CorelDRAW 2.01 Vs. Arts & Letters Graphics Editor 3.1

ComputerCraft

August 1992 $2.95
(Canada $3.95)

THE PRACTICAL MAGAZINE FOR PERSONAL COMPUTERS & MICROCONTROLLERS

- **Intel's Speed-Doubling Chip**
- **Computer-Controlled Frequency Synthesizer**
- **EPROM Programming For SBCs & Microcontrollers**

0 74820 08559 08

Robotics, Control & Monitoring

Yacco's Impressions Of Microsoft Windows 3.1

How Interrupts (IRQs) Work

ComputerCraft

Volume 2, No. 8 August 1992

45

88

ON THE COVER: Intel debuts its user-installable OverDrive chip that doubles the internal processing speed of 486SX computers. For details, see Ted Needleman's column beginning on page 61.

Cover Photo: 1992 Intel Corporation/Richard Wahlstrom Photography

Computer User Groups

Personal Computer User Groups abound across the nation. They consist of groups of all sizes and varied computer, software or application focuses. These like-minded people are enthusiastic computer users, of course.

A picture of club members in broad-type clubs (as compared to narrower groups that might focus on only "Paradox" software or the like) was recently gathered by a User Group Demographic Study prepared by Dr. Arthur Saltzman, Department of Marketing, California State University. It surveyed 12 user groups and was sponsored by IBM, Intel, Lotus Development and the Association of Personal Computer User Groups.

Since we recently completed a study of *ComputerCraft* readers, I naturally compared both results. For example, 91.8% of members of the dozen clubs surveyed own an IBM or compatible computer, whereas 86.5% of all *ComputerCraft* readers own at least one. And while club members own an average of two IBM/compatible computers, *ComputerCraft* readers average 1.3 IBM/compatible machines each. Furthermore, club members average about 20% laptop computer ownership, while *ComputerCraft* readers reported 11.5% ownership.

Our methodology was different in some respects; so certain data aren't fully comparable, but they're close enough. For example, the Club study asked respondents, to describe the IBM or compatible they used most frequently, whereas our study requested information on all IBM/compatibles owned.

Accordingly, everything comes up with a 100% total in the Club study, whereas our total exceeds a 100% base when multiple responses are issued. Nevertheless, the user-group survey is a very rich collection of valuable data that was heretofore unavailable.

With the foregoing in mind, the Club study reveals that 19.5% of all members responding own a PC or XT (8088 or 8086 CPU), 27.2% an AT (80286 CPU), 14.8% a 386SX, 32% a 386DX, 4.8% a 486SX or DX, and 1.7% don't know. (This totals 100%.) In contrast, 45% of *ComputerCraft* IBM/compatible owners own a PC/XT; 49.3% have an AT; 15.8% a 386SX; 22.6% a 386DX; and 3.4% a 486SX or DX. (This totals 136.1% due to multiple responses that reflect ownership of more than one machine.)

VGA and SVGA color video displays were owned by 57.2% of club members, whereas 69.8% of *ComputerCraft* readers owned one.

In the case of software, the Club study indicates that 96.2% are competent with word-processing, 84.8% with spreadsheet, 83.6% with communications, 80% with database, 70.6% with desktop publishing, 51.3% with computer-aided design software. Our study concludes that 88.9% work with word-processing, 58.5% telecommunications, 56.5% database, 52.7% spreadsheet, 39.4% CAD and 31.5% DTP software.

Also, the Club study asked members if they were knowledgeable or better in language skills; 62.6% reported they were. *ComputerCraft*, in turn, asked readers if they do programming, whereby 86.2% responded affirmatively by noting which languages they work with. Additionally, 28.5% of all club members reported they use DOS with *Windows*, while 40.4% of *ComputerCraft* readers who own IBM/compatibles affirmed this. (Both figures are likely much higher now that Microsoft introduced *Windows* 3.1 in April, after both surveys were completed.)

Occupations differed significantly. Among club members, 21.9% were employed in the computer industry, compared to 43.6% of *ComputerCraft* respondents. For club members' occupations, 28.3% were owners or managers; *ComputerCraft* readers numbered 10.6% in this category. Other occupations were: Club: 33.7%, professional staff; 6.8%, educator; 1.7%, student; 2.1%, unemployed and 14.5%, retired, which totals a rather high 16.6%. *ComputerCraft*: 55.6%, professional staff; 5.1%, educator; 7.5%, student; 3.9% retired, disabled or unemployed.

Interestingly, the studies revealed virtually the same percentage of respondents who recommend, specify or approve PC purchases at work: 76.4%, Club; 75.4%, *ComputerCraft*.

Average club member age was reported to be 48.3 years, while *ComputerCraft* readers' average age was about 39 years.

Both studies bring its people into sharper focus. You can get a copy of the Club report, which contains much in-depth information (PC-use patterns to shareware to likes about user groups) from Lotus Development Corp. (617-693-1554) or by downloading it from APCUG's bulletin board (408-439-9367). If you're really into computing, you should seriously consider joining a PC user group. ■

EDITORIAL STAFF

Art Salsberg
Editor-in-Chief

Alexander W. Burawa
Managing Editor

Dorothy Kehrwieder
Production Manager

Emily Kreutz
Production

Elizabeth Ryan
Art Director

Barbara Terzo
Assistant Art Director

Susan Reale
Artist

Pat Le Blanc
Florence V. Martin
Linda Romanello
Phototypographers

Hal Keith
Illustrator

Bruce Morgan
Photographer

Jan Axelson, Tom Benford, Hardin Brothers, Joe Desposito, Nick Goss, Ted Needleman, Peter R. O'Dell, SF Sparrow, Stan Veit, Wayne Yacco
Contributing Editors

BUSINESS STAFF

Richard A. Ross
Publisher

Art Salsberg
Associate Publisher

Dorothy Kehrwieder
General Manager

Frank V. Fuzia
Controller

Catherine Ross
Circulation Director

Melissa Kehrwieder
Carol Minervini
Data Processing

Denise Pyne
Customer Service

ADVERTISING SALES

Margaret Milanese

(516) 681-2922
FAX: (516) 681-2926

Karen Nauth
Sales Assistant

Offices: 76 North Broadway, Hicksville, NY 11801. Telephone: (516) 681-2922. FAX (516) 681-2926. ComputerCraft (ISSN 1055-5072) is published monthly by CQ Communications, Inc. Subscription prices (payable in US Dollars only): Domestic—one year $18.97, two years $36.00, three years $53.00; Canada/Mexico—one year $21.00, two years $40.00, three years $59.00; Foreign—one year $23.00, two years $44.00, three years $65.00. Foreign Air Mail—one year $76.00, two years $150.00, three years $224.00.

U.S. Government Agencies: Subscriptions to ComputerCraft are available to agencies of the United States government, including military services, only on a cash with order basis. Requests for quotations, bids, contracts, etc. will be refused and will not be returned or processed.

Postmaster: Please send change of address notice to ComputerCraft, 76 North Broadway, Hicksville, NY 11801.

DR DOS Booster

•The June issue of *ComputerCraft* was chock full of useful information. I take issue only with the conclusion drawn in the Software Review article on DR DOS 6.0. DR DOS deserves much more credit than allowed.

In my business at home, I use DR DOS 6.0 exclusively. I don't miss DOS 5.0 at all. Like the reviewer, I also couldn't configure my 386SX to run under *Windows* 3.0 in enhanced mode under DR DOS. Rather than spending too much time juggling parameters and devices, I chose an easier way. As the reviewer states, DR DOS allows you to CHAIN different config.sys files for different configuration choices at boot-up. However, DR DOS has a SWITCH command that can be added to prompt you to make a choice within a single config.sys file. The benefits of this command are: (1) all configuration information is contained in one file (as opposed to two or three very small files consuming far greater hard-drive resources), and (2) by keeping all configuration information in one file, all options are accessible by *Windows*' System Edit function.

Windows 3.0 comes with its own memory manager. Simply use its memory manager if you need the enhanced 386 features. Some programs work better in standard mode, such as *Battle Chess*. So give yourself this choice by using DR DOS' memory manager with the /WINSTD switch. Finally, for DOS-specific applications (when *Windows* isn't run), give yourself the choice of using DR DOS' sophisticated MEMMAX utilities. Since I use this choice primarily for programming (text), I end up with 680K of usable RAM. If I exit *Windows*, I end up with only 570K of usable RAM. A copy of my current config.sys file's choices is as follows:

```
ECHO SELECT CONFIGURATION:
ECHO 1. WINDOWS ENHANCED 386 MODE
     [WINDOWS MEMORY MANAGER]
ECHO 2. WINDOWS STANDARD MODE
     [DR DOS MEMORY MANAGER - NO
     HIGH MEMORY]
ECHO 3. DR DOS APPLICATION MODE
     [DR DOS MEMORY MANAGER - HIGH
     MEMORY]
SWITCH CONFIG1, CONFIG2, CONFIG3
EXIT
:CONFIG1
DEVICE=C:\WINDOWS\himem.sys
DEVICE=C:\WINDOWS\mouse.sys/Y
DEVICE=C:\WINDOWS\smartdrv 768
SET M=ENHANCED
RETURN
:CONFIG2
DEVICE=C:\DRDOS\EMM386.SYS/F=AUTO/
     K=AUTO/B=AUTO/V/R=AUTO/WINSTD
DEVICE=C:\DRDOS\PCKWIN.SYS
DEVICE=C:\SSTORDRV.SYS
DEVICE=C:\DEVSWAP.COM
DEVICE=C:\DRDOS\TVGA\TANSI.SYS
DEVICE=C:\DRDOS\KRAFT\MOUSE.SYS
SET M=STANDARD
RETURN
:CONFIG3
HIDOS=ON
DEVICE=C:\DRDOS\EMM386.SYS/
     F=AUTO/K=AUTO/B=AUTO/V/R=AUTO
DEVICE=C:\DRDOS\PCKWIN.SYS
DEVICE=C:\SSTORDRV.SYS
DEVICE=C:\DEVSWAP.COM
HIDEVICE=C:\DRDOS\TVGA\TANSI.SYS
HIDEVICE=C:\DRDOS\KRAFT\MOUSE.SYS
SET M=DIGDOS
RETURN
```

The SET M = " " lines in each choice sets an environment variable (M) that's then used by my autoexec.bat file to run *windows* as "WIN" for enhanced, "WIN/s" for standard or run a DOS menu.

MS-DOS can't match DR DOS' flexibility. Just having the SWITCH command available makes DR DOS a superior product. All the other extras DR DOS offers,

(Continued on page 83)

IBM Software Moves. IBM announced that it enhanced its DOS 5.0 to run on other vendors' hardware. The "refreshed" IBM DOS Version 5.00.1 will also run in a mixed LAN environment, and additionally has the latest corrective service diskettes fixes, including QBasic. Price is $165, and users of any make DOS 2.0 or higher can upgrade for $85.

PRODIGY telecommunications service users can now order IBM's OS/2 2.0 directly from its screens for $139; DOS users can upgrade for $99 and current Windows users can get it for $49. The new operating system also supports DOS, Windows and OS/2 applications in one package. The true multitasking software ships with over 25 utility, entertainment and personal productivity mini-applications, such as a NotePad, Calculator, etc. To obtain the software package, which can also run new 32-bit applications being developed (ore than 1,000 applications in the next year, says IBM), you can order via a toll-free number, 1-800-3-IBM-OS2.

References on Disk. The 1992 IC Master Alternate Source Directory Version 4.1 has been released. The industry-wide cross-reference database of integrated circuits is delivered on four 5-1/4" 360K disks or one 5-1/4" or 3-1/2" high-density disk. The program allows users to add comments up to 19 digits long. By simply typing in a manufacturer and the current or discontinued device number, the program displays all available pin-for-pin and functional equivalent devices in seconds. It's priced at $179.50 from Hearst Business Communications (516-227-1300).

The latest VCR Cross Reference on disk from the International Society of Certified Electronics Technicians (ISCET) allows the user to search by manufacturer for model numbers and description of part numbers, including an on-screen view of all substitutes. Price is $69.95 plus $2 shipping for two 5-1/4" program and data disks or one 3-1/2" disk. To order, call 817-921-9101 (Fort Worth, TX).

Computerized Devices Aid Disabled. University of Washington students, with funding by the National Science Foundation, have created a bevy of devices to assist the disabled in an upper-division class in the design of computer subsystems. The electrical engineering course by Professor Yongmin Kim earns five credits for its students, who often labor up to 20 hours per week for three months, in addition to six lecture hours and six lab hours each week.

PC Software Modem/Fax/Answer Machine Switching. Electronic Technologies' newest version of RemoteControl software automatically directs phone calls to a modem, PC fax board or answering machine. A Ring-Back version adds security to the host computer system. $35. The Rochester, MI-based company also added postal bar codes to its Postal Service software, which allows volume mailers to take a 2-cent discount for presorted letters and a 1.3-cent discount for post cards that include a nine-digit ZIP code and a preprinted postal bar card. A companion program, Bar:ReplyMail, prints camera-ready artwork for Business Reply Mail in less than one minute on any HP/compatible laser printer. Each costs $99. Call 313-656-0630 for more information.

Printer Cleaner

Scrubex and Printex from Aspect are based on a new approach to cleaning the paper path of laser printers, impact printers, copiers and fax machines. These systems utilize a multilayer porous sheet treated with a solvent that feeds through the paper path. Each sheet has a number of "tails" on each side with brush-like fringe on it that gently "scrub" the printer components. The product corrects problems associated with toner, label adhesives, paper dust, dirt and other foreign matter. Each

package consists of five sheets hermetically sealed in individual foil packs. $25. *Aspect, Inc., 57 Eisenhower Lane S., Lombard, IL 60148; tel.: 708-627-9600; fax: 708-627-9601.*

CIRCLE NO. 22 ON FREE CARD

New Notebook Computer

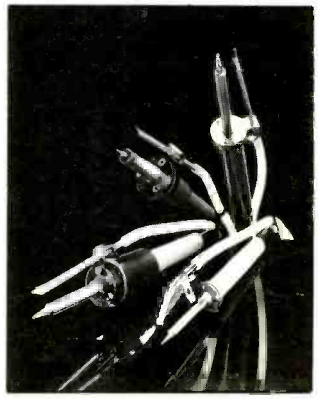
Bondwell's new 386SLF notebook computer, built around the Intel 386SL CPU, features an automatic rest mode, "any key resume," 3 hours of battery operation between charges, 2M of RAM expandable to 8M, cache memory, 80M hard drive and a 2,400 bps V42.bis modem. A second version is available with a 60M hard drive. The unit is housed in a 11.7″ × 8.6″ × 1.5″ aluminum case and weighs 5.2 pounds (without external floppy drive). It has a full-stroke 81-key keyboard and an LCD

VGA display with 64 levels of grayscale. MS-DOS 5.0 and some utilities are included. $2,499. *Bondwell Industrial Co., 47485 Seabridge Dr., Fremont, CA 94538; tel.: 510-490-4300; fax: 510-490-5897.*

CIRCLE NO. 24 ON FREE CARD

Fume Extractor

PACE's TIP-EVAC is a new extraction system that addresses the problem of hazardous fumes created during hand soldering operations. Toxic fumes are extracted right at the soldering iron's tip, before they reach an operator's breathing zone. Contaminants are then removed by a high-efficiency central filtration unit. TIP-



EVAC extractor tubes are available in a variety of clip-on and screw-on versions that fit most popular brands of soldering irons. *PACE Inco., 9893 Brewers Ct., Laurel, MD 20723; tel.: 301-490-9860; fax: 301-498-3252.*

CIRCLE NO. 26 ON FREE CARD

PC DVM Card

Prairie Digital's Model 70 is a complete 5½-digit DVM virtual instrument including IBM/compatible software, cable and power supply. It's a single-channel measuring device but has provisions for multi-station capability. Up to 32 Model 70s can be daisy-chained together and addressed independently from one RS-232 port. Standard input range is from −2 to +2 volts dc (optional custom ranges are available at additional cost). The software features relative, max/min, accurate 5.5 or fast 4.5 modes, data logging to printer and disk data files, time/date stamping, adjustable sampling times, and 32-channel display capability. $239. *Prairie Digital, Inc., 846 17 St., Prairie du Sac, WI 53578; tel.: 608-643-8599 or 608-643-6754.*

CIRCLE NO. 25 ON FREE CARD

Large Video Monitor

ViewSonic's Model 6FS 15″ multi-frequency noninterlaced video display monitor offers resolution up to 1,024 × 768, 72-Hz refresh rate at highest resolution and 0.28-mm dot pitch. Its flat-square display gives up to 36% more viewing area than a typical 14″ monitor. Actual display area is 280 × 210 mm. Like other advanced monitors in the ViewSonic line, the Model 6FS has an intelligent digital control system with 32 display formats (16 preset and 16 user-programmable). Circuitry automatically adjusts to vertical frequencies from 50 kHz to 90 kHz, horizontal frequencies from 30 to 60 Hz. Video bandwidth is 80 MHz. $799. *ViewSonic 12130 Mora Dr., Santa Fe Springs, CA 90670; tel.: 800-888-8583; fax: 310-946-1618.*

CIRCLE NO. 23 ON FREE CARD

Free Programming Guide

Building Block Software has issued a technical report that provides a blueprint for writing parametric-design programs. *How To Write A Parametric Program* describes the steps to follow in creating a script of CAD instructions to automatically design members of a family of parts. Following the recipe described, a programmer can write any type of parametric design program. An example program creates drawings of flat patterns for a family of boxes and illustrates the design and programming steps involved. The report is valuable for software product developers, engineers and consultants who write custom programs or enhance or augment existing CAD/CAM systems. *Building Block Software, 371 Moody St., Waltham, MA 02154; tel./fax: 617-899-4350.*

CIRCLE NO. 28 ON FREE CARD

Laptop-To-TV Video Converter

Laptop TV from Willow Peripherals is a VGA-to-TV-video converter designed as a portable peripheral. With Laptop TV, portable PCs can now generate computer text and graphics presentations and output to TV sets, monitors and VCRs in resolutions up to 640 × 480 with full 24-bit color. Used with a desktop computer, Laptop TV saves a valuable expansion-slot since it attaches to the VGA port.

The converter is available with either NTSC or PAL standard video outputs. It's designed to work with any VGA-equipped computer and Apple Macintosh computers, the latter when equipped with an accessory adapter. $1,195. *Willow Peripherals, 190 Willow Ave., Bronx, NY 10454-3596; tel.: 212-402-9500; fax: 212-402-9603.*

CIRCLE NO. 27 ON FREE CARD

Low-Cost LAN

Data Spec's Let's Talk peer-to-peer LAN can link together as many as 200 personal computers into a network. It enables the user to transfer files from one computer to another, send messages to other users on the network and share printers. Files in any computer are accessible to any user on the system (confidential files can be protected). Electronic messages can be easily exchanged with other users on the network without interrupting the current application program. When Let's Talk's Chat feature is activated, a double window allows the user to type in messages on one side while receiving responses on the other.

Installation consists of plugging a small interface module into the parallel port of the IBM/compatible. Any printer connected to the computer is still active, and a simple daisy-chain of modular telephone cord connects the computers. Copying a file that runs under DOS or *Windows* completes the installation.

Let's Talk can ultimately expand the network to as many as 200 stations and 600 printers. $140 per computer. *Data Spec, 9410 Owensmouth Ave., Chatsworth, CA 91311; tel.: 818-772-9977; fax: 818-718-8626.*

CIRCLE NO. 29 ON FREE CARD

Battery-Operated EPROM Eraser

The DE-1 from UVP is a portable EPROM eraser designed with the field technician and hobbyist in mind. Because it's battery operated, work won't be limited by the availability of a wall outlet. EPROMs can be completely erased in as short a time as 2 minutes. Measuring

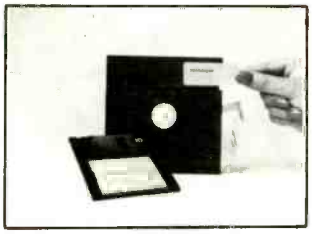
1.5" × 3" × 6.75", the DE-1 weighs just under 9 ozs. Battery life, from four AA cells, is estimated at 5 hours. $60. *UVP, Inc., 5100 Walnut Grove Ave., San Gabriel, CA 91778; tel.: 800-452-6788.*

CIRCLE NO. 33 ON FREE CARD

Removable Labels

Merritt Computer Products' "Labelit" is a removable labeling system for diskettes. Affix the specially designed labelholder on your diskette and insert a card label that lists the contents of the diskette. If modifications are needed, just push the label out of the holder with a flick of the thumb and replace it. If one side of the label isn't enough, use both sides of the reversible label.

Each Labelit kit consists of a quantity of label-holders plus specially designed card labels in five colors. Labelit has custom



designed versions for 3½" and 5¼" diskettes, as well as data cartridges and VHS tapes. Label refills are also available. $8 and up. *Merritt Computer Products, Inc., 5565 Red Bird Center Dr., Ste. 150, Dallas, TX 75237; tel.: 214-339-0753; fax: 214-339-1313.*

CIRCLE NO. 30 ON FREE CARD

Instant Business Cards

CardsNOW from Topitzes & Associates enables users to generate business cards instantly with a personal computer and laser printer. The program comes with 10 styles of type and 25 clipart symbols for personalizing cards. You can scan

in company logos and place them on a card (PCX file format), or Topitzes can be used to scan them in.

The program comes with 25 sheets of paper stock with microedge perforations for 10 cards. Other stock can be used. Topitzes can also provide preprinted cards that lack only the name of the company.

Requirements are DOS 2.0 or later, HP LaserJet Series II or compatible printer and 256K of RAM memory. A hard disk is recommended. $39. *Topitzes & Associates, 6401 Odana Rd., Madison, WI 53719-1158; tel.: 800-233-9767; fax: 608-273-8804.*

CIRCLE NO. 32 ON FREE CARD

Personal Weather Station

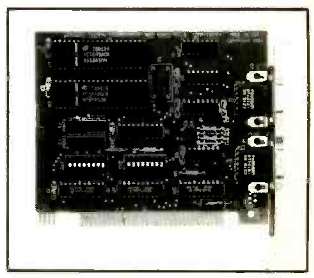
WeatherPort's WindStation plugs into the game port of an IBM/compatible to measure local weather conditions. It includes a rugged wind-vane/anemometer that mounts on an exterior rooftop pole or antenna mast. Average wind speed, wind direction and peak gust value are shown with the basic model. Sensors for outside temperature (displaying 24-hour extremes) and wind chill are available. Historical data can be maintained on-disk for subsequent plotting or trend analysis using most spread-

sheet programs. Specifications: wind speed range, 0 to 125 mph; wind speed resolution, ±0.1 mph; wind direction resolution, 22.5°. *$130. WeatherPort, Inc., 12036 Nevada City Hwy., Grass Valley, CA 95945; tel.: 916-274-8100; fax: 916-274-6429.*

CIRCLE NO. 31 ON FREE CARD

Serial I/O Card With Extended Interrupts

Sealevel's model Comm+232/EX provides two additional serial ports for XTs/ATs/compatibles with the AT extended interrupts. Features include two independent RS-232 channels, selectable addresses (COM1 through COM4, or any other I/O address through 3FF hex), selectable/shareable interrupts (2 through 5 on XTs and 10, 11, 12 and 15 on AT connector), all RS-232 modem



signals, dual male DB-9 connectors, 16550 buffered UART available, SCO XENIX and interrupt buffering drivers available. $179. *Sealevel, 102 W. Main St., Liberty, SC 29657; tel.: 803-843-4343.*

CIRCLE NO. 34 ON FREE CARD

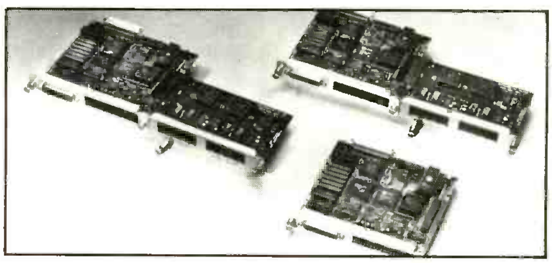
Remote-Console Version 3.0

Remote-Console from the company of the same name is a remote-access communication and control system designed to link two IBM/compatible or one computer and one H/Z-19 terminal via modem or direct cable connection. It's said to provide complete control of the host computer from almost anywhere, and it can run virtually any program on an unattended host computer from a remote location.

Remote-Console is also LAN compatible. A fast file-transfer feature allows a user to link desktop 5¼" and remote 3½" floppy drives to copy files at transfer speeds up to 115,200 bps. Version 3.0 adds remote printing, "call-back," and "auto-call" capabilities and high-speed remote screen updates. A new diagnostic tool is included as well. Requires DOS 3.0 or later and 22K of RAM and support COM1 through COM4 ports. It ships on single 360K disk. $69. *Remote-Console, PO Box 888, Oceano, CA 93445; tel.: 805-481-5687.*

CIRCLE NO. 35 ON FREE CARD

LaserJet IIISi Print Sharing



Belkin Components has three new printer-sharing devices specifically for the HP LaserJet IIISi. Each is designed to be installed in the IIISi Optional I/O (MIO) port. The Laserlink IIISi allows multiple-computer users to send print jobs simultaneously. It queues the jobs in its buffer and sends the data to the IIISi on a FIFO basis. The print buffer includes 1M of RAM, expandable to 4M. Low-cost modular telephone-style cables are used between serial port adapters and the Laserlink IIISi. Buffer memory is dynamically allocated on an as-needed basis. Laserlink functions with Macintosh, mini and mainframe computers, as well as IBM-type computers. Models range from six serial ports and one parallel port to 14 serial ports and one parallel port. From $795. *Belkin Components, 1303 Walnut Park Way, Compton, CA 90220; tel.: 213-898-1100; fax: 213-898-1111.*

CIRCLE NO. 36 ON FREE CARD

WordPerfect Add-In

Perfected! from Key Functions automates more than 150 routine administrative tasks inside *WordPerfect* (DOS) 5.1, all at the touch of a keystroke. It maintains a database in *WordPerfect* format with easy sorting and selecting. Users can prepare mass or specialized mailings in short order. The program has pre-defined forms for invoicing (automatic sequential numbering), name tags, filing systems, petty cash, financial reports and fax cover sheets, among others.

Perfected! includes sample laser printing supplies for creating labels, rolodex cards, routing slips, filing labels and name tags. Requirements: IBM/compatible with DOS *WordPerfect* 5.1 (286 or later recommended), 3M hard-disk space and HP/compatible laser printer. $150. *Key Functions Inc., 60 Pleasant Blvd., Toronto, ONT M4T 1K1, Canada; tel.: 416-928-1943; fax: 416-928-1116.*

CIRCLE NO. 37 ON FREE CARD

What's New!

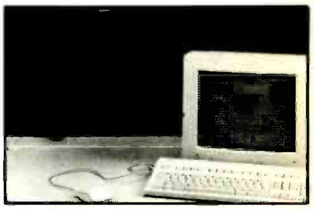
Remoter Power On/Off Switch

Server Technology's new Power On/Off is an improved telephone-activated power control unit that sits in-line between a telephone outlet and remote host PC's modem. It monitors standard telephone line ring and on-hook signals. When Power On/Off detects a programmable number of ring signals on an incoming call, it activates power to the host PC. During power-up, the host PC loads the appropriate communications software via its autoexec.bat. When the unit detects the on-hook signal, it pauses for a programmable delay and then powers down all hardware devices under its control. $169. *Server Technology Inc., 2332-B Walsh Ave., Santa Clara, CA 95051; tel.: 800-835-1515; fax: 408-988- 0992.*

CIRCLE NO. 38 ON FREE CARD

CAD Task Light

The CAD-Room Vienna from Waldmann Lighting is specifically engineered for the CAD workstation to keep overall illumination levels low enough to maintain VDT contrasts while providing adequate light for reference documents. It uses a built-in parabolic louver that focuses light on the work area to prevent stray light from creating glare on VDT screens or irritating nearby workers.



This is reported to result in less eye fatigue and improved productivity. *Waldmann Lighting Co., 9 W. Century Dr., Wheeling, IL 60090; tel.: 708-520-1060; fax: 708-520-1060.*

CIRCLE NO. 39 ON FREE CARD

Three Large Monitors

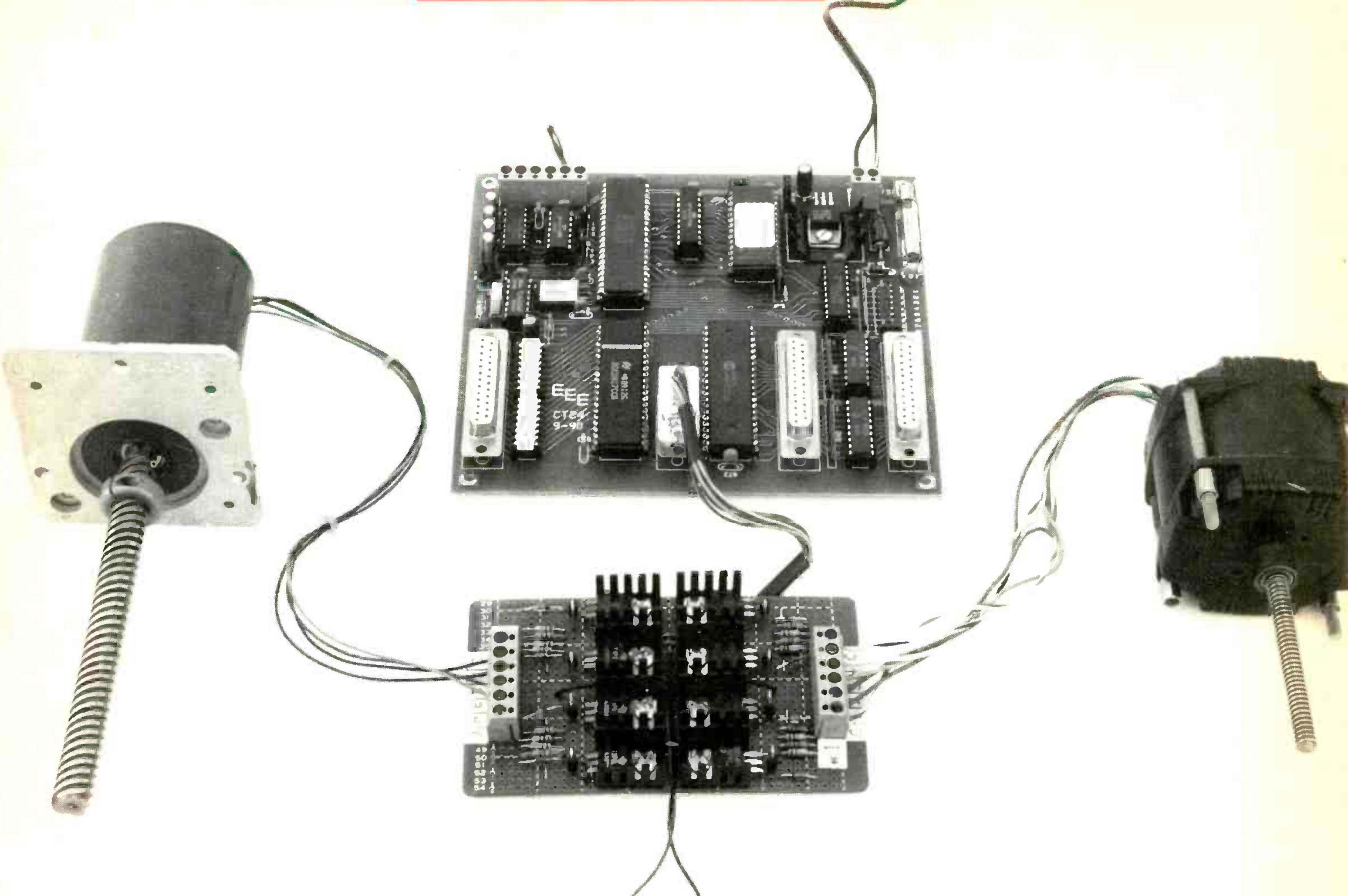
IOcomm has three new monitors that are aimed at the graphics-intensive world of CAD, CAM and DTP. ThinkSync 5 is a 15″ color monitor that provides 1,280 × 1,024 noninterlaced resolution with 0.28-mm dot pitch. It utilizes the entire screen by producing edge-to-edge overscan images, while its flat square tube eliminates glare and distortion. This results in 35% larger active display area than traditional 14″ color monitors. ThinkSync 5 is noninterlaced in 1,024 × 768, as well as 1,280 × 1,024 to minimize flicker, and it supports 70/72-Hz VESA refresh rates to provide absolute picture stability. $795.

ThinkSync 4E is a 14″ color monitor that provides noninterlaced operation at 1,024 × 768 maximum resolution with 0.28-mm dot pitch. The most needed controls are located on the front panel for convenience. $495.

ThinkSync 7M is a 17″ flat square color monitor that produces approximately 50% more area than on a conventional 14″ monitor. On-board microprocessor technology permits the monitor to auto-sync and auto-size video signals ranging from MCGA to VGA to 1,280 × 1,024. This monitor allows you to define and program display mode data including, sync, size and positioning, for maximum viewing comfort in various graphics modes. Multiscan range is 30 to 64 kHz to accept a maximum noninterlaced resolution of 1,280 × 1,024 with 0.28-mm dot pitch. *IOcomm, 12700 Yukon Ave., Hawthorne, CA 90250; tel.: 213-644-6100 or 213-644-6068.*

CIRCLE NO. 40 ON FREE CARD

 By Daniel N. Eggert



Robotics, Control and Monitoring With an Embedded Controller

Using the 8031 and your personal computer as a platform for a wide variety of real-world control applications

Nowadays, every new device or appliance seems to have at least one microcomputer chip buried within it. In most cases, it's an embedded controller like the Intel 8031 eight-bit microcomputer system on a chip. The embedded-controller circuit built around the 8031 featured can be used for robotics, data-acquisition and control applications. With this circuit and a BASIC language interpreter, you can use your personal computer for many other applications besides word processing and number crunching.

A program in EPROM for the circuit and several BASIC program examples are available on floppy disk from the author. These provide a versatile control system commanded by your personal computer via one of its serial ports. Although this article focuses on the use of the BASIC programming language to command and control the system, any programming language that can send and receive data in the formats to be described can be used. The control system's 13 commands allow your personal computer to read the state of any of the circuit board's digital or analog inputs and to change the state of any of the digital outputs.

Two different modes of operation are available for the control system. One circuit assembly can be used in the system if 16 analog inputs and 24 digital I/O lines are adequate for a given application. Up to 16 circuit boards can be chained together in a network to provide a total of 256 analog inputs and 384 digital I/O lines in one system. For robotics and motion-control applications, eight system commands are dedicated to controlling two stepper motors. Any combination of three- or four-phase stepper motors can be used with the stepper driver interface.

About The Circuit

Shown in Fig. 1 is a block diagram of the circuit. Because of its large size, the schematic diagram is shown here in two parts as Fig. 2(A) and Fig. 2(B). Figure 2(A) includes the embedded controller and its program memory, serial interface, clock oscillator for the A/D converter and voltage regulator. Figure 2(B) includes the analog and digital interface and configuration DIP switch port.

The heart of the circuit is 8031 embedded controller *IC1* in Fig. 2(A). Any embedded controller in the 8051 family can be used with this circuit board if jumpers *JP1* through *JP5* are properly configured. This makes the circuit useful for many custom applications.

Because the 8031 doesn't have internal program memory, the program for it would reside externally in EPROM *IC3*. Octal latch *IC2* latches the lower eight address lines during external program and data memory operations. Address lines A0 and A1 latched by *IC2* are also used by 8255 programmable peripheral interface *IC5* in Fig. 2(B). The reset signal for *IC1* and *IC5* is provided by *R1*, *C1* and inverter *IC9E*. Pins 1 and 2 of *J5* can be momentarily shorted together to reset the embedded controller. Jumper *J5* also provides connections to timer inputs T0 and T1 and the interrupt input to the embedded controller INT1.

The RS-232 serial port circuitry in Fig. 2(A) is simplified by the use of MAX232 *IC7*, which has built-in charge pump voltage converters and two complete RS-232 receivers and transmitters. Controller *IC1* has a built-in serial interface with a programmable baud rate generator.

A frequency of 11.0592 MHz was chosen for crystal *Y1* so that *IC1* can generate communication rates from 150 to 19.2K baud. The 8031 applications program would typically read the baud rate selected via the settings of DIP switch *SW1* in Fig. 2(B) immediately after the circuit is powered up or reset. The RS-232 serial interface connections are made at terminal strip *TS1*.

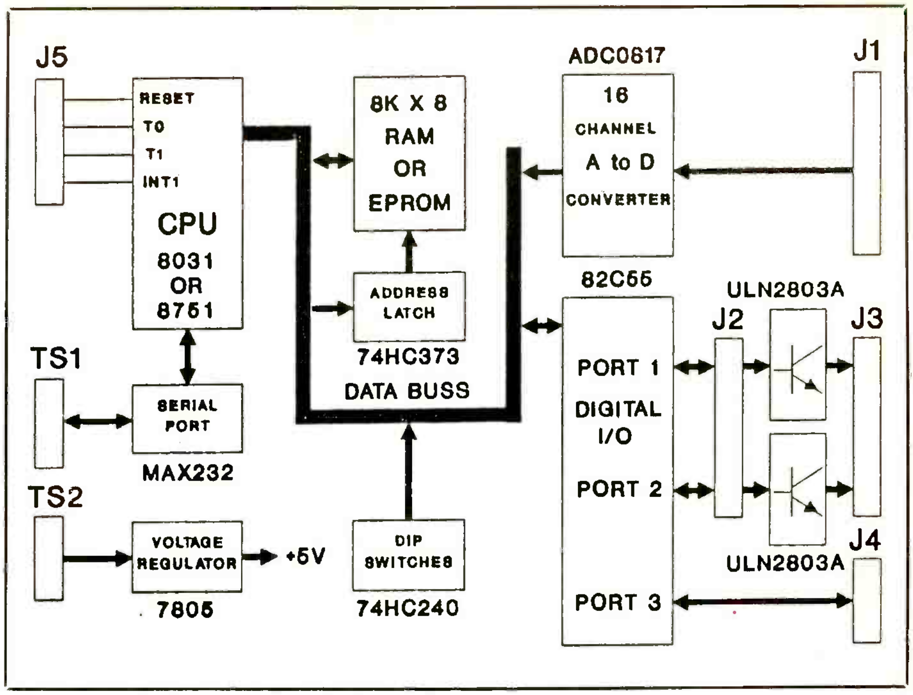


Fig. 1. Block diagram of embedded controller system.

Notice that *TS1* has master and slave port connections. The master RS-232 port connections are always used when interfacing to the host PC. The slave RS-232 port repeats the receive and transmit RS-232 signals through the circuit. The master RS-232 input at lug 4 of *TS1* is converted to a TTL level signal by *IC7*. This signal enters *IC7* at pin 13 as RS-232 and exits at pin 12 as TTL.

This TTL-level signal is the received data to *IC1*. This same signal is converted back to RS-232 by *IC7*. It re-enters *IC7* at pin 10 as TTL and exits at pin 7 as RS-232. Therefore, the master RS-232 received data is repeated through the circuit board from lug 4 of *TS1* to the slave RS-232 output at lug 1.

Slave RS-232 received data from lug 3 of *TS1* repeats through the circuit to lug 6 of *TS1* in a slightly different way. This signal is repeated through *JP2* is jumpered from lug 2 to lug 3. The slave receive data is ANDed with the transmit data from *IC1* by gate *IC8C* when they're both at TTL signal levels.

Figure 3 illustrates how several circuits can be wired together in a chain-like manner to form a network. When a network of circuits is interconnected in this way, all circuits simultaneously receive the same serial data from the host PC, and the circuits must take turns transmitting data to the host PC.

Programmable peripheral interface *IC5* provides three eight-bit digital I/O ports. Connections to digital I/O Ports 1 and 2 are made at *J2* and *J3*, as shown in Fig. 2(B). Connector *J2* can be used for TTL-level input or output. The digital I/O at *J2* is pulled up to +5 volts so that a switch contact can be used as an input source.

The digital outputs for Ports 1 and 2 at *J3* are open-collector from Darlington-transistor arrays *IC10* and *IC11*. These Darlington arrays are capable of sinking 500 milliamperes and withstand up to 50 volts in the off condition. Maximum continuous power dissipation of one output is 1 watt, and total power dissipation for the complete IC is 2.25 watts. Connector *J4* is for digital I/O Port 3, which is a general-purpose eight-bit TTL I/O port.

