exposed bare copper leads, cover it with a layer of masking tape to avoid shorting to the pc foil pattern. Mount the display and switches so that the tops of the switch bodies are flush with the upper surface of the display.

All wire connections to the board are made through the blank side of the board using slender flexible insulated wire. The batteries are wired in series and formed into a small bundle.

For the prototype a small plastic case was used for the timer. It measured 2⅜" W × 1½"H × 1½"D, with a fitted cover. Drill holes for *S3* and *S4* on one side of the case with sufficient spacing for the crystal between them. With the switches mounted on the case and connected to the proper points, solder the power leads to the AAA cells. Then fit the cells into the bottom of the case and put the pc board in the case with the display between the two switches. The edges of the pc board (and possibly the display) may have to be trimmed to make a proper fit. Insert the pc board until the upper surface of the display is just slightly below the rim of the case. The operating handles of *S1* and *S2* should stick above the case rim.

Once the board has been properly positioned, determine the locations of the readouts and *S1* and *S2* and cut the necessary slots in the cover. Install the cover and identify all the switches with a white dry-transfer lettering kit. Using the same careful techniques, apply a decimal point on the upper surface of the display just to the left of the two digits on the right end. Then apply a colon to the left of the second pair of digits.

If desired, an spst switch can be added between *IC1* pin 19 and the negative side of the battery. Operating this switch will permit the display to show the running clock at any time.

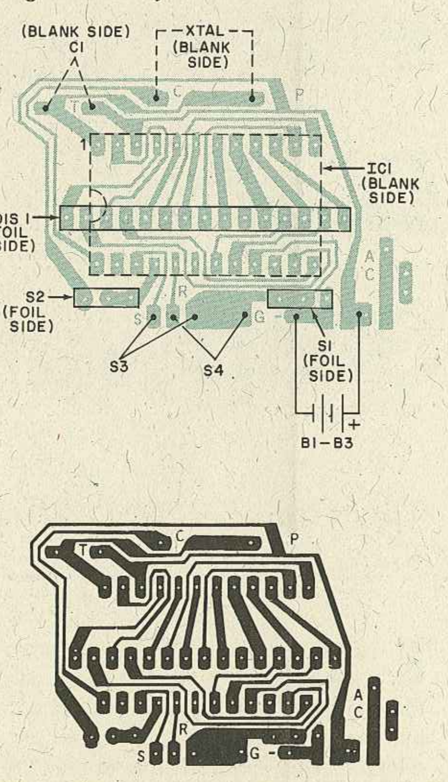


Fig. 2. Etching and drilling and component installation guides.

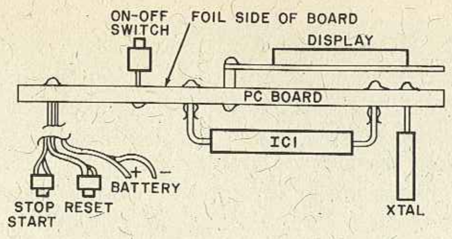


Fig. 3. Top and bottom of board.

Testing. Place *S1* (ON-OFF) in the ON position and note that the display is .00. Depressing *S3* (START-STOP) should cause the display to start counting in hundredths of a second. The IC has built-in automatic power-on reset and leading zero blanking so that the other digits will not be displayed until they are needed. Depressing *S3* again should cause the display to stop and indicate some elapsed time.

If only one digit comes on and it is very bright, the internal oscillator is not working. Examine the crystal circuit. If segments are missing, check the two pc boards for solder bridges or broken traces. If a segment and digit-driver line are shorted, that particular segment will not glow.

Once the timer operates correctly, it can be checked by listening to WWV or CHU. Place *S2* in the SPLIT position and start the timing on the minute signal. Check the time against the signal every few minutes to determine the timer accuracy. If the displayed time is more than it should be, mount a small trimmer capacitor (8 to 40 pF) on the pc board as indicated in Fig. 2. Adjust the capacitor and continue the timing tests until the desired accuracy is obtained.

Using the Timer. The START-STOP pushbutton is used as in a mechanical stopwatch. The RESET button zeros the counter.

In the SPLIT mode, operate the START-STOP button at the beginning of an event; then once more when the desired interval (half or full lap, etc.) is finished. The display will indicate the elapsed time. However, the counter is still operating, so depressing the START-STOP switch again causes the display to indicate the total elapsed time since the start of the event. If desired, operate the RESET pushbutton to stop the counter and return it to zero.

The TAYLOR mode automatically resets the counter to zero with each operation of the START-STOP switch, and the display shows the time interval between depressions. This is a useful function when you want to time each lap in a race without resetting the timer to zero. ◇

Digital Fuel Gauge

BY GREGORY BAXES

Replace your old analog meter with a bright LED display indicating percentage of fuel remaining.

THE FUEL gauge in most motor vehicles is a simple electrical meter-type movement that constantly monitors a changing current through a sensor located in the fuel tank. It is a simple matter to convert the monitoring system to a digitally generated numeric display and eliminate the uncertainties involved with reading and interpreting meter-type displays. Furthermore, a numeric display is much easier to read at a glance, which adds up to greater driver safety on the road or highway.

The digital fuel gauge described in this article can be installed in just about any motor vehicle to display the quantity of fuel remaining in the tank in 10's of percent. It uses readily available low-power TTL logic and linear IC's and large, easy-to-read seven-segment LED displays. The entire project can be built for about $25.

About the Circuit. The block diagram of the basic gas gauge circuit is shown in Fig. 1. Note that although the system is rigged to display three digits (to represent from empty, or 0, to full, or 100%), the units digit is a dummy seven-segment display that is always powered to show a 0; it is not driven by the circuit's logic as are the 10's and 100's digits. Since only 11 increments are actually displayed by the system, only 1½ di-