In Fig. 2(B), *IC4* is an ADC0817 16-channel multiplexed eight-bit analog-to-digital (A/D) converter. The analog input connector is *J4*. Analog inputs are optionally pulled up to +5

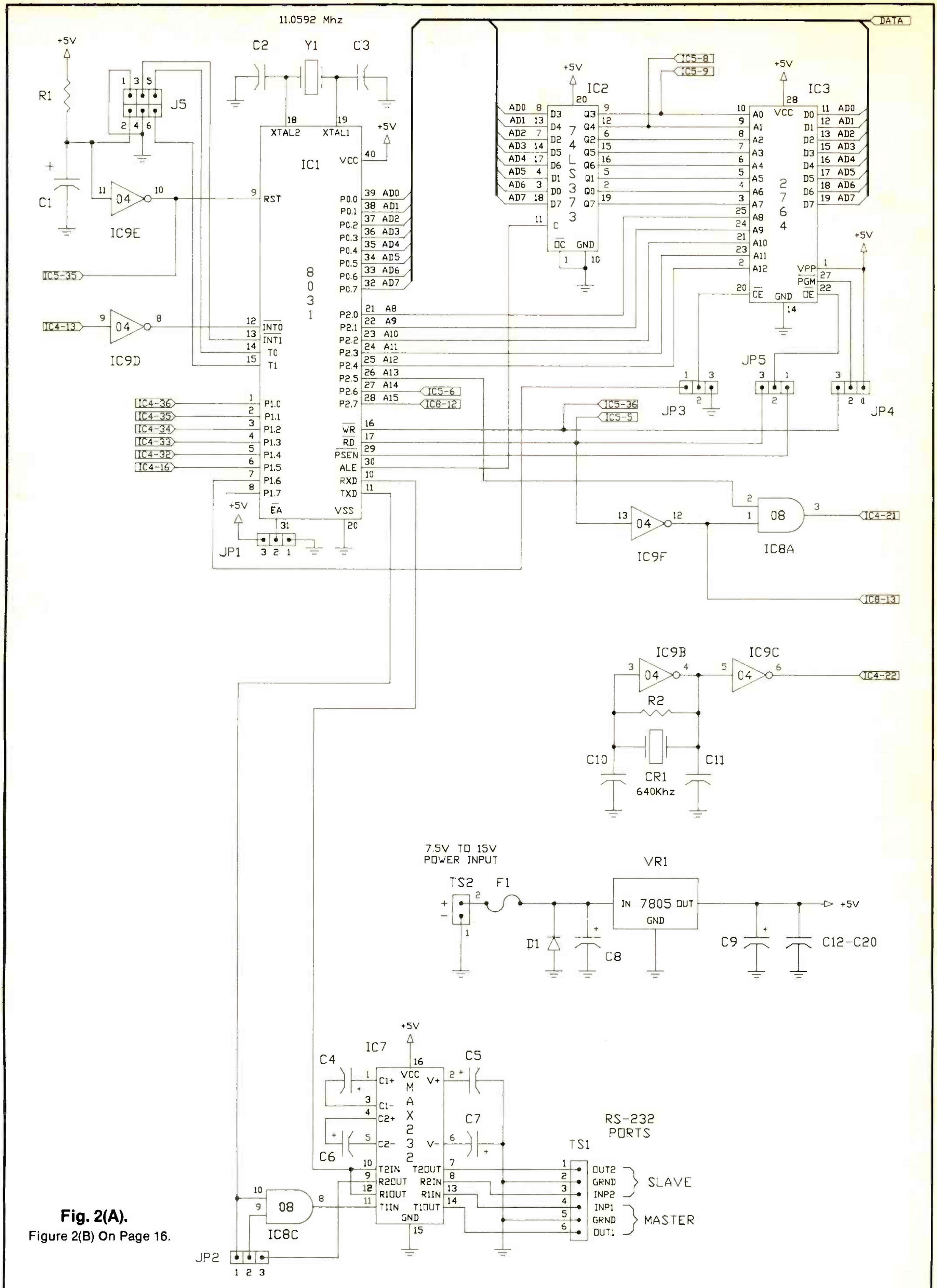


Fig. 2(A).
Figure 2(B) On Page 16.

Fig. 2. Complete schematic diagram of project, shown in two parts.

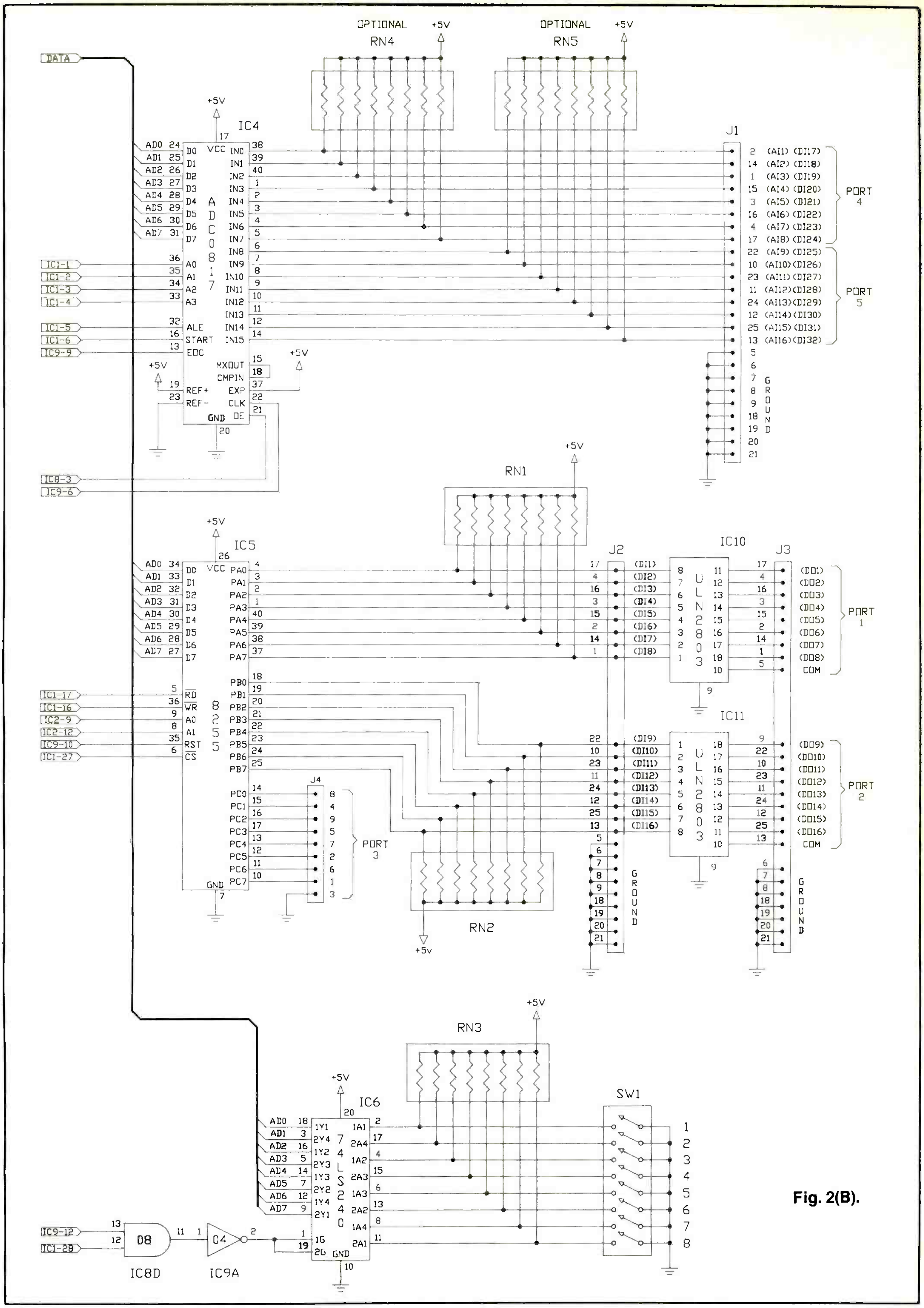

DATA
OPTIONAL
RN4
+5V
OPTIONAL
RN5
+5V
IC4
ADC0817
J1
PORT 4
PORT 5
GROUND
IC1-1
IC1-2
IC1-3
IC1-4
IC1-5
IC1-6
IC9-9
IC8-3
IC9-6
RN1
IC5
8255
J2
IC10
ULN2803
J3
PORT 1
IC11
ULN2803
PORT 2
IC1-17
IC1-16
IC2-9
IC2-12
IC9-10
IC1-27
J4
PORT 3
RN2
RN3
IC6
74LS240
SW1
IC9-12
IC1-28
08
04
IC8D
IC9A

Fig. 2(B).

volts by resistor networks *RN4* and *RN5*, which allows switch contacts to be used as an input source when these inputs are substituted for digital inputs. The clock input signal for A/D converter *IC4* is generated by a 640-kHz oscillator circuit consisting of ceramic resonator *CR1*, inverter *IC9B* and capacitors *C10* and *C11* shown in Fig. 2(A).

Controller *IC1* in Fig. 2(A) controls the A/D converter with I/O lines P1.0 through P1.5 and INT0. Jumpers *JP1*, *JP3*, *JP4* and *JP5* make the circuit usable with all embedded controllers in the 8051 family. Jumper *JP1* selects external or internal program memory. Pins 1 and 2 of *JP1* must be jumpered together to enable use of an 8031 embedded controller. Pins 2 and 3 of *JP1* must be jumpered together if an 8751 or 8752 is used. Pins 2 and 3 of *JP3* must be jumpered together if EPROM *IC3* is used as program memory to keep the EPROM enabled.

In other applications where the 8751 or 8752 embedded controllers are used, pins 1 and 2 of *JP3* must be jumpered together to permit P1.6 of *IC1* to enable optional external RAM or EPROM. Pins 1 and 3 of *JP4* must be jumpered when an EPROM is used for *IC3*. This applies +5 volts to pin 27 of the EPROM. In other applications where RAM or EEPROM is used, pins 2 and 3 of *JP4* must be jumpered together to connect the write control line from *IC1* to pin 27 of *IC3*. Pins 1 and 2 of *JP5* must be jumpered together when an EPROM is used for external program memory to select pin 29 of *IC1* (PSEN) for use as the output enable for the external program in *IC3*. In other applications when the program is internal to the embedded controller, as in the 8751 and 8752, pins 2 and 3 of *JP5* must be jumpered together to connect the read control signal to optional external RAM or ROM *IC3*.

The circuit uses a simple external data memory addressing scheme to access the three parallel I/O ports, analog inputs and DIP-switch port. Addresses of the three digital I/O ports and their associated control port are decoded by P2.6 at pin 27 of *IC1*. The 8255 is selected when P2.6 goes low during execution of an external data memory read or write instruction. P2.6 is external memory address line A14. Whenever any of the other I/O devices are accessed, P2.6 must be set high to keep the 8255 disabled.

Parallel port addresses are: PORT 1 = 0000H, PORT 2 = 0001H, PORT 3 = 0002H and CONTROL = 0003H. The analog input port address is decoded by *IC9F* and *IC8A* in Fig. 2(A). An analog input value is read when P2.5 (A13) at pin 26 of *IC1* goes high during execution of an external memory read instruction. The address used to read an analog input after a complete conversion is detected by *IC1* is 6000H. Note that bit 14 in the address is also set, which sets P2.6 (A14) high to keep the 8255 disabled.

The configuration DIP switch port address is decoded by *IC8D* and *IC9A*. The DIP switches are read when P2.7 (A15) at pin 28 of *IC1* goes high during execution of an external

PARTS LIST

Semiconductors
D1—1N4004 silicon rectifier diode
IC1—80C31BH 12-MHz embedded controller
IC2—74HC373 octal latch
IC3—27C64-15 8K × 8, 150-ns EPROM (must be programmed—see Note below)
IC4—ADC0817 16 channel A/D converter
IC5—82C55A programmable periheral interface
IC6—74HC240 octal buffer
IC7—MAX232 dual RS-232 interface
IC8—74HC08 quad AND gate
IC9—74C04 CMOS hex inverter
IC10,IC11—ULN2803A Darlington array
VR1-7805 fixed +5 volt fixed regulator

Capacitors
C1,C4 thru C7—10-µF, 16-volt radial-lead electrolytic
C2,C3—22-pf ceramic disc
C8—47-µF, 35-volt axial-lead electrolytic
C9—4.7-µF, 16-volt radial-lead electrolytic
C10,C11—470-pF ceramic disc
C12 thru C20—0.1-µF ceramic disc

Resistors (¼-watt, 5% tolerance)
R1—10,000 ohms
R2—1 megohm
RN1,RN2,RN3—9 × 10,000-ohm single-row resistor network (Digi-Key Cat. No. 770-101-R10K or similar)
RN4,RN5—8 × 10,000-ohm dual-inline resistor network (Digi-Key Cat. No. 761-3-R10K or similar)

Miscellaneous
CR1—640-kHz ceramic resonator (Digi-Key Cat. No. P9946 or similar)
F1-½-ampere fuse
J1,J2,J3—DB-25SC female pc-mount connector
J4—DB-9SC female pc-mount connector
J5—Six-pin dual-row header (2 × 3)
JP1 thru JP5—Three-pin single-row header (1 × 3)
SW1—Eight-position DIP switch
TS1,TS2—Four each two-position pc-mount screw-type terminal strip (Digi-Key Cat. No. ED1601 or similar)
Y1—11.0592-MHz Type HC18/U crystal (Digi-Key Cat. No. X078 or similar)
Printed-circuit board or perforated board with holes on 0.5" centers and suitable Wire Wrap hardware (see text); sockets for all DIP ICs; five shorting socket jumpers; two pc-mount fuse clips for F1; suitable enclosure (see text); heat sink for VR1 (optional—see text); 4 × 40 hardware; hookup wire; solder; etc.

Note: The following items are available from D. Eggert, 3527 E. Edgemere Dr., Appleton, WI 54915: Silk-screened and solder-masked double-sided pc board with plated-through holes, $20; 27C64 EPROM programmed with control system, $7.50; 11.0592-MHz crystal, $1.80; 640-kHz ceramic resonator, $1.50. Also available are: kit with all circuit-board components, except programmed EPROM but including pc board, $87.50; a floppy disk containing BASIC program examples that exercise all system commands, $5; floppy disk containing 8031 assembly source code listing for control system and a binary file for use in programming your own EPROM, $10. For diskettes, specify 3½" or 5¼" format. Add 3.5% of total order for shipping (minimum $2.50). Wisconsin residents, please add 5% sales tax.

Component Sources

Digi-Key Corp.
701 Brooks Ave.,
P.O. Box 677
Thief River Falls, MN 56701-0677
1-800-344-4539

Jameco Electronics
1355 Shoreway Rd.
Belmont, CA 94002
415-592-8097

data memory read instruction. The address used to read the DIP switch port is C000H. Note that bit 14 in the address is set here also to keep the 8255 disabled.

If optional RAM or ROM *IC3* is used in applications where an 8751 or 8752 is used in this circuit, pins 1 and 2 of *JP3* must be jumpered together. Pins 1 and 2 of *JP4* must be jumpered together if the memory device used is EPROM, while pins 2 and 3 must be jumpered together if the memory device is read/write (RAM or EEPROM). The application program must keep digital I/O line P1.6 high, except during an external data memory read or write to *IC3*. When addressing *IC3*, the three highest address bits must also keep the I/O ports disabled. Address bit 13 and 15 must be cleared, and bit 14 must be set whenever memory at *IC3* is accessed. The addresses of the optional 8K of external data memory at *IC3* are 4000H to 5FFFH.

The circuit is powered by any dc source capable of delivering 7.5 to 15 volts. It connects to *TS2*. Fuse *F1* and diode *D1* protect the circuit against accidental input voltage polarity reversal. Voltage regulator *VR1* provides the +5 volts needed to power the ICs. Circuit power consumption is significantly lower if CMOS devices are used for *IC1*, *IC3* and *IC5*. A heat sink isn't needed for *VR1*; the circuit uses only CMOS devices.

A circuit made up of only CMOS devices draws about 45 milliamperes with 12 volts applied to *TS2*. A circuit that uses CMOS devices for only *IC4*, *IC7* and *IC9* draws about 275 milliamperes with a 12-volt input. Low power ("LS") or CMOS ("HC") series ICs can be used for *IC2*, *IC6* and *IC8*. For the A/D converter's ceramic resonator clock oscillator to function correctly, *IC9* must be a 74HC04.

Pins 5 and 13 of *J3*, labeled COM in Fig. 2(B) aren't the common grounds for the Darlington-array open-collector outputs. All open-collector outputs from *IC10* and *IC11* have an internal clamping diode. These eight clamping diodes, inside each array package, have a common connection at pin 10. These diode connections are available at *J3* and are labeled COM. Use the pins labeled GROUND for output common ground.

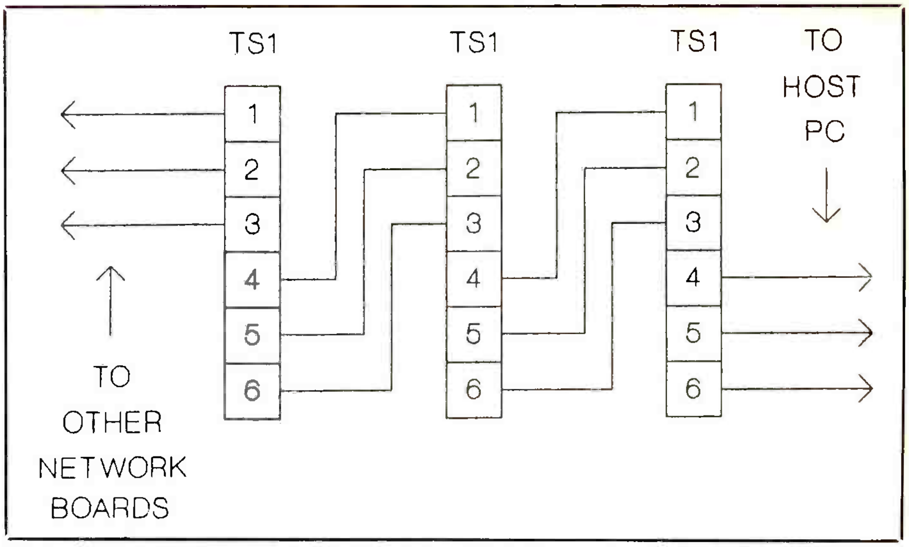


Fig. 3. Multiple circuit cards (up to a total of 16) can be connected together as shown here to form a network.

Take note of the digital I/O pin assignments at the right side of Fig. 2(B). As you can see, pin assignments for the I/O lines of Port 1 are the same at *J2* and *J3*. However, an offset exists between connectors *J2* and *J3* for the digital I/O lines of Port 2.

Up to 16 project boards can be chained together to form a network. If position 4 of *SW1* is ON when a given project card is reset or power is turned on, the board starts up in network mode. When in network mode, positions 5 through 8 of *SW1* determine the address of the board. Every board in the system must have its own unique address between 0 and 15.

Position 5 of *SW1* is the least-significant bit (LSB) of the four-bit binary address, position 8 the most significant bit (MSB). The host PC's transmit RS-232 signal is repeated through each board in the chain by via the slave RS-232 port. Therefore, all boards in the network receive commands simultaneously. All transmit RS-232 signals of the circuit boards in the network are ANDed together by gate *IC8C* of each circuit board in the chain, except the last board on the far end of the chain.

Pins 2 and 3 of *JP2* must be jumpered together to AND the transmit signals of each pc board. Pins 1 and 2 of *JP2* must be jumpered together if the board is the last in the chain in network mode or the only board in single-board mode.

Figure 3 illustrates how all the boards in a network are to be interconnected. All system commands function exactly the same in both single-board and network modes. In network mode, the address must be exchanged between host PC and the board in the network that possesses that address before the command word is sent. First, the host PC sends the address of the circuit board for which the command to follow is intended. The PC then waits for the address to be echoed back to it. If address exchange is successful, the PC sends the command word and any appropriate data. Boards in the network that don't possess the address being exchanged skip the command and any associated data transfer.

The control system offered in EPROM (see Note at end of Parts List) for the embedded controller circuit is a powerful building block for data-acquisition and control system design. The control system hardware interfaces to your computer via one of its serial ports. No circuit boards have to be installed inside your personal computer. This allows the use of any personal computer to command and control the system. Even the smallest of laptop computers could be used.

A BASIC programs diskette (see Note at end of Parts List; also see Command Summary box) has several routines that can be used to exercise the control system. These programs can be used as building blocks

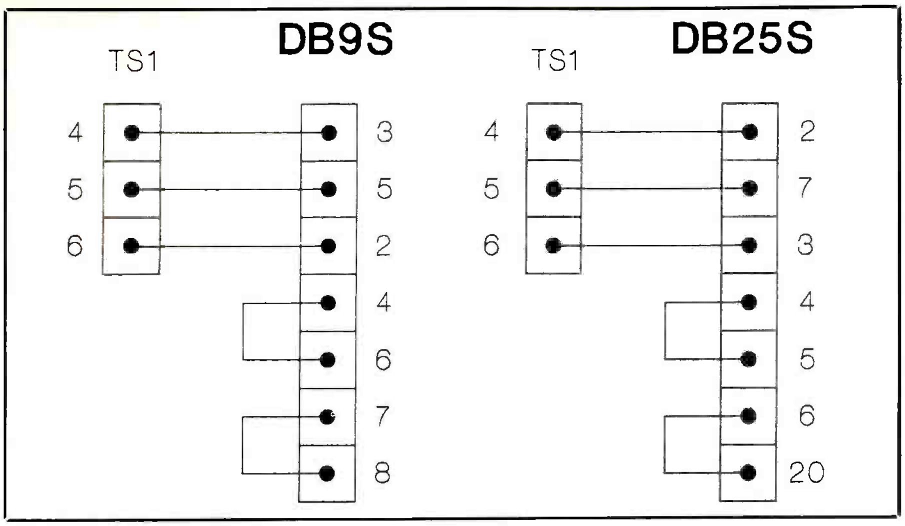


Fig. 4. Interface cable wiring details.

when developing your own applications software.

Construction

If you're experienced with Wire Wrap techniques, you can build this circuit on perforated board that has holes on 0.1″ centers, using suitable Wire Wrap hardware. No fabrication guides for a printed-circuit board are given here because such a board is virtually impossible to make at home due to the fact that it requires plated-through holes with which to bridge its double-sided trace pattern. If you wish pc construction, you can purchase a ready-to-wire board from the source given in the Note at the end of the Parts list. This board has plated-through holes, silk-screened component guide and solder masked top and bottom.

If you use a pc board, begin construction by installing the sockets for the DIP ICs. If you Wire Wrap the circuit, arrange and orient the components in a manner similar to that in the lead photo. Whichever method of construction you use, do *not* plug the ICs into the sockets until after you've conducted preliminary voltage tests.

After mounting the IC sockets, install and solder into place the resistors, capacitors, crystal, ceramic resonator and jumper pins. Use double-sided foam mounting tape to secure *Y1* to the surface of the board. Make sure that the electrolytic capacitors are properly polarized before soldering their leads into place. Then mount screw terminal strips *TS1* and *TS2*, DB-9S connector *J4* and DB-25S connectors *J1*, *J2* and *J3* in their respective locations.

You can use any type of enclosure that will comfortably accommodate the circuit-board assembly. If you plan on including a power supply in the same enclosure instead of using an external one, select an enclosure that's large enough to accommodate its circuitry as well. The enclosure could also accommodate analog conditioning circuitry and terminal strips for connection to I/O devices in your particular application. As an alternative, you can mount the circuit-board assembly in an existing equipment enclosure.

When you finish wiring the circuit board and before mounting it in its enclosure, check to make sure all components (except the DIP ICs, which shouldn't be installed at this time) for correct values/types and orientations.

When you're satisfied that all is okay, turn over the board and carefully examine it for missed and poorly soldered connections and solder bridges, the last especially between closely spaced IC pads and conductors. Solder any missed connections and reflow the solder on any connection that appears suspicious. Use desoldering braid or a vacuum-type desoldering tool to remove any solder bridges located.

Install the shorting jumpers at JP1 through JP5 for your particular application. When you're certain that the circuit is properly wired and soldered, mount it inside the enclosure using ½″ metal spacers and 4-40 machine hardware. Then prepare the cables that are to interconnect the project with your computer. Your PC will have either a 9- or 25-pin serial-port connector. Use Fig. 4 as a guide when fabricating one of the two possible types of cables needed for interconnect to your PC's serial port.

Initial Checkout

A dc voltmeter or a multimeter set to the dc-volts function is the only test instrument you need to check out the project. Clip the common lead to a suitable circuit ground point, such as one of the mounting holes at the corners of the pc board. With no ICs installed in the sockets and the interface cable not connected from your PC, power up the circuit via *TS1*. Make certain you properly polarize the connections.

With power applied to the project, touch the "hot" probe of the meter to each of the *IC1* through *IC9* V_{cc} socket pins. You should obtain a reading of +5 volts in all cases. If you fail to obtain the proper reading at any one or more points in the circuit, power down the and rectify the problem. Check to make sure that all components are in their correct locations and are properly oriented and make sure that all components are soldered into place.

Once you obtain the proper readings, power down the circuit and allow the charges to bleed off the electrolytic capacitors. Then plug the ICs into their respective sockets. Make sure each is in its proper socket and is properly oriented and that no pins overhang the sockets or fold under between ICs and sockets.

System Configuration

Configuration jumpers *JP1* through *JP5* on the circuit board must be correctly installed for the project to operate properly as follows:

JP1—Jumper from pin 1 to pin 2 for external program memory.

JP2—Jumper from pin 1 to pin 2 for single-board mode or for the last circuit board on the far end of the chain in network mode. Jumper pins 2 and 3 for all other boards in network mode.

(continued on page 79)

Interrupts

What they are and what you should know about how they affect normal system operation

Interruptions happen inside a computer just as they do in the real world. When they occur, the computer stops what it's doing, takes care of the interruptions and then returns to its previous activity. The Intel family of CPUs recognizes four kinds of interruptions, or interrupts, as they're more commonly called. To keep things as confusing as possible, there are two vaguely-related numbering schemes for these interrupts.

Clearing Them Up

Whenever an interrupt occurs, which is a condition that originates outside the CPU, the CPU must stop what it's doing and run a special program in memory called an interrupt handler or interrupt service routine (ISR). Many of these routines are contained in the computer's ROM BIOS. Others are installed by such hardware as hard drives and video adapters. Still others are part of special drivers that are installed when you boot your computer, like the mouse drivers supplied with every mouse. In an important way, almost all of DOS is an ISR, as I'll explain below.

To keep all of these ISRs straight, the CPU gives each one a number between 0 and 0FF hex (255). The CPU reserves room for the addresses of these 256 ISRs in a 1,024 byte look-up table that starts at the beginning of memory (in memory location 0:0). When the CPU needs to execute ISR number 8, for example, it reads the eighth set of four bytes from the ISR look-up table, interprets those bytes as the segment and offset address of the necessary ISR routine and then jumps to that address.

To install an ISR, a program simply puts the appropriate address in the look-up table or uses a DOS service to do the same. The only thing special about an ISR is that it must end with an IRET (interrupt return) instruction. Otherwise, it's simply a routine that's written to do a special task.

ISRs are included in many utility and application programs. All memory-resident programs, for example, include one or more ISRs that determine when they're supposed to "pop up" and get to work. Most top-quality application programs include ISRs that allow them to handle errors like trying to read an empty disk drive or detecting when a printer is out of paper. And communications pro-

CPU-Generated Interrupts*

Interrupt Number**	Processor Type 8088	80286	386	486
00	Divide Error	Divide Error	Divide Error	Divide Error
01	Single Step	Single Step	Single Step Debugging Exceptions	Single Step Debugging Exceptions
02	Non-Maskable Interrupt	Non-Maskable Interrupt	Non-Maskable Interrupt	Non-Maskable Interrupt
03	Breakpoint	Breakpoint	Breakpoint	Breakpoint
04	INTO Detected Overflow	INTO Detected Overflow	INTO Detected Overflow	INTO Detected Overflow
05		Bound Range Exceeded	Bound Range Exceeded	Bound Range Exceeded
06		Invalid Instruction	Invalid Instruction	Invalid Instruction
07		Coprocessor Not Available	Coprocessor Not Available	Coprocessor Not Available
08		Double Exception Detected	Double Exception Detected	Double Exception Detected
09		Coprocessor	Coprocessor Protection Error	(Reserved) Protection Error
0A		Invalid Task State Segment	Invalid Task State Segment	Invalid Task State Segment
0B		Segment Not Present	Segment Not Present	Segment Not Present
0C		Stack Fault	Stack Fault	Stack Fault
0D		Protection Fault	Protection Fault	Protection Fault
0E			Page Fault	Page Fault
10		Coprocessor Error	Coprocessor Error	Coprocessor Error
11				Alignment Check

*Also called "Exceptions." **All Interrupt numbers are in hexadecimal format.

BIOS and DOS Interrupts

Interrupt	Usage
05	Print Screen
10	Video Control
11	Get Equipment List
12	Get Memory Size
13	Disk Services (Floppy & Hard Drive)
14	Serial-Port Services
15	Various Operating-System Hooks
16	Get Keystroke & Set Keyboard Options
17	Printer Services
18	Start Cassette BASIC (IBM Only)
19	Reboot System
1A	System Real-Time Clock
1B	Control-Break Handler
1C	System Timer Tick
20	DOS Terminate Program
21	Most DOS Services
23	DOS Ctrl-C/Ctrl-Break Handler
24	DOS Critical-Error Handler
25	DOS Disk Read
26	DOS Disk Write
27	DOS Terminate & Stay Resident
28	DOS Idle Interrupt
29	DOS Fast Console Output
2E	DOS Execute Command
2F	DOS Multiplex Interrupt

grams almost always have ISRs that can collect from a modem each character as it arrives.

Sorting Out Interrupts

You probably do interrupt your own work occasionally with a thought like "Omigosh, I have to call Mom." Similarly, the CPU can create interrupts (and use interrupt service routines) when specific conditions occur. Usually, these self-interrupts are the result of an error a program might not be able to handle.

For example, the CPU can perform integer division, but it can't divide by 0. If it tried to do so, it would enter an infinite loop that would lock up the system. So, if the CPU receives a division instruction, and if the divisor is 0, the CPU generates an Interrupt 0. Normally, the ISR for division by 0 is installed by DOS. This ISR prints an error message on the screen and immediately aborts the current program.

Because the program aborts so abruptly, the computer is in an undefined state. If you ever see the division by zero message, your best bet is to reboot and then try to recover whatever files the application was in the process of using.

The CPU generates an Interrupt 1 after any instruction if it's running in single-step or trace mode. In this mode, a debugger can let the CPU execute an instruction and then get back control of the system as soon as that instruction is finished. Some computers require that debuggers emulate the CPU in order to trace through a program one step at a time. Intel added the single-step interrupt to the 8086 family to simplify debuggers and make them more reliable.

Interrupt 3 is also used by debuggers. The CPU generates this interrupt whenever it's asked to execute a special single-byte instruction. Debuggers use that byte to mark breakpoints in a program. They can then ask the CPU to execute a section of code at full speed, confident that the CPU will stop when it gets to the breakpoint. Of course, a debugger has to include ISRs for Interrupts 1 and 3 in order to support single-stepping and breakpoints.

The other CPU-generated interrupts occur on special errors, like an invalid machine instruction or a coprocessor instruction if no coprocessor is available. These rarely occurring interrupts are usually handled by the ROM BIOS or by DOS without any message to the user.

Intel reserved the first 32 interrupts for the CPU, but it used only the first five (Interrupts 0 through 4) on the 8088. When IBM designed the original PC, the company ignored Intel's warning that other interrupts could be used with future chips and allocated many of the reserved interrupts to PC-specific functions. The 80286 can generate 13 interrupts internally; the 386 and 486 add a few more. Therefore, there are some conflicts between "normal" use of interrupts (the way IBM intended) and exceptions caused by a CPU error.

Most of these conflicts never show up on the majority of users' machines because they're the result of bugs found before software is shipped. But if your computer repeatedly does something very strange at a specific point in an application program, it may be because of one of these interrupt conflicts.

For example, IBM and all clone makers use Interrupt 5 for the "Print Screen" ISR. But an 80186, 80286, 386 or 486 will call the same interrupt on an internal error called "Bound Range Exceeded." Therefore, if your computer seems to be pressing the Shift-Print Screen keys by itself in a particular application, it's likely generating this error internally. There's not much you can do when this occurs, except report the bug to the application publisher.

Software Interrupts

Any piece of program code—whether it's in an application, utility, DOS or ROM BIOS—can create an interrupt and invoke an ISR. At first, you might think that this is a foolish practice because the program could call the ISR directly without an INT (or Interrupt) instruction. But the use of interrupts adds flexibility to the system because it means that different programs can communicate without knowing each other's address in memory.

For example, programs normally request DOS services with an Int 21 hex instruction. The values in the machine registers when the call is made define which of hundreds of services the program is requesting. It makes no difference to the calling program whether you're using PC-DOS, MS-DOS, or DR DOS. Also, as long as the requested service is supported, the calling program doesn't care whether you are using DOS version 2.1 or version 5.0.

If programs made DOS calls by specific addresses, as they had to do in some early microcomputers, DOS writers would have to guarantee that the same service routines were always at the same address. Later versions of DOS would be a tangled mass of jumps from one location to another as they tried to remain compatible with earlier versions while adding more features.

Also, use of INT instructions instead of direct calls allows programs to substitute their ISRs for original ISRs. It's relatively easy, for example, to write a program that records all DOS service requests as they occur. You simply have to write an ISR routine that installs itself as Int 21 hex and records the values in the registers each time it's

Standard IRQ Assignments & Related Interrupts

Interrupt Request	Related Interrupt	Use On XT	Use On AT And Later
IRQ0	08h	System Timer	System Timer
IRQ1	09h	Keyboard Input	Keyboard Input
IRQ2	0A	LPT2**	***
IRQ3	0Bh	COM2 (and COM4)	COM2 (and COM4)
IRQ4	0Ch	COM1 (and COM3)	COM1 (and COM3)
IRQ5	0Dh	Hard Disk	LPT2
IRQ6	0Eh	Floppy Disk	Floppy Disk
IRQ7	0Fh	LPT1	LPT1
IRQ8	70h		CMOS Clock
IRQ9	71h*		EGA & VGA Retrace Signal
IRQ10	72h		Available
IRQ11	73h		Available
IRQ12	74h		PS/2 Mouse
IRQ13	75h		Math Coprocessor
IRQ14	76h		Hard Disk
IRQ15	77h		Available

*Redirected to INT 0Ah on AT. **On original PC only. ***PIC -2 Cascade (see IRQ9 above).

called and then jumps to the original Int 21 hex ISR.

Several programs use software interrupts besides DOS. Many BIOS routines, including a wide range of video services, are available through software interrupts. So are all mouse services if a mouse driver is installed. Programs share expanded and extended memory with software interrupt calls, and they sometimes request disk, printer and serial port services with INT commands. Nearly all modern programming languages have some facility for making interrupt calls as well, because it's almost impossible to write a professional-quality application without using them.

Hardware Interrupts

Imagine that you are designing the world's first telephone system. How do you let the receiver know when someone is calling? You could require that everyone pick up the telephone every minute or two to check for an incoming call. Or you could install a bell that interrupts whatever the recipient of a phone call is doing when a call actually arrives.

In the same way, a computer can be designed to either "poll" its hardware (check each piece of hardware frequently to see if it needs attention) or wait for the hardware to interrupt its normal tasks. A computer based on interrupts is usually more efficient, but it's also more expensive to design and build.

The CPU (8088 through 80486) has two pins that the rest of the system can use to claim attention. The non-maskable interrupt (NMI), is like a fire alarm going off. It's for use in emergency situations when hardware needs the CPU's attention no matter what else is happening at the moment. This kind of interrupt is called "non-maskable" because it can't be ignored or turned off inside the CPU with software instructions.

The effect of an NMI is immediate. As soon as the CPU finishes its current machine instruction, it executes the Interrupt 2 ISR. On most IBM-compatible computers, the NMI is connected to the RAM parity circuitry. Each byte of RAM is held in nine bits. The first eight bits hold the value; the ninth bit is used to record whether the eight data bits contain an even or odd number of logic 1s.

The parity bit is set each time the RAM byte is written and checked each time the byte is read by electronic circuitry in the computer. If the parity bit is incorrect, at least one of the bits has changed by itself. The parity circuitry then creates an NMI and Interrupt 2. Normally, this ISR displays an error message on-screen. It also locks up the computer to force a reboot on the theory that a parity error could cause a bug in a program that must, at all costs, be avoided.

The NMI is also used by breakout switches, which are debugging devices that let a programmer press a button to stop execution of a program. The debugger installs its own ISR as Interrupt 2 and uses this approach to start debugging mode.

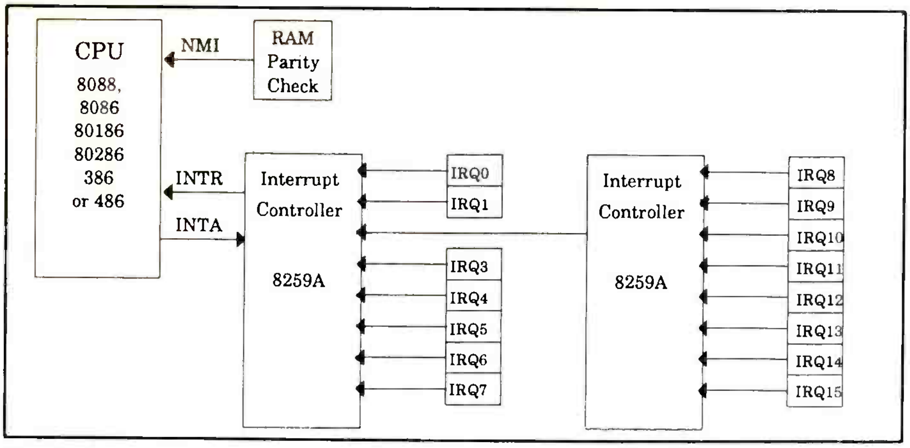
The final type of interrupt is generated by the computer's hardware during normal operation of the computer, not in an emergency. For example, the system timer needs attention 18.2 times per second to update the time-of-day clock, the keyboard needs attention every time the user presses or releases a key, and a mouse needs attention every time it moves or the user presses or releases a mouse button.

These are interrupt requests (IRQs), not demands like an NMI, and can be temporarily turned off or masked by a software command when a program enters a critical section of code that shouldn't be interrupted. The hardware signals that it needs attention on the Interrupt line to the CPU. If IRQs aren't turned off, the CPU finishes its current instruction and then signals that it's ready for the interrupt by sending an "interrupt acknowledge" signal back to the hardware device. The hardware then puts the number of the appropriate ISR on the data bus, and the CPU branches to that ISR.

If there were only one hardware device that could request an interrupt, or if only one device asked for service at a time, this scheme would work fine. But a PC has many devices and no guarantee that they won't need simultaneous service. To arbitrate conflicts between hardware devices, IBM-compatible computers use one (on an 8088 computer) or two 8259A microprocessors called a Programmable Interrupt Controllers (PIC).

Each hardware device that needs interrupt service has access to an interrupt request (IRQ) line on the computer's bus. And each IRQ line goes to a different input pin on a PIC. The first PIC (and the only one on an XT-compatible) is connected to IRQ0 through IRQ7, the second to IRQ8 through IRQ15. When there are two PICs, the second is run as a slave to the first and its output is connected to the master as IRQ2.

Each pin (or IRQ) on a PIC has a priority that can be changed with software instructions. Normally, pin 0 has highest priority and pin 7 has the low-



Block diagram of interrupt circuitry found in an Intel 80xx6-based computer.

est. When an interrupt request shows up at an input pin, the PIC checks to see whether any higher-priority interrupt is currently being serviced. If not, the PIC sends an interrupt signal to the CPU, followed by that pin's Interrupt number. If a higher-priority interrupt is being serviced, the PIC remembers the interrupt and sends it to the CPU when all interrupts with higher priority have been serviced.

The Interrupt Service Routine number associated with each IRQ can be changed with software. Normally, the BIOS sets up the relationships between IRQs and ISRs during initialization and DOS (and nearly all applications) leave them unchanged). On an IBM-compatible computer IRQ0 through IRQ7 are associated with Interrupts 8 through 0F hex. On 80286 and later computers, IRQ8 through IRQ15 are associated with Interrupts 70 hex through 77 hex. Because CPU exceptions can also generate Interrupts 8 through 0F hex, operating systems like *DESQView* and OS/2 re-map the first set of IRQs, usually to the range of 50 through 57 hex.

At least two devices are assigned permanent IRQs by the circuitry in every IBM/compatible computer. The system timer is always attached to IRQ0. It interrupts the computer about 18.2 times per second to allow the BIOS to update the system time and date. Many memory-resident programs also hook into the system timer to check if they should "pop up" or perform background activity.

IRQ1 is permanently assigned to the keyboard. Each time you press or release a key, the keyboard's internal processor sends a code to the keyboard controller in the computer over the keyboard serial cable. When the controller has assembled the entire code, it makes an interrupt request so that the CPU, using BIOS instructions, can read the code and place the keystroke in the type-ahead buffer.

On an 80286 or later computer, IRQ2 (on the master PIC) is tied to the secondary or slave PIC. The slave PIC handles IRQ8 through IRQ15.

The COM2 serial port is connected to IRQ3, and the COM1 port is connected to IRQ4. By convention, COM3, if you have a third serial port, usually shares IRQ4 with COM1, and COM4 shares IRQ3 with COM2. When devices share an IRQ, there's no way for software to determine which device is requesting service. This explains why you can't use COM1 and COM3, or COM2 and COM4, simultaneously on a computer with an XT or AT bus. The Micro Channel bus used in IBM's PS/2s and the EISA bus used in high-end compatible computers get around this problem if the software is written correctly. Therefore, devices in these machines can share IRQs without conflict.

Because COM2 is connected to a lower-numbered IRQ than COM1, it gets priority over COM1. If you have a modem and a mouse connected to two different serial ports, and if you use a *Windows* terminal program or another communication program with mouse support, it's wise to put the modem on COM2 and the mouse on COM1. Otherwise, mouse movements may interrupt data arriving through the modem.

When you want to add a hardware device to your computer, you'll often have to set a switch to tell it what IRQ to use. The IRQ chart accompanying this article will help you determine what IRQs are free. So will most diagnostic programs. It's a good idea to keep a notebook with a list of the devices you have added and the IRQs, DMA (direct memory access) channels and other resources each uses so that you have something to help you decide how to set each new device as you install the hardware.

If you need a list of the interrupts and IRQs used by hardware, DOS, DOS extenders, *DESQView,* memory management (EMS, VCPI, and DPMI), most networks and other software, the best reference is a book titled *PC Interrupts* by Ralf Brown and Jim Kyle (Addison-Wesley Publishing Co., 1991). This 1,000-page book consists almost entirely of lists of Interrupts, the program that uses each, the purpose served by each and possible conflicts with other programs. It's an invaluable reference for programmers and for anyone who needs to know as much as possible about what's going on inside his PC-compatible computer. ■

Microcontroller EPROM Programming

Easy and economical programming options for single-chip computers and microcontrollers

Single-chip computers are simple and elegant. Each contains all the elements of a computer on one integrated circuit, including an arithmetic logic unit (ALU) for calculating, memory for storing programs and data, and input/output (I/O) ports for communicating with the outside world. Single-chip computers are sometimes called microcontrollers because they often provide automated control of machines and other physical devices or embedded controllers because they're often embedded, or fixed, in the devices they control.

Because they're simpler, microcontroller circuits are easier to design and build than larger, more complex computer circuits. Microcontrollers are perfect for projects that require the ability to store information, calculate, compare and decide, but don't require the disk drives, video display, keyboard, operating system and other resources of a full desktop computer.

When a microcontroller's on-chip memory is large enough to fill all of a circuit's memory needs, you're freed from the effort, expense and complexity of adding external memory and its support logic. If program memory is EPROM, or another form of user-programmable memory, you can revise and alter a stored program when you need to. But copying a program into a microcontroller's memory isn't always as simple and straightforward as you might hope.

Although EPROM programmers are widely available, many of them program only standard EPROMs in 24-, 28- and 32-pin packages. Many of those that do program microcontrollers do so only with special adapters, at extra cost. And since different devices often require different adapters, working with a variety of devices quickly gets expensive.

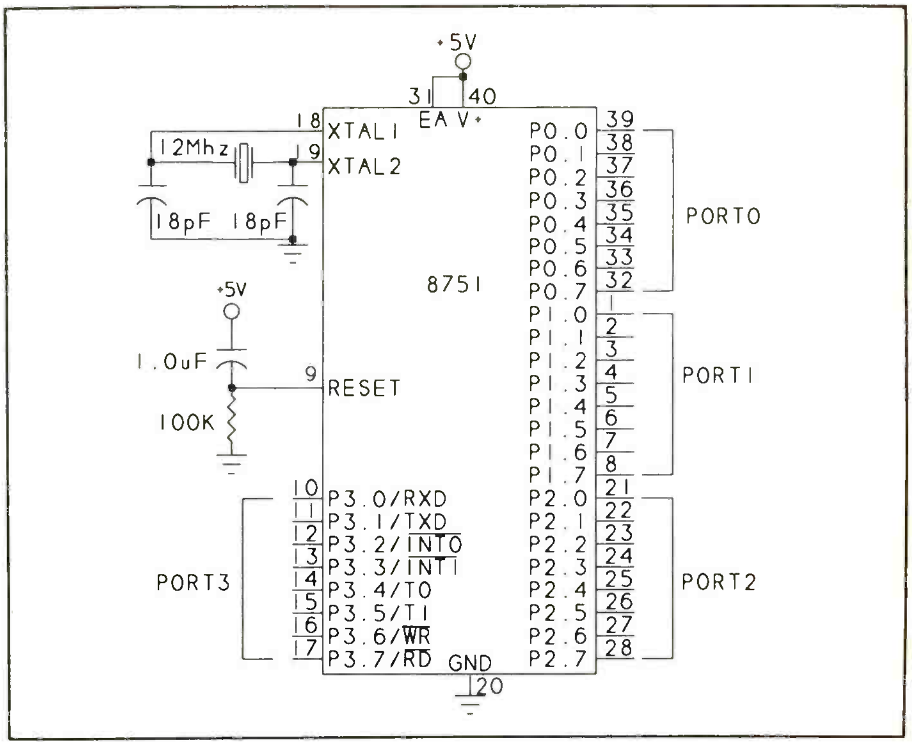


Fig. 1. The 8751 is a true single-chip computer with embedded EPROM for program storage whose 32 port pins can connect to devices it monitors or controls.

In this article, we'll investigate programming options for single-chip computers, including Intel's 8051 family, Motorola's 6801 and related families, and more, with the focus on easy and economical programming options.

Single-Chip Circuits

What kinds of circuits can you design using a single-chip computer? Their circuits can do many of the same types of things other microcomputers do, although the program must be small enough to fit into the limited memory that's available. Popular uses for single-chip computers include data loggers, motor controllers, environmental monitors and controllers, device interfaces and more. If a single chip can do the job, there's no reason to use a more-complex circuit.

Figure 1 shows an example single-chip computer circuit based on the 8751 microcontroller. All that's required are the microcontroller, power supply, timing reference and whatever unique components you connect to the port pins as inputs or outputs.

The power supply for this circuit must deliver stable dc, usually 5 volts. The crystal or other timing reference provides a clock signal that times and controls internal operations of the microcontroller as it executes instructions.

The 8751's EPROM holds the program the circuit executes on power-up. Program memory must be non-volatile, which means that the information it stores is retained when power is removed. The amount of program memory varies from device to device. A basic 8751 has 4K of EPROM, but versions with up to 32K are available.

The 8751 also contains 128 bytes of RAM, which the microcontroller can read from and write to. Unlike EPROM, RAM is volatile memory. Information it stores disappears, or "evaporates," on powering down. Some microcontrollers also have non-volatile read/write memory for more permanent storage. On-chip read/write memory in most devices is limited to a few hundred bytes.

The 8751 has 32 port pins that are arranged as four eight-bit ports. These ports enable the 8751 to access and control the world outside itself. When the 8751 accesses external memory, ports 0 and 2 hold the data and address, and pins 6 and 7 of port 3 provide read and write signals. When there's no external memory, you're free to use all port pins for other functions.

Some common components that might interface to the port pins include motors, switches, relays, displays, sensors, keypads and analog-to-digital (A/D) and digital-to-analog (D/A) converters. Many devices can interface directly to the port pins. If a device requires a high current or voltage, or has other special requirements, you may have to add interfacing circuitry between it and the microcontroller.

In eight-bit microcontrollers, the port pins are usually arranged in groups of eight. You can read and control the pins as a group or singly. For example, when an A/D converter's output is an eight-bit word, you can read the value of the port pins as a single number with a value from 0 to 255. When each bit functions individually—to control a relay, for example—you can read from or write to each bit independently. Many microcontrollers have special bit-oriented instructions that are used for controlling individual port pins.

Because the 8751's port pins are bidirectional, they can be made to function as inputs or outputs. Other microcontrollers have different quantities and combinations of port pins. Some devices have port pins that aren't bidirectional but are dedicated as inputs or outputs.

Memory Options

A program memory in a microcontroller can be any of several types, including ROM, EPROM, OTP EPROM, EEPROM and NV RAM. Each has a particular niche that it fills.

ROM (read-only memory) is mask-programmed into the chip during the manufacturing process. Once the chip has been manufactured, the contents of ROM can't be changed. Microcontrollers with ROM are used in high-volume applications, when a design has been debugged and is ready for production. Just about all microcontrollers are available in a ROM version. In this article we'll ignore the ROM option and instead concentrate on user-programmable types.

EPROM (erasable programmable ROM) is user-programmable, with the aid of an EPROM programmer. EPROMs are manufactured in ceramic packages with a quartz window over the EPROM's storage cells. The EPROM's contents are erased and readied for re-programming the on-chip cells by exposing them to ultraviolet energy through the chip's built-in quartz window.

Most microcontrollers are available with EPROM. EPROM versions are used for experimenting and production of single- and small-quantity projects and as a prototyping tool for programs that will later be mask-programmed into ROM.

EPROM microcontrollers are more expensive than microcontrollers that have no embedded program memory. For example, a basic 8751 sells for $20 to $35 in small quantities, while an 8031, which is identical except that it's in a plastic package and has no on-board EPROM, costs only $3 to $4. But the convenience of on-chip, erasable memory is often worth the extra cost. Because prices vary from seller to seller, it's a good idea to shop around before you buy.

OTP EPROM (one-time programmable EPROM) is a low-cost EPROM in a plastic package and has no window. These devices are programmed like windowed EPROMs, but, in spite of their EPROM name, OTP EPROMs can't be erased. They're useful for small-volume production, where ordering a ROM version would be too expensive and the ability to erase is unnecessary.

EEPROM (electrically erasable PROM) is easier to program and erase than EPROM. Current EEPROMs require no special programming voltages. Their contents can be re-programmed simply by writing new data to the desired locations. Thus, they require no ultraviolet energy for erasing old contents. A few microcontrollers are available with on-board EEPROM program memory.

NV RAM (non-volatile random-access memory) doesn't lose its contents on powering down. One popular form of NV RAM contains a conventional static RAM chip with an embedded lithium cell for battery back-up. Dallas Semiconductor offers a microcontroller with on-board NV RAM program memory.

Table 1 lists some popular devices with user-programmable program memory. This is only a small selection of the hundreds of devices available, and new devices are being released all the time.

Developing a Project

To develop a project using a single-chip computer, you need at least a microcontroller or other single-chip computer, whatever other components your circuit requires, an assembler or compiler to create an executable program file for your microcontroller, and a device programmer to write the executable file into the microcontroller's program memory.

Sometimes, just finding a source for a microcontroller in which you're interested can be a challenge, especially if you need only one or two chips for a single project. The Sources box at the end of this article lists vendors that carry a selection of microcontrollers and have minimum orders of $30 or less. On request, chip manufacturers will provide a list of distributors, although not all of these accept small orders. Another sometimes-successful

Table 1. Popular Eight-Bit Microcontrollers With Embedded User-Programmable Program Memory

Device	EPROM	Process	Comments
8051 Family			
8751H	4K	HMOS	Basic 8051; 12 MHz, 21-volt programming
8751H-8	4K	HMOS	8 MHz maximum; 21-volt programming
87C51	4K	CMOS	Basic CMOS EPROM version of 8051; 21-volt programming
8751BH	4K	HMOS	12.75-volt Quick-Pulse programming
8752BH	4K	HMOS	Enhanced 8751BH with 256 bytes of RAM and three timers
87C51FA	8K	CHMOS	Includes programmable counter array, up/down counter, enhanced serial channel
87C51FB	16K	CHMOS	Similar to 87C51FA
87C51FC	32K	CMOS	Similar to 87C51FA
80C550	4K	CMOS	Includes A/D converter, watchdog timer
87C751	2K	CMOS	24-pin 0.3" DIP; I²C bus; (Signetics/Philips)
87C752	2K	CMOS	28-pin DIP; A/D; I²C bus; (Signetics/Philips)
DS5000	8/32K	CMOS	Program memory is NV RAM; can also be used as data memory (Dallas)
6800 Families			
MC68701	2K	HMOS	Enhanced 6800
MC68701U4	4K	HMOS	Similar to MC68701
MC68705P3	1.8K	HMOS	28-pin package
MC68705P5	1.8K	HMOS	Secured EPROM
MC68705R3	3.7K	HMOS	Includes A/D converter
MC68705R5	3.7K	HMOS	Includes A/D converter, secured EPROM
MC68705S3	3.7K	HMOS	28-pin package; includes A/D converter, SPI*
MC68705U3	3.7K	HMOS	Similar to 68705P3; 40-pin package
MC68705U5	3.7K	HMOS	Secured EPROM
MC68HC705C4	4K	HCMOS	Includes SCI**, SPI; runs at 3 to 5.5 volts
MC68HC705C8	8K	HCMOS	Similar to 'HC705C4
MC68HC805B6	6K	HCMOS	Program memory is EEPROM
MC68HC805C4	4K	HCMOS	Program memory is EEPROM
MC68HC711D3	4K	HCMOS	Includes SCI, SPI
MC68HC711E9	12K	HCMOS	Includes 512 bytes EEPROM
MC68HC811E2	2K	HCMOS	Program memory is EEPROM; includes A/D converter, SCI, SPI
Z8 Family			
Z08603	2K	HMOS	Z8 protopack with piggyback 2716 socket
Z08613	4K	HMOS	Z8 protopack with piggyback 2732 socket
Z08822	8K	HMOS	Super8 protopack with piggyback 2764 socket

*Serial Peripheral Interface; **Serial Communications Interface

approach is to contact the manufacturer's nearest local office and request a sample.

If you want erasable EPROM, be sure to order the chip in a ceramic package that has a window, not the plastic OTP EPROM version. Package type is usually indicated in the part number, although there's no standardization in the method manufacturers use to indicate such. For example, Motorola indicates a ceramic DIP with an "S" suffix, while Intel uses a "D" prefix.

Assemblers for assembly-language programming and compilers for programming in C and other languages are available from a variety of sources. Check the ads in this magazine for sources and try programming-oriented BBSs for freeware or shareware assemblers and compilers. Some data books also list companies that offer software and other products for use with the products described.

The assembler or compiler reads the source file, which is the program you write with a text editor, and translates it into a format the microcontroller can understand. Two popular formats that include error checking are Intel Hex and Motorola S-record.

The device programmer programs the microcontroller's EPROM or other program memory with the information contained in the assembled or compiled file. When the device is programmed, you can insert it into your circuit, power up and the program will execute.

Of course, the program probably won't execute without a flaw on the first try, and you'll have to analyze what went wrong, erase the EPROM and repeat the process until everything works as it should. To avoid having to program and erase repeatedly, you can use development tools like simulators, emulators and evaluation boards.

Programming 8751s

One of the most popular single-chip computers has been Intel's 8751, an EPROM version of the 8051 microcontroller. As with the basic 8051, many derivative versions of the 8751 have been developed by Intel and others. EPROM versions usually contain a "7" in the part number. The still-available early version 8751H contains 4K of EPROM. Its EPROM is programmed much like other older EPROMs, with a programming potential of 21 volts and 50-ms programming pulses. Newer 8751 versions—including the 87C51FA, -FB and -FC—have more program memory, and can be programmed with a variation of Intel's Quick Pulse programming algorithm, using 12.75 volts and 100-μs programming pulses.

Many EPROM programmers offer adapters for programming at least some versions of the 8751, usually at an extra cost of $50 or more. A few EPROM programmers can handle 8751s without an adapter.

At $149.95, Target Electronics' Programmer Series is the lowest-cost 8751 programmer I was able to find. It plugs into a PC expansion slot and has a 2-foot cable that goes to the microcontroller socket. It programs eight 8748/8751 versions. This programmer doesn't do standard EPROMs, though. If you want to program EPROMs as well as 8751s, a low-cost option is Needham's Electronics PC EPROM programmer with 8751 adapter, which sells for $189.95.

If you want to use the 8751 and already have a programmer that doesn't offer an 8751 adapter, a solution is shown in Fig. 2. Logical Systems has designed a series of smart adapters that enable you to program 8751s and other microcontrollers with a "generic" EPROM programmer. To use an adapter, you plug it into the EPROM socket on your programmer, set functions with the adapter's DIP switches and configure your programmer to program the EPROM that the adapter emulates. There are three adapters that sell for $95 to $159. All program basic 8751s, the higher-priced ones programming newer versions as well.

Many EPROM programmers are designed to program devices in dual in-line packages (DIP) that have 40 or fewer pins. They feature 40-pin ZIF (zero-insertion force) DIP sockets for inserting the devices to be programmed. These sockets aren't much use when you want to program a device in a square LCC (leaded chip carrier) or other package type. The solution is to use Logical Systems' socket adapters for programming LCC and other packages in conjunction with a DIP programmer.

Two 8751-family microcontrollers with EPROM and special programming requirements are made by Signetics/Philips. The 24-pin 87C751 and 28-pin 87C752 each have 2K of EPROM. Because of their smaller package sizes, some of the pins do double duty.

In both devices, 11 address bits are multiplexed on a single eight-bit port, with the high byte latched first during programming, followed by the low byte. In addition, before the device can be programmed, the programmer must send a 10-bit code to the device's RESET pin to put it into a special programming mode. Several companies, including Logical Systems, offer programmers and adapters for use with these devices.

Pure Unobtainium is a source for the 87C751 and 87C752 in small quantities. The Signetics/Philips' 80C51 and Derivative Microcontrollers handbook has details on these microcontrollers, including programming information.

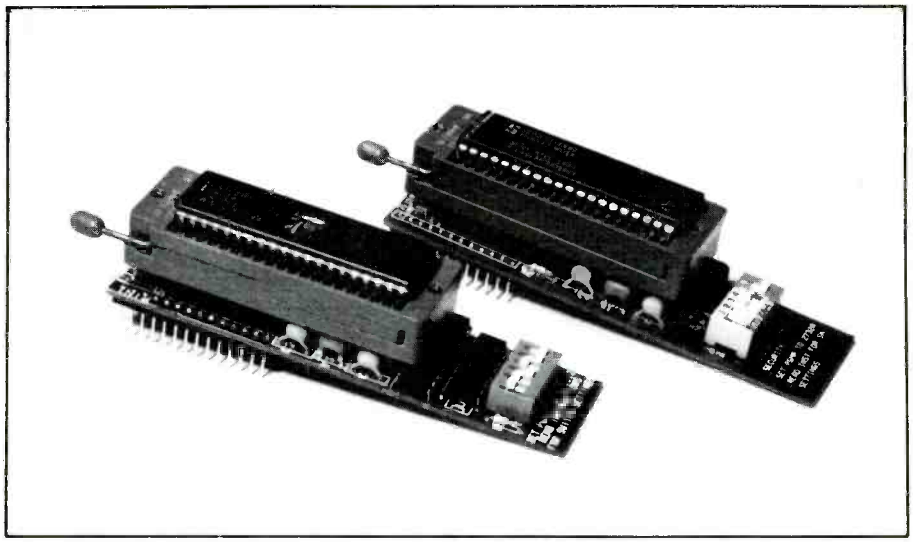


Fig. 2. With these adapters from Logical Systems, you can program an EPROM in an 8751 with any generic EPROM programmer.

A final 8051 derivative is Dallas Semiconductor's DS5000, which dispenses with EPROMs and stores its program and data in NV RAM. You can program the RAM with Dallas' DS5000TK development kit.

To erase information in a windowed EPROM, you need an EPROM eraser. This is typically an enclosed source of ultraviolet energy, with a slot or drawer into which you place the device to be erased. Drawer-type erasers are more versatile, since they aren't limited to DIPs. Some slot-type erasers won't even erase 40-pin DIP EPROMs, since the windows on the larger packages don't line up with the hole through which the ultraviolet energy passes in the eraser.

Motorola Microcontrollers

The 8751 family is by no means the only family with EPROM microcontrollers. Motorola offers several microcontrollers with EPROM. These include the 6801, 6805, 68HC05 and 68HC11 families. Motorola's approach to programming on-chip EPROMs was to design programming routines right into the chips to permit you to program the EPROMs with a minimum investment in programming hardware and software.

There are two approaches to programming Motorola microcontrollers: (1) first program a generic EPROM with the desired code and then insert the programmed EPROM into a special programmer that copies EPROM contents into the microcontroller's EPROM; (2) download a file from a personal computer to a microcontroller programmer, bypassing the intermediate EPROM.

For the first method, Motorola offers a series of application notes in which are described circuits that program a microcontroller's EPROM from a generic EPROM. Each note includes a circuit schematic, programming instructions and pc-board layout for building your own board. If you don't want to fabricate a pc board, you can Wire Wrap or point-to-point wire the circuits.

Circuits in the Motorola literature are relatively straightforward to put together. For example, the circuit for 68705 programming requires sockets for the microcontroller and EPROM, 4040B 12-stage binary counter, 78L12 12-volt regulator and a couple of dozen discrete components (capacitors, resistors, crystal, switches, LEDs and diodes). The circuits require + 5- and + 26-volt power supplies or a 26A05 5-to-26-volt converter. Since the programmer transfers information only from a standard EPROM to the microcontroller's EPROM, you also need access to a generic EPROM programmer to make use of the circuits. Application notes are available free on request from a Motorola office.

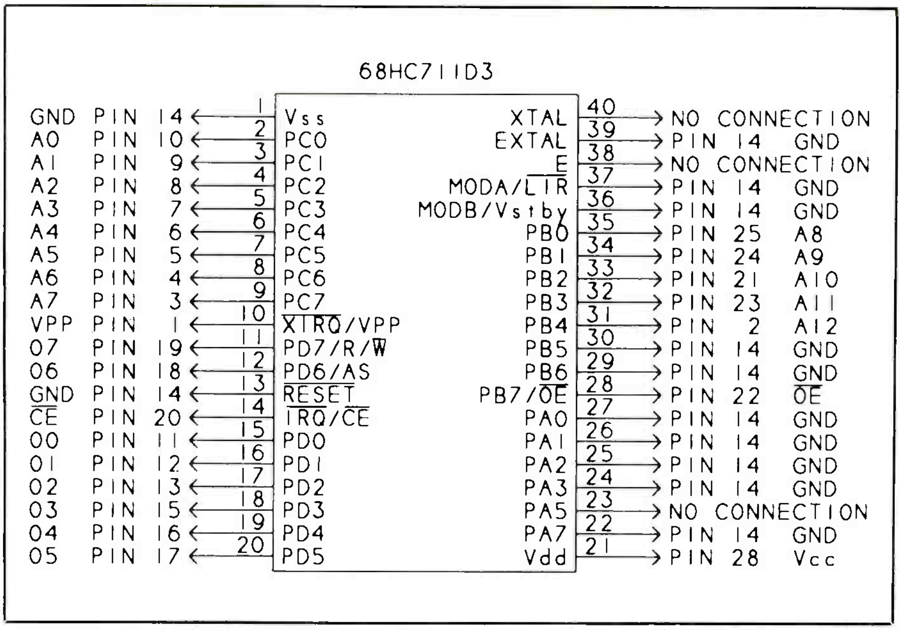


Fig. 3. All you need to program Motorola's 68HC711D3 is an EPROM programmer and a socket adapter wired as shown here. The adapter puts the '711D3 into its EPROM emulation mode and routes address, data and control signals to pins that emulate a 28-pin 27256 EPROM package.

Motorola's Applications Literature booklet (BR135/D) lists the following:

AN857. MC68705P3/R3/U3 8-Bit EPROM Microcomputer Programming Module

AN906A. Self-Programming the MC68701 and the MC68701U4

AN907A. Programming the MC 1468705G2 Microcomputer EPROM

AN942. MC68704P2 8-Bit Microcomputer Programming Module

AN966. MC68HC805C4 8-Bit EEPROM Microcomputer Programming Module.

AN1009. Programming the MC 1468705F2 EPROM Microcomputer Unit

In addition, Motorola offers a series of programmer boards for programming the EPROMs in many of its microcontrollers. For example, the M68HC05PGMR board programs an MC68HC05 microcontroller either from a programmed EPROM or via a serial link to a PC. Motorola's on-line BBS (512-891-3733) has freeware assemblers for its devices.

Other companies have come up with products that expand or improve on what Motorola has to offer. Let's look at a few.

Single Chip Solutions offers programming boards for 68705, 68HC705 and 68701 microcontrollers. Each communicates over a serial link with a desktop computer and programs the microcontroller with a file stored in S-record format, with no intermediate EPROM required. Kit versions of the programmers are also available.

TECI also offers serial-link EPROM programmers for 68705 and 68HC05 chips. Also available are optional cross assemblers and simulator/debuggers for use with the programmers.

See the June 1992 *ComputerCraft* for an article describing a programmer you can build for the 68HC705C8 microcontroller. Pure Unobtainium is a source for this chip.

Easy Programming

For new designs, Motorola has been using a new approach to programming its microcontrollers. The theory is as follows. If a microcontroller can be made to look and act like a generic EPROM, you can program the device like a standard EPROM, using any EPROM programmer. All you need is a socket adapter that routes the microcontroller's pins to the pinout of a standard EPROM. This great idea makes feasible programming microcontroller EPROMs, without requiring you to spend a fortune on specialized programmers and adapters.

Motorola's first device with EPROM emulation is the 68HC711D3. This 68HC11 microcontroller has 4K of EPROM. Holding low the MODA, MODB and RESET pins puts the microcontroller into PROG mode, in which it emulates a standard 27256 EPROM.

Figure 3 shows what's involved in making a programming adapter. In PROG mode, the pins of the '711D3 perform the functions of the pins on a 27256 EPROM. Port C and bits 0 through 4 of Port B receive the address to be programmed, Port D holds the data to be programmed (or read), and other pins perform the functions of the V_{pp}, CE and OE pins of a 2756.

Pin 31 (A12) isn't used but is included to allow for a future upgrade to an 8K EPROM. A13 and A14 are left open, on both the '711D3 and the 27256 socket. Pin numbers shown are for the 40-pin '711D3 and 28-pin 27256 DIP devices.

Using Figure 3 as a guide, I wired an adapter on perforated board and programmed the EPROM in a '711D3, using an ordinary EPROM programmer. I used two 14-pin SIP (single in-line package) headers spaced 0.6″ apart to emulate the 28-pin EPROM package, and a 40-pin IC socket to hold the '711D3. To make the connections shown, I soldered wires from the SIP headers to the 40-pin socket.

To program the '711D3, I plugged the SIP headers into the socket on my EPROM programmer and instructed the programmer to program a 27256 EPROM. Most 27256s program with the Intelligent programming algorithm, with V_{pp} = 12.5 volts, V_{cc} = 6 volts and a series of 1-ms programming pulses, which is what I selected to program the '711D3.

If you make your own adapter, be sure to position the components so that the pins fit into your EPROM programmer's ZIF socket without interfering with its lever. And be sure to wire the 27256 socket so that its pin 1 orients correctly when it plugs into your programmer's socket. Alternatively, you can buy a tested and assembled adapter for '711D3 programming from Logical Systems.

The '711D3's 4K EPROM uses just ⅛ of a 27256's 32K capacity. If possible, configure your programmer to program only the first 4,096 bytes of a 27256 (locations 0 through FFFh). If the programmer tries to program the entire 32K, it will program the '711D3's EPROM eight times, since

Sources

All of the following sources offer microcontroller programmers for $500 or less.

Andratech
P.O. Box 222
Milford, OH 45150
Tel.: 513-831-9708; fax: 531-831-7562
Universal programmer.

BP Microsystems
10681 Haddington, Ste. 190
Houston, TX 77043
Tel.: 713-461-9430 or 1-800-225-2102
Universal programmer.

B&C Microsystems, Inc.
750 N. Pastoria Ave.
Sunnyvale, CA 94086
Tel.: 408-730-5511; fax: 408-730-5521
Universal programmers.

Dallas Semiconductor
4350 S. Beltwood Pkwy.
Dallas, TX 75244-3292
Tel.: 214-450-0400 or 1-800-336-6933
Evaluation kit for DS5000.

DIP Industrial Products
P.O. Box 9550
Moreno Valley, CA 92552
Tel.: 714-924-1730; fax: 714-924-3359
87C751/2 programmer.

GTEK, Inc.
P.O. Box 2310
399 Hwy. 90
Bay St. Louis, MS 39521-2310
Tel.: 601-467-8048; fax: 601-467-0935
68705 programmer.

Logical Systems Corp.
P.O. Box 6184
Syracuse, NY 13217
Tel.: 315-478-0722; fax: 315-475-8460
Socket adapters for EPROM programmers, 87C751/2 programmer.

Motorola Semiconductor Products Inc.
P.O. Box 20912
Phoenix, AZ 85036
Tel.: 1-800-521-6274

Programmer boards for Motorola microcontrollers.

Needham's Electronics
4539 Orange Grove Ave.
Sacramento, CA 95841
Tel.: 916-924-8037; fax: 916-972-9960
Universal programmers.

Jameco
1355 Shoreway Rd.
Belmont, CA 94002
Tel.: 1-800-831-4242; fax: 1-800-237-6948

JDR Microdevices
2233 Samaritan Dr.
San Jose, CA 95124
Tel.: 1-800-538-5000; fax: 1-800-538-5005

Newark Electronics
4801 N. Ravenswood Ave.
Chicago, IL 60640-4496
Tel.: 312-784-5100

Pure Unobtainium
P.O. Box 285
Tolland, CT 06084
Tel. voice/fax: 203-870-9304

Unicorn Electronics
10010 Canoga Ave., Unit B-8
Chatsworth, CA 91311
Tel.: 1-800-824-3432; fax: 818-998-7975

Single Chip Solutions
P.O. Box 680
New Hartford, CT 06057-0680
Tel.: 203-496-7794
68701/705/HC705 programmers.

Target Electronics
P.O. Box 400
Post Falls, ID 83854
Tel.: 208-773-1962; fax: 208-773-0894
8748/51 programmer.

TECI
Rte. 3, Box 8C
Barton, VT 05822
Tel.: 802-525-3458; fax: 802-525-3451
68705/HC705/HC805 programmers.

Xeltek
764 San Aleso Ave.
Sunnyvale, CA 94086
Tel.: 408-745-7974; fax: 408-745-1401
Universal programmers.

All of the following sources have minimum orders of $30 or less.

Arrow Electronics, Inc., Catalog Div.
1860 Smithtown Ave.
Ronkonkoma, NY 11779
Tel.: 1-800-932-7769

EasyTech, Inc.
2917 Bayview Dr.
Fremont, CA 94538
Tel.: 1-800-582-4044; fax: 1-800-582-1255

A12, A13 and A14 are ignored. For example, location 0 will also be programmed as 1000h, 2000h, 3000h, and so on up to 7000h.

Some programmers don't allow you to program part of a device. If you must program the entire 32k, instruct the programmer to write FF to locations 1000h through 7FFFh. Since the programmer can change only 1s to 0s, not 0s to 1s, this technique will prevent it from corrupting previously programmed data. Locations beginning with 1000h won't verify, since they don't exist in the '711D3.

One caution about the '711D3: in an early version of the chip, PROG mode did not work correctly. Any device with the mask number C45A should be programmed only in the alternate way, using the microcontroller's test or bootstrap mode and an appropriate programming board. The mask number is visible on most packages. Another clue is the date code. Date codes after 9135 are okay to program in PROG mode, with the first two digits indicating the year and the remaining digits the week in the year (9135 = week 35 in the year 1991).

For more details on '711D3 programming, see the MC68HC711 Technical Data Book (#MC68HC711D3/D). Another Motorola product with PROG mode is the 68HC711E9, which has 12K of EPROM as well as 512 bytes of EEPROM. An early mask for this device (1C47M) also had some programming problems in PROG mode.

Zilog Approach

One company that has taken a different approach to adding EPROM to its microcontrollers is Zilog. For prototyping and low-volume applications, Zilog offers its Z8 and Super8 protopacks, which are microcontrollers with piggyback EPROM sockets.

To use the piggyback microcontrollers, you program a generic EPROM with your code and simply plug the EPROM into the socket provided on the top of the microcontroller package. The microcontroller accesses the EPROM as if it were masked ROM or other on-chip memory.

The piggyback socket makes it very easy to use ordinary EPROMs and still produce a single-chip design. Available devices include the Z08603 Z8 with 2716 EPROM socket, Z08613 Z8 with 2732 EPROM socket and Z08822 Super8 with 2764 EPROM socket.

Product News

A source for information and files to download relating to 8051 microcontrollers is the Signetics/Philips Customer Support BBS at 1-800-451-6644 or 408-991-2406, the purpose of which is to assist design engineers who are using Signetics PLD and microcontroller products. It includes a message area with answers to design questions and a file area from which you can download files, including assemblers, interpreters, compilers, monitors and applications examples for 8051 products. If you're interested in the I²C serial bus, you'll find information and files on it here as well. ■

Send your comments, suggestions or questions on topics relating to designing, building and programming microcontrollers and other small, dedicated computers to Jan Axelson, ComputerCraft, *76 N. Broadway, Hicksville, NY 11801. For a personal reply, please include a self-addressed, stamped envelope.*

Jan Axelson

Putting the Controller/ Actuator to Work

Assembling an X-Y table for drawing and computer-aided manufacturing

Last month, we discussed the theory behind and gave complete construction details for a computer-driven linear controller/actuator that serves as the foundation for a wide range of robotic devices. Now we show you how to arrange two or more of these NEURACTORs to make useful devices, among them a complete computer numerical control (CNC) machining system that you can put to use in practical real-world applications.

NEURACTOR Theory

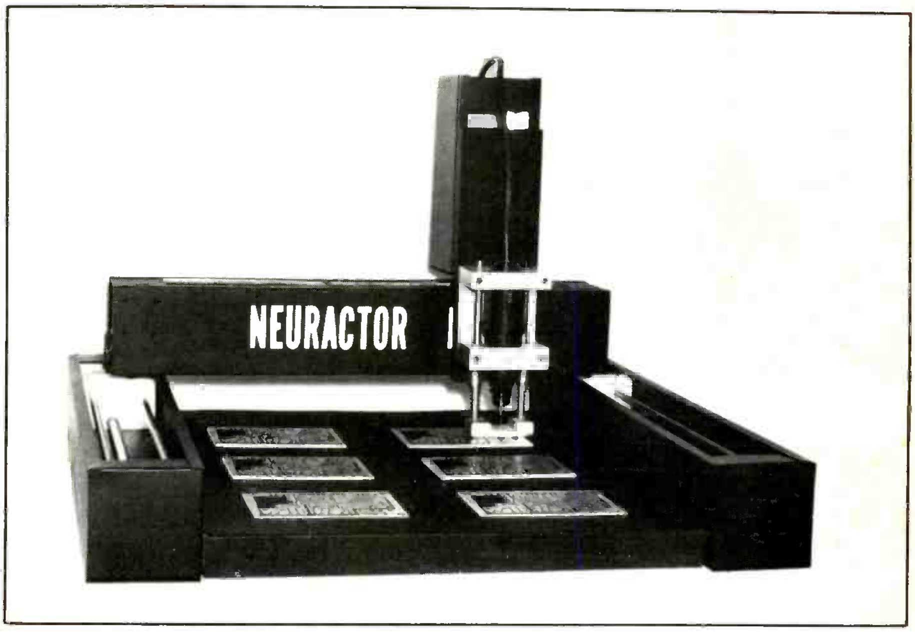
The NEURACTOR is a self-contained linear actuator that uses internal microcomputer electronics and a stepper motor to control motion along a single axis. If you built the NEURACTOR from details given last month, you're probably already familiar with the basic features of the unit, which include:

(1) Fully self-contained, with an internal four-phase stepper motor, controller card and lead-screw actuator mechanism.

(2) Use of true "distributed" intelligence that, coupled with a special SPI (serial peripheral interface) makes connecting together several NEURACTORs an easy process.

(3) Use of a neural-network IC to convert the on-board stepper motor into a "neural" servo to allow you to use a 0-to-5-volt signal to move the slide mechanism to any relative position along the axis of travel.

Power is supplied to the NEURACTOR from a commonly available wall-type transformer. A standard RS-232 cable connects the NEURACTOR to its host computer. Serial data is transferred from the host computer to the NEURACTOR's internal control card.



An MC68HC705C8 microcontroller on the control card converts motion data into precisely-controlled stepper motor pulse-streams.

On power-up, the NEURACTOR control card always moves the slide mechanism to the "home" position by stepping the motor and moving the slide back toward the motor end of the NEURACTOR. A Hall-effect sensor monitors the proximity of a small magnet mounted on the slide block. With the NEURACTOR in home position, the control card establishes a zero reference from which to count.

The stepper motor used in the NEURACTOR requires 200 pulses to make one revolution. Each revolution of the ACME-thread shaft moves the slide block exactly 0.100". Consequently, the control card knows that it takes 2,000 pulses to move 1" or that one step equals about 0.0005".

Using the control card as a sophisticated stepper-motor controller, the host computer can read disk data files and transmit motion values via its RS-232 interface. An extended feature of the NEURACTOR control card is its ability to operate several other NEURACTORs that are "slaved" from its SPI communication port. This distributed architecture is a big advantage for multi-axis robotic configurations because the host computer needs only one serial port instead of a separate port for each axis and, thus, also greatly simplifies the host computer control program.

While we'll be concentrating on CNC applications for the NEURACTOR here, remember that the basic strength of the NEURACTOR system is its neural-network interface. If you want real-world output for your experimental neural network project, the NEURACTOR is the perfect device for it.

By translating the 0-to-5-volt input at the USC-2240 IC into relative positions along the motion axis, a NEURACTOR can be directly controlled (like a servo motor) by an analog voltage. The USC-2240 also outputs a matching analog "feedback" voltage that corresponds exactly to the position of the slide block. This allows you to form "closed-loop" servo-type systems without the need for an expensive linear position encoder.

X-Y Axis CNC Table

One of the more attractive possibilities for this project is to use it to make an X-Y table with which you can indulge your artistic abilities for drawing, painting and photo-plotting using pencils, pens, air-brushes, low-power lasers and LEDs. You can also experiment with computer-aided manufacturing or "desktop" manufacturing with an X-Y table, performing machining operations like drilling, routing and profiling, using rotary tools, lasers, water-jets, etc.

The simplest X-Y table you can configure with the NEURACTOR is shown in Fig. 1. This easy-to-build arrangement uses only two NEURACTORS. By mounting the X-axis NEURACTOR on blocks (called a "stage") above the table, you can mount the drawing device or cutting head directly to the slide block. The X axis (left and right, facing the front of the machine) moves only the head itself.

The Y axis (moving away from and toward the front of the machine) is controlled by the Y-axis NEURACTOR. The stage bolts to the slide block.

In operation, a BASIC program translates input from a joystick, mouse, lightpen, tablet or disk data file into X and Y coordinates for the NEURACTOR control card. Even though your host computer uses only one serial cable, the host NEURACTOR forwards Y-axis motion data directly to the Y-axis NEURACTOR. This way, both NEURACTORS can be simultaneously moving to their relative positions.

In the simplest and least-expensive X-Y table configuration, you'd simply attach the drawing device or machining head directly to the X-axis NEURACTOR slide block, without provision for the Z axis (up and down). However, if you wish to be able to control the Z axis, you can simply add another NEURACTOR to control it. You probably won't need a "stroke" of more than 6″ for the Z axis.

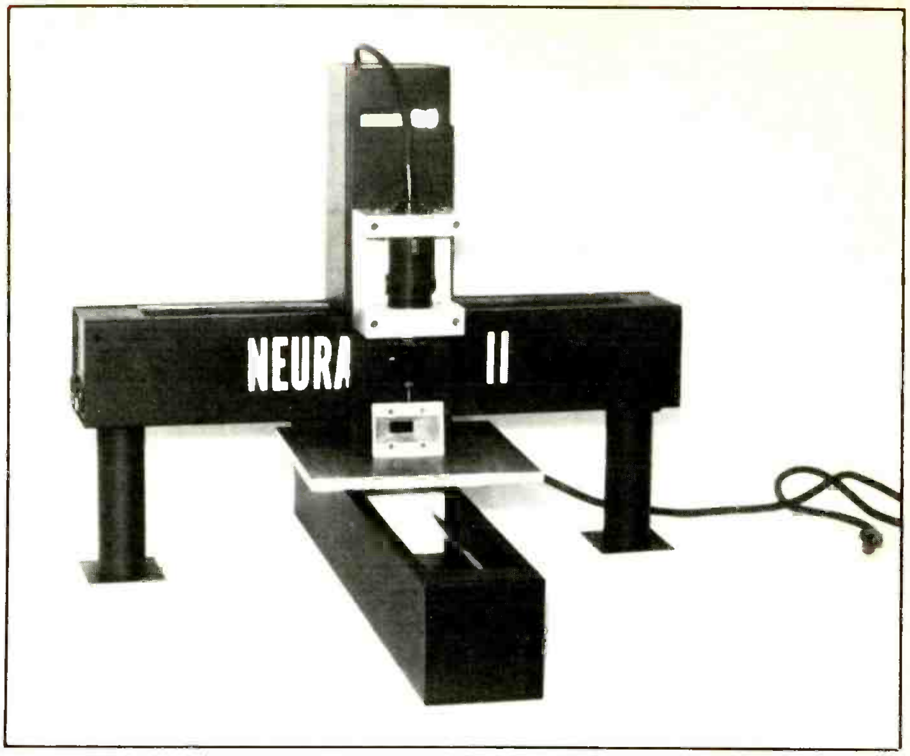


Fig. 1. The simplest X-Y table consists of just two NEURACTORS, one each for the X and Y axes.

For the Z-axis NEURACTOR, you can make a shorter version of the standard NEURACTOR or obtain one in kit form from the source given in the Note at the end of the Parts List. Just as the Y-axis NEURACTOR is controlled by the host computer via the X-axis NEURACTOR, the Z-axis NEURACTOR is controlled in much the same manner.

Your drawing-device and machining-head mounts can be as simple or complex as you need. Be sure to keep in mind that the mount you use must be rigid enough to hold the device firmly, but it must permit easy adjustment and removal of the device.

You can attach paper, cardboard or film to the table stage with alligator clips, clamping bars or paper clips. For machining operations, you can use a miniature vise or make "hold-downs" from machine screws. Always remember that your final product will be only as accurate as the method of attachment is secure.