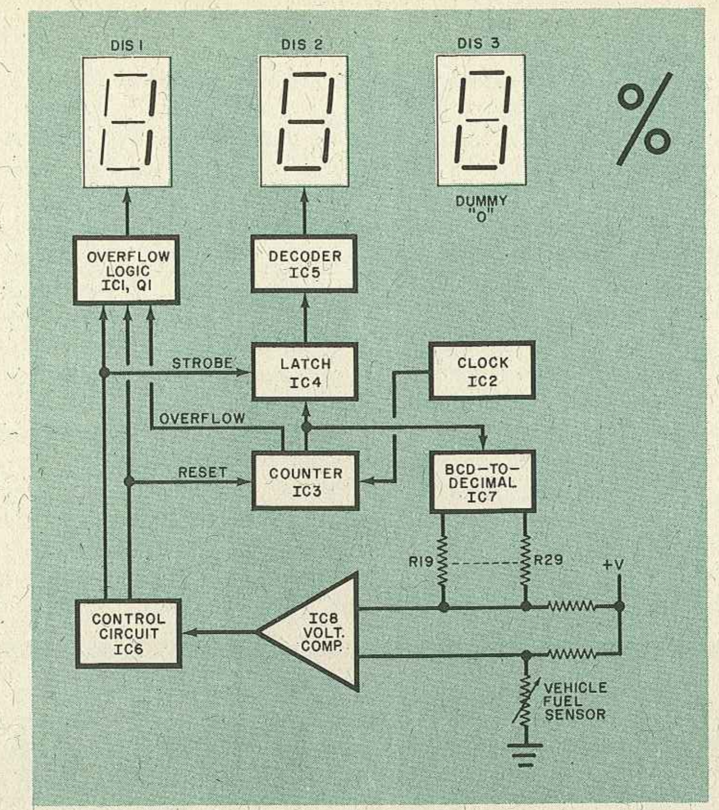


Fig. 1. Block diagram showing how gauge works.

gits are required. Hence, *DIS1* is either blanked or displays a 1.

Integrated circuit *IC2* serves as the clock generator for the system, operating at about 1 Hz. It drives a conventional decade-counting system consisting of counter *IC3*, latch *IC4*, decoder/driver *IC5*, and display *DIS2*. Under normal conditions, this counter simply cycles, with the clock pulses, from 0 to 9 and then generates an "overflow" pulse. In this system, however, the BCD outputs from *IC3* are also coupled to BCD-to-decimal decoder *IC7*. The 0-through-9 outputs from *IC7* and *R19* through *R29* generate a voltage that is proportional to the count at any instant. This voltage and a second voltage that is determined by the amount of fuel left in the gas tank are summed in voltage comparator *IC8*. The output of *IC8* is either high or low, depending on the differential between the two input voltages.

The values of weighting resistors *R19* through *R29* are selected to provide 10% changes in the display count. If, for example, the tank is 50% full, when the *IC3* through *IC7* circuit "sees" a 5, *IC8*'s output changes state to activate the *IC6* control circuit. Dual monostable multivibrator *IC6* generates a strobe pulse to cause a 5 to be displayed by *DIS2*. Shortly after this, *IC6* generates a reset pulse to allow the circuit to cycle again. In our example, the display system will indicate 50%.

The only time the system displays 100% is when the gas tank is full. At this time, *IC3* counts through 9 and goes to 0, generating an overflow, or "carry," signal. The carry signal passes to the overflow logic and is used to turn on the 1 in *DIS1* when the strobe pulse appears.

The reset pulse will return the system back to 0 so that the cycle can repeat. Thus, the display is updated every second or so, depending on the rate of the clock. The display will not flicker, however, because the latch in the used 1½ digits will maintain power to the digits between strobe pulses.

Although, with slightly more logic, the gas gauge could have been designed to provide a full 100-step resolution, an 11-step resolution was selected for practical reasons. A greater than 10% resolution would have resulted in an annoying fluctuation of the numerals displayed by *DIS3* as the motion of the vehicle caused the level of the gas in the tank to rise and fall.

The complete schematic diagram of the gas gauge is shown in Fig. 2.

Construction. You can use either a printed circuit board of your own design or perforated board to assemble the gas gauge. In either case, it is recommended that you use sockets for all IC's to obviate the possibility of heat damage to these components during soldering. It is also suggested that you use two boards—one for the display and a second for the rest of the circuit. Use color-coded hookup wire for the interconnections between the boards.

There are three external connections to be made for the gas gauge: +12 volts to the vehicle's electrical system, vehicle ground, and the "hot" side of the fuel sensor. This is most practically accomplished with the aid of a three-lug screw-type terminal strip mounted on the rear of the case in which the circuit is housed. Additionally, if you prefer, voltage regulator *IC9* can also be mounted on the case, provided the case is metal, for heat-sinking purposes.

Note in Fig. 2 that the values of resistors *R19* through *R29* are not specified. These resistor values must be determined for the specific fuel sensor with which the gas gauge is used. To determine the values of these resistors you must first locate the "hot" lead of the fuel sensor going to the meter on the vehicle's dashboard. Break this lead so that you can measure the sensor's resistance between the lead and ground.

There are two ways to obtain a relatively accurate list of fuel-sensor measurements. First, you can drive your car into a gas station with your car's meter-type fuel gauge still connected and registering empty. (Do *not* completely empty your tank of gas. If you assume an empty tank when the gauge reads empty, you will have a margin of safety when your tank runs low.) Have the attendant fill your tank to full and keep a record of the amount of fuel required to fill the tank. Divide the number by 10. You now know gallons at each 10% point. Then the next time you go for a refill, disconnect the "hot" sensor lead from the meter and have the attendant fill your tank in the previously noted 10% increments while measuring and logging the sensor's resistance at each 10% point.

The second way is to estimate the 10% marks on your car's meter-type fuel gauge, marking these points on the meter's faceplate with a grease pencil. Fill your tank to full, install a switch in the sensor's "hot" lead, and (with the switch closed) drive around until the meter's pointer registers 90%. Pull over, open the switch, and measure the sensor's resistance. Repeat this procedure until the meter registers empty, keeping a log of your measurements. Again, do *not* drive your car until the tank is completely empty. Remove the switch from the sensor's hot lead.

The resistance measured for an empty tank is the value of *R19* at pin 1 of *IC7*,