When you want greater accuracy

PARTS LIST *

2**—18″ to 36″ NEURACTOR actuators with four-phase stepper motor and controller
1—6″ NEURACTOR actuator with four-phase stepper motor and Controller
1—Drawing device or machining-head holder (see text)
1—¼″ 5052T aluminum stage table (size to suit your application.

*For two-NEURACTOR configuration table.
**Increase to 3 for three-NEURACTOR configuration table.

Note: The following items are available from Cyance Kit, Rte. 2, Box 284, Cyber Rd., West Fork, AR 72774 (tel.: 501-839-8293): Complete three-NEURACTOR configuration 18″ CNC Kit, $499; 6″ Z-axis NEURACTOR actuator, $139; Dremel tool holder, $29.95. Also available are individual NEURACTORS, $199.95, complete with all machined mechanical and electronic parts; tested 58 oz.-in. six-wire, 4-volt, 1-ampere Sanyo-Denki stepper motor, $19.95; USC-2240 NEURACTOR, $29.95; complete control card electronics lab kit, $99.95. Call for information and prices for longer NEURACTOR actuator kits. Other available items are: Cyber HC5 development system with assembler/programmer software, $89.95; MC68HC705C8S (EPROM version) microcontroller, $22.95. Add $8.95 S&H for larger items, $4.95 for small items. Arkansas residents, please add 5% sales tax. MasterCard/Visa accepted.

and control over drilling, milling, routing, profiling, etc., operations, you'll have to move to a three-NEURACTOR configuration. Thus, two Y-axis NEURACTORs move the entire X-axis assembly across the work-piece. This "gantry-robot" approach has been used in industry for years with considerable success. You again have the option of using the Z axis to control your drawing device or machining head. Alternatively, you can go "bare" to cut costs.

Software control of the three-NEURACTOR configuration is the same as for the two-NEURACTOR configuration, except that the separate X-axis NEURACTORS are "slaved" together using different SPI communication addresses. This lets both devices synchronize digitally, as well as mechanically.

Table Construction

One of the greatest design strengths of the NEURACTOR concept is interchangeability. You can always re-use your NEURACTORs by simply configuring them into new machines.

To construct either of the two tables described above, simply bolt them together. That's all there is to it! Drill mounting holes in the bottom or sides of the NEURACTOR cover ends. Then use four screws at each end to mount the NEURACTOR to the table top or the other NEURACTORs in your setup.

In the three-NEURACTOR configuration, you may have to run the X-axis NEURACTOR with the mounting bolts slightly loose to allow for "play" in the ACME thread shafts. Even though both of the Y-axis NEURACTORs are digitally "locked," mechanical binding might occur along the length of the Y axis. You'll have to experiment with your particular unit to determine the tradeoff point between high accuracy and mechanical reliability.

The NEURACTOR CNC system was specifically designed to use the Dremel-brand high-speed drill, which can be used for drilling and routing a variety of different materials. When you mount the tools, make sure to leave enough slack for the power cord to travel the full length of the table.

Software Requirements

The NEURACTOR CNC controller card can be driven with a BASIC program. Most CAD programs, including *AutoCAD* and *EASYPC*, provide plotter and printer outputs to disk file. You simply need to write a small routine that reads the various X, Y and Z coordinates from the disk and output them to the serial port on your computer.

The CYBER HC5 micro-controller development system (featured in the November and December 1991 issues of *ComputerCraft*) is the perfect way to develop custom software for the NEURACTOR control card. With the HC5, you can tailor the control card's performance with respect to stepper-motor parameters like step rate, acceleration, deceleration, single and half step, etc., as well as the SPI communication protocol.

Applications

When you have your NEURACTOR CNC X-Y table on-line, you'll probably want to use it for interesting and/or utilitarian applications. As a small sampling of what you can do with it, you might want to try these activities:

- To use your NEURACTOR CNC as the ultimate X-Y plotter, try making drawings of different sizes by changing the coordinate-to-step ratios in software. Use different styles of pens and pencils and different types of paper, Mylar and other flat materials.
- Try painting with the NEURACTOR CNC. I originally developed the NEURACTOR to do computerized paint-by-number art, using an airbrush.
- Drill your own printed-circuit boards with the NEURACTOR CNC. I use a NEURACTOR to drill circuit boards every day. It's easy to operate from a BASIC program. You can use the files created by your pc-board CAD system directly with the NEURACTOR CNC. If you really want to try a super software project, program the NEURACTOR CNC to rout out the spaces between the traces on bare pc boards.
- Rout and mill wood, plastic and metal with your system. You can create works of art and useful commercial products.
- The ultimate is to use your NEURACTOR CNC to actually "manufacture" prototype wood, plastic and metal parts right on your desktop or lab bench. Using a Dremel Moto-Tool with the right milling and routing tool bits, you can create three-dimensional models from your CAD software.
- Construct a measurement probe for your NEURACTOR CNC to precisely digitize three dimensional objects, directly from your computer, via the serial port. A microswitch and probe rod are often all you need to make a highly accurate digitizer. Use data created by the NEURACTOR CNC directly in your CAD and graphics programs.

As you can see, the NEURACTOR CNC can be a very useful device. I've spent many hours just watching my NEURACTORs cranking-out everything from mechanical drawings and pc boards to prototypes and models of future projects. ■

Nick Goss

By Ronald J. Portugal

Computer-Controlled Frequency Synthesizer

Covers a range from 1 Hz to 10 MHz in 1-Hz increments for less than $50 in parts

Every so often, you need a precision frequency source, say 6.144 MHz for a baud-rate generator or 7.373 MHz for a microprocessor chip, and so on. Either you make do with a signal-generator/frequency-counter combination that usually doesn't quite provide the frequency or stability needed or hunt around for a crystal to build an oscillator. If you're really lucky, you set up your frequency synthesizer.

Frequency synthesizers are instruments that provide lots of discrete, stable and accurate frequencies from the below-audio range to the gigahertz range. They usually cost several hundreds or even thousands of dollars and are worth the money if your work can justify the cost. My need for a stable, adjustable and accurate frequency source in the below-10-MHz range resulted in the four-digit 10-MHz Frequency Synthesizer described here, which you can build for less than $50.

This frequency synthesizer employs a straightforward design. It uses standard off-the-shelf components, doesn't have tricky circuits and can be programmed in any language (QBASIC, TBASIC, C, C++, etc.) that can write data to a printer port. Like most synthesizers, this one relies on a single crystal oscillator, phase-locked loop and a few counter dividers to generate outputs—in this case, 54,000 discrete output frequencies.

Unlike most synthesizers, this one consists of a single 4″ × 3⅜″ printed-circuit card and doesn't have any controls—the PC with which it's used takes care of the frequency-setting, hardware-control and computation functions. All you have to do is punch in a frequency from 1.000 Hz to 10.00 MHz, and the software and hardware do the rest.

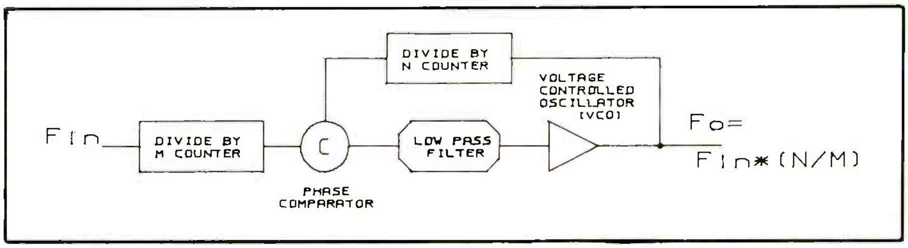


Fig. 1. Block diagram of a typical frequency synthesizer.

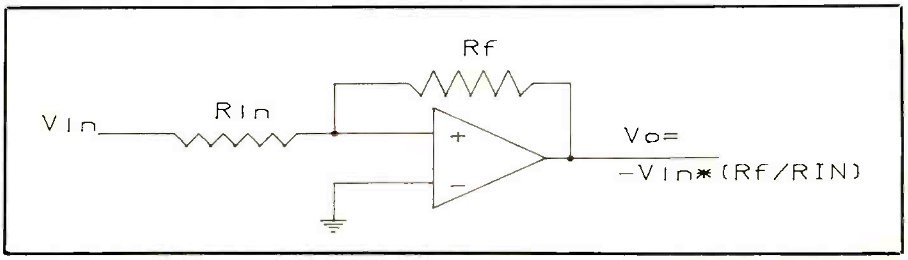


Fig. 2. Schematic representation of a unity-gain inverting operational amplifier.

Because the interface to the computer is through a standard parallel printer port, there's no need to add a card to your computer's expansion bus. Programming the Synthesizer is really easy, too, consisting of first a group of output instructions that are available in just about every programming language I've seen.

For simple byte-size inputs and outputs, using the printer port is a fast and easy way of doing things. Since the Frequency Synthesizer is just an output device (it doesn't have to send data back to the PC), it's not really easy to program.

About the Circuit

Before getting into a discussion of the actual circuitry that makes up our Frequency Synthesizer, you should become familiar with the basics of PC parallel printer ports. If your knowledge of this subject is limited or you're new to it altogether, you should read the Parallel Printer Port Basics box before proceeding.

To understand how this Frequency Synthesizer works, you must know something of how the phase-locked loop (PLL) works. Shown in Fig. 1 is a block diagram of the classic PLL. It consists of an input signal, a divide-by-m counter, phase comparator, low-pass filter, voltage-controlled oscillator (vco) and divide-by-n counter. With this arrangement, the PLL can be thought of as a frequency-domain operational amplifier.

Shown in Fig. 2 is the schematic representation of the classic op amp. If the values of R_{in} and R_f are equal, input voltage V_{in} appears at the output of the op amp as $-V_{out}$. To understand how this is accomplished, first assume that you're dealing with an ideal op amp, infinite input imped-

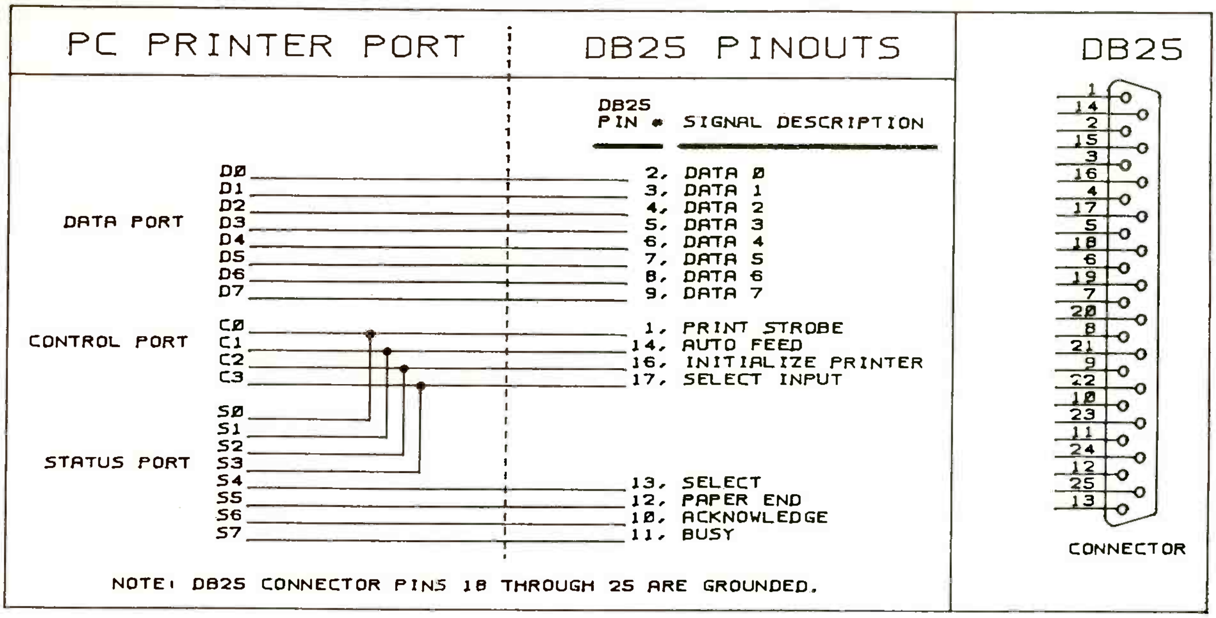


Fig. A.

Two topics with which you should be familiar to be able to understand how the Frequency Synthesizer in the main article works are how a PC parallel printer port works.

Printer Port Basics

PC printer ports consist of three registers: data, status and control. Port addresses depend on the parallel printer port used and are summarized as follows:

Port	Data	Status	Control
LPT1	03BC	03BD	03BE
LPT2	0378	0379	037A
LPT3	0278	0279	027A

The addresses here are in hexadecimal notation.

To send a byte to the port, use the OUT instruction. For example, OUT &H03BC (any number between 0 and 255) sets data-port lines to the states specified by the output number. Therefore, if you want to send a byte to the Frequency Synthesizer, send it to the data register.

Note that all program segments are compatible with QBASIC. If you use TBASIC, change the SLEEP instruction to the DELAY function. Otherwise, the instruction listings are okay for TBASIC.

Since we don't have to "read back" from the Synthesizer, we can ignore the status port. On the other hand the control port makes it possible to transfer data between the Synthesizer and PC with which it's used.

The status port isn't a full eight-bit port. Rather, it's a four-bit (nibble) port. Just as with the data port, you can write to the status port with an OUT instruction, with appropriate change in address, naturally.

A minor problem with the four bits from the control port is that three of the bits are inverted to meet the requirements of printer interfaces in which the lines had to be logic 1s.

Shown in Fig. A are the pinout details for a standard DB-25 printer-end parallel printer port. The four control-port lines are on pins 1, 14, 16 and 17 with only the one on pin 16 not inverted. Thus, when you send 0 to the control port, out comes a hex B instead of 0. This can easily be corrected by using the XOR (exclusive-OR) function on the nibble to be sent to the control port prior to executing the OUT instruction.

To send hex C to the control port, for example, first XOR hex C with hex B. Then OUT the results to the control port. The example shown in Fig. B should clarify things.

PIN NUMBER	17	16	14	1
	C3	C2	C1	CØ
OUTPUT DATA, HEX C	1	1	Ø	Ø
CORRECTION, HEX B	1	Ø	1	1
C XOR B	Ø	1	1	1
INVERTED OUTPUTS	X		X	X
RESULTING OUTPUTS	1	1	Ø	Ø

Fig. B.

It's easy to control peripheral hardware through a printer port. Just use a port instruction and send data to the data port. Reset the four control lines emanating from the control register with OUT instructions to take care of the hardware handshakes.

Suppose you have a D/A converter on the data lines of a printer port. Assume that the D/A needs a negative-going strobe pulse to write data into its internal register. Here's how you'd accomplish this:

(1) Set the control port to hex F:

CP = &HF XOR %HB ____ Prime D/A
OUT Control Port Address, CP " "

(2) Load the D/A byte into the data register with an OUT instruction:

OUT Data Port Address, Data Word

(3) Assuming bit 0, pin 1, of the DB-25 connector is used to strobe the D/A register,

CP × &HE XOR &HB 'Pin 1 to 0V
OUT Control Port Address, CP 'Pin 1 to 0V
CP = &FH XOR &HB 'Pin 1 to +V
OUT Control Port Address, CP 'Pin 1 to +5V

So the name of the game is: send the byte to the data port, send the control word or pulse to the control port. That's it. The only catch is that the control "pulse" line(s) to the output hardware must be clean (no noise or extraneous pulses).

ance, zero offset voltage between the inputs and infinite gain.

Since the inverting (−) input of the op amp is tied to ground, it must be at 0 volt. Under ideal conditions, the current flowing through R_{in} is calculated using the formula $I_{in} = V_{in}/R_{in}$. Since input impedance of the op amp is infinite, the input current can't flow into the amplifier. It must flow through feedback resistor R_f to the output of the amplifier.

The output voltage of an op amp is calculated by multiplying I_{in} by R_f, which yields the classic formula $V_o = -V_{in}(R_f/R_{in})$ inverting op-amp gain equation. The minus sign is used because current flows away from the input node, making the node positive with respect to the output terminal of the op amp.

Returning to Fig. 1, note that the two inputs to the phase comparator must "see" equal-frequency signals for the system to be "locked," just like two inputs of an ideal op amp are at the same voltage. One input of the phase comparator "sees" a frequency of F_{in}/M, the other a frequency of F_o/N. Thus, $F_o = F_{in}(N/M)$.

M and N are positive integer values that can be as large or as small as you like. The range of F_{in} and F_o depends on the hardware you're using. PLLs have been set up to run at frequencies that span from the low end of the audio into the uhf range.

Assuming that M is unity, you have $F_o = N \times F_{in}$. Since you want a four-digit synthesizer, N will vary between 1,000 and 10,000. Letting F_{in} be 1,000 Hz, for 1,000-Hz frequency steps, and plugging in the maximum and minimum values of N, $F_{o(max)} = 1{,}000 \times 10{,}000 = 10{,}000{,}000$ Hz and $F_{o(min)} = 1{,}000 \times 1{,}000 = 1{,}000{,}000$ Hz. Thus, with this set of values for M, N and F_{in}, you can generate 9,000 discrete frequencies between 1 MHz and 10 MHz in 1-kHz steps.

Let's say you want to go down to N = 100 and extend the frequency range of the PLL down to 100 kHz. There's no reason why you can't do this, except that you might have a bit of trouble getting an easy-to-build and inexpensive vco to cover this range, and you'd end up with only a three-digit Frequency Synthesizer.

Suppose you wanted to reduce F_{in} to, say, 1 Hz and raise resolution by a factor of 1,000. Again, you could do this, but the same type of problem arises. Now you get into trouble in the phase-comparator and filter circuits.

A 10:1 vco swing and a 1-kHz resolution in the 1-to-10-MHz range turn out to be reasonable parameters when you consider component cost, circuit complexity, simple construction, etc.

You may be wondering how you get down to lower frequencies, like 1 Hz. To do this, you simply divide the output of the vco with six decade counters, such that 1,000,000 divided 10^6 (the division factor of the six decade counters) to obtain 1.000 Hz.

When two waveforms are compared in the phase-comparator and filter portion of the PLL, the term "phase" has no meaning when the frequencies of the waveforms aren't the same. Hence, the phase comparator is really a frequency comparator that provides a means for measuring the phase relationship between two input signals of the same frequency. It's like the error voltage in an op amp that's driven toward zero by the feedback loop.

A phase comparator can be—and is in this project—a fancy edge-triggered phase/frequency detector that's independent of signal duty cycles and has a charge-pump-type output stage that simplifies filtering.

I'm not going to attempt a detailed explanation of how the values and frequency breakpoints of the filter were selected for the Frequency Synthesizer. Suffice it to say that the breakpoints are at 0.66 and 7.32 Hz, which are well below the 1,000-Hz frequency reference and keep "jitter" in the output signal to an almost imperceptible level. If you want to know more about the design of the filter, refer to the *Signetics High-Speed CMOS Data Manual*, which devotes about 25 pages to explaining and analyzing the 74HCT4046 PLL.

The output of the PLL's M counter provides the reference frequency to the comparator and F_{in} should be a stable crystal-controlled signal. You need a 1,000-Hz reference signal. The particular counter chip used in the Frequency Synthesizer, an 82C54-2 triple 16-bit counter, is limited to inputs of 10 MHz maximum. Therefore, any crystal rated at less than 10 MHz and specified to 1,000 Hz will do the job.

Just set the counter to divide the crystal frequency by the appropriate number. Once set, the counter never has to be changed. For example, a 6-MHz crystal is fine. Just divide it by

PARTS LIST

Semiconductors
Q1—2N4401 silicon npn transistor
U1,U2—74HC04 hex inverter
U3—74HC390 dual decade counter
U4—74HC151 eight-input multiplexer
U5—82C54-2 tripe 10-MHz 16-bit counter
U6—74HC4046 phase-locked loop and phase comparator
U7—74HC75 four-bit latch
U8—LM7805 fixed +5-volt regulator

Capacitors
C1—8-to-50-pF single-turn trimmer capacitor with leads on 0.2″ centers
C2—20-pF npo ±% ceramic disc
C3,C6—100-pF ceramic disc
C4—150-pF npo ±% ceramic disc
C5,C8—10-μF, 16-volt ±% tantalum
C9 thru C12—0.1-μF ceramic disc

Resistors (¼-watt, 5% tolerance carbon-film)
R1,R5,R11—2,200 ohms
R2—220,000 ohms
R3—47 ohms
R4,R12—100 ohms
R6—10,000 ohms
R7,R10—22,000 ohms
R8—3,000 ohms
R9—8,200 ohms
R13—680 ohms

Miscellaneous
J1—Panel-mount coaxial BNC connector
P1—Right-angle, pc-mount DB-25 connector
Y1—6-MHz crystal (see text)
Printed-circuit board (see text); 9- to 12-volt dc, 200-mA plug-in wall-mount power supply; suitable 3- to 4-foot cable terminated at both ends in DB-25 connectors; suitable enclosure (see text); red panel-mount light-emitting diode; power jack (optional; see text); machine hardware; solder; etc.

Note: The following items are available from RJP Electronics Co., 52 Susan Lane, North Haven, CT 06473: Ready-to-wire pc board, $15; complete kit of parts, excluding power supply; power connector and enclosure, $35; 1.2M 5¼″ IBM PC-compatible program disk, $10. Please add $3 per order for P&H.

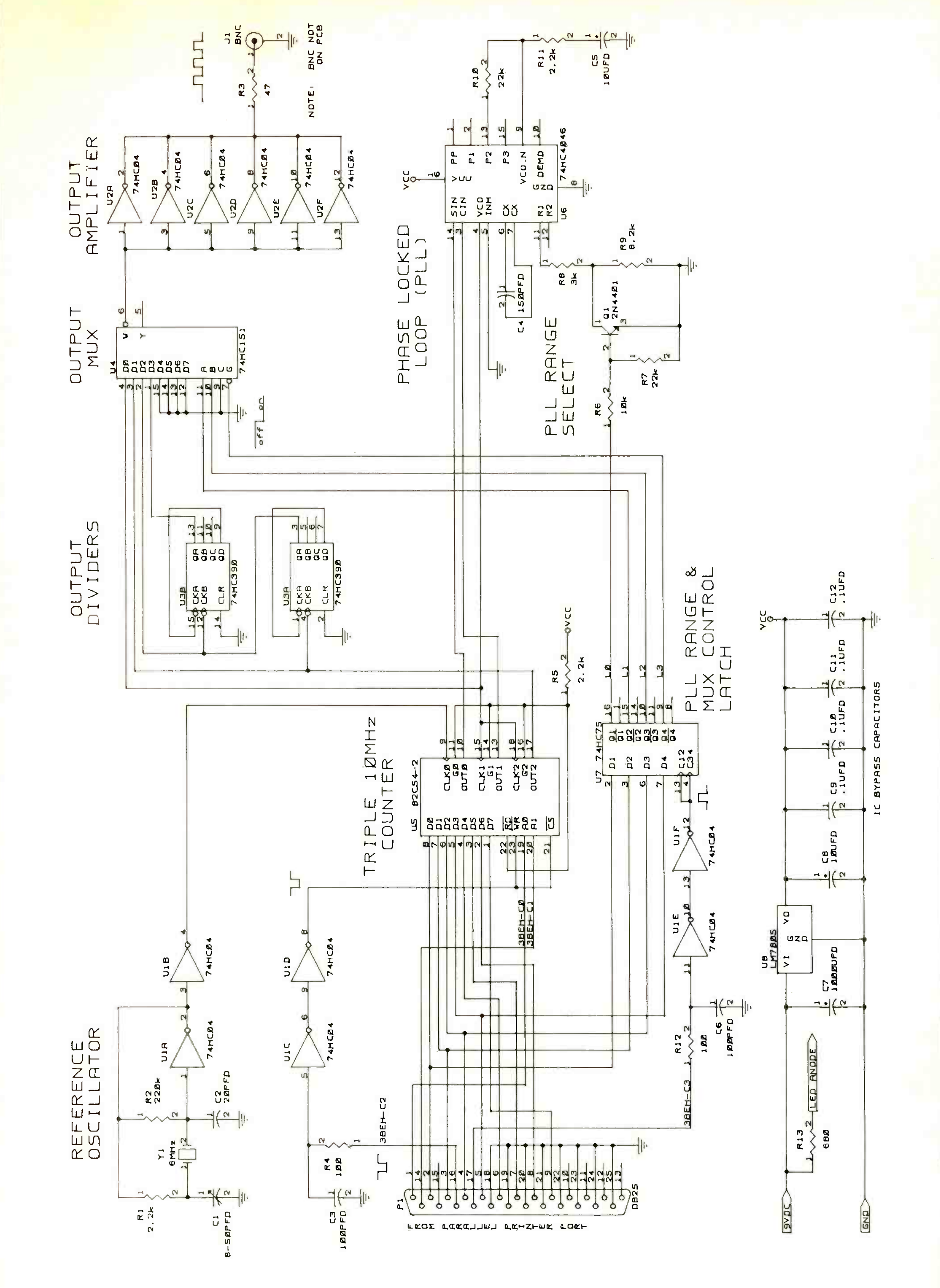


Fig. 3. Complete schematic diagram of PC-Controlled Frequency Synthesizer.

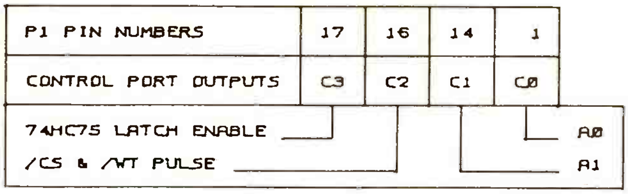


Fig. 4. Details of the P1 control lines connected to the counter and latch chips.

Fig. 5. Latch bit functions.

LATCH PIN #	NAME	FUNCTION
16	LØ	+5V, VCO HIGH RANGE ØV, VCO LOW RANGE
15	L1	MUX DECODER, INPUT A
1Ø	L2	MUX DECODER, INPUT B
9	L3	+5V, MUX OFF, OUTPUT OFF ØV, MUX ON, OUTPUT ON

6,000. However, a 3,579,545-Hz color-burst crystal isn't so fine because it's frequency can't be divided by any integer to yield 1,000 Hz.

Although it isn't shown in Fig. 2, a third counter chain is needed to bring the output of the vco down to the 1-Hz range. The third counter, an 82C54-2, can be programmed to divide by 10, 100, 1,000 and 10,000. This isn't good enough to get down to 1 Hz, though. Thus, another chip, this time a 74HC390 dual decade counter (*U3A* in Fig. 3), is used to generate the last two ranges.

I chose the 74HC4046 PLL for this project because it includes three phase comparators and a 20-MHz vco. The comparator I used, Phase Comparator 2, is a a very easy-to-use edge-triggered phase/frequency detector, but requires a lengthy explanation.

Although the specifications of the 74HC4046 indicate that a 10:1 vco range is possible, I found that not all such chips are created equal. Some covered the 1-to-10-MHz range easily, while others couldn't manage it at all. Therefore, to ensure that the vco does what it's supposed to do, I broke the 1-to-10-MHz range into two parts: 1 MHz to 3 MHz and 3MHz to 10 MHz. This solved the hardware problem, but it introduced a software annoyance.

The vco in the 74HC4046 is controlled by a resistor and capacitor, both external to the chip. The resistor is the obvious candidate to manipulate to obtain the ranges. This is accomplished with PLL RANGE circuit *U7* in Fig. 3. Resistors *R8 and* R9 are connected in series to ground. Transistor *Q1* shunts *R9* to ground when the high range is in use and is controlled by a latch output from *U7*.

PLL *U6* provides clean 50% duty-cycle square waves over a 1-Hz-to-10- MHz range. In lab tests, I've gotten the unit to run as high as 12 MHz, and it could probably go to 20 MHz or more if the divide-by-N counter could handle the frequency.

With the vco having a 1-MHz-to-10-MHz output range, I used two counters—*U3* and one of the three in *U5*—to obtain the full 1-Hz-to-10-MHz range. With four sources for the output signals, I've included a way of selecting one of four, using *U4*. Only four of the eight inputs are used. The remaining four connect to ground and are never addressed.

Chip *U4* is controlled by 74HC75 latch *U7*. This latch also controls the frequency range of PLL chip *U6*.

Instead of designing a high-speed operational amplifier or discrete-component variable-amplitude output stage, I opted for a single fixed-amplitude TTL-compatible output driver. The circuit uses 74HC04 hex inverter *U2*, with all six outputs connected in common and fed by the output of *U4*. The common inverter outputs couple through *R10* to *J1*.

With the *U2* inverter arrangement, you can drive a 50-ohm load without blowing the 74HC04 chip. You can even short the output to ground at *J1* for prolonged periods of time without damaging the output stage.

Keep in mind that a coaxial cable running from the output of the Frequency Synthesizer to a circuit should be terminated, ideally in the characteristic impedance of the cable, which is usually 50 ohms. For most applications, a 1,000-ohm terminating resistor will do the trick. If you don't terminate the output cable, you'll get ringing, overshooting and undershooting in the output signal. Some circuits don't take kindly to such electrically "noisy" signals.

The reference oscillator consists of one of the six inverters in the 74HC04, *U1A*, which is biased into its active region by *R2*. The output of the inverter connects to crystal *Y1* through *R1*. The other terminal of *Y1* connects to the input of the inverter. Capacitor *C1* provides a means for trimming the oscillator's operating frequency to within a few hertz of the crystal's stamped value, while capacitor *C2* completes the Pierce crystal oscillator configuration to ground. The output of the oscillator is buffered by inverter *U1B* and is then fed to counter *U5*.

The remaining four inverters in *U1* are used to clean up the two pulse lines from the PC to the Frequency Synthesizer. The remaining PC lines are more or less passive in that they're set and left to settle down. The two buffered lines are used to activate the counter chip's write and chip-select lines and the latch chip's strobe line. These lines must be clean because, if they aren't, spurious signals could activate either the latch or the counter inputs and cause the system to "crash."

I didn't include a full power supply for the Frequency Synthesizer. Instead, I used on-board regulator chip *U8* through which you can supply power from a standard 9-volt dc supply, plug-in or bench-type as you choose. Resistor *R13* provides current limiting for driving a power-on LED for the circuit.

How It Works

Plug *P1* in Fig. 3 tells most of the story. Pins 2 through 8 run to data line D0 through D7 of *U5*'s data port. Note, too, that pins 2 through 5 of *P1* also connect to data inputs D1 through D4 of four-bit latch *U7*.

To control the range of the vco, you select the output frequency source and turn on/off the output of the MUX circuit during programming. You then load the desired bit pattern into the lower four bits of the PC's printer data register with an OUT instruction. Then you strobe the data into latch *U7* when the PC's printer-port control register at pin 17 (C3) of *P1* generates a positive-going pulse.

In addition to the data lines, the counter chip has five control lines. Since nothing is being read from the chip, the /RD (read) line is tied high to +5 volts. This leaves just four lines with which you have to contend.

/CS and /WT, chip-select and write, must both be set low when transferring a byte from the data bus to the chip. This is accomplished by tying them together and driving them from pin 16 of *P1* through a filter made up of *R4, C3, U1C* and *U1D*.

Four internal registers are used to program the three counters in 82C54 counter *U5*. Registers 0, 1 and 2 correspond to counters 0, 1 and 2. Register 3 is used to program operating modes and direct preset numbers to the appropriate counter. Inputs A0 and A1 address the internal registers in binary and must be set to correspond to the operation being programmed. A0 and A1 connect to pins 1 (C0) and 14 (C1), respectively.

By sending different bit patterns to the PC's printer control register, A0, A1, /WT and /CS can be manipulated, along with data lines D0 through D7, to perform all operations needed to program the counter (Fig. 4).

Fig. 5 summarizes the *P1* lines that are connected to the counter and latch chips. The function of each of the latch bits is illustrated in Fig. 6 L0 selects the range of the vco (1 MHz to 3 MHz or 3 MHz to 10 MHz), and L3 turns on and off the outputs of the PLL and MUX. The reason for turning off the outputs of the PLL and MUX circuits is to prevent any circuit connected to the Frequency Synthesizer from responding to the unpredictable things that can be generated by the counter/MUX/output-amplifier chain during programming.

Counter 0 in *U5* divides the crystal frequency down to feed to one input of the phase comparator. Counter 1 divides down the output of the vco to 1,000 Hz to provide the second input of the phase comparator. Counter 3 provides the first four decades of division for the output signal and is driven by the vco.

The output from counter 2 feeds decade divider *U3A*, which, in turn, feeds decade divider *U3B*. The outputs from *U3A* and *U3B* generate the 10-to-100-Hz and 1-to-10-Hz ranges.

The outputs from the vco, counter 2, *U3A* and *U3B* are fed to MUX *U4*. L1 and L2 of *U7* select which signal is fed to the output amplifier.

Construction

Printed-circuit construction is recommended for this project, though it's possible to assemble it on perforated board using suitable Wire Wrap or soldering hardware. You can make your double-sided pc board from the actual-size artwork given in Fig. 7, but you'll need special hardware to be able to solder IC sockets (recommended) and *P1* into place. Alternatively, you can purchase a ready-to-wire board from the source given in the Note at the end of the Parts List.

Assuming pc construction, place

(continued on page 73)

CONTROL WORD BITS	D7	D6	D5	D4	D3	D2	D1	D0
CONTROL WORD NAMES	SC1	SC2	RW1	RW0	M2	M1	M0	BCD
COUNTER 0	0	0						
COUNTER 1	0	1						
COUNTER 2	1	0						
READ/WRITE LSBYTE FIRST, MSBYTE SECOND			1	1				
SQUARE WAVE MODE					0	1	1	
BINARY COUNTING								0

Fig. 6. Control word format.

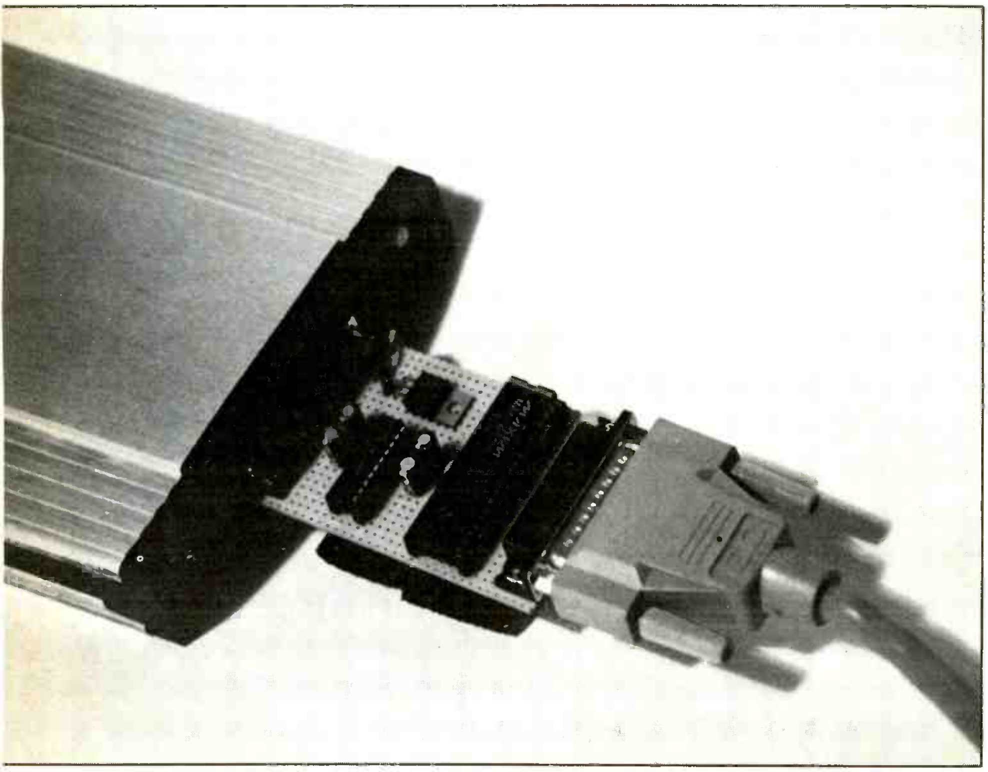
Serial-Port Surge Protector

This two-chip build-it-yourself circuit provides serial-port ground isolation and surge protection

Most of us accept surge protection circuits on the input to the power-supply circuit of a computer platform as a given. However, if you use a modem or fax board, any events that occur on the telco network can reach your computer through the serial port to which your modem is connected. Consequently, unless proper measures are taken to isolate the outside world from entering your serial port, you run a very real risk of damage to your computer from out-of-spec electrical disturbances.

One way to ensure your computer against such damage is to incorporate into its serial port the circuit described here. Our Serial-Port Surge Protector is completely transparent to data streams up to 9,600 bits per second (bps). Installing it between the serial port on your PC and the device connected to this port provides 1,520 volt rms isolation between computer and serial device for a period of 1 second. This level of isolation makes it less likely that potentially catastrophic voltages originating outside the PC will reach your computer.

Though this circuit is designed to protect a modem/serial-port interface, it can be applied in other situations where transient voltages and high differential ground potentials may exist, such as in industrial control environments. The isolation circuit is able to withstand a continuous ground differential up to 130 volts rms and can provide isolation for a full minute when the differential is 1,260 volts rms.



About the Circuit

Shown in Fig. 1 is the complete schematic diagram of the Serial-Port Surge Protector circuit. This two-chip isolation circuit is possible because of a new device from Maxim. The MAX252, shown as *IC2*, provides the isolated data interface that the Serial-Port Surge Protector uses. It's a multi-chip module packaged in a 40 pin DIP.

Contained in the MAX-252 are two separate RS-232 chips, each of which provides a pair of RS-232 transmitters and a pair of RS-232 drivers. These two chips are isolated from each other by optical isolators built into the 40-pin DIP package. Data is transmitted between the two RS-232 chips through the optical isolators. This arrangement of functional modules creates the data isolation between the input and output sides of the MAX-252.

The chip also contains a small transformer (there's actually a small toroid transformer in the chip), four capacitors and a pair of diodes. These components create a switching power supply that's used to provide isolated power to one of the RS-232 chips in the device. This power supply also provides an isolated supply and ground that creates an electrical barrier between the ground circuit on the PC side and the ground circuit on the externally connected device.

With the above arrangement of data isolation and ground-path isolation, the PC and modem (or other device) have no electrical connection between them at all, yet are able to transfer data between each other.

The Serial-Port Surge Protector also employs a Motorola MC145407 RS-323 receiver/driver (*IC1* in Fig. 1). Notice that the RS-232 receivers and drivers in both the Motorola MC-145407 and the MAX-252 invert the data stream. Connecting these two de-

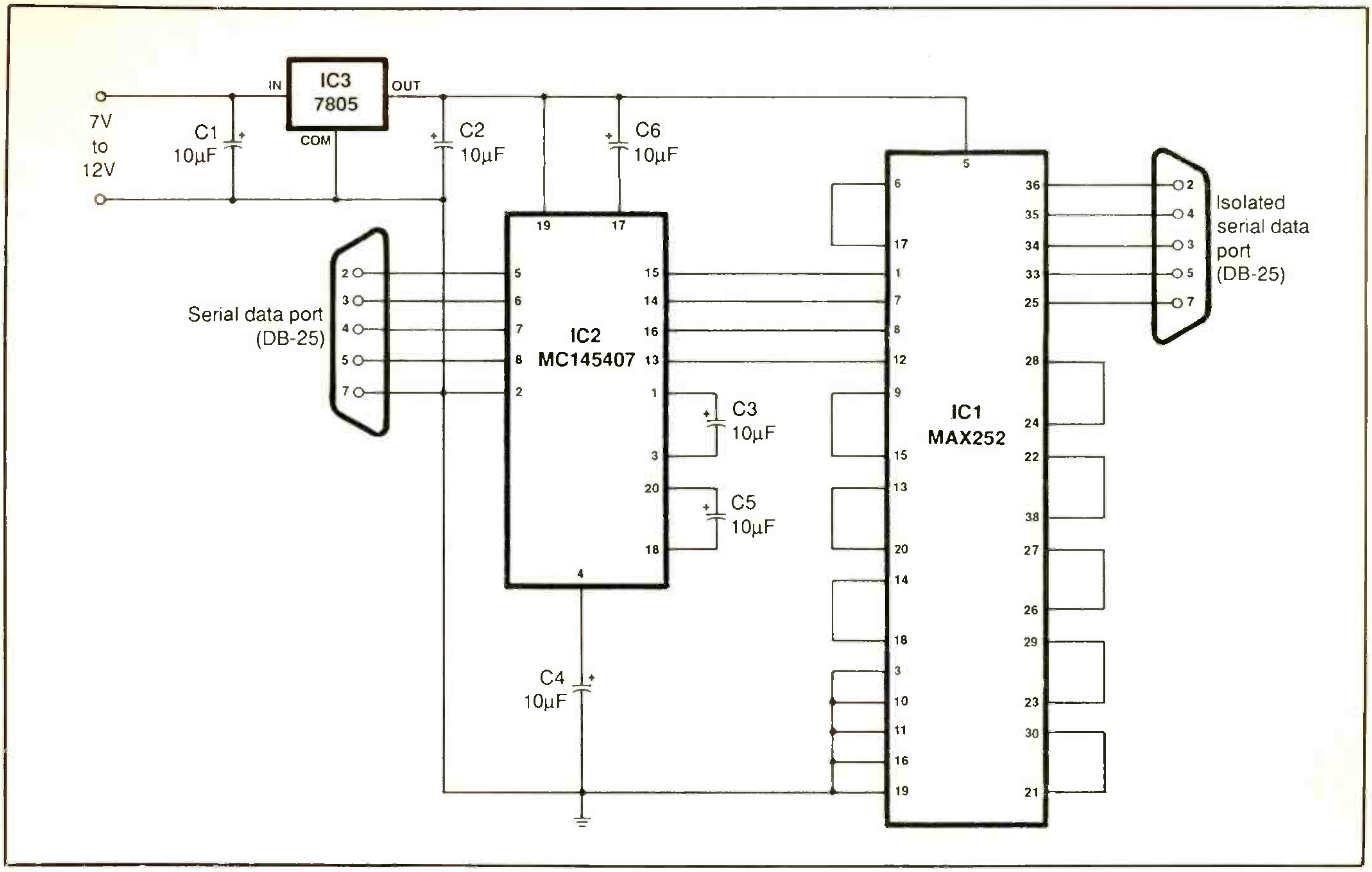


Fig. 1. Complete schematic diagram of the Serial-Port Surge Protector circuit.

PARTS LIST

Semiconductors
IC1—MAX-252 (Maxim)
IC2—MC145407 (Motorola)
IC3—LM7805 fixed +5-volt regulator
Capacitors
C1—10-μF, 35-olt tanatalum
C2 thru C6—10-μF, 16-volt electrolytic
Miscellaneous
Printed-circuit board or perforated board with holes on 0.1" centers and suitable Wire Wrap hardware (see text); sockets for IC1 and IC2; male and female DB-25 connectors (see text); suitable enclosure; hookup wire; solder; etc.

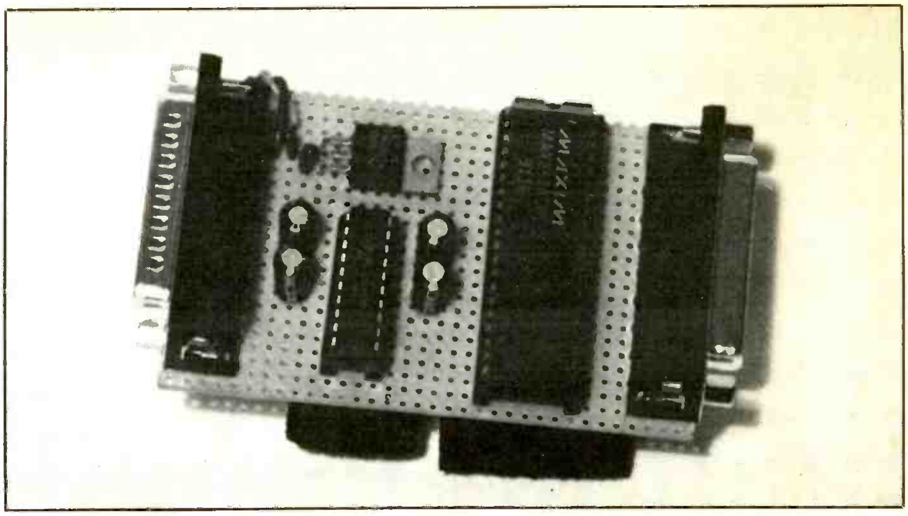


Fig. 2. Author's prototype was wired on perforated board using the Wire Wrap technique. DB-25 connectors mount at opposite ends of the board.

vices in series in this manner provides a double inversion, so net inversion takes place within the circuit. This makes the circuit completely transparent to the data. In fact, as far as the data is concerned, the Serial-Port Surge Protector is just a "cable."

Choice of the MC145407 for this circuit was based on the fact that it also contains an internal switching power supply that provides to its drivers voltage levels that conform to EIA RS-232D and CCITT V.28 standards. Electrolytic capacitors *C2* through *C5* attached to the MC145407 are used by the internal charge-pump circuitry to generate ±10 volts. This simplifies the power-supply requirements of the entire circuit.

Since both ICs used in the Serial-Port Surge Protector are able to operate from a single +5-volt supply, all that's required is to use regulator *IC3* in a straightforward manner.

Construction

If you'd like to fabricate a printed-circuit board for the Serial-Port Surge Protector, get the MAX-252 application notes from Maxim. However, the circuitry is simple enough for you to Wire Wrap it instead, as was done for the prototype shown in the Fig. 2.

Use of sockets is highly recommended for the integrated circuits. When wiring the circuit, of course, pay careful attention to polarization of electrolytic capacitors, the basing of regulator *IC3* and orientations of *IC1* and *IC2*.

Use a female DB-25 connector on one port and a male DB-25 connector on the other port to make the Surge Protector gender-transparent. Since the circuit is able to send and receive data bidirectionally, using one male and one female connector on the circuit assembly gives the Serial-Port Surge Protector the capability to hook into any serial connection, regardless of the gender of the connector on that device.

Summing Up

Placing the Serial-Port Surge Protector in-line between your PC and any serial device lessens the possibility that the PC can be damaged by any high voltages or transients that may originate from the outside world. Use of this project is very sensible in areas where electrical storms occur frequently and in applications where ground level differentials are likely to exist for any reason. The data transparency of the Serial-Port Surge Protector makes it useful in all applications where the transmission rate doesn't exceed 9,600 bps. ■

CorelDRAW 2.01 Versus Arts & Letters Graphics Editor 3.1: Evaluating and Comparing These Popular Draw Programs

If you're interested in creating images on an IBM/compatible computer running *Windows* 3.0, you have to choose between sticking with *Windows Paint*, the freebie program Microsoft throws in with *Windows* or spending about $700 on a full-featured drawing program. Many users have undoubtedly wasted a lot of time exploring the first option, only to discover that *Paint* is a veritable toy. For serious image creation, you need a real tool.

Here we compare two of the finest drawing tools available for the PC: *CorelDRAW* 2.01 and *Arts & Letters Graphics Editor* 3.1. Each package is full of disks, manuals and other paraphernalia (see Table 1). Both programs have similar requirements: an IBM PC/AT, PS/2 or compatible computer with a hard disk, DOS 3.0 or later, *Windows* 3.0, at least 1 M of RAM (640K for *CorelDRAW*), EGA or better video and a pointing device.

Owing to the high number of disks and compressed files on the disks included in the packages, installing each program took some time. Though the installation process was time-consuming, the installations themselves went smoothly.

Table 1. Contents of CorelDRAW and Arts & Letters Graphics Editor

Contents	CorelDRAW	Arts & Letters
User's Guide	Y	Y
Tutorial	8 Lessons	12 Lessons
Clip Art Handbook	Y	Y
Technical Reference	Y	N
Color Chart	Y	Y
Character Reference Chart	Y	N
Quick Reference Card	Y	N
Publisher's Type Scale	Y	N
Total Disks (3½")	7	17
Program Disks	3	2
Video Tape Tutorial	Y	N
Stand-Alone Programs	WFN Boss Mosaic CorelTRACE	Decipher

"Y" = Yes; "N" = No.

About Draw Programs

Because draw programs in general are very different from paint programs, it's important that you don't confuse the two. Paint programs create images with bitmaps that give complete control over an image down to the pixel level (all paint programs have a "fat-bits" feature that lets you edit drawings at the pixel level). The problem with bitmapped images is that they can't be scaled very well and often show jagged edges on-screen and when printed out.

Draw programs create images in a completely different way. Unlike paint programs that work at the pixel level, producing bitmapped graphics, drawing programs work with objects that can be symbols, blocks of text or freeform lines. An object isn't visible on-screen until it has outline and fill characteristics. Objects contain nodes and some sort of line to connect them. You can think of a node as a critical point along the line where directions change. A simple example is a box that has four nodes, one at each corner.

This means images created with a draw program are scalable without the problems associated with bitmapped paint programs. Smooth lines remain smooth, regardless of how much you lengthen or shorten them. Corners and bends don't get the "jaggies." Curves are true curves and can be manipulated as such. Special effects are relatively easy to create. And objects can be rotated, skewed, stretched, mirrored and manipulated in various ways.

Both *CorelDRAW* and *Arts & Letters* are certainly powerful drawing tools. If you're a skilled artist, you can use either program to produce phenomenal drawings. As an example, the steam locomotive shown in Fig. 1 (a sample drawing included with *CorelDRAW*) incorporates many of the capabilities of the program (the printout is black and white, but the original drawing is in color). If you're a skilled artist, you can create comparable drawings with *CorelDRAW*.

If your artistic skills aren't of a high caliber, your ambitions may have to be more realistic. The more you know about drawing and painting, the more effectively you'll be able to use a draw program. Even if you aren't an artist, though, you'll still be able to do some pretty nifty things with a draw program.

A variety of special effects are offered by powerful draw programs, which may include creating 3D appearance, stretching text any which way, curving text to go around an illustration, shifting objects to create perspective, and so on. Coupled with boundless color choices, various pattern fills and imported illustrations, photos and fonts, the results obtained can be spectacular.

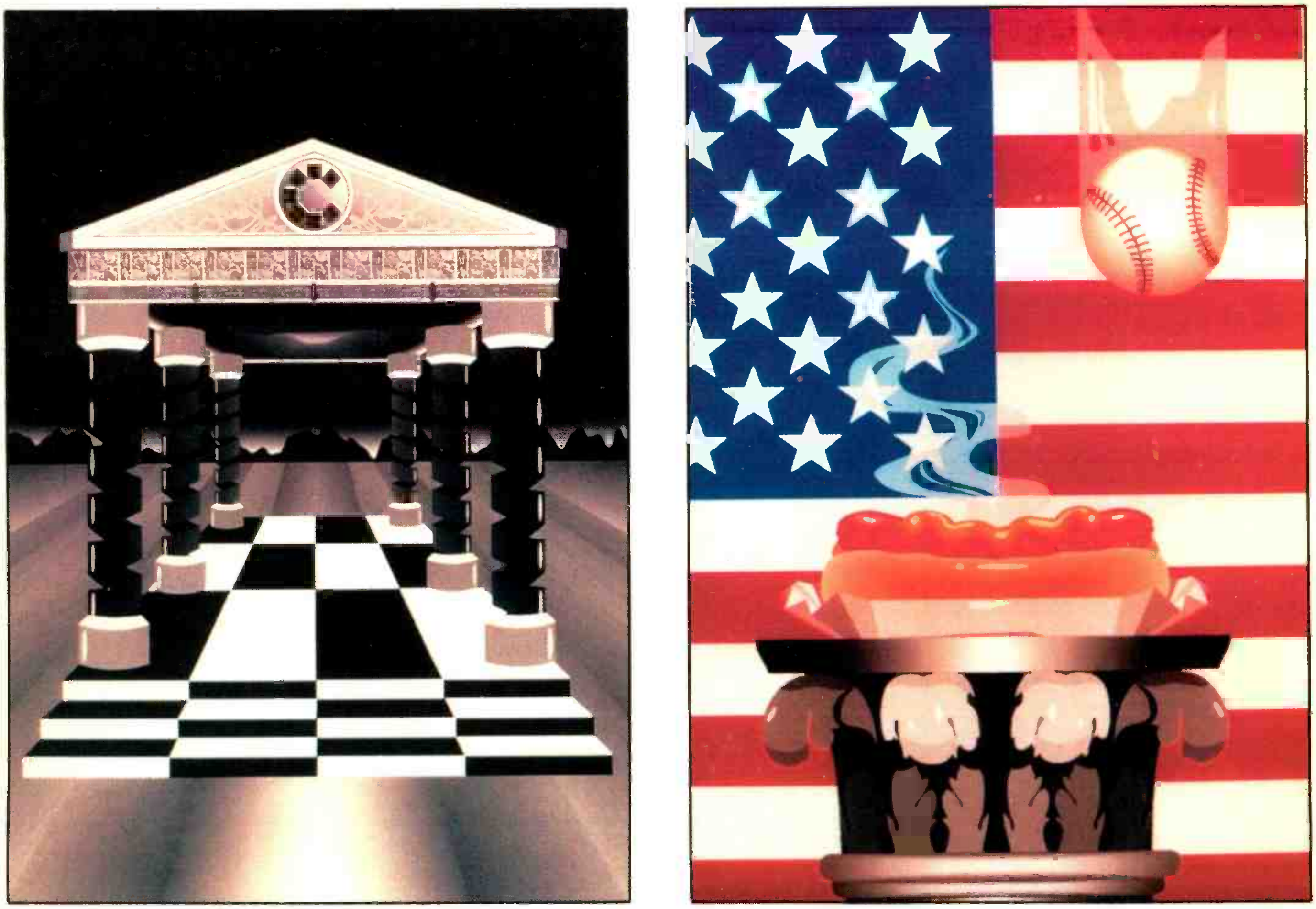
Corel Draw 2.01

When you click on the *CorelDRAW* icon, the opening screen shown in Fig. 2 appears. The main components of this screen are the menu bar along the top, status line just below it, editing window with printable page area in the center, color palette along the bottom and toolbox at the left. Not shown is the preview screen.

The Toolbox

CorelDRAW's toolbox menu extends down the left edge of the screen. The first of nine options shown is the "Pick" tool. Its operation is similar to many other graphics-oriented programs, such as *Pagemaker*. You use the Pick tool to size or scale objects in the horizontal and vertical planes. You can also use this tool to produce a mirror image of an object.

Pressing the Ctrl key while performing an operation has the general effect of constraining the operation in *CorelDRAW*. In the case of the Pick tool, constraining can be used to duplicate an existing object or make an exact mirror image of it.



Examples of some of the things you can create with CorelDRAW.

Double-clicking on the Pick tool presents new options. These let you skew objects vertically and horizontally. Also, you can rotate an object about its center and adjust the center of rotation to any other point on the screen.

A Shape tool is located beneath the Pick tool. As its name implies, the primary function of this tool is to alter the shape of objects or groups of objects. This tool works in conjunction with nodes that appear on the object when you select the tool. You can make circles and ellipses into arcs and pie wedges with the Shape tool. Clicking on and dragging a node causes the object to change shape accordingly. Using the Constrain option limits movement to horizontal or vertical displacement.

There's a second means for affecting the shape of a curve. Nodes have control points, which you can think of as graphical levers you move around to cause the curve to change shape (see the "Working With Curves: First Steps" box). Depending on the type and position of a particular node, it can have up to four control points. Each control point affects the shape of the curve differently.

Screen shot shows flexibility of Arts & Letters.

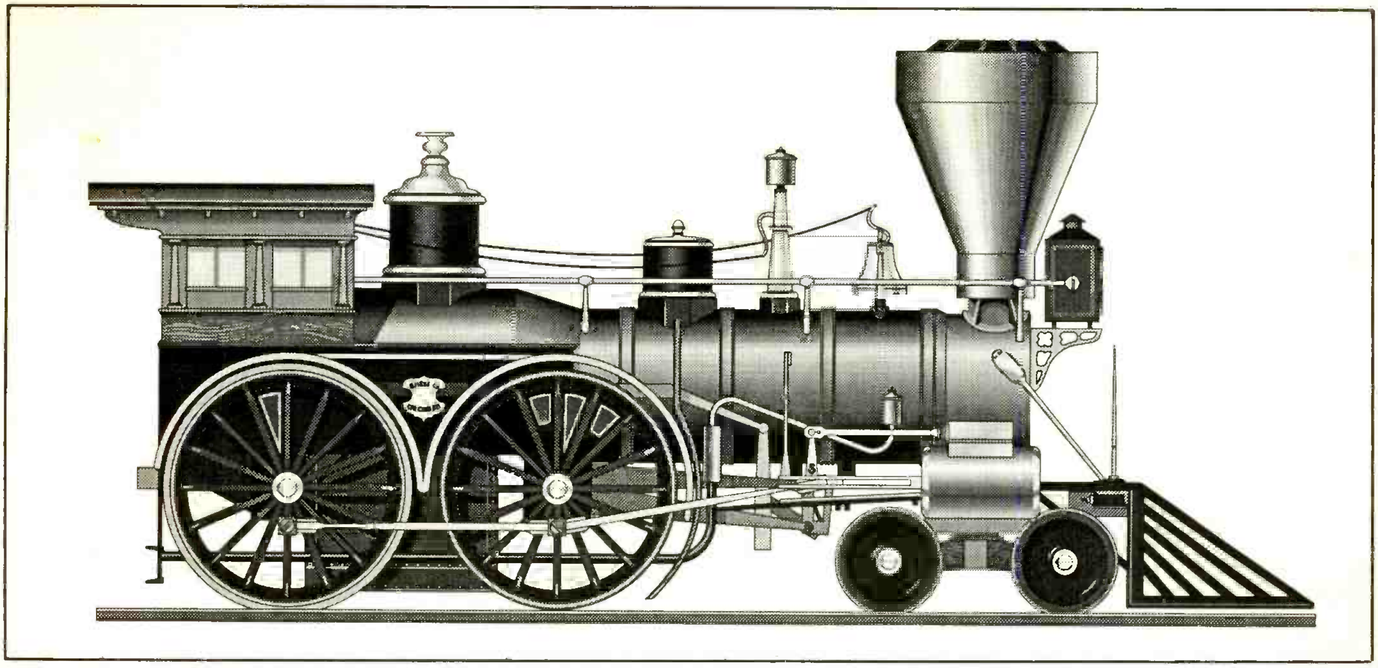


Fig. 1. Sample drawing from CorelDRAW incorporates many of the capabilities of this program.

If you double-click on one node of a selected object, the node becomes highlighted and you're presented with a menu that offers several choices: Delete, Add, Break, Join, toLine, toCurve, Cusp, Smooth, Align and Symmet. For the most part, using these options is intuitive. Double-clicking on a line segment with the Shape tool highlights the segment. The same menu pops up as before.

Even though *CorelDRAW*'s native mode of operation is with objects, you can also edit bitmapped graphics. In this case, you use the Shape tool primarily for cropping (trimming) the bitmap. The Shape tool is also useful for working with text.

Next on the tool menu is Zoom. When you click on its icon, a submenu juts out horizontally, offering five options. A magnifying glass with a plus sign in it is used to zoom in on some portion of the drawing. *CorelDRAW* automatically displays the area selected in an area as large as possible without changing the aspect ratio. The exact amount you can zoom in depends on the type of monitor you use. In general, you should be able to zoom in to about 10 times the size of the original. The preview screen automatically updates to show the same area as the work screen.

A magnifying glass with a minus sign in it reverses the steps of the one with the plus sign. Click on this icon, and the screen zooms out one level. A 1:1 option displays the page on the screen at the same size as it will print. The User's Manual points out that this is dependent on *Windows* having an accurate idea of your screen size. A "Fit in window" option causes every object in the drawing to be shown on-screen in the largest size that fits. Finally, a Page option shows the entire page with the picture in

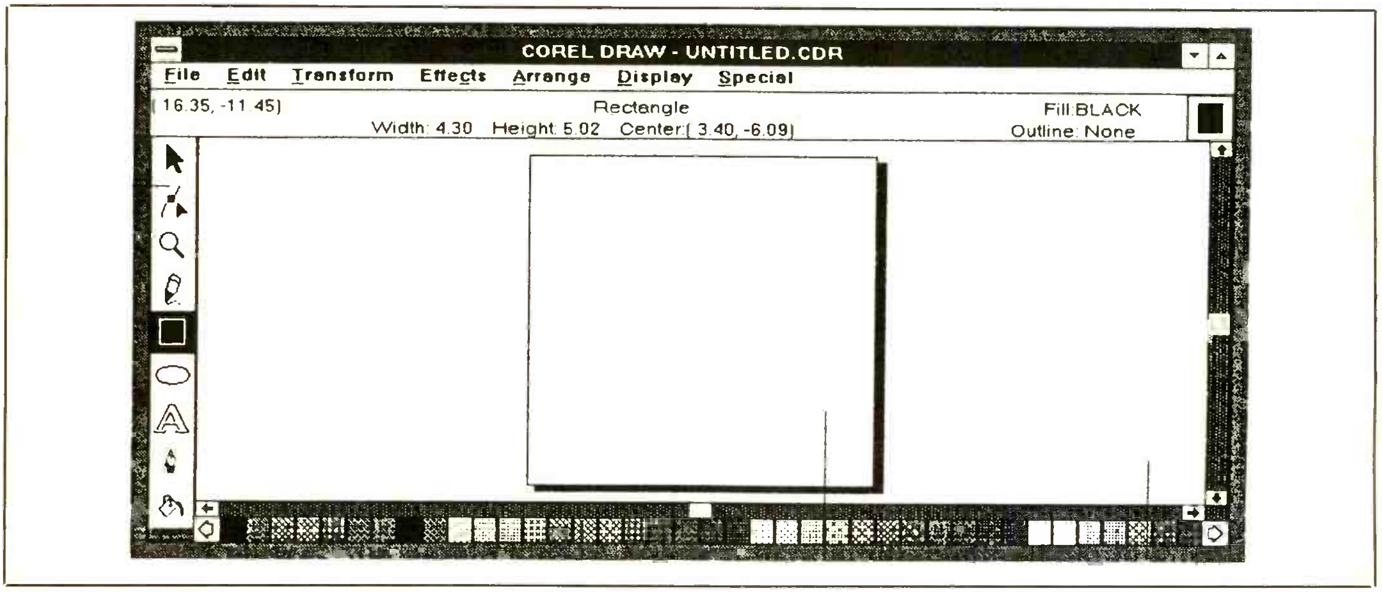


Fig. 2. CorelDRAW's opening screen.

the position where it would print.

You use the next four tools to create new objects in a drawing. A pencil is the icon for the Line/Curve tool. Constrain this tool, and you're limited to straight lines. With Constrain on (pressing the Ctrl key), straight lines are limited to 15° increments from the 0° horizontal line. When drawing curves, you can choose to draw in a freehand mode in which the curve follows the pattern you create while dragging the mouse. You can also use Bezier curves, in which you set down the nodes and *CorelDRAW* fills in the curves in a connect-the-dots fashion.

You can also use the Line/Curve tool to access an Autotrace feature (this is in addition to the more powerful CorelTRACE utility included in the package). This tool handles TIFF- and PCX-format files. Results from this option are useful so long as the drawing is simple.

The Box tool is useful for drawing all kinds of boxes. Hold down the Ctrl key while drawing the box, and *CorelDRAW* forces the object to be a square. Regardless of the aspect ratio, the object will be a rectangle of one sort or another (90° angles in each of four corners).

The Ellipse tool is similar to the Box tool, except that you use it to draw ellipses and circles. Holding down the Ctrl key constrains the drawing to a circle.

Text is one of *CorelDRAW*'s most-powerful tools. In fact, the-text handling capabilities are so good that you could use *CorelDRAW* for rudimentary desktop publishing if need be. However, text formatting and positioning isn't strong enough, even for moderately complex documents. If you're dealing with more text than you would find in a glitzy ad, you'll need a page-layout program, too.

CorelDRAW comes with more than 150 typefaces. Moreover, you can adjust thickness, nib shape and fill to create effects that give a typeface a different appearance. Furthermore, you can create your own typefaces with the tools provided.

Two modes are available for creating text. If you want headline (display) text, you select the Text tool and click once in the work-screen area. A new window opens over the existing one and presents you with a pull-down menu for selecting typeface and text attributes. Since *CorelDRAW* is object-based, type is fully scalable. Not only can it be increased or decreased in size, it can be rotated, skewed, stretched, mirrored, etc.

The second method of creating text is similar to the first, but it's oriented toward blocks of text. You select the Text tool and drag the mouse to create a box where you want to place the text. When you release the mouse button, the Text window opens with the same options as in the display-text mode. Other options are available in this mode. For instance, you can import ASCII text and create columns.

You can set point size in the Text menu (both modes), as well as alignment (left, center, right, full justification or none). Depending on typeface, you may be able to select bold, italic or bold-italic attributes. Also, you can modify inter-character, inter-word, inter-line and inter-paragraph spacing.

A subset of the Text tool is a library of symbols you can place in a document and use much like text. *CorelDRAW*'s Symbol Library contains nearly 3,000 symbols that range from electronics to business to the environment. You could, for instance, create an electronic schematic diagram from the symbols available.

The next option on the tool menu is Outline, which has a fountain-pen icon. Clicking on the icon produces a slide-out menu with a second fountain-pen icon. Clicking on this icon causes a window to open with several options. One lets you define the shape of the Outline tool's nib. Once you define the shape of the nib, *CorelDRAW* produces lines as if drawn by pen with a nib shaped like the Outline nib. Solid or dotted lines (15 variations), corner shapes behind fills and arrowheads are available, too, though these are four levels down on the selection menu.

The balance of the options on the first line of this slide-out menu are for selecting the thickness of the stroke line. Thickness ranges from 0" (no line) to 0.333". The second level of the slide-out menu is concerned with the color of the outline. You click on the paint-brush icon for a menu of color options, from which you can select between CMYK (cyan, magenta, yellow, black—used in four-color printing), RGB (red, green, blue) and HSB (hue, saturation, brightness). If you're going to print your drawing on a PostScript device, several PostScript options are available.

A word of warning: *CorelDRAW*'s color capabilities far exceed those of the programs to which you may want to export drawings. For instance, most page-layout programs don't adequately support the color-separation process. Until more sophisticated packages are available for the IBM platform, you might have to transfer your work to a Mac for color processing—or do it the old-fashioned way and let a service bureau strip film.

Each *CorelDRAW* object contains an inner area that can be filled with a pattern, color or gray shade or be left empty. The final tool gives control over these "fills." As with outlines, Corel offers a wide range of sophisticated options for fills: colors, shades of gray, PostScript textures, etc. Fountain fill, though, is one of the more useful ones. It lets you define a fill that blends two colors or tints of color.

You also have a choice between radial and linear fountain fills. A radial fill changes color from the center of the object outward in concentric circles. Linear fill changes color in one direction. You specify starting and ending colors, as well as the angle (direction) of the fill for linear fills. Fountain fills (shading) are particularly useful for enhancing the depth of a drawing on which you're working.

With *CorelDRAW*, you get a free Pantone license, enabling you to select the long-time printing industry's standard for printing colors. You also have the option of choosing process color mixing.

Program Menus

CorelDRAW's File menu is standard *Windows* issue for the most part, with "New," "Open," "Save" and other options. A couple of features are worth noting, though. A "Print Merge" option provides for mail-merge-type functions. You could, for instance, create a certificate of appreciation for volunteers who helped out at the local heart association 10-km run. Then create a merge file with the names of all participants and select the Print Merge option. *CorelDRAW* would then print a personalized copy of the certificate for each participant.

"Import" and "Export" options do what you'd expect them to do. If you import a bitmapped graphic for tracing, you must select that item from the submenu before starting the import. Although *CorelDRAW* has become one of the standard graphics packages in the IBM environment, its proprietary file format hasn't. The program exports files in several "standard" formats that can be used easily by other programs. Encapsulated PostScript (EPS) is the most useful if you'll ultimately be printing on a PostScript device.

CorelDRAW allows you to include an image header in an EPS file. *PageMaker*, for instance, can work with EPS files, but it can't display them on-screen. Including the low-resolution bitmapped image header allows you to see on-screen what the graphic will look like and size it as needed. PCX and TIFF (bitmapped) formats are available as well.

CorelDRAW's Print option has a couple of interesting features. With it, you can elect to print only selected items from a drawing. This feature can greatly speed up "proofing" a complex drawing. If you haven't dealt with printing complex graphic pages, PostScript or otherwise, you may be surprised at how long it can take to do. As an example, the drawing in Fig. 1 took just over 17 minutes to print on a Hewlett-Packard LaserJet IIIP printer with the HP PostScript cartridge installed and driven

Fig. 3. An example of what you can do with CorelDRAW's blend capability.

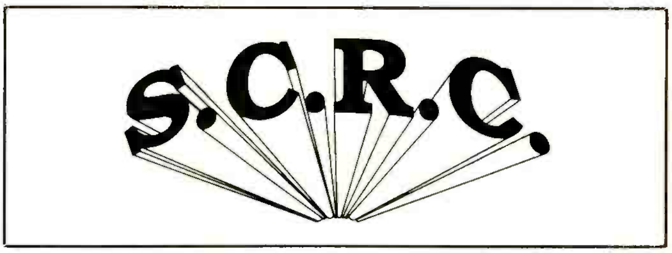


Fig. 4. An extrusion is actually a separate group of objects.

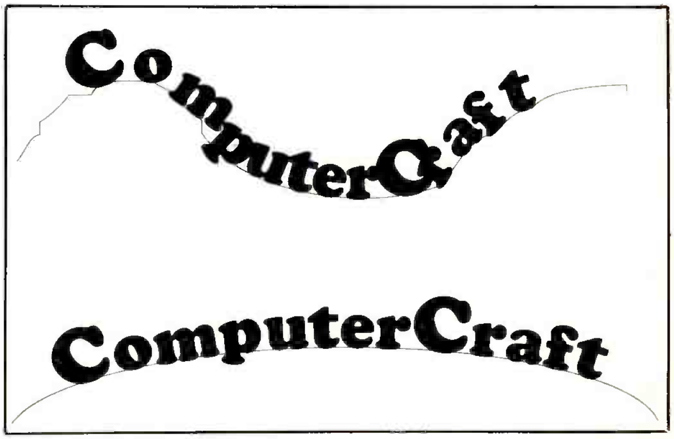


Fig. 5. Result of using CorelDRAW's Fit Text To Path ability.

by an Everex 20-MHz 386SX notebook computer. The same picture took well over 25 minutes to print when using a Heath 8-MHz 286 computer connected to an HP LaserJet III with the HP PostScript cartridge installed. Both printers have the full complement of memory.

The Edit menu presents few surprises. "Undo" and "Redo" options toggle between the last two operations. "Cut," "Copy," "Paste," "Clear" and "Duplicate" perform the operations you would expect them to from their names. "Copy Style From" allows you to copy an object's style from one object to another. Style items include outline pen, outline color, fill and text attributes (if it's a text file). If you select a block of text and choose "Edit Text," you're taken back to the text menu, where you can change any of the text attributes in the same manner in which they were selected with the Text tool.

A Transform menu provides alternative means for moving, rotating, skewing, stretching and mirroring objects. You can perform all operations here with the Pick tool, except for "Clear Transformations." This option resets all parameters back to those of the original object. It's useful for situations where you've applied numerous operations—"Undo" works with the last activity only.

The Effects menu is where *CorelDRAW* really shines. From it, you can edit an object's envelope (general shape). The envelope has eight "handles" that allow you to pull part of the object in one direction. Four options (straight line, single arc, double arc and not constrained) are available for editing an envelope. Since type is defined as an object by *CorelDRAW*, you can use an "Edit Envelope" option to create dramatic effects. You can also add a new envelope to an object.

A Perspective option allows you to create one- or two-point perspective views of an object. After selecting "Edit Perspective," you can drag one of the pop-up handles to create perspective. You can also manipulate the vanishing points if they're onscreen (depending on the particular drawing and how much perspective was introduced). You can align pairs of vanishing points, and you can copy perspective from one object to another.

You can create dazzling graphics with a "Blend" option. In simple terms, blend graphically transforms one object into another. For example, a square with a brick fill can be gradually transformed into a circle (or other object) with a different fill. Another point of interest is that *CorelDRAW* has the capability of blending some fills—colors and shades of gray, for instance. Had the square been filled with one color and the circle with another, the intermediate steps would have been intermediate shades between the two colors.

Blend works with text objects, too. In Fig. 3, the word *ComputerCraft* seems to be falling down and turning upside-down. To create this effect, you first copy the word *ComputerCraft* and move it to the bottom of the screen and then turn the word upside-down. You then select these two objects for blending. Next, you instruct *CorelDRAW* to rotate the intermediate steps 180° between beginning and end. Finally, you map the first node of the "C" on top to the first node of the "C" on the bottom.

Another effect, called extrusion, gives objects an illusion of depth. You have a choice of producing either parallel or perspective drawings. Extrusions with perspective are much more realistic in appearance than those that don't have extrusion. You can extrude objects of any sort (text, boxes, ellipses, curves).

An extrusion is actually a separate group of objects. This becomes evident if you attempt to move an extrusion on the page. You get two items on the page, the original object and some very strange looking lines! This isn't all bad. Notice in Fig. 4 that the face of the type is black while the fills in the extrusion are white. This wouldn't be possible if the original object and extrusion were made into a single object. The slight rounding effect of the type is achieved by pushing up the bottom and top of the type envelope before extrusion is applied.

The Arrange menu offers numerous options for arranging objects that overlap each other. You can push to the front, push to the back, move one item forward or backward and reverse the order. Also, you can group and un-group options using the features in this menu.

Two other options on the Arrange menu deserve comment. "Convert To Curves" permits selection of a block of text and converting it to curves. This further enhances *CorelDRAW*'s ability to manipulate text. The other option, "Fit Text To Path," lets you wrap text around some object. The text bends and twists to fit the path of the object, as shown in Fig. 5.

The Display menu offers many options that directly affect on-screen information as you work with *CorelDRAW*. This version of *CorelDRAW* includes new items with which users of *PageMaker*, for instance, are already familiar. You can set up grids for aligning objects. You have the option of "snapping" objects to the grid—the "magnetic" grid pulls the object to it when it gets close. Rulers are now available and can be toggled on and off for precisely sizing an object on the screen.

A "Special" menu rounds out *CorelDRAW*'s menu bar. You can set up certain defaults through this system—freehand or Bezier curves, for instance. You can also extract text from a *CorelDRAW* drawing for editing in an outside word processor and then merge it back (ASCII only) when the changes have been made. You can create your own bit-mapped fill patterns or special symbols.

Using the Program

Working with a *CorelDRAW* drawing can be a bit frustrating. Outlines—what's displayed on your work screen—are nothing more than lines of equal width that make up all objects in the drawing. You can't look at the work screen and know what outline or fill attributes are used. Some of this information appears below the menu bar on the status line when an object is selected, which is great for reference but requires far too much attention to be useful while you're working with a drawing.

You have the option of toggling on a preview screen that shows how the drawing will appear either in a presentation graphic or on a printed page. The preview screen is toggled on and off with the Shift + F9 combination. Having on the preview screen is handy unless the drawing is extremely complex, such as the one of the train in Fig. 1. Since the preview screen must be redrawn each time the work screen is modified, it adds substantially to screen

Working With Curves: First Steps

One of the more difficult tasks to master with a drawing program is curve manipulation. In a drawing program, curved lines are different from straight lines. In *CorelDRAW*, you draw a curve by selecting the Pencil tool, holding down the mouse button and moving the mouse around (to draw a straight line with the Pencil tool, you click on one spot and then on another). With *Arts & Letters* you select the Curve tool, hold down the mouse button and move the mouse around (to draw a straight line, you select the Line tool, hold down the mouse button and move the mouse around).

When you draw a curve, small squares appear on the curve. In *CorelDRAW* these squares are called nodes, while in *Arts & Letters*, they're called point handles. Both nodes and point handles have associated control points. In a drawing program, you have to learn how to manipulate these features of a curve to achieve a desired effect.

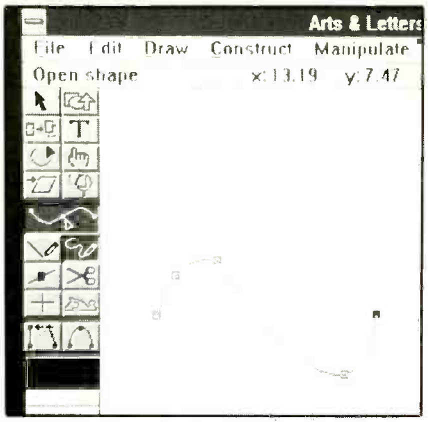


Fig. A. Freehand sketch isn't the best way to begin drawing a curve.

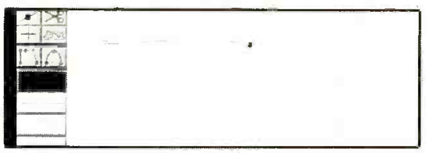


Fig. B. A sine curve starts with two nodes or point handles.

To give you a better understanding of how these nodes or point handles and control points work, I'll show you how to create a simple sine curve. When creating a curve, the "intuitive" thing to do is to draw it freehand. In a drawing program, this is a mistake since the curve will contain too many nodes or point handles (Fig. A). When drawing a curve, you

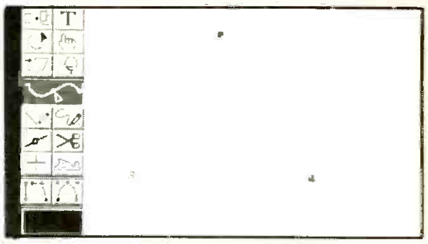


Fig. C. Curve shape is determined by the distance and direction of the control point from the node or point handle.

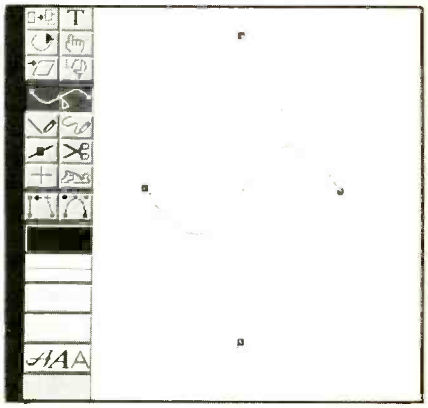


Fig. D. Placing control points along the same y coordinate yields the symmetry needed to create sine curve.

should try to create as few of these tiny squares as possible.

The best way to draw a sine curve is to draw a relatively straight line (using the program's method for drawing a curve), as in Fig. B. This gives just two nodes or point handles. Now you can concentrate on manipulating the control points. (In *CorelDRAW*, you may have to double click on the right node and select "To Curve" from the menu for the control points to display.)

The shape of a curve is determined by the distance and direction of the control point from the node or point handle (Fig. C). To create a sine curve, you must move each control point the same distance from each node or point handle. The direction of the left control point should be up and to the right, and the direction of the right control point should be down and to the left. If you place the control points along the same y coordinate, you obtain the symmetry needed to create the curve (Fig. D). Of course, as your curves become more complex, you have to deal with more nodes and their interactions with each other.

—Joseph Desposito

re-draw time. Even with a fast computer, this can be a bit of a problem.

Drawing curves and lines with *CorelDRAW* can be confusing at first. You draw curves as expected, by dragging your mouse. Drawing a straight line is done differently. Though your impulse is to drag the mouse across the screen, this technique doesn't work in *CorelDRAW*. Instead, you must click in one place, move the mouse, and then click in another place. You can easily learn to do this, but it certainly isn't intuitive.

If you need to create drawings with precise measurements, you may run into problems. Although *CorelDRAW* has a display that indicates x-y coordinates, you can't always hit the number you want. To work around this, you must magnify the screen. However, magnifying the screen reduces the work area. In one instance, I attempted to go beyond the limits of the display in the magnified area, expecting the screen to scroll. It didn't. Instead, the line segment moved about on its own like a snake dancing to music.

A problem occurred with a single file I transferred from one machine to another. Attempting to open the file caused *CorelDRAW* to give an error message indicating that the file had been corrupted. The program, however, did allow the file to be opened. After attempting to perform an operation on it, though, *Windows* gave an "Unrecoverable Application Error" message (the dreaded UAE) and shut down *CorelDRAW*. Such a problem with a *Windows* application isn't unusual and can be easily worked around.

Exporting EPS files for use with *PageMaker* (or any other program that accepts this format) is easy and painless. Compared to printing an EPS file, exporting is virtually instantaneous, taking only a few seconds. It's the mode of choice if you have a PostScript printer. You can easily scale and print EPS drawings without undue distortion. If you prefer, exporting popular bit-mapped graphics formats is also easy and fast.

Modifying drawings with *CorelDRAW* is a lot easier than drawing from scratch. You may find that you can modify a piece of clipart provided with the package to convey a certain message. One drawing in the *CorelDRAW* clipart files was perfect for a brochure I was making, with one small exception—the lady in the drawing was holding a drink. It took only about 20 minutes or so to alter the drawing to make the drink disappear. I created some special effects with type and imported them into my drawing. Within 30 minutes, I had a "custom" drawing for the brochure on which I was working.

Text manipulation is far better in *CorelDRAW* than in most drawing and paint programs. If you'll never be doing anything more than a simple single-page flyer or something of this magnitude, *CorelDRAW* could substitute for a page-layout program. However, if you're doing complex flyers with lots of text, booklets or anything else that requires significant amounts of text, you'll probably need a page-layout program. Also, complicated formatting, such as you might find on a restaurant menu, would tax *CorelDRAW*'s text capabilities.

If you use a slow computer, it can take quite some time to load a picture, regardless of how fast your hard disk may be.

The Documentation

In general, the documentation supplied with *CorelDRAW* is excellent. The User's Manual is spiral-bound and runs a tad more than 300 pages. There's also a Technical Reference manual that's a fountain of useful information. Each of the standalone utilities has its own manual, and a directory of clipart is included in the package. There's also a spiral-bound tutorial booklet that covers the main features of *CorelDRAW* (it takes a couple of hours to work through the examples). There's also a 60-minute VHS video tutorial that gives a good overview of the program.

The manuals do a good job of illustrating the features included and procedures used in *CorelDRAW*. If you can't find in the manuals the answer to a question you have and you're a registered user, you can contact Corel technical support via telephone, fax or mail at no service charge.

Extras

To help you become productive with the program as fast as possible, *CorelDRAW* includes symbol and clipart libraries (9M worth!) and their own separate manual. To access the clipart, you need to use a separate Mosaic program, which is a visual file manager that displays thumbnail sketches of graphic files on-screen. There are 14 different categories of clipart, many of which are samples supplied by third- party vendors. At the end of the manual is a directory that gives information on where to purchase more clipart from these vendors. You can use the clipart as-is or modify it with *CorelDRAW*.