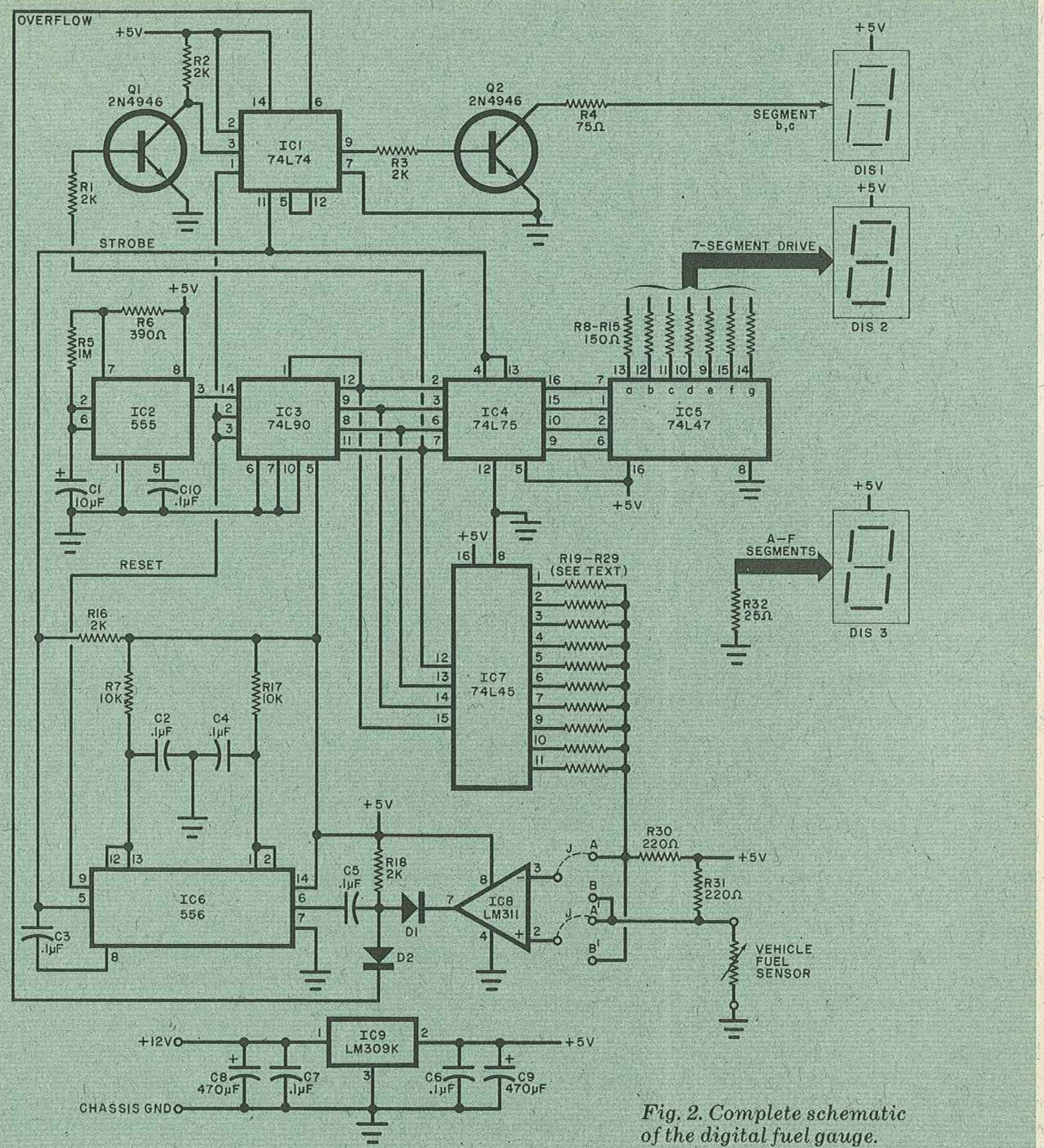


Fig. 2. Complete schematic of the digital fuel gauge.

PARTS LIST

C1—10-μF, 10-volt electrolytic capacitor
C2 through C7, C10—0.1-μF, 10-volt capacitor
C8, C9—470-μF, 15-volt electrolytic capacitor
D1, D2—1N4454 diode
DIS1, DIS2, DIS3—MAN-52 (Monsanto) or any common-anode seven-segment LED display
IC1—74L74 dual D flip-flop
IC2—555 timer
IC3—74L90 decade counter
IC4—74L75 latch
IC5—74L47 seven-segment decoder/driver
IC6—556 dual timer
IC7—74L45 BCD-to-decimal decoder/driver
IC8—LM311 voltage comparator
IC9—LM309K 5-volt regulator
Q1, Q2—2N4946 transistor
The following resistors are ¼ watt, 10%:
R1, R2, R3, R16, R18—2000 ohms
R4—75 ohms
R5—1 megohm
R6—390 ohms
R7, R17—10,000 ohms
R8 through R15—150 ohms
R30, R31—220 ohms
R19 through R29—Trimmer potentiometer (see text)
R32—25 ohm, 10%, ½-watt resistor
Misc.—Perforated or pc board; sockets for IC's (optional); chassis box; white dry-transfer lettering kit; red plastic display window; hookup wire; solder; machine hardware; etc.

while the resistance measured for a full tank is the value of *R29* at pin 11 of *IC7*. All other resistances are the values of *R18* through *R28* and fit into the circuit in consecutive order between pin 2 and pin 10 of *IC7*. (Note that pin 8 of *IC7* goes to ground; skip this pin when installing the resistors.) You can use miniature pc-type trimmer potentiometers for *R19* through *R29*.

If your tank's fuel sensor resistance increases as the fuel decreases, connect the inputs of *IC8* to pins A and A′ as shown with a jumper. If the tank sensor's resistance decreases as the fuel level decreases, connect the inputs of *IC8* to B and B′

Once the digital fuel gauge project is

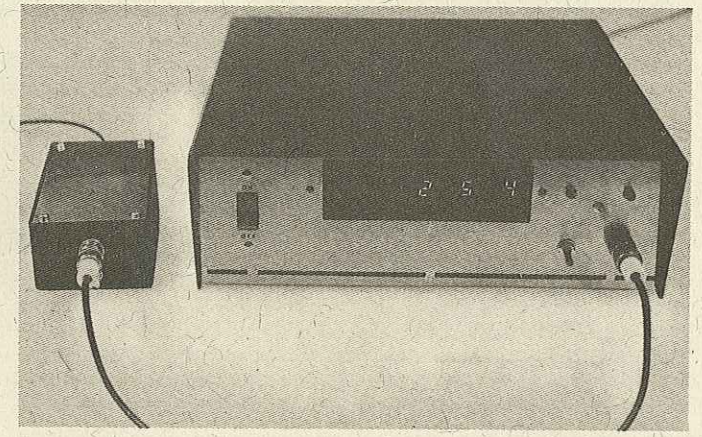
assembled, mount a red plastic window in front of the displays. Then, using a white dry-transfer lettering kit, label the legends FUEL above and PERCENT below the displays. (If you prefer, you can paint a white % sign on the window.)

Once the project has been assembled, it can be installed in your vehicle on top of the dashboard or in any location where it provides an unobstructed view of the displays. Make the three connections to the vehicle's electrical system: (1) +12-volt power input line to any point in the system that is powered when the ignition is on and off when the ignition is off; (2) ground to the vehicle's chassis ground; (3) sensor to the "hot" side of the fuel sensor. ◇

An A/D Temperature Converter

BY W. J. PRUDHOMME

Use your frequency counter to measure temperature to 0.1°C resolution.



THIS project is a low-cost analog-to-digital converter which allows you to make accurate temperature measurements with a frequency counter. Its range is 0° to 100°C, with a resolution of 0.1°C and an accuracy of 0.5°C.

The circuit uses an inexpensive silicon signal diode as a temperature sensor, a dual operational amplifier IC, a unijunction transistor, and a handful of resistors and capacitors. Parts cost is less than $10. No warm-up period is required, and the project is easily calibrated. Several sensors can be switched into the circuit to provide temperature readings at various locations.

About the Circuit. The converter's schematic diagram is shown in Fig. 1. When power is applied to the circuit, zener diode *D1* and resistor *R6* set up a reference 1-mA current through temperature sensor *D2*, a 1N914 silicon signal diode. When *D2* conducts, it exhibits a forward voltage drop of approximately 0.7 volt at room temperature (25° C). But this voltage drop is temperature dependent. For each 1° C increase in ambient temperature, the forward voltage drop decreases 2.2 millivolts. Conversely, for each 1°C decrease, the voltage drop increases 2.2 millivolts. This voltage signal is applied to the noninverting input of *IC1A*, an op amp integrator.

When the voltage across integrating capacitor *C1* reaches a certain value, unijunction transistor *Q1* turns on, discharging *C1*. Potentiometers *R2* and *R5* set the minimum and maximum charge/discharge rates, respectively. Each time *C1* is discharged, an output pulse is generated. This pulse is coupled to the noninverting input of *IC1B*, an op amp buffer whose gain has been selected to produce a pseudo-square-wave output. The output signal, appearing at *J2*, is then coupled to the frequency counter input by a short jumper of coaxial cable.

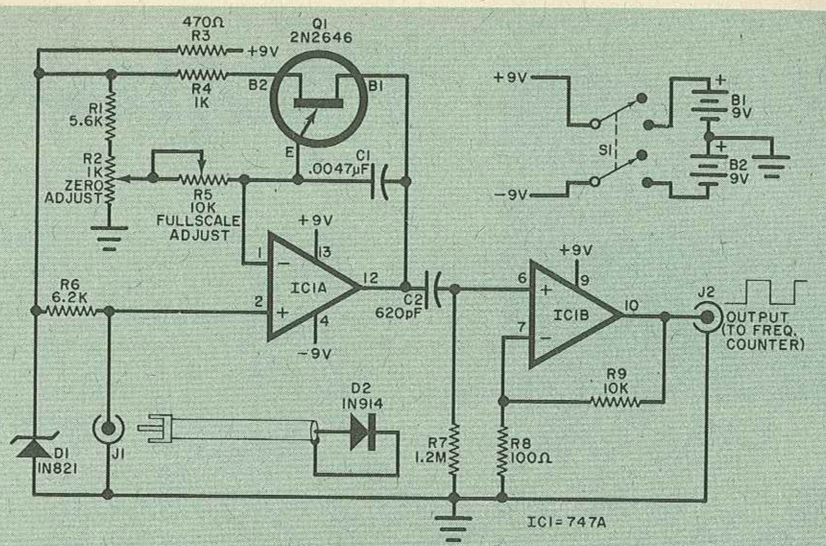
The conversion ratio of the A/D converter is 10 Hz per degree C when properly calibrated. That is, when the measured temperature is 25.4° C, the counter will indicate a frequency of 254 Hz.

A bipolar (±9-volt) power supply is required. A line-powered dc source can be used, but two 9-volt transistor batteries connected in series are also suitable.

Construction. Circuit layout is not critical, so printed circuit or perforated board can be used. Etching and drilling and component layout guides for a suitable pc board are shown in Fig. 2. Trimmer potentiometers *R2* and *R5* can be fashioned from vertical mounting types. Simply bend the three lugs on each so the controls can be mounted flat against the circuit board. Be sure to observe proper polarity and basing on the IC, and other semiconductors. The use of an IC socket or Molex Soldercons is recommended.

Temperature sensor *D2* should be connected to a length of shielded cable terminated with a phono plug. Be sure to connect the diode so that its cathode is grounded. Otherwise, false readings will be obtained. Also, it is recommended that you dip the diode in clear epoxy cement after it has been soldered. Allow the epoxy to cure for 24 hours before using the sensor. This will give a protective coating around the diode.

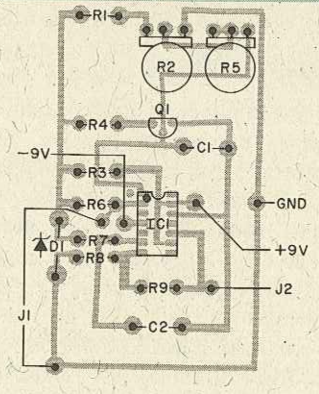
Connection to the frequency counter should also be made with a jumper of shielded cable terminated with proper plugs. You may want to use the same jack for *J1* as is on the counter, such as a BNC jack. Alternatively, a phono jack can be used.



PARTS LIST

B1, B2—9-volt transistor batteries
C1—0.0047-μF silver mica or polystyrene capacitor
C2—620-pF silver mica or polystyrene capacitor
D1—6.2-volt, 1-watt zener diode (1N821 or equivalent)
D2—1N914 silicon signal diode
IC1—747A dual operational amplifier
J1, J2—phono jacks
PL1—phono plug
Q1—2N2646 or Radio Shack 276-111 unijunction transistor
The following resistors are 10%, ¼-watt unless otherwise specified:
R1—5600 ohms
R3—470 ohms
R4—1000 ohms
R6—6200 ohms, 5% tolerance
R7—1.2 megohms
R8—100 ohms
R9—10,000 ohms
R2—1000-ohm printed circuit trimmer potentiometer
R5—10,000-ohm printed circuit trimmer potentiometer
S1—Dpdt toggle switch
Misc.—Battery clips, suitable enclosure, hookup wire, shielded cable, IC socket or Molex Soldercons, machine hardware, solder, etc.

Fig. 1. In converter circuit, signal diode D2 acts as a linear temperature sensor.