CorelDRAW includes two other standalone utilities, CorelTRACE and WFN Boss. CorelTRACE changes bit-mapped images to vector images. It's easy to use and powerful, much more so than the tracing program built into *CorelDRAW*. WFN Boss is a typeface conversion utility used to convert typefaces from many different manufacturers into WFN fonts, *CorelDRAW*'s native typeface format. This lets you add any number of typefaces to the over 150 currently packaged with the *CorelDRAW* program.

Professional Results

To indicate to new users how much can be accomplished with *CorelDRAW*, Corel includes a selection of very high-quality sample files. Most of these files were created by experienced *CorelDRAW* users.

—Peter R. O'Dell

Arts & Letters Graphics Editor 3.1

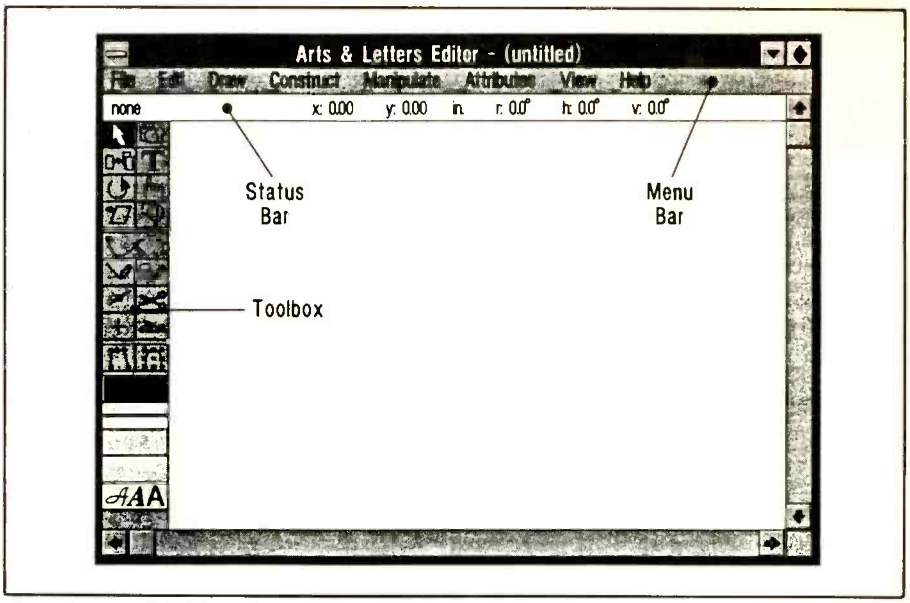


Fig. 6. Arts & Letters' opening screen.

Arts & Letters' opening screen, shown in Fig. 6, has its menu bar along the top and status bar just below it. Its toolbox is arranged vertically along the left edge of the screen. Because *Arts & Letters* doesn't use a separate preview screen, a drawing and all its characteristics are always displayed on the screen.

The Toolbox

The *Arts & Letters* toolbox is divided into four sections, the top one containing basic tools for manipulating and viewing objects. Four tools let you scale, duplicate, rotate and slant objects; four others let you add or replace symbols, add or edit text, select a block and zoom in. Most powerful of these is the "Add Symbol" tool.

In *Arts & Letters*, a symbol can be anything from an arrowhead to a map of the world. More than 5,000 clipart symbols are included in the package. After you select Add Symbol, you type in the symbol's number from the *Clip Art Handbook* supplied with the package.

For text, *Arts & Letters* provides 81 typefaces and has excellent type-handling capability. You can adjust line thickness and fill to create special type effects.

The next part of the toolbox contains the tools for freeform drawing and editing. The "Edit Freeform" tool lets you adjust the shape of an object and text by moving control points or adjusting point handles. Line and Curve tools let you draw straight lines or curves. *Arts & Letters* differs from *CorelDRAW* in this respect. *CorelDRAW* has one tool to draw lines or curves. You draw one or the other, depending on how you use the tool.

Add Handle is a tool that lets you add additional handles to curves, which increases the control you have over the shape of a freeform object. A Split tool, in the shape of a scissor, lets you split an object at any point. A Make Horiz/Vert tool lets you make line segments perfectly horizontal or vertical. A Make Smooth Join tool is useful for joining separate shapes into a single object.

The third section in the toolbox contains two toggle switches that affect such features as snap to points, snap to grid, show control points and show freeform points.

The final section in the toolbox gives quick access to the attribute dialog boxes. These tools let you change line color, line type, fill color, fill pattern and type attributes. *Arts & Letters* supports CMYK, RGB and hues. However, it doesn't support Pantone colors.

Many functions available from the toolbox are also available from the program's menus. You may have noticed that the toolbox lacks tools to draw rectangles, squares, ellipses and circles. With *Arts & Letters*, these shapes are included as symbols in the clipart library.

The Menus

For the most part, the File and Edit menus in *Arts & Letters* are typical *Windows* menus. One noteworthy item on the Edit menu is Lock/Hide/Name. It lets you assign a name to any object in a document. Once you give an object a name, you can lock or hide it. A locked object is frozen in place so that it can't be selected, moved, sized or otherwise manipulated. A hidden object becomes locked and invisible.

A Draw menu contains many features also available from the toolbox. Three that aren't are Custom Symbol, Chart and Trace Bitmap. Custom Symbol lets you add symbols to the plethora of symbols provided in the *Arts & Letters* package. Selecting Chart brings up a mini-spreadsheet program for entering numbers to produce line, pie and bar charts.

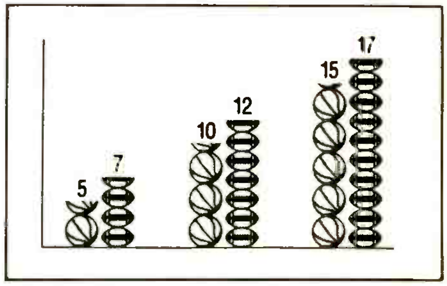


Fig. 7. A&L lets you easily produce a bar chart with stacked or stretched symbols.

You can easily produce a bar chart with stacked or stretched symbols, as in Fig. 7, by including the number of the symbol when constructing the chart. Selecting Trace Bitmap lets you convert bitmap images to vector representation, though this feature isn't nearly as powerful as it is in CorelTRACE.

Construct is another menu that duplicates many of the toolbox functions. Otherwise, the main use of this menu is to specify length and absolute angle of a line segment and x and y coordinates of a point by typing in values.

Manipulate is the menu that gives *Arts & Letters* most of its power. On this menu you'll find selections like Blend, Bind to Shape, Warp/Perspective and Flip. Bind to Shape lets you fit text to a path, as in Fig. 5. Warp/Perspective can produce dramatic special effects like that in Fig. 8.

An Attributes menu is essentially a duplication of the bottom portion of the toolbox. The View menu works in conjunction with the Zoom tool. For example, you can zoom in with the Zoom tool,

Windows Printing Problem Solved

Initially, I had difficulty printing even relatively simple graphics from *CorelDRAW*. I found out why in a discussion on one of the CompuServe forums. Microsoft *Windows* 3.0 has a time-out option for sending material to a printer. The default setting is 45 seconds. When transmission time exceeds 45 seconds, *Windows* terminates the print job and places a cryptic message on the printer control panel. Chances are, you'll miss this message, or not understand it if you do stumble across it.

A simple solution is to change the default—in two separate places! First you must access the printer control window from the Control Panel. Next, you have to select "Configure" and change "Transmission Retry" from the default of 45 to 999, which is the maximum allowable value. Then, you return to the Program Manager. Next, you must pull down the File menu and select "Run." In the Command Line box, type sysedit and hit Enter. Then select the WIN.INI file by double clicking on its menu bar. Fifteen items or so down the page you'll see "Transmission Retry Timeout" and some numeric value. You must set the value to 999 here as well. Then exit by closing the window. Finally, you must exit *Windows* and restart it. Your defaults are now changed to something that allow you to send complex graphics to your printer.

If there's a complaint here against Corel, it's that this problem isn't anticipated or mentioned in the User's Manual. I should point out that Corel does acknowledge the problem and offers a similar solution in its Technical Reference manual. I stumbled across it long after I had solved the problem by a different route.

—Peter R. O'Dell

but you need to use the View menu if you wish to zoom out.

Using the Program

As with *CorelDRAW*, creating images with *Arts & Letters* is easiest when you can make use of the program's clipart collection. I found the *Arts & Letters* collection to be more broad-based than *CorelDRAW*'s. For example, I was able to find symbols like sine curves and logarithmic scales, which sped up some of my work. On the other hand, I couldn't find any electronic symbols in *Arts & Letters*, but I did find a bunch in *CorelDRAW*.

When drawing lines and curves with *Arts & Letters*, you use two different tools.

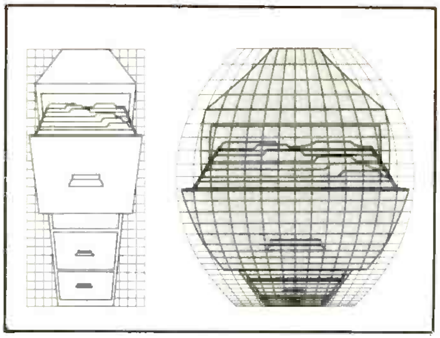


Fig. 8. With A&L's Warp/Perspective, you can produce dramatic effects.

This approach differs from the one employed in *CorelDRAW*, in which you manipulate your mouse one way for a line and another way for a curve. I made far fewer mistakes with the *Arts & Letters* method, though some people may prefer not to go back and forth to change tools.

I encountered problems converting a bitmap of a printed-circuit foil pattern into a vector representation. The Trace Bitmap feature simply traced a rectangle around the outer edge of the layout. CorelTRACE did a much better job with the conversion.

The Documentation

Arts & Letters' documentation consists of a user's guide and a 12-lesson tutorial. Although clearly written and amply illustrated, I felt the documentation lacked depth. For example, the tools aren't adequately explained, and there's very little information on creating three-dimensional effects, though several illustrations are given in the manual. No instructional video is available with the program.

Extras

Decipher, a stand-alone program included with *Arts & Letters*, is used to manipulate bitmapped images. It gives *Arts & Letters* users access to PostScript, PCX, GIF, TARGA and other files. Among other things, you can use *Decipher* to convert bitmap images from one format to another, capture images from *Windows* applications and change bitmap images using color reduction, dithering and color-to-grayscale tools.

Professional Results

Arts & Letters includes a selection of sample files to help you understand what you can accomplish with the program. In my opinion, the sample files aren't of the quality of those included with *CorelDRAW*. The sample program shown in Fig. 10 took 22 seconds to draw on-screen with an ALR PowerFlex 486 computer.

—*Joseph Desposito*

Conclusions

Drawing programs like *CorelDRAW* and *Arts & Letters Graphics Editor* offer tremendous power for anyone who wants to produce graphs, technical drawings, illustrations and other kinds of artwork. Using both products for our applications, we found *Arts & Letters* to be easier to use than *CorelDRAW*. When the necessary clipart was available in *Arts & Letters*, we were able to complete drawings at least ten times faster than with *CorelDRAW*. Even in the case where we had available *CorelDRAW* symbols (and *Arts & Letters* didn't), the procedure for using them hindered us from working at a fast pace.

One of *CorelDRAW*'s big advantages over *Arts & Letters* is its tracing program. The stand-alone CorelTRACE program is a much more powerful utility for turning a bitmapped image into a vector image than the one built into *Arts & Letters*. Another *CorelDRAW* advantage is its support of Pantone colors.

A more subtle advantage of *CorelDRAW* over *Arts & Letters* is the momentum it has in the marketplace. The company's ability to encourage users to share ideas through special promotions is a subtle benefit a serious user may not want to pass up.

In choosing between these two powerful programs, *Arts & Letters* gets highest marks for ease of use, while *CorelDRAW* wins on overall depth of features. These views are impressions generated by early use of the subject programs. They may change with heavier work, of course. For example, a person we spoke to who has used both regularly said he prefers *CorelDRAW* because he finds it *easier* to use since he doesn't have to switch out of one part of the program to do another drawing task. ■

In Brief

CorelDRAW 2.01, $695
Corel Systems
1600 Carling Ave.
Ottawa, Ontario, Canada KIZ8R7
Tel: 613-728-8200; Technical Support: 613-728-1851

CIRCLE NO. 161 ON FREE INFORMATION CARD

Arts & Letters Graphics Editor, $695
Computer Support Corp.
15926 Midway Rd.
Dallas, TX 75244
Phone: 214-661-8960; Technical Support: 214-661-8960

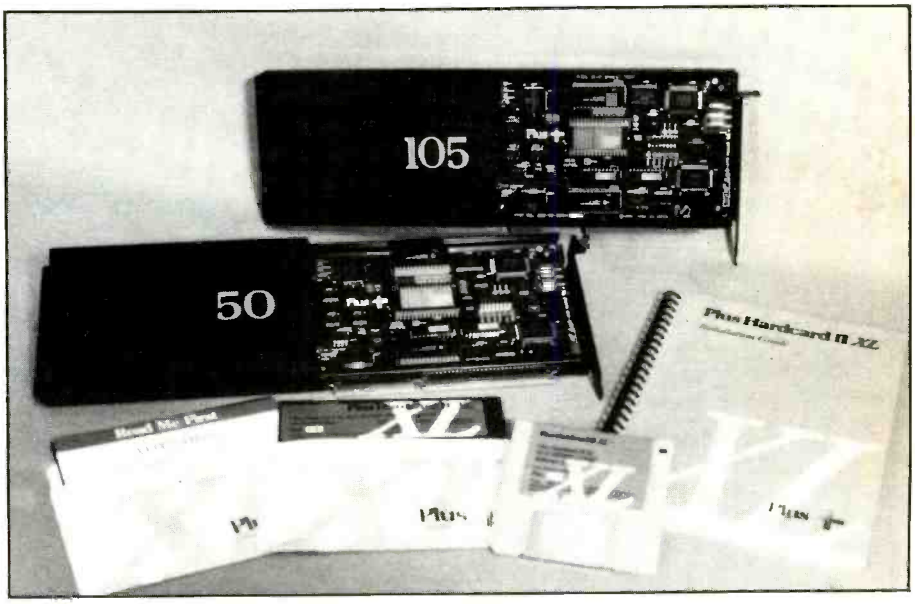
CIRCLE NO. 162 ON FREE INFORMATION CARD

Hardcards: Sensible Storage Solutions for PC Compatibles

Plus Development's expansion-slot hard drives come in 52M and 105M capacities, while being slim, fast and transportable

Today, a 40M hard drive is the bare minimum realistic capacity for sufficient program and data storage. If you run *Windows* 3.0 and several *Windows*-based applications, you'll exceed its capacity unbelievably fast. If you find yourself in this situation and want to increase hard-drive storage capacity but don't have a drive bay in which to mount a second hard drive or don't want to struggle with mounting brackets and making connections, you're a good candidate for a "hardcard." These handy devices can give you 50M or more of hard-disk space without requiring an open drive bay. They combine a hard drive and its interface electronics on a 16-bit card that simply plugs into any open expansion-bus slot in an AT-compatible computer.

In this review, I'll detail my experiences with a pair of hardcards for AT-bus computers from Plus Development Corp. that feature a blazing 9-ms disk access speed.



Plus Development's Hardcards II XL for IBM PCs and compatibles come in two models: 50M and 105M, which yield 52M and 107M formatted capacities.

ISA & EISA Hardcards

I use a "slimline" PC as my everyday office workhorse. Its small footprint and low profile (about 4" high) provide little in the way of expansion room, though, and it's already expanded to the max. There's no room inside the system unit to mount another hard drive. So with my 85M Seagate hard drive filled to capacity and an urgent need for more hard-drive storage, I had a real problem on my hands. Fortunately, I still had an unoccupied full-length 16-bit expansion slot, which decided me to investigate the feasibility of installing a hardcard to get additional mass storage. I'm glad I did.

Plus Development offers two capacities for its Hardcard II XL models for PC/AT and compatible computers, one with 52M and the other with 105M capacity. Apart from storage capability, virtually no difference exists between the two drives. So unless I make a definite distinction, everything you read here applies to both Hardcard II XL models.

These hardcards require one full-length slot in a PC's expansion bus. On each is mounted an ultra-thin IDE hard drive and controller electronics. The drive nestles comfortably between the top and bottom rails of an aluminum chassis, and a black cover that bears the model/size designation of the drive and hides it from view.

Located next to the drive is a dedicated IDE controller card with Plus Development's proprietary PROMs and other circuitry. The latter includes a 64K-byte disk cache. Edge connectors that plug into a 16-bit expansion slot protrude below the lower chassis rail. Because the hardcards are less than 1" thick, they fit comfortably in any full-size slot without extending into an adjacent slot's area as some other make 1½-slot-type hardcards do.

Drawing only 7 watts, these Plus Development Hardcard II XLs are easy on a power supply. The IDE drives feature automatic head parking when power is shut off, of course.

Installation

Installation of the Hardcard II XL couldn't be simpler. If ever a device existed that "self-installs," this is probably it. To install the hardware, I powered down my PC, removed the cover of the system unit and blocking plate for the full-length 16-bit slot in which I was installing the hardcard, and plugged the Hardcard II XL into the slot. (A plastic snap-in rear-edge card guide is provided if required.) After securing the card's bracket in place with the screw that held the blocking plate in place, I replaced the cover of the system unit.

Installing the supplied driver software is even easier. First, I told the system that the Hardcard was present simply by adding DEVICE = ATDOSXL.SYS to my CONFIG.SYS file and copying an ATDOSXL file from the Plus floppy disk onto my boot drive. This done, I rebooted my system to prepare the drive using partitioning options.

If the DOS version being used is earlier than 4.01, maximum partition size is 32M. This means that a 52M drive must be split into 32M C: and 20M D: partitions. In turn, a 105M drive must be split into three 32M partitions with logical identifiers of C:, D: and E: and one smaller logical drive F:. You can create any partition sizes you desire, as long as you stay within the 32M ceiling.

Users of MS-DOS 4.01 or 5.0 (and DR DOS 5.0 or 6.0) don't have a 32M restriction and can keep the entire capacity of either drive as a single, large partition if

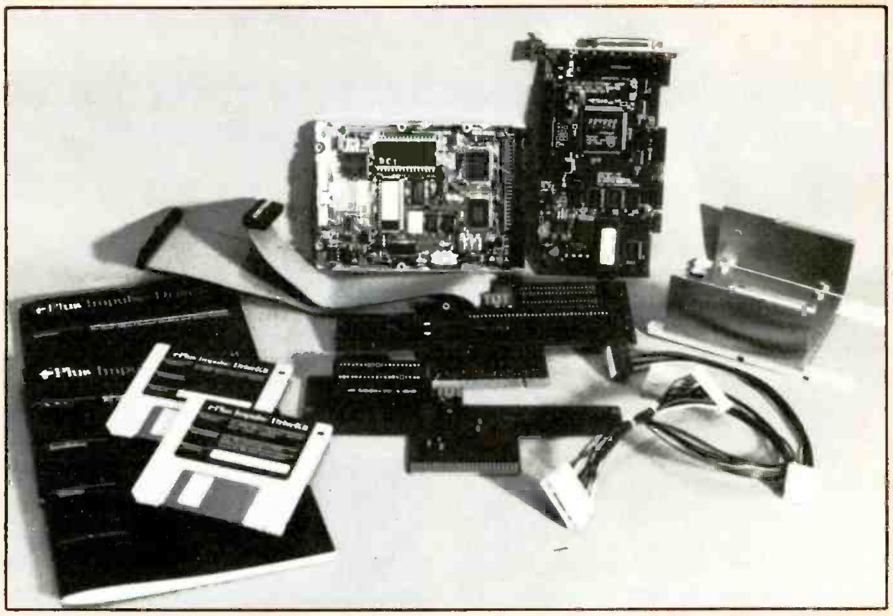
Hard Drives for the MCA Bus

Users of IBM Microchannel PCs don't have anywhere near the selection or freedom of choice for expanding/upgrading their machines that are available to ISA and EISA PCs, particularly if you want to add a second drive to your machine. Plus Development comes to the rescue once again with a line of Impulse drive kits in capacities ranging from 52M to a whopping 425M capacities.

Impulse hard drives can be purchased with or without a separate SCSI interface card. The kit we used for this review had a capacity of 105M and came with an adapter. IBM PS/2 computers that are compatible with Plus Development's Impulse drives are Models 50, 50Z, 55SX, 60, 65, 70, 80, 90 and 95.

Our kit consisted of the Impulse drive, mounting bracket with "sled" attached, drive bus adapters for Models 50, 50Z and 70, dc power connector for Model 55SX, mounting screws, hardware installation manual, customer enrollment card and PC decal. It also contained the Plus Impulse MC-200S SCSI host adapter, SCSI interface cable, installation software for DOS, OS/2, NetWare, UNIX and Xenix system, and a software installation manual.

Unlike the Plus Hardcard II XLs discussed above for PCs and compatibles, installation of the Impulse drive kit took quite a bit of doing in my IBM PS/2 Model 50Z. Although the installation procedures differ slightly for the different PS/2 Microchannel models, steps for installing the drive kit are typical.



Plus Development's Impulse hard-drive kit for Microchannel computers contain everything needed to upgrade a PS/2: Impulse drive, SCSI adapter card, mounting brackets, drive bus adapters, power connectors, other hardware and software.

Installation

I began installation of the Impulse kit by powering down my computer, disconnecting all cables from the system unit and removing the cover. Before plugging it into the expansion bus, I configured the MC-200S SCSI adapter by setting terminating resistors for the SCSI bus to reflect actual device connections. Two socketed resistor networks on the MC-200S terminate the internal and/or external bus connectors.

If both internal and external SCSI devices are connected to the host adapter, the two resistor networks must be removed. If internal-only or external-only devices are connected, you leave the terminating resistor networks in place. The users manual provides tables and diagrams for positioning terminating resistors on multiple devices connected to the SCSI interface, as well.

Three two-pin jumpers on the Impulse hard disk govern assignment of the SCSI device ID number. Here again, the manual provides detailed information for setting the jumpers for various ID configurations. The default setting of SCSI ID is 5. If the drive is the first SCSI device connected to the adapter, both jumpers must be removed to set the ID to 0. If other SCSI devices are in use on the system, an appropriate ID number ranging from 1 through 7 must be selected to avoid conflicts with other devices along the SCSI daisychain. (SCSI ID 6 is used by the MC-200S adapter itself and can't be changed.)

Additional jumpers on the drive may or may not require setting, depending on system configuration and operating preferences. Jumper WS controls the Wait/Spin option that determines whether the Impulse drive's motor will or won't spin on power-up until the host sends a START/STOP UNIT command across the SCSI bus. This permits power-up sequencing so that the system's power supply isn't overloaded by many devices demanding peak start-up current at the same time.

Jumper EP controls parity checking across

they wish. Of course, you can create multiple partitions in selected sizes under these later versions of DOS. Whichever way you go, you must execute the DOS SHARE command on boot-up (preferably from the AUTOEXEC.BAT file) to prepare your system for using large media.

I prepared the hardcard with a supplied XLINSTAL program that, when run, automatically checks for the presence of each Hardcard II XL in the system and displays an on-screen description of each for verification purposes. Any discrepancies between displayed parameter list and actual physical configuration can be amended on-screen.

When the list matches the hardware, the installation program continues. At this point, I was given a choice between automated and manual installation. In most systems, automated installation is preferred. The entire process took less than 5 minutes. I was also given the opportunity to go with default partitioning (a single large partition for the entire drive) or to enter any partitions I desired. The program proceeded automatically to prep the drive according to these specs.

Hardcard II XLs are configured with factory defaults for hardware and software that works the vast majority of DOS, OS/2, XENIX and UNIX computer configurations without modification. In some instances, you might have to change some defaults—like when you're installing a second hardcard in the same system. Default settings are preset to BIOS address C8000h, I/O address 170h and IRQ11 interrupt level. You can alter these settings to resolve conflicts or for special circumstances, such as making the Hardcard II XL the boot drive in an OS/2 system.

the SCSI bus. When installed, this jumper enables parity generation/checking of data. Installed is the factory default setting for this jumper.

Impulse hard drives also have three socketed resistor networks that are used to terminate the SCSI bus. Optional configurations detailed in the manual apply if more than one SCSI device is on the same daisychain. Factory default is all three resistor networks in place for a single SCSI-drive system.

When all resistor networks and jumpers have been set for the appropriate configurations, actual installation can begin.

The Impulse hard disk attaches to the mounting bracket with a mounting sled held in place with four supplied screws. Next, I removed the 3½" diskette drive(s) from my computer. I then removed the retaining clip from the IBM Drive Bus Adapter, which I also removed from my computer. Next, I installed the supplied Plus drive bus adapter for the Model 50/50Z in place of the IBM adapter. On single-floppy systems, you can reinstall the A: drive in its bay at this point. If a second drive was also installed in the B: bay, you can't re-install it because the new drive must occupy this bay.

With installed mounting bracket and sled, I slid the Impulse drive about half-way into the drive bay and connected the supplied dc power cable between the connector on the rear of the hard drive and edge connector on the drive bus adapter. Then I slid the drive all the way into the bay until the locking tab on the mounting bracket clicked into place.

Attachment of the SCSI interface cable to the SCSI card and the hard drive came next, followed by installation of the interface card in an open Microchannel expansion slot in my computer. Replacing the system unit cover and reconnecting all cables completed the hardware phase of the installation.

The manual provides numerous diagrams of kit components in relation to the host system. The instructions are clear and easy to understand. Though this installation is admittedly more involved than simply popping a hardcard into a slot, as with the II XL models above, it isn't really difficult and shouldn't deter any Microchannel owner from doing the upgrade himself. A little patience and Plus Development's excellent manual more than compensate for any lack of experience or technical prowess. If you follow instructions and refer to the diagrams, you won't have any problems getting up and running.

With Microchannel machines, you have to tell the system when hardware is installed in an expansion slot for it to be recognized. I accomplished this with the IBM Reference Diskette supplied with my computer. Software installation is a multiple-step process, which begins with placing the IBM Reference Diskette into the floppy drive and booting the system by turning it on.

I then selected "Copy an option diskette" from the displayed menu. When prompted to insert the option diskette, I had to replace the Reference Diskette with the supplied Plus Impulse installation software disk, after which the system copied a @6127.ADF file from the diskette. When prompted, I again replaced the disk in the drive with the IBM Reference diskette and pressed ENTER to update it with the @6127.ADF file.

When the menu reappeared, selecting "Run automatic configuration" completed the update of the computer system configuration.

Running the Impulse INSTALL program came next. This program allows you to partition and format SCSI hard disks connected the the MC-2205 adapter. If the Impulse drive is to be the boot drive, the DOS system files and the device driver required for it are transferred to the root directory of the drive and a new CONFIG.SYS file is created with the device driver statement and transferred to the root directory. If the Impulse drive is to be the second hard drive in your system, its device driver is transferred to the root directory of the boot drive and the device driver statement is added to the existing CONFIG.SYS file. The INSTALL program is menu-driven, and you're prompted for appropriate responses.

INSTALL first finds each hard disk in a system and lists those it finds for verification of existing system conditions. Any disparities between physical devices and the list displayed can be rectified through editing, although a list that doesn't agree with the installed drives indicates that something isn't properly connected. If this is the case, all connections must be rechecked and any other remedial actions must be taken to clear up the problem.

When the list checks out with the installed drives, you can proceed with the INSTALL program. I was asked to select automatic or manual installation. Remember that a 32M limit exists for hard-disk partitions with DOS versions earlier than 4.0, which means you must manually partition the drive if you use DOS 3.3 or earlier. With DOS 4.1 or 5.0, you can select automatic installation to set up the disk for a single large DOS partition. Of course, you can also choose manual installation to create smaller partitions if you wish.

Automatic installation takes only a few minutes and is entirely self-running. Upon completion, a screen prompt advises you to hit RETURN to exit INSTALL, at which point, your new Impulse drive is ready for use.

User Comment

This drive is extraordinarily fast, thanks to an on-drive disk caching chip that reduces average access time down to the 9-ms range from the 17-ms time for the same drive without caching enabled. In operation, this is a very quiet drive. Another plus is that it coexists in perfect harmony with the built-in 60M drive on my Model 50Z. If you want more mass storage on your PS/2 Microchannel machine, Plus Impulse drives are certainly a good choice.

Plus Impulse Drive Kit, $699 (52M); $799 (80M); $999 (105M); $1,749 (210M); $2,999 (425M)
Plus Development Corp.
1778 McCarthy Blvd.
Milpitas, CA 95035-7421
Tel.: 408-944-0410

CIRCLE NO. 147 ON FREE INFORMATION CARD

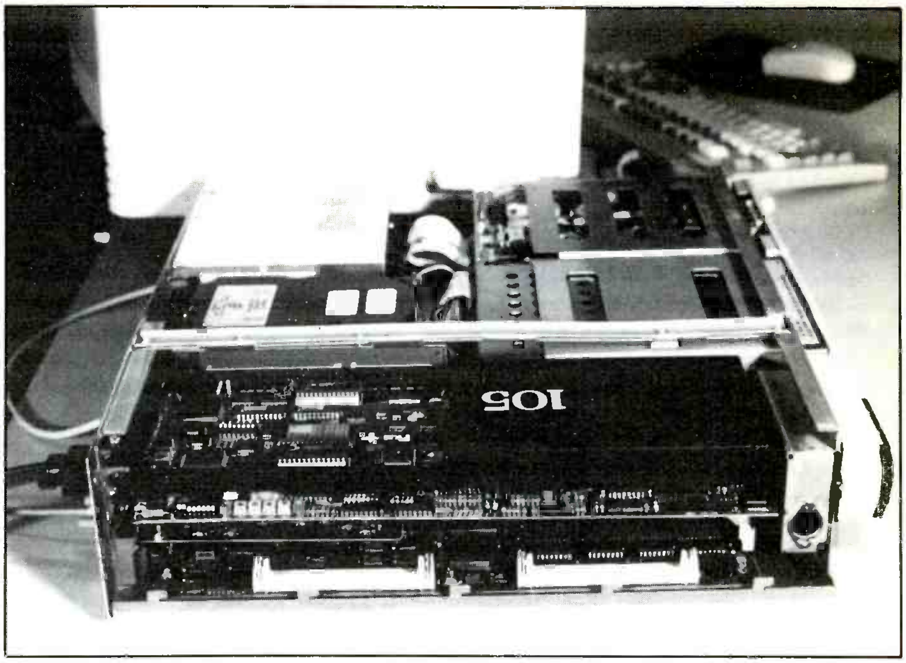
All settings are governed by the positions of an eight-position DIP switch located on the hardcard's controller board. Positions 1 through 4 control the BIOS and I/O addresses. Positions 5, 6 and 7 command the IRQ settings. Position 8 is reserved.

Manual installation permits adjusting the partitions (both primary and extended) and creating logical drives and assigning them DOS or non-DOS status. Menu screens with available options are presented at each step of the way, and extensive on-line help can be accessed to guide you.

The Hardcard II XL's installation guide is excellent from cover to cover. It gives detailed instructions for installing the hardcard in DOS, OS/2, XENIX and UNIX environments and for changing default configurations. Diagrams that show switch locations and positions augment the text. The manual also includes a thorough troubleshooting section.

These hardcards install with no visibility to the outside world. So no drive activity indicator is visible on the outside of the computer to signal when data is being read from or written to them. However, a supplied LIGHT.EXE utility provides and controls an on-screen prompt that informs you when the hardcard is being accessed. When you run this utility, a small plus sign displays in the upper-right corner of the screen to signify disk activity. You can turn off this prompt with this utility. (Other programs for OS/2, UNIX and XENIX installations are included on the supplied program disk.)

One of the nicest features of these Plus Development hardcards is that they peacefully coexist with virtually any other existing type of hard drive in a computer without causing conflicts. This is good news if



Hardcards are often the only feasible way to expand storage capacity, as is the case in the author's fully-loaded slimline 386SX PC.

you're using an older MFM or RLL drive as your primary drive and the hardcard as a secondary drive.

When I tried the Hardcard II XLs on several systems that had older drives in them, as well as in newer PCs with IDE boot drives, I experienced no conflicts or other problems whatsoever. Plus Development's hardware and software engineers should take a bow for the excellent design of these hardcards.

Conclusion

Since Hardcard II XLs are so compatible and easy to install, they make excellent "transportable" drives for moving lots of information from one PC to another. This is especially welcome if you take work home from the office, as I do. These drives are also great for backing up the data from your PC's primary hard drive. If your host PC goes down, you just remove the hardcard, plug it into another PC, install the device driver, and you're up and running again in almost record time.

While all of the above features are attractive, they don't come cheap. Hardcards II XLs sell for about $100 more than standard IDE hard drives of the same capacities. (A 50M XL has a mail-order street price of about $298, while the 105M goes for about $438.) But you won't find standard hard drives in these capacities that can give the fast average disk access time of these Hardcard II XLs. At 9 ms average, they're among the fastest drives you can get at any price. The fact that they can almost effortlessly be moved to my next PC when I upgrade was a bonus that justified the additional cost for me. If you're looking to expand your PC's hard-drive capacity, the Plus Hardcard II XLs may be just what you're looking for. They're masterpieces.

Plus Development, which pioneered hard drives on a card, has brought to market a truly first-rate drive that's rugged and features superb workmanship. Moreover, its 50M Hardcard costs half as much as its 40M one did some five years ago, while providing 9-ms access speed compared to its earlier model's 40 ms. ■

Plus II XL, $389 (50M); $579 (105M)
Plus Development Corp.
1778 McCarthy Blvd.
Milpitas, CA 95035-7421
Tel.: 408-944-0410

CIRCLE NO. 147 ON FREE INFORMATION CARD

All photos by Liz Benford

Tom Benford

Ted Needleman

Making Your PC More Productive

This time around, I'm going to give you a look at a number of different approaches for making your PC into a more-productive tool. I'll begin with a retailer who has some interesting and unique products to help you get a lot more out of your laser printer.

Laser-Printer Products

A couple of the applications I run most frequently are word processing and graphics. Because of this, I've sometimes become a bit preoccupied with the output I'm able to generate. One of the most useful things I've run across in getting the most out of my laser printer isn't a gizmo or gadget or even a piece of software. Rather, it's the PaperDirect catalog.

PaperDirect is a difficult company to categorize. It has just about any kind of paper you might think of running through your laser printer (and quite a few that you'd never have known existed), as well as a few really neat software packages, and even folders and easels in which to mount your finished work.

I'll discuss several of PaperDirect's products I've used, but the company's 80-page catalog makes absolutely fascinating reading. In addition to very-high-quality laser papers from a variety of manufacturers, there are dozens of specialty papers in all sorts of colors, textures and even gradients that go from light to dark. Do you need to run off some business or Rolodex cards? A line of "Perfs & Scores" makes this a snap to do.

One of the papers that really impressed me is PaperDirect's extensive line of Brochure/Mailing forms. You run the preprinted papers through your laser printer to add text and then fold. In seconds, you have a brochure the equal of anything I've seen done commercially. Again, as with the other types of unique papers, there are dozens of styles and patterns from which to choose.

PaperDirect also sells a product called Desktop ColorFoil that lets you add spot color to your laser output. I've covered similar products here in the past, like Letraset's Color Tag and KroyColor. These and ColorFoil, are carrier sheets that contain a color material that you melt into the laser toner on a printed page by covering the printed area and heating it. Color Tag uses a special heat applicator and costs a bit under $100. Until now, it was the least-expensive method I'd run across to add spot color to laser output.

ColorFoil is pretty much the same material. It comes in 10 different metallic colors and a half-dozen different patterns. You can buy it in a 2″ wide × 25-foot-long roll or 8″ × 11″ sheets, accompanied by several dozen small adhesive dots.

To use the system, you snip off a piece of ColorFoil sized to fit the area you want to add color to a laser-printed page and secure it in place with several of the adhesive dots. Next, you run the whole thing through your laser printer using manual feed while printing a blank page. The heat and pressure rollers in the printer melt the toner so that the ColorFoil binds to it. When the page comes out of your printer, you peel off the ColorFoil and you have colored output.

At $19.95 for a roll or 16-sheet pack of foil, it's a really inexpensive way to add color to your output. ColorFoil also works on some photocopier output, if the machine uses a dry toner system.

I've used ColorFoil (and its related "cousins," Color Tag and KroyColor) with great results. Most laser printers have a straight-through path for heavier materials that minimizes the number of turns a piece of paper must make as it passes through the printer. If you don't have this feature (or don't use it), your beautifully colored laser output comes out of the printer with a definite curve to it. If you must make several passes to add several colors, the output becomes so curved that it's almost impossible to straighten out.

When using this spot-color technique, don't forget to print a *blank* page when you're adding the color. It's really frustrating to get a brochure or certificate just the way you want it, only to ruin it at the last minute by overprinting something you don't want on it while adding spot color. To guard against this, I have a blank document I created in my word processor, appropriately named BLANK.DOC, that I'm careful to bring up before starting the spot-color process.

Two other PaperDirect products I use frequently are Laser Note Cards and "One-Up" single mailing labels. Laser Note Cards are 5½″ × 8½″ heavy paper cards that come in 10 different colors and cost $19.95 for 50 sheets and 50 matching envelopes. You run them through your laser printer and then fold them to create your own high-class note cards, invitations, announcements and the like. You can, for example, use them for personal cards, printing your name and whatever else you want on the front and then color the print with ColorFoil. This generates stationery as nice as I've ever seen at a printers.

One-Up single mailing labels mount a 4¼″ × 3⅞″ label on a carrier the size of a No. 10 business envelope that goes through a laser printer's manual-feed slot. You can use almost any envelope-printing program to generate a terrific looking mailing and packaging label. One-Up labels aren't cheap at $26.95 for a box of 250, but they're certainly handy.

PaperDirect also offers a number of its own software packages and packages from major vendors, like Microsoft *Publisher*. If you get into using some of the specialized papers, you'll really want to get a copy of the PaperTemplates disks. The one I have is for Aldus *PageMaker* on the PC (it's also available for the Mac). By the time you read this, Paper Direct will also have versions for *WordPerfect, Quark Express*, Microsoft *Word* and Microsoft *Publisher*.

For $39.95, you get more than 100 templates for most of PaperDirect's specialty products. To use them, just open the template for the particular product you use, and replace the placeholder text with your own. PaperTemplates even comes with two free BitStream fonts and 20 free pieces of T/Maker clipart.

If you use a laser printer, PaperDirect's catalog is a must-have item. It's entertaining, informative and, best of all, free! Call 1-800-A-PAPERS to request a copy. If your first order exceeds $30, PaperDirect also sends you a terrific PaperKit that contains samples of every paper the company offers. Alternatively, you can buy the kit for $19.95). I've been really pleased with PaperDirect. The company ships quickly, and the quality of its products is outstanding. Give PaperDirect a try, I think you'll like it as much as I do.

There Must Be a Cache

A while back, I discussed the reasons why I use the *Power Meter* benchmarks to measure the performance of equipment I review here. I made the point that the term "performance" isn't always a straightforward measurement. As with the old "weak link in a chain" theory, your PC can be only as fast as its slowest element. In many applications, especially when you have a fast CPU and memory, that slow component is your hard-disk drive.

As the technology has matured, hard disks have gotten faster, of course. Where early PCs sported drives with almost 70-ms access times, today's IDE and ESDI drives are frequently in the 15-ms range. Even so, with applications that are very disk-intensive, like *Windows*, even these fast hard drives can seem awfully slow.

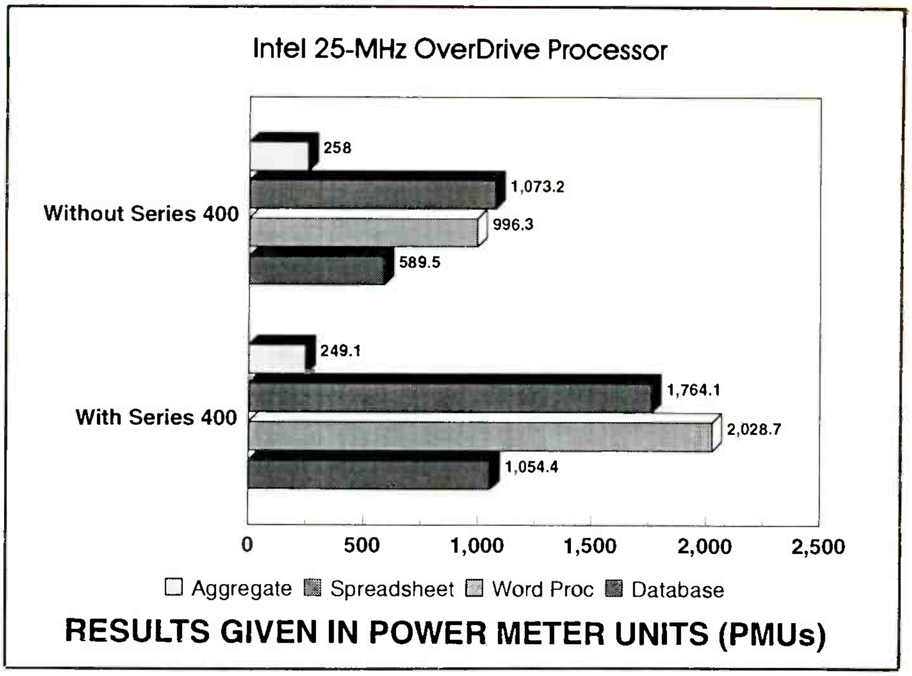
Caching, the process of moving information you frequently access from slower memory into faster memory, is an effective way to speed up things. If you've ever created a RAM disk to emulate a hard disk drive in RAM memory, you know that disk operations that formerly took several seconds or more can be executed almost instantaneously. The i486 CPU makes use of caching technology, using its on-board 8K RAM cache to buffer information between the fast CPU and much slower motherboard bus.