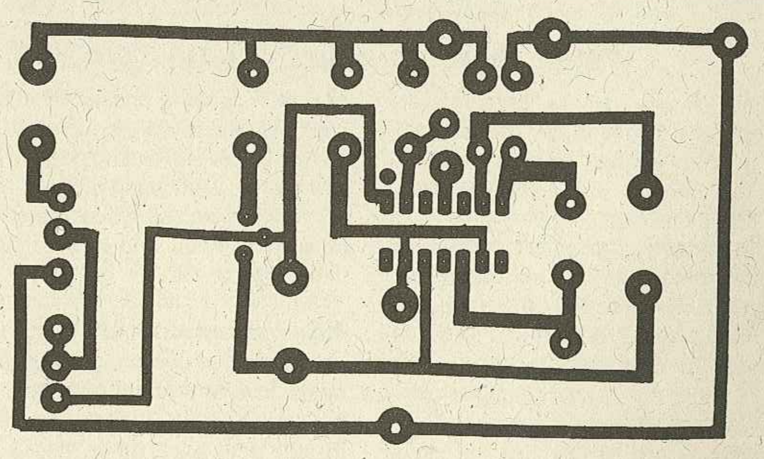


Fig. 2. Actual-size etching and drilling guide at right. Parts placement guide above.

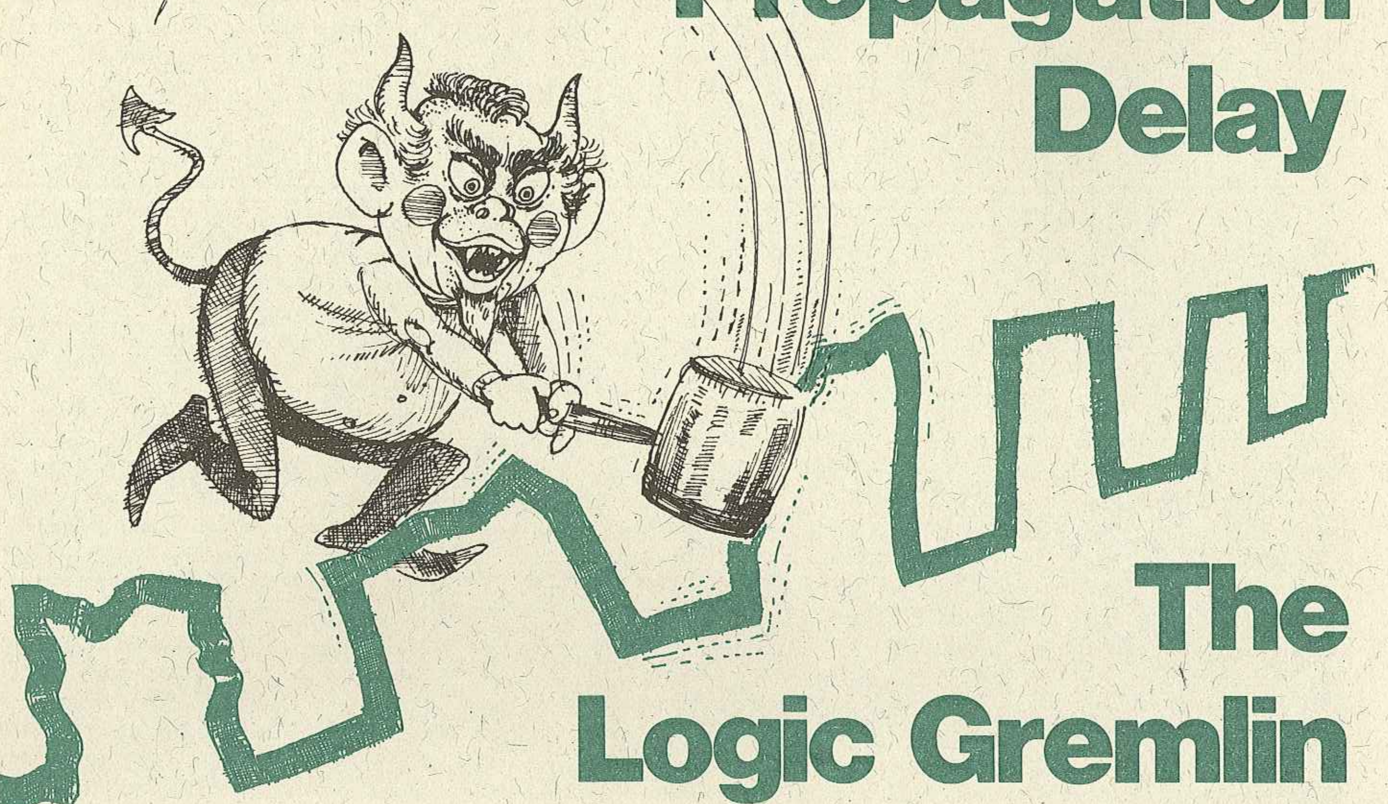
Calibration. Once the circuit has been built and is operating, it should be calibrated at both extremes of its range. With the converter connected to the frequency counter, insert the temperature sensor into a bath of boiling water. Adjust *R5* so that the frequency counter reads 1000 Hz (100°C). Then insert the sensor in a container of crushed ice and adjust *R2* for a 0-Hz (0°C) reading. Because there is a degree of interaction between the two controls, the procedure must be repeated several times until proper readings are obtained at both temperature extremes.

Operation. Once the project has been calibrated, it will exhibit excellent linearity over its entire temperature range. A rotary switch can be added if remote sensing at several locations is desired. The shortest possible length of shielded cable should be used with each diode. A voltage drop in the wire of even a few millivolts (that has not been compensated for in calibrating the project) will affect the converter's accuracy. If the diode sensor is damaged at temperature extremes, simply replace it with another. The cost of signal diodes is low enough for you to keep many spares on hand. ◇

"That's the third paragraph you've started with 'according to our computer.'"

BY RALPH TENNY

Propagation Delay The Logic Gremlin



Where those "glitches" come from and what to do about them.

LOGIC circuits usually behave very logically. For example, trigger a flip-flop and its outputs change state; or drive an inverter and the signal flips over. What could be simpler?

Unfortunately, "glitches" (undesired signals) sometimes get into a circuit and cause it to misbehave. When you look into the problem, you find that all the digital logic IC's are good; the clock is fine and healthy; the power supply is clean and well-regulated; and the wired interconnections are all OK. But the circuit still produces erratic results!

If you are blessed with a high-quality oscilloscope, it is possible to spot mysterious glitches wandering around the circuit, appearing like that shown in Fig. 1. If this signal is applied to a logic circuit, the circuit will trigger on the glitch as well as on the leading edge of the real signal. This produces an erratic result. In this article, we will discuss the sources of such glitches and how to eliminate them, if possible.

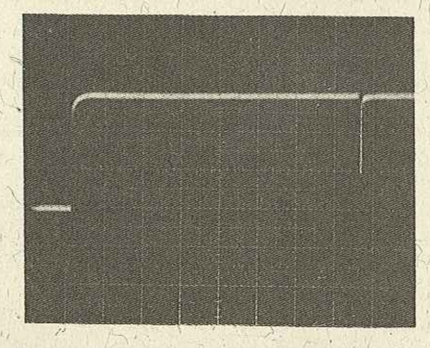


Fig. 1. Retouched photo of glitch.

Basic Element. Let's begin by considering the simplest logic element—the basic inverter. Although it seems that the input and output of an inverter occur in step with each other, this is *not* the case. These devices require a finite amount of time to respond to a signal input. This "propagation delay" is shown in Fig. 2. Specification sheets for the logic device give the amount of delay to be expected. Interestingly, propagation delay is not related to waveform rise and fall time and is different for positive-going and negative-going waveforms. To further complicate matters, many TTL specifications sheets list both minimum and maximum delay times, with both specified for a standard resistive and capacitive load. Any extra capacitance in the load will simply produce more delay.

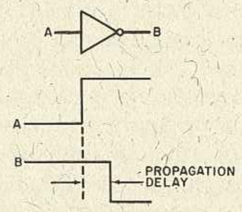


Fig. 2. Inverter output delay.

For example, if two TTL devices such as the 7400 quad 2-input NAND gate and the 7404 hex inverter are combined in a circuit that depends on propagation delay (sometimes we can make the glitch work for us), we can observe the effect of typical and maximum delay times. The circuit, a propagation delay one-shot, is shown in Fig. 3 with its associated waveforms. The positive-going edge (A) causes the output to fall one delay time later (B). At about the same time the output at (C) falls. One more delay later, output (D) rises. Finally three delays after input (A), output (E) drives output (B) high. The solid lines show an ideal situation while the dashed lines show how propagation delay affects the width of the output pulse. Note that the output pulse width depends on the combined propagation delays of the devices used.

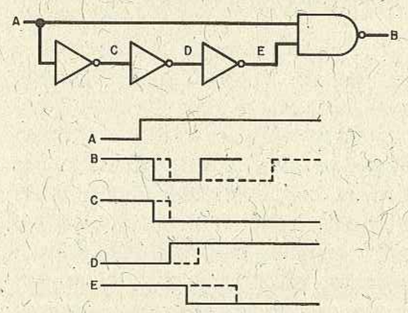


Fig. 3. Making use of the glitch.

The question then is: which delay figures should be used? The answer is: the worst-case figures. That is simple enough, but which is the worst case? The answer to that depends on the application. The designer must decide what effect a slow or fast (responding) device will have on each part of the circuit under consideration.

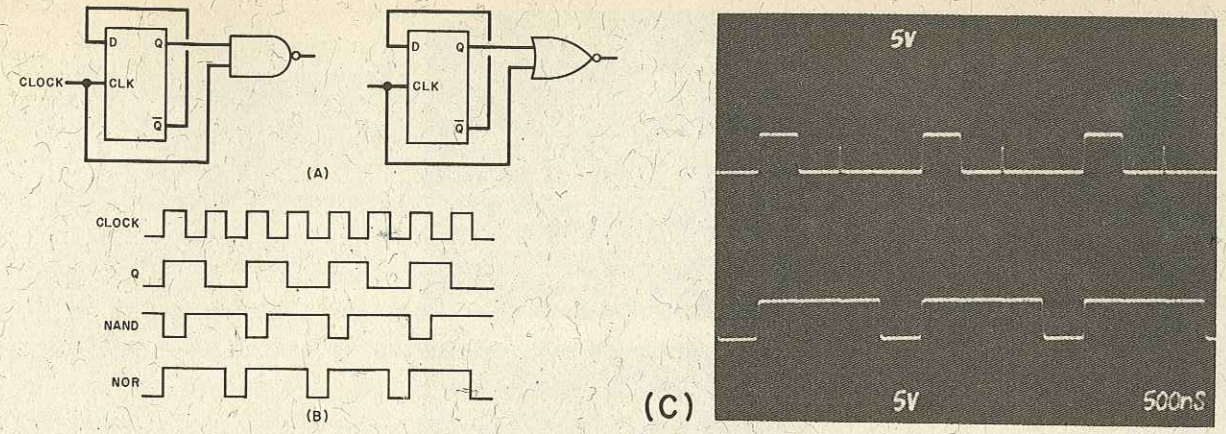


Fig. 4. Circuits in (A) should produce waveforms at (B). But (C) shows a glitch.

Predicting Propagation Delays. Here is one method that can be used to predict whether propagation delays will cause an unwanted glitch. Two divide-by-two circuits are shown in Fig. 4A. In the circuit on the left, a NAND gate is operated by a flip-flop. In the other circuit, the gate is a NOR. Theoretically the output waveforms of the two circuits would be similar and would look like those in Fig. 4B. However, Fig. 4C shows the actual output as viewed on a scope, except that AND and OR gates were used and the top waveform was inverted from that shown in the NAND portion of Fig. 4B. Now, where did that glitch come from?

To answer this question, we will redraw Fig. 4B using a time scale of 50 nanoseconds per division and take propagation delay into account. Fig. 5A is the result. Note that the glitch is produced by the overlap between the clock and the Q output caused by the propagation delay in the flip-flop. In the case illustrated, it was assumed that the flip-flop had maximum delay and the gates were typical devices—which makes the largest glitch. Fig. 5A does not look exactly like Fig. 4B because the latter was drawn as if the waveform transitions were instantaneous, which they are not. The effects provided by rise and fall times are shown in Fig. 5B.

So far we have considered only simple circuits. Clearly, by choosing the OR gate in Fig. 4A, we avoid the glitches. Now, suppose the design requires a source of timing signals derived from a counter. In the circuit in Fig. 6A, a 74197 counter drives a 74154 decoder to produce the waveforms shown in Fig. 6B. The circuit produces 16 sequential timing pulse outputs, but only five are shown in Fig. 6B. So far, so good. Unfortunately, if you look at the output on a scope, the waveform in Fig. 7A is the result. This is not a pretty picture! What went wrong?

The 74197 is a ripple counter. This means that the input clock toggles the first flip-flop, which in turn toggles the second flip-flop, etc. Eventually, the signal propagates to the output. The spec sheet for this device indicates a maximum of 60 ns delay, with a minimum of 10 ns for each stage. Next, in the 74154, the inputs are buffered by an inverter and then inverted again as necessary for the final decoding. The interlocking arrangement of inverters and gates produces differential delays and thus permits the occurance of glitches—even if the decoder inputs are synchronized. In this case, the solution is to feed a narrow clock pulse to the *enable* inputs of the 74154, then invert the clock to drive the 74197 counter. This "de-glitcher" is shown by the dotted lines in Fig. 6A. If the clock pulse is wider than the counter delay, the output signals become as clean as those shown in Fig. 7B.