By and large, software products let you cache a hard drive in an area of main RAM memory work pretty well. However, the best solution I've found so far when it comes to the higher-performance PCs like 33MHz 386s and the entire 486 line is to use a caching hard-disk controller. The hardware approach, which puts between 512K and 4M or more of moderately fast 80-ns RAM right on the disk controller card, just seems to work better in applications on which I've tried it. It also seems to be a lot more resistant to line glitches that sometimes find their way through my surge protector.

The biggest problem I've had in the past with caching disk controllers is that they've always required me to reformat my hard disk. This turns what's essentially a 10-minute card-swap job into a several-hour ordeal with backing up and restoring over 100M of files. Because of this, the only time I'd install a caching disk controller was when I was setting up a new machine.

That was then; this is now. With Alpha Research's Series 400, the first "plug-and-play" cache disk controller reduces the operation to a simple card-swap. Available in IDE, ESDI, SCSI and local-bus versions, these boards can contain up to 16M of cache RAM (even more, depending on model). It just plugs into an available bus slot, where it replaces an existing hard/floppy controller card.

Installation of the Series 400 card is simple. Using the included cables to connect the drives to the card completes the physical part of installation. Depending on the type of PC you're using, you may have to disable an internal IDE controller, if one resides on the motherboard. On the Epson Equity 486SX/25 Plus with which I tested the card, this wasn't necessary.

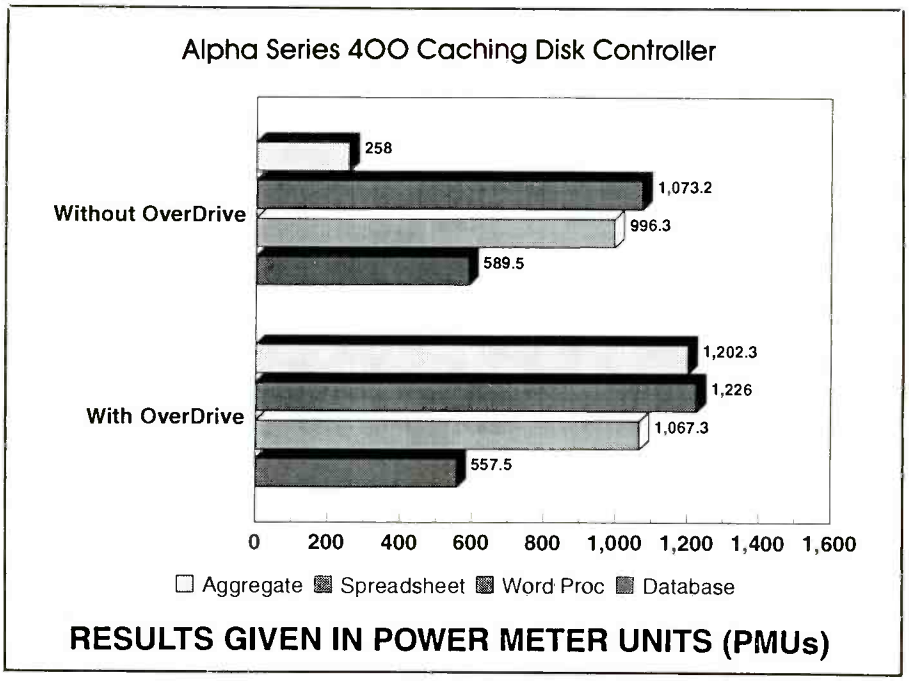


The Series 400 features automatic installation. It determines the type of hard drive connected to it and configures itself. With the IDE model I tested, I had to run the setup procedure to set the computer's CMOS drive table to Type 01 after I physically installed the card.

The whole installation took less than 10 minutes to complete, with a good part of that used to remove and replace the lid on the computer. I didn't have to reformat the hard disk to effect installation. Of course, you should always back-up your drive whenever you plan on doing something that could potentially trash your disk.

Is a Series 400 the answer to your problems? It is if your problem is a disk I/O bottleneck. In Chart A that accompanies this column, the most telling before and after are the Database and Word Processing benchmarks, which heavily weight disk performance. The increase achieved by installing the Series 400 shows just about double the performance. Even the boost in the Spreadsheet score is very impressive.

While the Series 440 card retails at about



$400 without memory, it should have a street price closer to $250. Add another $40 or so for a 1M SIMM, and for less than $300 you've substantially beefed up your PC's hard-disk performance, bringing average access time down into the sub-millisecond range.

First Gear, Second Gear

One of the most obvious ways to get a performance boost out of a PC is to move to a faster and/or more powerful processor. This is the theory behind CPU upgrade products like Intel's SnapIn, SOTA's Express/386, and AOX's Stax SX 25, all of which let you substitute a 25-MHz Intel 386SX CPU for a 286 in AT-class PCs. Having just completed taking a close look at these three products, I can vouch that this approach does work, though the boost is most obvious in computationally-intensive applications (remember the "weakest link" mentioned above.)

Having been in the CPU-upgrade business for a while with its InBoard product line of 8088 and 286 upgrades, and most recently with its SnapIn 386, Intel is well aware of the potential offered by CPU upgrades. This potential lies in two areas: revenues from providing upgrades and customer satisfaction from being provided with a path that staves off obsolescence.

Recognizing this potential, Intel anticipated that 486SX users might want to get a leg up without having to replace their motherboards. So the company designed a way for motherboard manufacturers to build in a CPU-upgrade path for users. You've seen the ads for more than a year now that show a vacant socket next to a 486SX with tag lines like "Room for the future." Until now, that socket has been pretty much dedicated to Intel's i487SX math coprocessor, which is really a full 486DX chip with a different pinout and that disables the 486SX CPU when installed.

The empty socket was really designed for Intel's recently-announced OverDrive chip, which is a consumer SX version of Intel's DX2 dual-clock-speed CPU. While it retains the SX chip's external 32-bit interface and 64-bit internal bus structure, it adds a number of things that really pump up performance.

Primary among the boosts provided by the OverDrive Processor is full numeric-coprocessor floating-point support. On the i486SX CPU, these functions have been disabled, requiring you to purchase an optional (and expensive) i487SX chip if you wanted the benefits.

Another enhancement is that the OverDrive Processor continues the 486-series' 8K internal RAM cache and adds dual clock speed features. The processor operates at one speed when communicating with the bus (on the SX, this is 16, 20 or 25 MHz), and twice the bus clock speed for all internal operations. Plugging in the OverDrive chip turns your 25-MHz 486SX system into 50-MHz PC. At least it does so some of the time. In reality, you don't get a full 100% boost in CPU performance because the CPU still has to slow down to communicate with the bus (again, the weakest link). Still, on computationally-intensive applications, the increase in performance is impressive.

I added an OverDrive Processor to an Epson Equity 486SX/25 Plus, which took about 5 minutes to do, most of which was opening and closing the system unit's case. You can see the before-and-after results in Chart B. The Aggregate benchmark, which weights all performance variables pretty much equally, shows almost a five-times increase in performance. The Spreadsheet test, a much more realistic indicator of the benefits the OverDrive Processor might have on applications, shows a much more modest jump in performance of a bit more than 20%. The other two benchmarks, which heavily weight disk performance, are affected to lesser degrees.

Lots of people bought and are continuing to buy 486SX systems because they like the promise vendors made that they can be upgraded when they start to push against the limits of performance. The OverDrive Processor delivers on this promise, and the cost isn't unreasonable. A 16/20-MHz chip retails for $549, the faster 25-MHz chip for $699. Street prices should be considerably less once the initial demand for the chip is met.

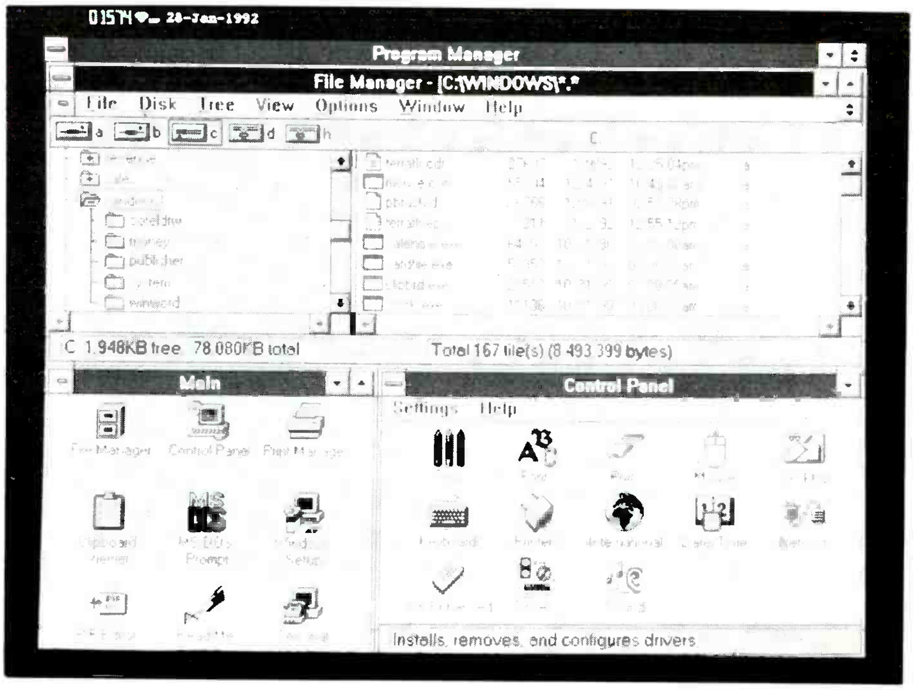
A DX version of the OverDrive Processor should show up sometime later this year. This will actually be the DX2 CPU already available to vendors. Pricing and availability on this clock doubler for the DX line have yet to be announced. ■

Products Mentioned

Series 400 Cache Disk Controller, $400 (0K RAM)
Alpha Research Corp.
8200 Mopac Expressway N.
Austin, TX 78759
Tel.: 512-345-6496
CIRCLE NO. 159 ON FREE INFORMATION CARD

OverDrive Processor, $549 (16/20 MHz); $699, (25 MHz)
Intel Corp.
3065 Bowers Ave.
P.O. Box 58065
Santa Clara, CA 95052-8121
tel.: 800-538-3373
CIRCLE NO. 160 ON FREE INFORMATION CARD

Windows 3.1



There are just a few telltale signs that distinguish *Windows* 3.1 from its predecessor. Some should remind you of OS/2 features. Some are significant. Some are largely cosmetic. For instance, the old wallpapers are gone, replaced by new fashions. This will please users who weren't amused by the old ones and disappoint those who liked them. *Windows* 3.1 gives you more control over colors and fonts.

Some of the old icons have been replaced by new designs, and you can select Wrap Title in the Control Panel Desktop to wrap very long icon titles to a second line. (Unfortunately, the title still disappears when you move icons, making it very difficult to align them by hand.)

There are also a couple of new icons. In the Control Panel, there's a new control for multimedia drivers. In the Accessories Group, you'll find Character Map, Media Player and Sound Recorder icons. The Character Map is a simple utility that makes it easy to clip and paste special characters from a table. Media Player and Sound Recorder provide controls for MCI (Media Control Interface) devices and sound boards, respectively. You'll also notice a new group in the Program Manager called StartUp. Instead of editing your WIN.INI file to run applications on startup, you can now simply drag and drop them into this group.

A built-in screen saver now offers four animations, control over their speed and frequency and a password option. The Marquee version even lets you compose a banner, with your choice of fonts and colors, that traverses the screen. Set and test these under the Desktop until you find your favorite.

On a laptop computer, you might also discover *Windows*' ability to create Mouse Trails that make it easier to follow the cursor on LCD panels. (*Windows* now supports the Advanced Power Management specification for laptops as well.)

Windows 3.1 has more than a thousand changes. Most are beyond the superficial things you spot with a cursory examination. Several add substantially to the basic functionality of *Windows*.

Object Linking and Embedding (OLE), the second generation of dynamic data exchange (DDE), is new for this version of *Windows*. It provides a hot link between *Windows* applications by embedding an actual binary object created by one program into another. A chart or spreadsheet, for instance, could be embedded into a word-processing document. When you double click on the object, the application that created it is launched, allowing you to make modifications.

Support for the new *TrueType* outline fonts makes a great addition to *Windows* 3.1. Large, inexpensive libraries of high-quality *TrueType* fonts for *Windows* 3.1 could make this feature alone worth the cost of an upgrade for many users. (See Key Fonts for Windows 3.1 in last month's column.)

TrueType fonts are generated on demand, and don't occupy the large amounts of disk storage space required by bit-mapped fonts. They can be output to any supported device without special cartridges or type managers, and they can be embedded in documents for display or printing on equipment whether or not the device has them installed. So, the new *Windows* can instantly provide WYSIWYG text, for both printers and monitors, in a wide variety of point sizes.

Because special interpreters aren't required, output devices can include dot-matrix as well as ordinary laser and PostScript printers. Since *TrueType* is supported by Apple's Machintosh computers, the fonts are WYSIWYG across platforms as well. And they don't conflict with or preclude other font technologies.

The MCI architecture and support for Musical Instrument Digital Interface (MIDI) sound boards are both part of Microsoft's multimedia strategy. MCI supports control of videodiscs, video tape, CD-ROM and other image/sound storage devices. MIDI provides support for synthesized audio, and *Windows* 3.1 also supports waveform or PCM audio in its programming interface. Moreover, sound objects can be embedded in documents by any application that supports OLE.

DOS has better support in *Windows* 3.1, which supports applications in VGA graphics mode in a window or in background. Mouse support is available in a window as well as in full-screen mode. And 32-bit disk access lets *Windows* 3.1 run more concurrent DOS applications than could *Windows* 3.0.

Other significant improvements include the Setup program, which now requires fewer steps. It also recognizes and sets up far more applications and TSRs and provides more ready-made program information files (PIFs) and more icons for them in the PROGMAN.EXE file, as before. Lamentably, a slip during installation still requires you to restart from the beginning and go through every intermediate step—even if you just want Setup to change the CONFIG.SYS and AUTOEXEC.BAT files at the very end.

One more important Setup change is the improved support for PC coordinators. Network installation has been enhanced, and it's possible to create custom Batch Install scripts that automate installation.

The File Manager has been so improved that I may even use it occasionally. Take printing. You can now print files by simply dragging and dropping them onto the Print Manager. And Print Manager can now automatically resume stalled print jobs. Files can likewise be opened by dropping them on a running application. Quick Format speeds up diskette formatting. Network re-connections are automatic and are based on the previous session.

Performance has also been improved in several areas, including the File and Program Managers and display drivers. Printing is faster, and control is returned to applications sooner. And the 32-bit disk access in *Windows* 3.1 speeds up virtual-memory paging by bypassing the operating system's basic input/output system (BIOS).

Windows can still go south unexpectedly. Nevertheless, this version of *Windows* has been widely reported to be more stable than version 3.0. In some ways it is. For example, it's now possible to reboot an application when it hangs, without having to boot your computer. After you hit Ctrl + Alt + Del, a pop-up message suggests the alternatives of closing only the offending application or restarting the system.

Other strategies *Windows* 3.1 has to help reduce the number of unrecoverable application errors (UAEs) include parameter validation, increased information and testing tools.

Windows 3.1 parameter validation checks handles, pointers and other resources. This prevents applications from requesting inappropriate resources that sometimes caused *Windows* 3.0 to write over system data in memory or send data to invalid addresses. In addition, *Windows* tests for incompatible TSRs and hardware configurations during setup and incompatible applications whenever they're run. This version of the environment also seeks to avoid the low-memory problems that plague *Windows* applications in version 3.0 through better allocation of system resources.

Instead of simply closing an application that experiences a potentially fatal error, *Windows* 3.1 identifies the actual source of the error and offers you the option of immediately closing the application or returning to it to save files before closing it. *Windows* then attempts to keep other applications running.

Furthermore, Microsoft has developed new tools, like Dr. Watson, to help developers produce more reliable code. Testing information is available for 268 printer drivers, 1,200 certified PCs and 18 video display drivers.

If you plan to run multimedia applications, to embed objects in your documents or you can benefit from *TrueType* fonts, you need to upgrade to *Windows* 3.1. Likewise, if you're running out of resources, find yourself bogged down in *Windows* 3.0 performance or use *Windows* on a portable computer, you should consider upgrading to *Windows* 3.1. Otherwise, you might not have a compelling reason to make this upgrade.

Neatness Counts

I must confess that I'm essentially sloppy, and I frequently have diskettes with no labels on them all over my desk. On at least one occasion, I've accidentally erased an un-labeled (but formatted and data-filled) Mac diskette, thinking it was an unformatted PC diskette. When I got up off the canvas, I realized that there must be a better way to live my life without having to give up being a slob. Now this kind of accidental knockout is a thing of the past. I solve the problem without firing up my Mac, merely to check. Instead, a program on my PC called *Mac-In-DOS* lets me quickly check diskettes for a Mac format.

If you work with a Macintosh in a mixed environment, you've probably used the Apple File Exchange program that allows the Mac to transfer files from diskettes formatted for the PC. You may also be familiar with more comprehensive third-party software like *AccessPC* and *AccessAT* from Insignia Solutions or Dayna Communications *DOS Mounter* that add cross-platform formatting and other features to the basic transfer of data. *Mac-In-DOS* is a software solution that does essentially the same thing, but on the PC.

Mac-In-DOS lets you directly read or write Mac files on a 1.44M diskette, and it formats diskettes for the Mac, too. Incidentally, *Mac-In-DOS* is actually a PC version of another Pacific Micro product, *Common-Link*, a Unix utility that exchanges files among Macs, PCs, and Unix systems like IBM RS/6000, HP, PC (Interactive, SCO) and PS/2 (AIX).

An inexpensive hardware solution, the Deluxe Option Board from Central Point Software, also reads, writes and formats Mac diskettes on the PC. It works quite well. Available for a number of years, the Deluxe Option Board has the ability to duplicate copy-protected files in several formats. It's slick and works with DOS-like commands. For example, you create folders with a special version of MKDIR (MD).

However, the file-conversion capabilities of *Mac-In-DOS* are slicker still because the program preserves both the data and resource forks, as well as finder data, in separate files when transferring from the Mac to the PC. And it can integrate them back to the proper format when returning a file to the Mac. This is a sophisticated feature that other products in this category simply don't offer.

While *Mac-In-DOS* isn't currently a *Windows* application, it will run in a DOS window if you observe one restriction. While *Mac-In-DOS* is writing to a floppy, make sure no other application, including those in the background, writes to it. The true *Windows* version is scheduled to ship with *Windows* 3.1. Until then, take solace in the program's uncomplicated interface. *Mac-In-DOS* is worth having if you work principally with a PC, transfer files to the Mac, share one of my bad habits and you want to preserve the life of your Mac and its monitors. ■

Products Mentioned

Mac-In-DOS, $99
Pacific Micro
201 San Antonio Cr., Ste. C250
Mountain View, CA 94040
Tel.: 415-948-6296

CIRCLE NO. 153 ON FREE INFORMATION CARD

Deluxe Option Board, $159
Central Point Software
9700 S.W. Capitol Hwy., Ste. 100
Portland, OR 97219
Tel.: 503-690-8088 or 800-445-4162

CIRCLE NO. 154 ON FREE INFORMATION CARD

Joseph Desposito

EIA/TIA-562 Transceiver, DRAM Controller, Improved Four-bit Microcomputer, Temperature Sensor, PC/AT Power-Supply Board

New standards are being developed all the time. One of the newest, the EIA/TIA-562 standard, was developed in response to a demand for physically smaller, lower-power interfaces that are more consistent with today's technology. This time around, I lead off with a device that supports this standard.

3-Volt RS-232 Transceiver

Maxim Integrated Products' (120 San Gabriel Dr., Sunnyvale, CA 94086) MAX561 implements the new EIA/TIA-562 standard that guarantees operation with outputs as low as + 3.7 volts. It's the ideal solution for any system that needs to save power while transmitting RS-232-type data, such as laptop computers and hand-held equipment.

Four drivers and five receivers are included in the MAX561, as shown in Fig. 1. The device is guaranteed to operate down to + 3.0 volts at a data rate of 20K bits/second. It consumes only 8 mA of quiescent current, compared to 15 mA for a typical four-driver/five-receiver, + 5-volt RS-232 device. Its low-power shutdown mode reduces supply current to 1 μA when the serial port isn't in use, saving even more power. Additionally, the MAX561 uses space-saving 1-μF external capacitors and is available in a space-saving 28-pin small outline package.

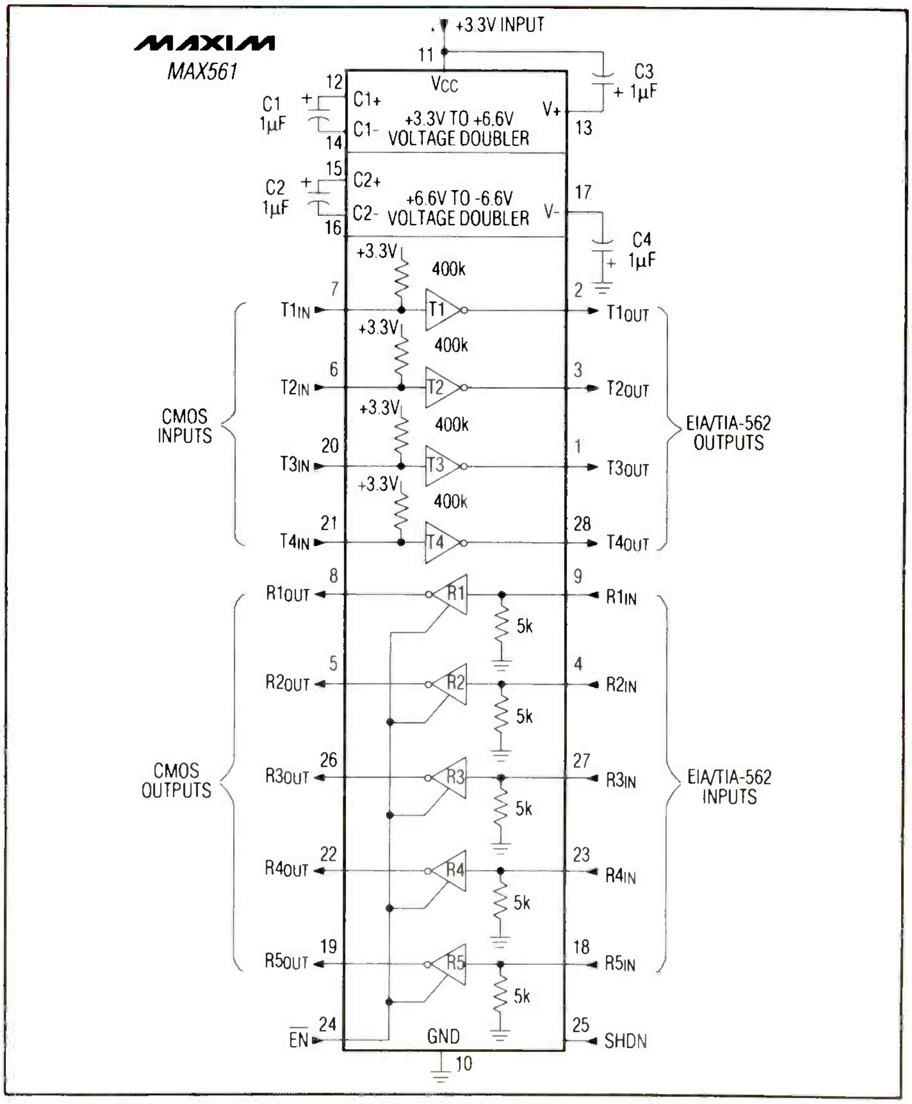


Fig. 1. Typical operating circuit configuration for Maxim's MAX561 EIA/TIA driver receiver chip.

The MAX561 meets all EIA/TIA-562 specifications. And, as stated in its forward, the new EIA/TIA-562 "... allows for electrical interoperation with equipment designed to conform to EIA/TIA-232D interfaces." Therefore, RS-232 transmitters will work with EIA/TIA-562 receivers, and RS-232 receivers will accept data from EIA/TIA-562 transmitters.

The MAX561 is offered over the commercial temperature range. It's priced at $4.19 each when purchased in 1,000-piece quantity.

Memory Enhancer

A new multi-chip module (MCM) DRAM accelerator from Cypress Semiconductor (3901 N. First St., San Jose, CA 95134) supports performance-oriented multiprocessing and uni-processing systems. The CYM7232 DRAM accelerator provides four high-level functions that make the system-to-DRAM interface run at maximum efficiency, accommodate cache-line burst transfers and support cache-to-cache transactions.

To simplify the design task, regardless of system configuration, a complete bus interface supports 50-MHz 32- or 64-bit multiplexed or separate address/data-bus systems. It incorporates transaction control, handshake and bus parity signals.

To permit processors to operate at full speed, without waiting for DRAM accesses, internal FIFOs can accept a 128-byte burst at full bus speed. To speed throughput while providing efficient correction of soft errors in the DRAM array, the 128-bit DRAM interface has four parallel 32-bit error-detection/correction paths, and a 156-bit pipeline data register and a 128-to-64 bit hardware multiplexer.

For seamless integration with sophisticated snooping multi-processor cache controllers, transaction transformation control supports inhibited reads and writes, reads for ownership and reflective reads for maintaining main memory coherency.

The DRAM Accelerator is totally generic. It works with processors as diverse as SPARC, i486, 88000, 680X0, i860, R4000 and their related caches.

By reducing the number of clock cycles required to fetch data from main memory during cache misses, the CYM7232 improves bus utilization. For example, it can make a memory built of 80-ns DRAMs as effective as one built of 50-ns DRAMs. This capability can be used to increase the number of processors sharing the bus or achieve economies in memory cost.

Efficient soft error correction is essential for large arrays of DRAMs, which may produce an error every several months. The CYM7232 includes error-correction/detection circuitry that corrects single-bit errors and detects two-bit errors.

In a typical system, the 7.8-square-inch CYM7232 DRAM accelerator provides better performance than a custom design while replacing 35 to 40 high-performance 22V10-type PLDs. These would occupy 30 to 35 square inches of board space but wouldn't include the write FIFO, parallel error-detection/correction paths or pipelining and multiplexing.

The CYM7232 consists of three high-performance integrated circuits, two for the data path and one for address and control. Functionally, these ICs incorporate an address path/control chip and two data path chips. The 20K gate (CMOS) address path/control chip includes four 100-MHz state machines; eight 50-MHz state machines; internal timing generation; and 33 programming registers. Each 20K gate (CMOS) data path chip includes four 8 × 32 data FIFOs, each with a 32-bit error checkbit generator; two 8 × 7 syndrome FIFOs; four parallel error-detection/correction circuits; and 128-to-64-bit data MUX/pipeline register.

The CYM7232 DRAM Accelerator is configured as a 400-pin pin-grid array. In quantities of 100, price is $327 each.

Improved Four-Bit Microcomputer

NEC Electronics (401 Ellis St., P.O. Box 7241, Mountain View, CA 94039) has expanded its 75X family of four-bit, single-chip microcomputers with four new devices that provide a number of powerful on-chip peripheral functions, including programmable controller/drivers for vacuum-fluorescent displays and eight-bit analog-to-digital (A/D) converters.

These new members of the 75X family provide functions needed to control almost any kind of semi-intelligent consumer product or light industrial process. Their best uses are in applications that require a bright vacuum-fluorescent display with many digits and segments. The most common applications are in the consumer-electronics and appliance industries.

The four new 75X devices are upgraded versions of less-powerful existing single-chip microcomputers, with expanded on-board ROM for program code and RAM for data. A number of on-board peripheral facilities have also been enhanced, including the display function of the vacuum-fluorescent display controller/driver, I/O ports, A/D converter and two-channel, eight-bit serial interface.

Vacuum-fluorescent displays work best and brightest when driven to about 35 volts, but few controllers have the ability to drive at this level. The new NEC devices have the ability to drive fluorescent displays directly.

Each device contains a CPU capable of one-four- and eight-bit data processing, 64 lines of I/O, 1K × 4 bits of RAM for data, and various amounts of ROM for program memory (from 16K × 8 bits to 32K × 8 bits). A separate group of pins is dedicated to controlling and driving a fluorescent indicator panel (FIP) consisting of 9 to 16 digits, each with 9 to 24 segments. Minimum instruction execution time is 0.67 μs, at a clock rate of 6-MHz.

The larger amount of ROM gives the user the ability to have multiple programs within a single code or to have a single very-detailed program. The result is one code in one microcomputer that can run a couple of different applications.

Each device provides an enhanced five-channel timer/counter function. Also provided is a timer/pulse generator capable of 14-bit pulse-width modulation (PWM) output. All on-chip peripherals are supported by vectored interrupts.

NEC provides a variety of development tools for these new devices. The μPD75P238, intended for prototyping, contains an on-chip PROM in place of the mask-programmed ROM. Various other development tools are available, including the IE-75000-R emulator and a macro-assembler with a structured assembler pre-processor that brings C-compiler-like features to assembly-language programming.

With V_{DD} input at 5 volts, the μPD75236, μPD75237 and μPD75238 typically draw only 4.5 mA of current. The PROM-based μPD75P238 version typically draws only 9 mA. Prices of the 75X family range from $6.15 each for the μPD75236, when purchased in 50,000-piece quantity, to $15 each for the μPD75P238, when purchased in 100-piece quantity. Prices for the μPD75237 and μPD75238 are quoted to customers on request.

Temperature Sensors

Teledyne Components (1300 Terra Bella

Ave., Mountain View, CA 94039) offers a family of solid-state temperature sensors, the TC620, TC621 and TC626, that directly convert temperature to a digital output format. This allows for direct control of relays and switches and an easy interface to any microcontroller. These devices provide a variety of functions and wide applications flexibility.

With the TC620, a user can program upper and lower temperature settings. The TC621 is designed to be used with an external thermistor for remote sensing applications. The TC626 is preset at the factory in 5 ° C increments, for applications that require a switch. Each temperature sensor has a circuit design that compares current flowing through two resistors.

The TC620, TC621 and TC626 compete with thermistors in a wide range of applications, such as temperature switches to turn on fans, open or close valves, actuate process-control systems or shut down power supplies and motors.

These devices are available in plastic and ceramic packages in both DIP and SOIC configurations. Pricing is $2.15 each in 10,000-piece quantity for the plastic DIP.

Power-Supply Board

DATEL (11 Cabot Blvd., Mansfield, MA 02048) has a precision programmable power supply and input monitor, configured on an IBM PC/AT board that delivers a full 22 watts of output power. The company offers the multi-channel isolated board for programmable linear test applications, custom PC-based automatic test equipment (ATE) and software-controlled manufacturing usage. The system can be used for testing power-supply variation rejection on development circuits.

The new board, model PC-462, includes two unipolar channels of 0 to +20.475 volts and 0 to −20.475 volts at 250 mA maximum intended for linear test circuits. Also included are two 0- to +6.1425-volt and 0- to −6.1425-volt channels, each delivering 1 ampere maximum, typically for logic circuits.

All four output channels are isolated from the host PC/AT bus by 250 volts rms. Each channel is controlled by an independent 12-bit D/A converter under software control. Sense feedback offers precise load regulation. The output channels are unconditionally stable under all load conditions, with 0.025% accuracy and linearity on the ±20-volt channels. Transient response is 200 μs.

Both current and voltage can be monitored on all output channels with an on-board, fully isolated 16-channel analog input system, using a 12-bit A/D converter. Four independent external analog inputs and four calibration channels can also be measured. Triggering can be either internal or external.

Four software-programmable isolated digital channels (two inputs, two power MOSFET outputs) can control discrete devices. One input can be configured as a PC/AT interrupt to flash an alarm or perform an emergency shutdown.

A comprehensive Model PC-4625ET graphic software package, in *Windows Visual BASIC*, offers multi-channel voltmeter display and control, engineering unit conversion, digital I/O calibration, time-stamped file save/display mode, automatic overlimit warning and shutdown. Prerecorded test sequences from disk can be played back using the system clock for automated ramp testing applications.

PC-4625ET runs under Microsoft *Windows* 3.0. The PC-462 is configured on a 4.5″ × 13.31″ board filling a long PC/AT slot. PC/AT power is used at +5 volts, 4 amperes maximum and +12 volts, 1.5 amperes maximum.

The PC-462 costs $1,195 in single quantity. Generous quantity discounts are available. A free disk of C subroutine library software is included. Prices for the PC-4625ET and the PC-462SRC source code are $95 and $395, respectively. ■

IBM and Microsoft square off with their offerings of OS/2 2.0 and Windows/Windows NT products and a whole lot more

The Spring Comdex has traditionally been basically a regional show that alternated between Atlanta and Chicago, with the real industry event being the Fall Comdex held in Las Vegas. This year, though, the tables have been turned, thanks largely to IBM and Microsoft. The Las Vegas Comdex in the fall will be anticlimactic because the real event took place at the Spring Comdex 92 and Windows World show at Chicago's McCormick Place on April 6 through 9. What made this a notable event was the simultaneous and long-awaited introduction of Microsoft's *Windows* 3.1 and *Windows NT* and IBM's OS/2 2.0.

The Arena

IBM dominated the main hall of the east building of McCormick Place with a large area where OS/2 2.0 and many demonstrations of applications for it were shown. The *Windows* exhibit occupied the entire west hall building. To emphasize the claim that IBM had developed OS/2 2.0 for all Intel-based PCs, the company set up an area with several hundred 386 and 486 computers of all types running every conceivable application, including MS-DOS and *Windows* 3.1 applications, all running under OS/2 2.0.

Attendees at the IBM exhibit could sit at the keyboard and experience for themselves the new OS/2 2.0 operating system. An IBM team dressed in white sweaters were on hand to help users. The sweaters seemed to make a statement. These IBM people were technical experts on OS/2 2.0 who could answer questions. When they didn't know the answers, they admitted it and directed the asker to someone who did know the answer. This was impressive, when you consider that the usual IBM representative at these shows is a staid sales type dressed in a gray suit.

OS/2 2.0's Workplace Shell (IBM's term for its GUI desktop) impressed me, mostly because it looks like the *Windows* desktop I'm accustomed to. The entire OS/2 2.0 system is a huge improvement over previous versions I used several years ago. It ran several DOS and *Windows* applications simultaneously and seamlessly. I had as many as five tasks running at the same time. This is true multitasking that by itself makes OS/2 2.0 a worthwhile product. While there weren't any real 32-bit applications running on the machine I used, the MS-DOS and *Windows* applications with which I'm familiar ran as well under OS/2 2.0 as they did in the Microsoft environment.

IBM claims that over 1,000 developers are working on 32-bit applications, including all the big DOS and *Windows* software developers.

One interesting aspect of both the Comdex and Windows World shows was the Multimedia Pavilion and other multimedia exhibits scattered throughout the hall. This had to be the largest exhibition of computer multimedia equipment and software ever assembled in one show. In keeping with this multimedia theme, IBM announced the Ultramedia Touch Activity Center, a self-contained interactive kiosk retailers can use to merchandise and sell products and museums and schools can use as an educational device.

At a recent exhibition of art at Atlanta's High Museum, visitors could use the IBM kiosk to get a complete multimedia preview of the exhibition at the touch of a finger. The presentation included music, narration and high-quality video images of the artwork on display.

Over at Windows World, Microsoft regarded the advent of *Windows* 3.1 as an accomplished fact. The company had previously loaded Federal Express warehouses with packages containing *Windows* 3.1. On April 6, the opening day of the show, while Bill Gates was delivering his keynote address, Federal Express delivered *Windows* 3.1 all over the world. It was a masterpiece of planning and logistics.

Is *Windows* 3.1 worth all the fuss? This question will debated for a long time to come. As far as I'm concerned, it is. It's a much more robust system than *Windows* 3.0. It has many desirable features that were lacking in earlier versions of *Windows*. For example, Object Linking and Embedding (OLE) provides a means for transferring and sharing information between applications. It also has scaleable *True Type* technology that displays on-screen exactly what you'll see when you print it.

If you're familiar with *Windows* 3.0, you'll find the File Manager in *Windows* 3.1 very different. It now displays the tree with the directories in a window on the left side of the screen and a tree with the files in the selected directory in another window on the right side of the screen. There are so many new features in *Windows* 3.1 that I can't begin to describe them in this column. You can be sure, though, this will be grist for the mill in many computer magazines for the balance of this year.

All the benefits of *Windows* 3.1 come at a small price for the upgrade software, but a big price if you aren't a power user. Gone from *Windows* 3.1 is the "real" mode of operation, which writes off all computers that have less than 1M of RAM. You really need 4M and a 60M hard disk minimum to even think about running *Windows* 3.1 and using it for any practical applications. To my mind, Microsoft has not only written off XT/compatibles but, for all practical purposes, 286/compatibles as well (although 286s can still run in "standard" mode if they have 1M of RAM). *Windows* 3.1 defines the new entry level as a 386SX computer with at least 2M (and preferably 4M) of RAM.

The demands of IBM's OS/2 2.0 are even worse. With it, you need at the very minimum 4M of RAM and 30M of hard-disk space, although you can run a stripped-down system with less than 30M. But keep in mind that this is for just the operating system and doesn't include any applications not included with the system software.

Obviously, there are a lot of perfectly good working computers out there that will have to get along without OS/2 2.0 and *Windows* 3.1. If you have one of these computers, you can still use *GeoWorks Ensemble* or *GeoWorks PRO*. Strangely, Geoworks didn't stand up and crow about the thousands of users who were dumped into its lap by the new offerings from IBM and Microsoft. Stranger still, though GeoWorks was at Comdex, it was hidden behind a curtain in a closed "room" with a "keep out" sign for casual show attendees. Talk about dumb marketing!

In the main, Microsoft's *Windows* exhibit was given over to the applications developers for *Windows* 3.1, of which there were hundreds at the exhibit.

Microsoft itself showed its new technology, such as the 32-bit version of *Windows* now known as *Windows NT 32*, the company's response to OS/2 2.0 and other true 32-bit systems that are just around the corner. *Windows NT 32* will provide a platform for high-end server systems like RISC-based computers and large LAN-based systems. It's designed to run applications written to the WIN32 API (Applications Programming Interface) as well as existing DOS and *Windows* applications.

Microsoft also exhibited *Windows* for Pen Computing and featured OEM vendors who have created applications for this new type of computer.

I could not begin to describe in this column all the new hardware and software on exhibit at the Spring show. I can only give impressions and describe a few of the outstanding products I did see.

Other Hardware & Software

For PC computers, the trend is rapidly going to the 486DX at the high-priced end of the market and the 486SX at the low end. Laptops are moving from 386SX-based models to low-power (3-volt) 386SL CPUs. Color for laptops is now a practical reality, with resolution as good as VGA on a flat screen, though it's costly and not likely to become less so because of manufacturing problems with this type of screen.

An outstanding piece of hardware at the show was Kyocera's new Ecosys a-Si printer that uses an "amorphous silicon" technology that's new to page printers. An amorphous silicon electrostatic printer uses a new ultra-long-life drum/developer system that eliminates the need for cartridge replacement. All you do is periodically add toner. The printer uses LED exposure, rather than laser. This combination of features results in a longer-lasting, lower-cost printer.

Print speed with the Kyocera Ecosys a-Si is rated at 10 pages per minute. The printer comes with 1M of RAM and is expandable to 5M. It has 79 resident fonts, and eight more fonts are optionally available. The printer also has 13 scaleable resident fonts and 46 optional ones. It can also generate 41 types of barcode.

This new printer will be available with several 500-sheet paper drawers and a sorter/stacker/mailbox assembly. The basic machine will sell for $2,300 and is scheduled to be available about the time you read this.

Multimedia News

Some very interesting products were shown in the multimedia section of the show. Although overshadowed by the IBM and Microsoft operating-system spectaculars, the entrance of the major optical photographic companies into the electronic imaging field is a sure sign of its growing importance. Nikon's Electronic Imaging Department was very much in evidence with products for both the PC and Macintosh platforms.

Nikon's LS-3510AF film scanner reads 35-mm color and monochrome film positives and negatives at a resolutions up to 5,000 × 5,000 pixels. It converts analog data to digital data while reading 256 gradations each of red, green and blue (RGB) in 24-bit code and reproduces more than 16.7-million colors (4,096 RGB gradations and billions of colors with a 12-bit option). The company's CP-300 full-color printer can reproduce color photos at a resolution of 1,024 × 1,270 pixels using thermal sublimation dye transfer technology.

Other offerings from Nikon included an NT-3000 35-mm color direct telephoto transmitter that transmits digital and analog 35-mm color negative and positive film. Nikon also showed an HQ-1500C high-definition still camera that snaps shots at a rate of three frames per second and features error-free registration. The Nikon system interfaces to computer equipment through a general-purpose interface bus (GPIB) or a small computer system interface (SCSI).

Precision electronic photographic devices are what industry has come to expect from Nikon, but the next item was completely unexpected. Nikon has developed the fastest 12″ electronic imaging magneto-optical (MO) disk drive ever. Approximately 4 gigabytes of data can be recorded on one disk. This drive operates at a data-transfer rate of 220K bytes per second and has a rotational speed of 1,500 rpm. The controller reads and transfers data at the rate of 1.43M per second!