Solutions. We can now summarize the points covered and learn a little more about de-glitching:

1. Glitches are caused by unbalanced propagation delays in the signal path. In theory, the glitch of Fig. 5 could be eliminated by adding a delay in the circuit as shown in Fig. 8. This would require that both inverters and flip-flop have "typical" delay specifications.

2. In general, decoding with OR/NOR

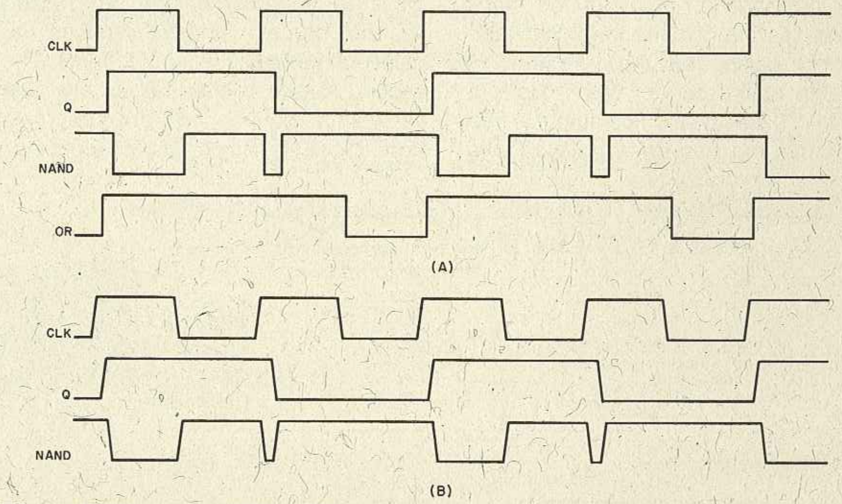


Fig. 5. Expansion of Fig. 4B shows propagation delay.

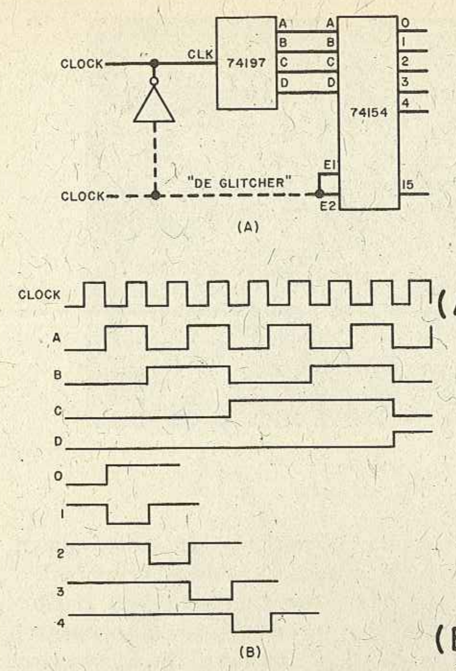


Fig. 6. Dotted circuit in (A) is added to remove glitches.

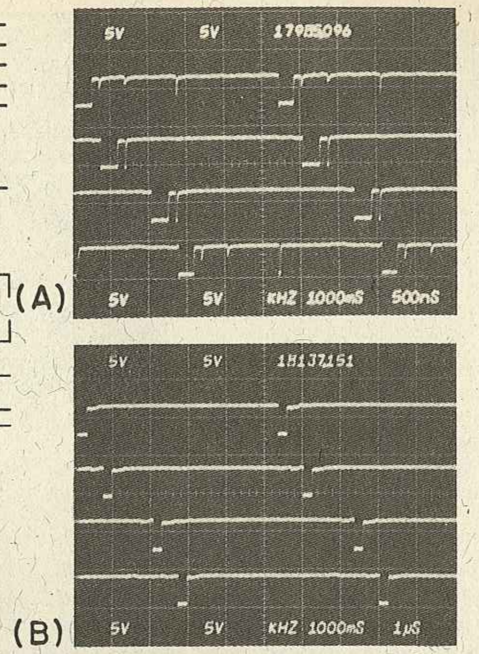


Fig. 7. Enable gating removes glitches in (A) to give (B).

gates, as in Fig. 4A, eliminates glitches.

3. Some logic devices incorporate enable inputs, which, if properly used, can eliminate glitches.

4. Some functions can be performed differently, such as by using synchronous counters like the 74193/74163 instead of ripple counters like the 7490/74197.

5. Though some logic families such as CMOS have slow rise times and slow operation (which should eliminate glitches), remember that any logic family will respond to glitches produced by that logic family.

6. In many cases, glitches can be eliminated by flip-flop sampling. If you have a glitched output that comes from a "black box" that can't be de-glitched by simple methods, use the circuit shown in Fig. 9A. The black-box output is fed to a D flip-flop that is clocked by the system clock. Propagation delay of the black box causes the glitches to fall between the clock pulses, but the real signal is available at the correct times. Note in the Fig. 9B that the flip-flop output is free of glitches but has been delayed one clock period plus the propagation delay of the flip-flop. If there are critical timing path considerations in the circuit, then it may be necessary to make some delay adjustment in one of the other "downstream" circuits.

7. An RC delay can be used to combat narrow glitches using the technique shown in Fig. 10A. Timing waveforms are shown in Fig. 10C. By proper selection of the RC time constant, the delay across the RC network is longer than the glitch time and the glitch disappears. For extra long glitches, it may even be necessary to use two RC networks separated by a logic gate as shown in Fig. 10B.

The RC de-glitching method is just barely acceptable for TTL logic for two reasons. The first is that since the TTL inputs require 1.6 mA drive, the resistor is limited to about 180 ohms. This requires the use of fairly large-valued capacitors—on the order of 1000 pF. Even so, the 180-ohm resistor reduces the noise immunity of the input it feeds. The second reason is that the large-valued capacitors require large drive currents from the TTL. Since CMOS logic has very high input impedance, the RC delay is very effective as a CMOS de-glitching method.

8. One particular type of TTL logic that will tolerate large values of RC time constants for de-glitching is the TTL Schmitt trigger. Figure 11 shows circuits and waveforms using the 7414 (hex Schmitt) and 74132 (quad 2-input Schmitt) for this purpose. It is still necessary to limit the resistor value to about 330 ohms using these devices. The CD4093 is a CMOS quad 2-input Schmitt trigger device that is very effective for de-glitching and delay using resistor values up to about 100,000 ohms.

9. A CMOS buffer (CD4010, CD4050) can be used for de-glitching, delay, and even switch debouncing with the circuit shown in Fig. 12. Feedback resistor *R2* determines the hysteresis of the circuit (the Schmitt trigger action) while the time constant of *R1/C* sets the amount of delay. ◇

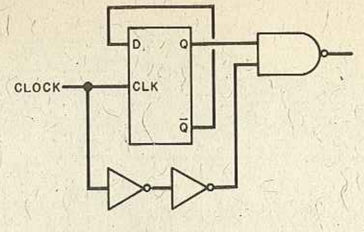


Fig. 8. Modification of Fig. 4.

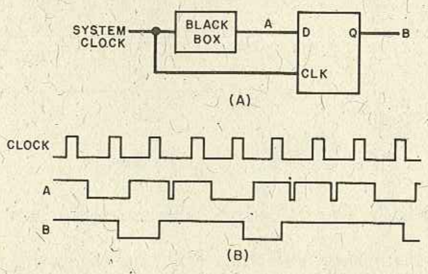


Fig. 9. Flip-flop removes glitches.

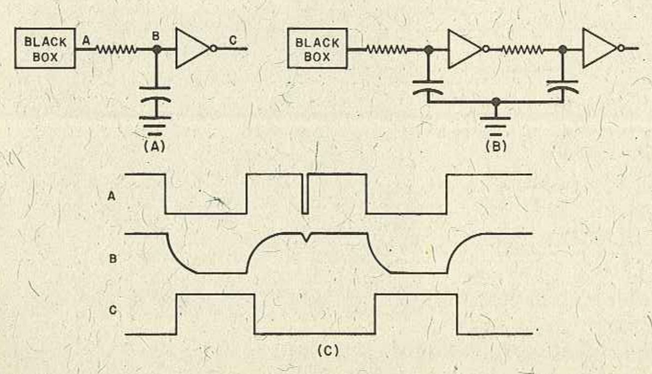


Fig. 10. RC delay can be used to remove narrow glitches.

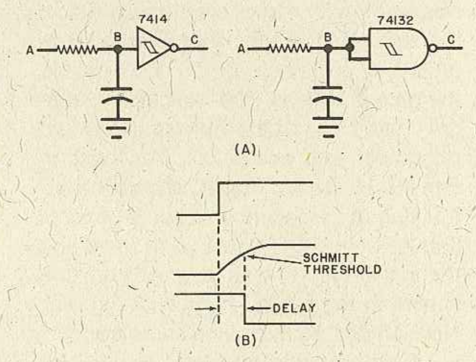


Fig. 11. Schmitt trigger solution.

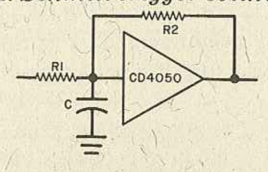


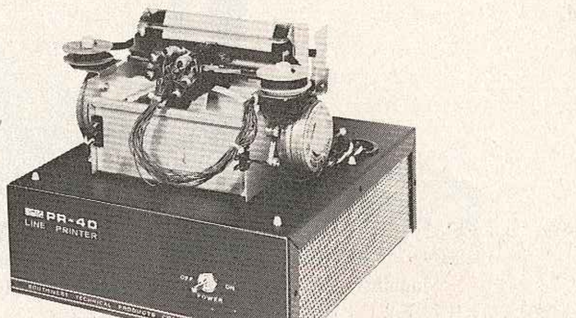
Fig. 12. CMOS de-glitcher.

NEED HARDCOPY?

If you are one of the many computer users who wants hardcopy printouts, but can't afford any of the available machines, your troubles are over. Our PR-40 is a universal printer that gives you clear easy to read hardcopy like the sample on the right with almost any computer. Our printer operates from any eight bit parallel I/O port. The printer has it's own character generator and memory buffer. This means that the computers only job is to feed data when the printer is ready. No special program is needed in the computer to convert the data to a form that the printer can use as each character is printed. The PR-40 is easy to use, easy to interface and easy to afford.

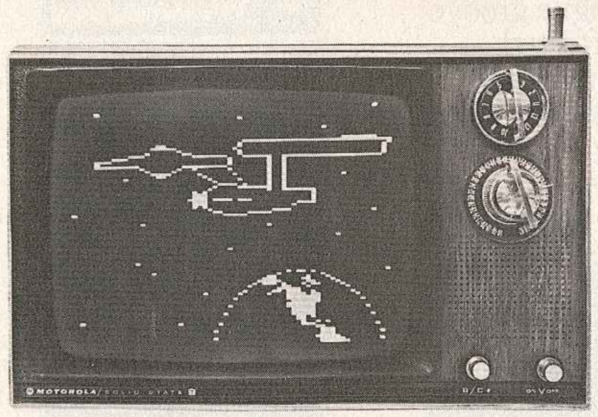
```
* SWTPC PR-40 ALPHANUMERIC PRINTER *

* 40 CHARACTERS / LINE
* 5 X 7 DOT MATRIX IMPACT PRINT
* USES STANDARD 3 7/8" CALCULATOR PAPER
* 75 LINE / MINUTE PRINT RATE
* AUTOMATIC RIBBON REVERSE
* 64 CHARACTER ASCII CHARACTER SET
* 40 CHARACTER LINE MEMORY
* TTL, SWTPC 6800, MITS COMPATIBLE
```



PR-40 LINE PRINTER KIT................$250.00 PPd

HOW ABOUT PICTURES?



Games are more fun with pictures. Now you can add graphics displays to your game programs and on any type computer. Our GT-6144 operates from any eight bit parallel I/O port. It has it's own self contained memory, so memory space for the display is not robbed from your computer. The 9½ x 13 circuit board contains all you need to produce a graphic display like the one of the starship "Enterprise" shown on the left. Kit is less power supply, or chassis.

GT-6144 GRAPHICS TERMINAL KIT . .$ 98.50 PPd

I know a bargain when I see it. Send the following:

☐ 6800 Computer $395.00 ☐ PR-40 Printer $250.00
☐ GT-6144 Graphics Terminal $ 98.50 ☐ Just data (free)

NAME
ADDRESS
CITY STATE ZIP

Southwest Technical Products Corp.
Box 32040, San Antonio, Texas 78284

CIRCLE NO. 61 ON FREE INFORMATION CARD