The Nikon 12″ MO disks are double sided and feature a unique double-layer magnetic thin-film structure on a reinforced glass substrate. Estimated life is greater than 10 years. Nikon obviously knows imaging and its needs. Its new MO drive offers the type of recording device that will be needed to keep many scanned high-resolution images on-line. Recording high-resolution electronic images requires an enormous amount of disk storage space that can render even the largest magnetic hard disks inadequate in short order. Nikon's new MO drive answers an urgent need in the electronic imaging industry.

IBM's OS/2 2.0 Plug

Microsoft released and delivered *Windows* 3.1 on the first day of Comdex, which was a dazzling coup in itself. IBM had also planned May 6 for a press conference at a downtown hotel to kick off OS/2 2.0. Even though this was to be the official introduction of OS/2 2.0 and the most-important product launch since it introduced the AT computer, IBM didn't have its logistics straight. As we members of the computer press passed through the entrance to the hotel, flyers were handed to us, inviting us to visit another hotel, where Bill Gates of Microsoft was to make some announcements to the same people who attended the IBM conference.

At the press conference, Jim Cannavino, V.P. of IBM and General Manage of Personal Systems, presided over a collection of top executives from IBM and major software companies. On hand were Alan Ashton, President of WordPerfect Corp.; Jack Blount, V.P. of Novell; Philippe Kahn, CEO and Chairmen of Borland International; John Warnock, Chairman and CEO of Adobe Systems; John Landry, V.P. of Lotus Development Corp.; Mike Copeland, President and CEO of Corel Systems; George Grayson, President of Micrografx; Gray Clow, President of Stac Electronics; and James Zuko, V.P. of MCI Telecommunications.

All the IBM people, Jim Cannavino included, wore white sweaters to show that they had changed and were no longer stuffed shirts in gray suits. The gist of Cannavino's speech was that OS/2 2.0 was the greatest thing for computers since IBM invented the first software because it was designed for use by everybody, not just those people who use IBM computers, and IBM had established a lab to test compatibility with all MS-DOS computers and application software.

Because it was so great they would almost give away OS/2 2.0. *Windows* users could upgrade for $49, DOS users for $99 and new users for only $119. Registered OS/2 users could update for free until July 31, 1992. Everyone got free telephone support for 60 days, after which, there will be an IBM Helpline at a reasonable cost.

Following Cannavino at the podium, the industry executives stood up and, one by one, delivered accolades about OS/2 2.0. Borland's Philippe Kahn confessed that he "...had tried *Windows* but had never inhaled!" The last speaker was Paul Pignatelli, owner of the Corner Store and an ordinary citizen. He delivered words of highest praise for IBM, PS/2 and OS/2 2.0 as if he had been sent down from heaven to mark this great event.

When the hype finished, it was Q&A time. One of the early questions from the floor was, "Can you install OS/2 2.0 from any drive?" The answer was, "No. You have to use Drive A." (All IBM PS/2 computers have a 3½″ drive A:.) Another question was, "If you must install from drive A:, why did [IBM] supply 3½″ disks when most non-IBM computers use a 5¼″ drive A: and can you get 5¼″ disks?" The answer to the second part of the question was: "Eventually, but for now we only have 3½″ disks." Later questions got more pointed and sharper.

At the end of the press conference, we received our copies of OS/2 2.2. I have mine in a closet, where it'll have to sit until I can find time to change the cables and alter my CMOS boot program so my 3½″ drive becomes drive A: from its present assignment as drive B:. I also have to move around a lot of files to free up 30M on my hard drive to accommodate the program. It would have been so much simpler if IBM had had the program ask me what drive I'd like to use for installing OS/2 2.0 than forcing me to use drive A:. I suspect that there will be quite a number of you who will feel the same as I do about this. Apparently IBM never considered that it would be used for a non-IBM computer!

Windows 3.1 and Microsoft clearly were the stars of this show. When the Fall Comdex in Las Vegas rolls around, the smoke will likely have cleared and the picture will be clearer as to where everybody stands in the continuing battle for operating-system dominance. Now that Microsoft has finally made the complete break from IBM, the world of personal computing may never again be the same. ■

Synthesizer *(from page 41)*

the board in front of you oriented as shown in Fig. 8. Begin populating it by installing and soldering into place the sockets for all DIP ICs. Do *not* plug the ICs into the sockets until after you've conducted preliminary voltage checks and are certain that everything is okay.

Next, install and solder into place the resistors, capacitors, transistors and crystal. Make certain that the electrolytic capacitors are properly oriented and that the transistors are properly based before soldering their leads into place. Then install and solder into place the 25-pin right-angle DB-25 connector in the *P1* location.

The only off-the-board components in this project are BNC connector *J1*, the POWER LED that connects between *R13* and ground and the optional jack for the external plug-in power supply.

Component problems you might encounter are few. Make sure you use the -2 version of the 82C54 chip. Use zero-temperature-coefficient capacitors for *C2* and *C4* to maximize temperature stability of the circuit, and use a low-leakage ±10% tantalum capacitor for *C5*.

When you have the circuit completely assembled, mount it in any enclosure that will accommodate it, preferably plastic for easy machining of the slot required for *P1*. Use a hot knife or other suitable tools to make the slot for the DB-25 connector, and drill suitable size holes for mounting the LED, BNC connector and plug-in power-supply connector. Then drill smaller holes for mounting the board to the floor of the enclosure.

If you prefer, you can eliminate the connector for the power supply. In this case, simply route the output cable from the supply through its own hole in the enclosure and wire it directly to the appropriate points on the board. Then use ½″ spacers to mount the circuit-board assembly inside the enclosure.

Mount the LED in its hole in the enclosure. If necessary, use a drop of fast-setting epoxy cement to secure it in place. Then wire it to the circuit-board assembly, lengthening its leads with stranded hookup wire as needed and insulating all exposed wiring to prevent short circuiting. The anode lead goes to the free end of *R13* the cathode lead to circuit ground.

Table 1. Control-Word Formats

Control Word	Numbering System Binary	Hex	Decimal
0	0011,0110	36	54
1	0111,0110	76	118
2	1011,0110	B6	182

Finish up by mounting the BNC connector in its hole and wiring it to the circuit-board assembly. Use coaxial cable for the connections from the BNC connector to the OUTPUT and nearby GND holes in the circuit board.

With no ICs, except regulator *U8* installed on the circuit-board assembly, power up the circuit. Use a dc voltmeter or a multimeter set to the dc-volts function to first check the polarity of the power-supply connections. If it's correct, clip the common meter lead to any circuit-ground point and use the "hot" lead to probe the V_{cc} contacts of the IC sockets (pin 14 of *U1* and *U2*; pin 16 of *U3, U4, U6* and *U7*; and pin 24 of *U5*). If you fail to obtain a reading of + 5 volts at any of these points, power down and correct the problem before proceeding.

Once you're sure your wiring is correct, power down and plug the ICs into their respective sockets. Make sure each IC is properly oriented and that no IC pins overhang the sockets or fold under between ICs and sockets.

Programming

To use the Frequency Synthesizer, you must create a program that asks for the desired output frequency and then automatically sets up the counter chip and latch to generate the specified frequency. You must perform a few preliminary procedures before you can get into actual setting of the counter. The counter chip provides six operating modes for each counting element. Two modes are repetitive; once set, they keep on running. The other four are one-shot modes. Since you're dealing with frequency dividers, you want the repetitive modes.

In the rate-generator mode, the counter is preset to some number between 2 and 2^{16}. When the counter indexes to 0, a single clock pulse appears at the output, and countdown starts again. The second repetitive mode, called square-wave mode, works with

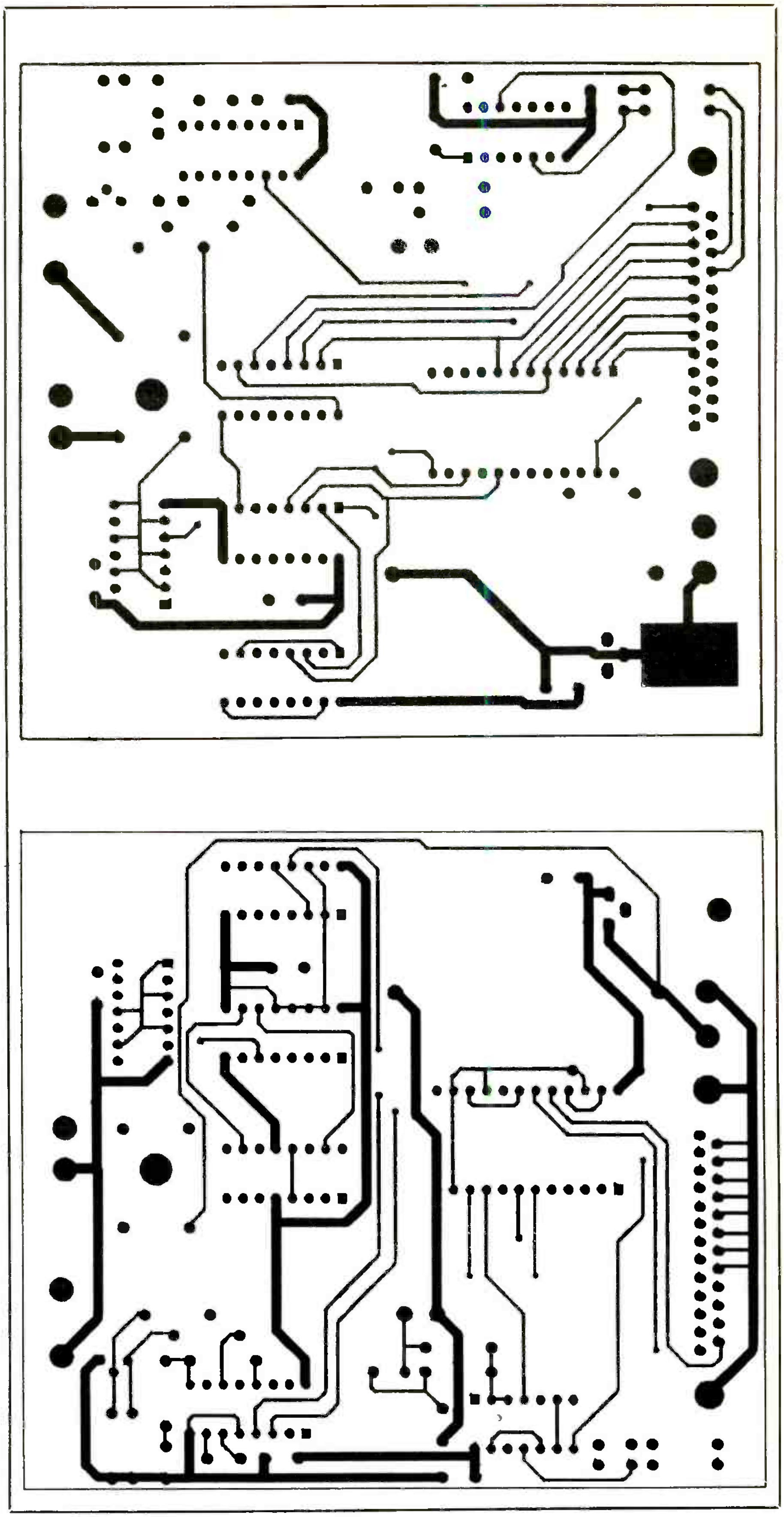


Fig. 7. Actual-size etching-and-drilling guide for making printed-circuit board for project.

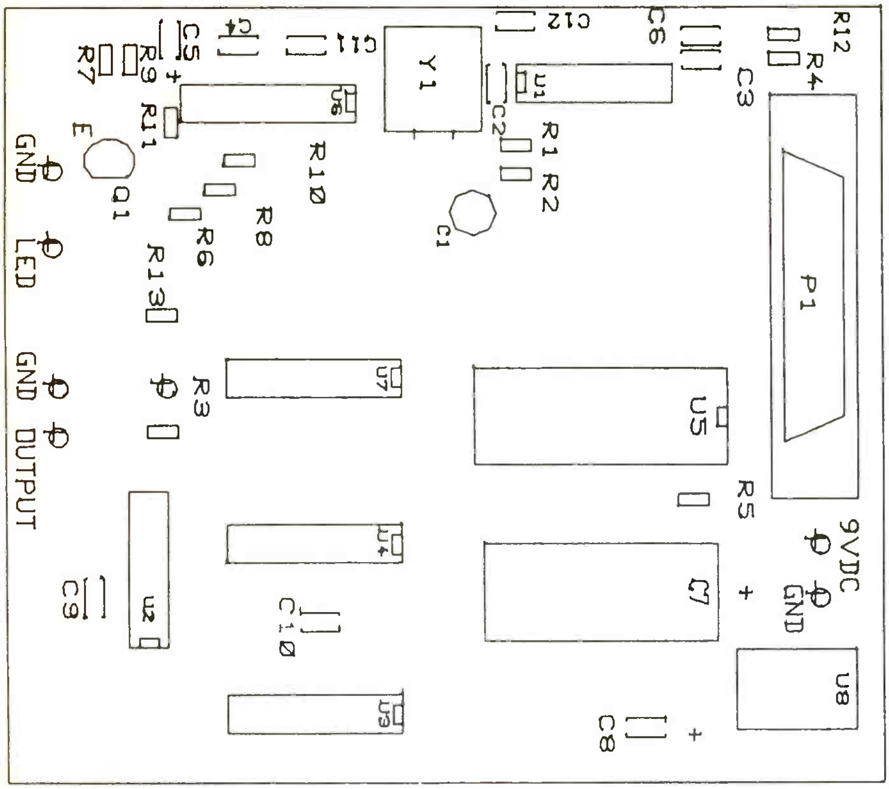


Fig. 8. Wiring guide for pc board.

the N, M and output divider counter of the Synthesizer. After priming the system, your first programming task is to set each of the three counters in *U5* to the square-wave mode.

You can then set up each counter to count in binary or BCD. Keeping things simple, I chose the binary mode. Next is the read/write least-significant-byte-first, most-signifi-

Listing 1

```
OUT &HBC, 8              'DATA TO DATA PORT
CTW = &HF XOR &HB        'CONTROL WORD FOR LOGIC 1 TO LATCH
OUT &H3BE, CTW           'ENABLES LATCH
CTW = &H7 XOR &HB        'CONTROL WORD FOR LOGIC 0 TO LATCH
OUT &H3BE, CTW           'DISABLES LATCH
```

Listing 2

```
   DT = &H3BC            'Printer data port address
   CR = &H3BE            'Printer control port address
   CT =   n              'Counter # (n = 0, 1, 2)
CW(0) = &H36             'Control word counter zero
CW(1) = &H76             'Control word counter one
CW(2) = &HB6             'Control word counter two
```

Listing 3

```
FOR I = 1 TO 3
     OUT DT, CW(I)          'Control word to data register
     CTW = &H3 XOR &HB      'Set "Write" to a zero
     OUT CR, CTW            'Output to control port
     CTW = &H7 XOR &HB      'Set "Write" to a one
     OUT CR, CTW            'Output to control port
```

Listing 4

```
N = 6,000
UPPER BYTE = NU = INT(N/256)
LOWER BYTE = NL = N - 256*NU
```

cant-byte-second mode. The control-word format for the various modes is illustrated in Fig. 6.

From Fig. 6, you can see that bits D0 through D5 of each control word are the same. Bits D6 and D7 differ because they specify different counters. Therefore, the control words are summarized in Table 1.

Before you can feed the control words to the counter chip, you must prime the system, make sure the control lines and latch output lines are at the correct levels. There are only two lines with you must concern yourself —/CS-/WT to the counter chip and enable to *U7*. These must be in the high and low states, respectively. Both lines come from the printer control register and can be set to the required levels by outputting an appropriate bit pattern to the printer control port. The bit pattern is as follows:

	C3	C2	C1	C0
	0	1	X	X
Latch Strobe				
Write Strobe				

The Xs are don't-care states. Keep in mind that outputs C3, C1 and C0 of the control port are inverted. Arbitrarily setting the Xs to 1s, the nibble pattern becomes 0111 binary, which is 7 hex and decimal.

Before you can send the control nibble to the control port, you must XOR it with hex B. So,

```
CTW = &H7 XOR &HB
OUT (control-register address), CTW
```

would do the trick if you knew the address of the control register. You do know it. Assuming you're using LPT1, the control-port address is 03BE hex.

You must now set up the latch to turn off the output signal. Since the MUX on/off control is connected to the L3 output of the latch and is active, you must set it high to turn off the output. To do so, first set the data port to 08 hex or decimal and then strobe printer control port C3 high and then low. The data on the data line then gets

Listing 5

```
    OUT DT, NL                   'Lower byte to data port
          CTW = &H0 XOR &HB       ' Write bit to zero
                OUT CR, CTW       ' Output to control port
          CTW = &H7 XOR &HB       ' Write it to one
                OUT CR, CTW       ' Output to control port
    OUT DT, NU                   'Lower byte to data port
          CTW = &H0 XOR &HB       ' Write bit to zero
                OUT CR, CTW       ' Output to control port
          CTW = &H7 XOR &HB       ' Write it to one
                OUT CR, CTW       ' Output to control port
```

Listing 6

```
SUB CTSET
            IF CT = 0 THEN S1 = 11: S2 = 15
            IF CT = 1 THEN S1 = 10: S2 = 14
            IF CT = 2 THEN S1 =  9: S2 = 13
      OUT DT, NL: OUT CR,S2: OUT CR,S1: OUT CR,S2
      OUT DT, NU: OUT CR,S2: OUT CR,S1: OUT CR,S2
END SUB
```

Listing 7

```
A:
'** ASKS FOR OUTPUT FREQUENCY
INPUT "Enter Frequency in Hz, (1.000 to 10,000,000)";Fo
'** CHECKS RANGE OF INPUT
            IF Fo<1 OR IF Fo>11^7 THEN
                  PRINT "OUT OF RANGE": BEEP
                  SLEEP (4):  CLS:  GOTO A
            END IF:
'** COMPUTES EXPONENT SCALING FACTOR
      EP = 6 - INT(LOG(Fo)/LOG(10)))    'Scaling constant
      IF EP <0 THEN EP = 0              '
      Fx = Fo*(10^EP)                   'Scale input to 1-10MHz
      N  = INT(Fx/1000)                 'Counter 1 divisor
      NU = INT(N/256)                   'Upper byte counter 1
      NL = N - 256*NU                   'Lower byte counter 1
      CT = 1                            'Selects counter 1
      CALL CTSET                        'Sets counter 1
'** SETS THE VCO RANGE
      IF Fo<3000000 THEN VCO = 0 ELSE VCO = 1
'** SETS DIVIDER SELECTION.  NOTE, DCO AND DC1 ARE USED BY
'** THE 'LATCH TO SET THE DECODER INUTS TO THE 74HC151
   SELECT CASE EP
      CASE 0
            DC0=0: DC1=0:  NU=0:                  NL=0
      CASE 1, 2, 3, 4
            DC0=1: DC1=0
            IF EP = 1 THEN NU = 0 NL=   10
            IF EP = 2 THEN NU = 0 NL = 100
            IF EP = 3 THEN NU=INT( 1000/256): NL= 1000-(NU*256)
            IF EP = 4 THEN NU=INT(10000/256): NL=10000-(NU*256)
      CASE 5
            DC0=0: DC1=1: NU=INT(10000/256): NL=10000-(NU*256)
      CASE 6
            DC0=1: DC1=1: NU=INT(10000/256): NL=10000-(NU*256)
   END SELECT
      CT = 2:                         CALL CTSET
      LTCH =8+4*DC1+2*DC2+1*VCO:      CALL LATCH
'** SLEEP DELAY LETS PLL LOCK AND SETTLE DOWN
      SLEEP (1)
'** TURNS OUTOUT SIGNAL ON
      LTCH = LTCH - 8:                CALL LATCH
GOTO A:

     SUB LATCH
           SHARED DT, CR, LTCH
           OUT DT, LTCH
           OUT CR,12: OUT CR, 4: OUT CR,12
     END SUB
```

Listing 8

```
*****************************************************************************
             SIMPLE PLL CONTROL PROGRAM, 3/30/92                   Page 1
*****************************************************************************

   DECLARE SUB SCRN ()
   DECLARE SUB FREQCOMP ()
   DECLARE SUB LATCH ()
   DECLARE SUB REFSET ()
   DECLARE SUB CTSET ()
   COMMON SHARED DT, CR, CT, CW(), NU, NL, LTCH, Fo, QT$, ORM$
   DIM CW(3):      CLS

** DEFINITIONS:
     DT = &H3BC                               'PRINTER PORT DATA REGISTER ADDRESS
     CR = &H3BE                               'PRINTER PORT CONTROL REG.  ADDRESS
  CW(0) = &H36                                'COUNTER 0 CONTROL WORD
  CW(1) = &H74                                'COUNTER 1 CONTROL WORD
  CW(2) = &HB6                                'COUNTER 2 CONTROL WORD
    QT$ = "PRESS <Q> KEY TO EXIT PROGRAM"
   ORM$ = "OUT OF RANGE"

   CALL SCRN
   OUT CR, 4                                  'PRIMES CONTROL REGISTER
   LTCH = 8:      CALL LATCH                  'TURNS OFF OUTPUT SIGNAL
   FOR I = 0 TO 3
       OUT DT, CW(I):
       OUT CR, 12: OUT CR, 8: OUT CR, 12     CONTROL WORDS TO COUNTER CHIP
   NEXT I
   C = 0:         CALL REFSET

*****************************************************************************
                          MAIN PROGRAM
*****************************************************************************
   DO
       LOCATE 7, 35:   LINE INPUT W$
       LOCATE 7, 35:   PRINT "________"
       X$ = LTRIM$(W$):     Fo = VAL(W$)
       IF Fo < 1 OR Fo > 11 * 10 ^ 6 THEN
               LOCATE 18, (80 - LEN(ORM$)) / 2
               PRINT ORM$:      BEEP:   SLEEP (2)
               LOCATE 18, 1:    PRINT STRING$(75, " ");
           ELSE CALL FREQCOMP
       END IF
   LOOP UNTIL X$ = "Q" OR X$ = "q"
   BEEP: LOCATE 24, 33: PRINT "END OF PROGRAM";
END
*****************************************************************************


SUB CTSET
   IF CT = 0 THEN S1 = 11: S2 = 15
   IF CT = 1 THEN S1 = 10: S2 = 14
   IF CT = 2 THEN S1 = 9: S2 = 13
   OUT DT, NL: OUT CR, S2: OUT CR, S1: OUT CR, S2
   OUT DT, NU: OUT CR, S2: OUT CR, S1: OUT CR, S2
END SUB

SUB LATCH
   OUT DT, LTCH: OUT CR, 12: OUT CR, 4: OUT CR, 12
END SUB
```

Listing 8 Continued
On Page 78

latched with the routine given in Listing 1.

With the Synthesizer primed, send the three control words to *U5*. This time, the control word nibble looks a little different. A0 and A1 must be 1s, and the latch strobe must be 0. The write strobe must be cycled from a 1 to a 0 and back to a 1.

The notation starts to become cumbersome here. Therefore, I defined a few mnemonics, as given in Listing 2. This done, you can write the instruction for setting up the counter modes, as enumerated in Listing 3.

The system is primed and the counting modes have been set. Now you must send the division constants to the counter. Let's begin with the crystal divider. Say you're using a 6-MHz crystal. Divide by 6,000, and out comes the 1,000-Hz reference you want. To load 6,000 into *U5* in two bytes, first break down the number into two parts, as shown in Listing 4. Then run the NU and NL into the counter, as detailed in Listing 5.

Note that &H0 in the second and seventh lines sets all bits of the control register to 0. Therefore, A0 and A1 are 0, selecting counter 0.

At this point, I became a little annoyed at all the XORing and OUTing business. So I set up a subprogram to take care of the bookkeeping. It's shown in Listing 6.

By declaring CR, CT, NL and NU common variables at the beginning of the program, all you have to do is set counter number CT and assign values to NL and NU and then call the subroutine. This saves a lot of typing.

Listing 8 Continued

```
'************************************************************************
'                  SIMPLE PLL CONTROL PROGRAM, 3/30/92              Page 2
'************************************************************************


SUB REFSET
   Fcy = 6002000: N = Fcy / 1000: NU = INT(N / 256)
   NL = INT(N - (NU * 256)):  CT = 0:  CALL CTSET
END SUB


SUB SCRN
   PS = (80 - LEN(QT$)) / 2: LOCATE 22, PS: PRINT QT$: LOCATE 1, 1:
   PRINT "                               ENTER FREQUENCY"
   PRINT
   PRINT "             (1.000Hz <= Fo <= 10,999,000MHz, 4 SIGNIFICANT DIGITS)"
   LOCATE 7, 35:   PRINT "__________"
END SUB

SUB FREQCOMP
   LTCH = 8: CALL LATCH                                   'TURNS OFF OUTPUT
   EP = 6 - INT((LOG(Fo)) / (LOG(10)))                    '   COMPUTES
   IF EP < 0 THEN EP = 0                                  ' HIGH & LOW BYTES
   Fx = Fo * (10 ^ EP): N = INT(Fx / 1000)                '  FOR COUNTER 1
   NU = INT(N / 256): NL = N - (NU * 256)                 '
   CT = 1: CALL CTSET                                     'SETS PLL DIVIDER
   IF Fx > 3000000 THEN VCO = 1 ELSE VCO = 0              'SETS VCO RANGE
   Fd = (INT(Fx / 1000)) * 10 ^ (3 - EP)
   LOCATE 15, 22:
   PRINT "OUTPUT FREQUENCY IS ... "; USING "##,###,###.### Hz"; Fd

  SELECT CASE EP
   CASE 0
        DCO = 0: DC1 = 0: NU = 0:                   NL = 0
   CASE 1, 2, 3, 4
        DCO = 1: DC1 = 0
           IF EP = 1 THEN NU = 0:                   NL = 10
           IF EP = 2 THEN NU = 0:                   NL = 100
           IF EP = 3 THEN NU = INT(1000 / 256):     NL = 1000 - (NU * 256)
           IF EP = 4 THEN NU = INT(10000 / 256):    NL = 10000 - (NU * 256)
   CASE 5
        DCO = 0: DC1 = 1: NU = INT(10000 / 256): NL = 10000 - (NU * 256)
   CASE 6
        DCO = 1  DC1 = 1: NU = INT(10000 / 256): NL = 10000 - (NU * 256)
  END SELECT
        CT = 2:                                  CALL CTSET 'SET OUTPUT DIVIDE
        LTCH = 8 + 4 * DC1 + 2 * DCO + 1 * VCO:  CALL LATCH 'SETS LATCH
        SLEEP (1)                                           'ONE SECOND DELAY
        LTCH = LTCH - 8:                         CALL LATCH 'TURNS ON OUTPUT
END SUB
```

The previous program segment for setting counter 0 can now be done with the following two lines:

CT = 0: NU = INT(6000/256): NL = 6000 – 256*NU
CALL CTSET

The next thing to do is set counter 1. If you want an output between 1 MHz and 10 MHz, this is easy to do. Just divide the frequency by 1,000 and then break the results down to NU and NL and send them off to the Synthesizer via the CTSET subroutine.

For the 1-Hz-to-1-MHz range you must do a bit of fancy foot-work. You have to scale the desired frequency up to the 1-MHz-to-10-MHz range, set counter 2 and then decide which divider output to use to get the desired frequency. Listing 7 shows one way of doing this. In looking over this Listing, you'll notice that one new subroutine (shown shaded) has been added.

A complete QBASIC listing for a *simple* Frequency Synthesizer program is given in Listing 8. Feel free to change, add to or rewrite it any way you choose. If you want to, you can write a pop-up program that activates when an unusual combination of keys is pressed and include a software switch to turn on and off the output signal. How about a sweep mode where the frequency changes by, say, 1,000 Hz every 5 seconds up to some predetermined frequency, after which the sweep reverses. As you can see, how you use the Frequency Synthesizer presented here is limited only by your imagination and ability to write routines to effect what you want. ■

Table 1. DIP-Switçh Baud-Rate Selection

	DIP Switch DS1		
Baud Rate	**Position 3**	**Position 2**	**Position 1**
150	Off	Off	Off
300	Off	Off	On
600	Off	On	Off
1,200	Off	On	On
2,400	On	Off	Off
4,800	On	Off	On
9,600	On	On	Off
19.2K	On	On	On

JP3—Jumper from pin 2 to pin 3 for external program memory.

JP4—Jumper from pin 1 to pin 2 for EPROM.

JP5—Jumper from pin 1 to pin 2 for external program memory.

Make sure that the baud rate used by your computer is the same as that selected on the control system circuit board(s). Use Baud-Rate Selection Table 1 when setting the baud rate of the system circuit board(s). Use the DIP-switch numbers silk-screened on the printed circuit board. Do *not* use any switch numbers that may be stamped or painted on the DIP-switch assembly.

Set position 4 of DIP switch *SW1* to OFF for single-board mode or ON for network mode. When using network mode, each board in the system must have its own unique address. Set the four-bit binary address positions 5 through 8 of DIP switch *SW1* for each circuit board in your system.

Digital I/O port control of each circuit board in the control system is automatic. When a circuit board is powered up or reset, all three digital I/O ports of the 8255 initialize as input ports. Whenever a circuit board receives a system command from the host PC that involves a read or write operation on any of the three digital I/O ports, the port control is automatically changed, if necessary, before the command is executed.

As an example of the above, if digital I/O Port 2 is used as an output port by your particular application, you want to initialize it as soon as the system starts up by writing a byte to it. This will switch port configuration of the 8255 so that Port 2 becomes an output port. If you decide to read the contents of Port 2 later, Port 2 would automatically switch back to an input port status.

It's important to know that all output ports reset low whenever port control is changed. Because this is an internal function of the 8255, it's not recommended that you mix input and output commands for the same digital I/O port unless you know the consequences.

Hardware Interfacing

Figure 5 is a schematic diagram of a simple stepper motor power driver circuit that can support two small-to-medium-size stepper motors. All components needed to build this circuit can be purchased at your local Radio Shack store. You can build the circuit on perforated board that has holes on 0.1″ centers using insulated hookup wire. Connector *P4* on the stepper power driver circuit connects directly to nine-pin connector *J4*, which is digital I/O Port 3 on the embedded controller circuit board.

A lot of inexpensive stepper motors are available for the robotics hobbyist. One source is American Design Components (815 Fairview Ave., P.O. Box 520, Fairview, NJ, 07022; tel.: 1-800-776-3700).

Figure 6 illustrates several simple hardware interfacing examples for those with little or no experience in interfacing computers to the outside world. Circuit (A) illustrates use of a LED indicator driven by one of the 16 digital outputs at connector *J3*.

Circuit (B) uses a relay as an isolated switch to turn on and off ac- or dc-powered devices. The resistance of the relay coil should be great enough that current drawn doesn't exceed the maximum power dissipation of the ULN2803A Darlington array IC. A coil resistance of 200 ohms and a power source of 12 volts wouldn't overtax the ULN2803A, unless more than three relays of the same type are used with the same

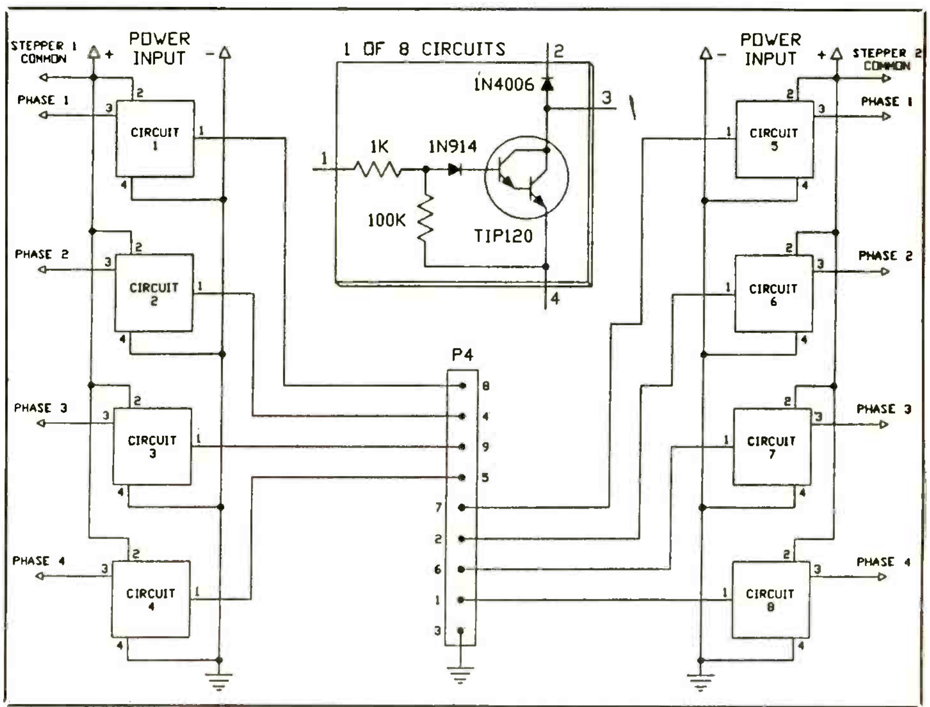


Fig. 5. Schematic diagram of a stepper-motor power driver circuit.

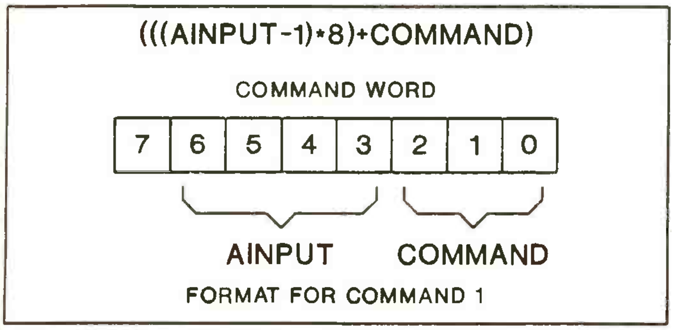
Command Summary

An illustration of the command word format and a BASIC program example statement are included in the software supplied by the source given in the Note at the end of the Parts List for each command description. Though the programs illustrate use of BASIC to command and control the system, you can use any programming language that sends and receives data in the system format described here. The command number is contained in bits 0, 1 and 2 of the command word. Commands 0 and 7 aren't used and are ignored by the system.

• **Command 1** (read analog input). There are sixteen analog inputs. The analog input number is contained in bits 3 through 6 of the command word. Bit 7 isn't used.

Example:

```
AINPUT = 4: COMMAND = 1
PRINT-1,CHR$((AINPUT – *8) + COMMAND);
I = ASC(INPUT$(1,-1))
```

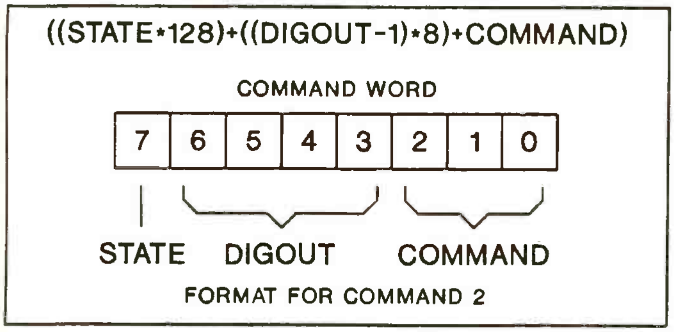


When the system receives this command from the host PC, it converts analog input 4 to an eight-bit value and sends this to the host PC. Analog input numbers are defined with a range of 1 through 16 but are converted to a range of 0 through 15 by the command output PRINT- statement. Variable "I" contains the analog input value received by the host PC.

• **Command 2** (write digital output bit). Sixteen digital outputs are associated with this command. Only Ports 1 and 2 are affected by Command 2. The digital output number is contained in bits 3 through 6 of the command word. Bit 7 contains the desired state of the digital output.

Example:

```
STATE = 1: DIGOUT = 13: COMMAND = 2
PRINT-1,CHR$((STATE*128) + ((DIGOUT – 1)*8) + COMMAND);
```

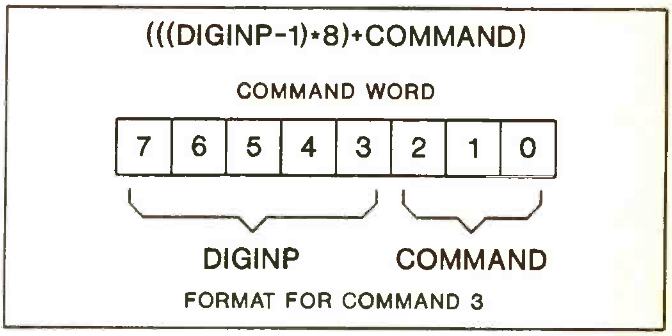


When the system receives this command from the host PC, digital output 13 is set to a logic-1 state. Digital output numbers are defined with a range of 1 through 16 but are converted to a range of 0 through 15 by the command output PRINT- statement.

• **Command 3** (read digital input bit). This command has 32 digital inputs associated with it. Digital inputs 1 through 16 are Ports 1 and 2. Digital inputs 17 through 32 are the analog inputs (Ports 4 and 5). The digital input number is contained in bits 3 through 7 of the command word.

Example:

```
DIGINP = 10: COMMAND = 3
PRINT-1,CHR$(((DIGINP – 1)*8) + COMMAND);
I = ASC(INPUT$(1,-1))
```

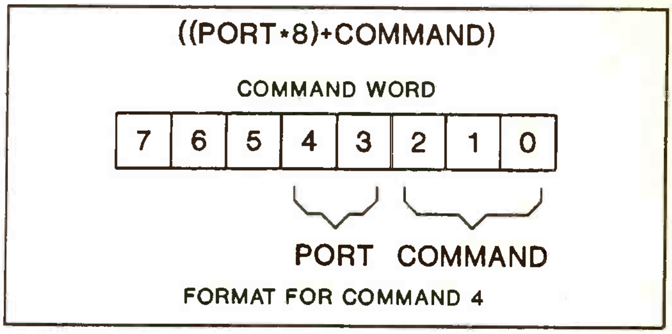


When the system receives this command from the host PC, the current state of digital input 10 is sent to the host PC as a decimal value of 0 or 1. Digital input numbers are defined with a range of 1 through 32 but are converted to a range of 0 through 31 by the command output PRINT- statement. Variable "I" contains the digital input value (0 or 1) received by the host PC.

• **Command 4** (write digital output byte). This command writes an eight-bit byte to Port 1, 2 or 3. The port number is contained in bits 3 and 4 of the command word. Bits 5, 6 and 7 aren't used. The eight-bit output value is sent immediately after the command.

Example:

```
PORT = 1: COMMAND = 4
PRINT-1,CHR$((PORT*8) + COMMAND);
PRINT-1,CHR$(129);
```



When the system receives this command and its associated output value from the host PC, digital outputs 1 and 8 are set to logical 1 and digital outputs 2 through 7 are cleared to logical 0.

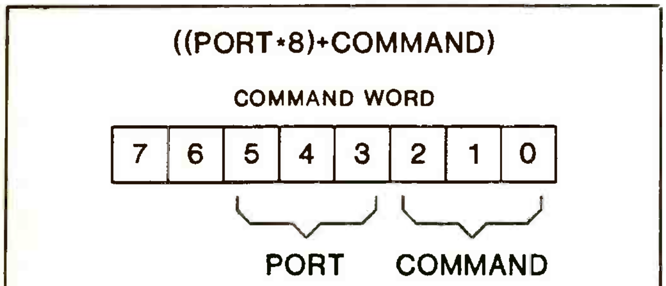
• **Command 5** (read digital input byte). Five ports are associated with this command. Ports 1, 2 and 3 are digital I/O ports; Port 4 is analog inputs 1 through 8; and Port 5 is analog inputs 9 through 16. The port number is contained in bits 3, 4 and 5 of the command word. Bits 6 and 7 aren't used.

Example:

```
PORT = 4: COMMAND = 5
```

```
PRINT-1,CHR$((PORT*8)+COMMAND);
I=ASC(INPUT$(1,-1))
```

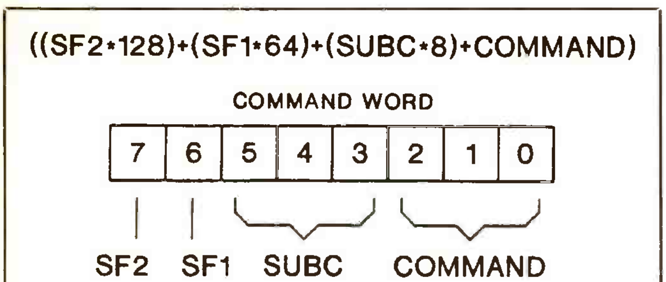
When the system receives this command from the host PC, analog inputs 1 through 8 are converted into 1-bit values. These bit values are all packed into one eight-bit byte and sent to the host PC. The current value of each analog input is tested using 2.0 volts as the threshold. If an analog input is equal to or greater than 2.0 volts, its associated bit value is set. If it's less than 2.0 volts, the bit is cleared.



FORMAT FOR COMMAND 5

Command execution time is considerably faster when Ports 1, 2 and 3 are used because just a digital input port must be read, but the eight analog inputs (Port 4 or 5) individually must be converted, tested for 2.0 volts and then packed into one byte.

• **Command 6** (stepper motor control). Eight subcommands are associated with Command 6. If the command number contained in bits 0, 1 and 2 of the command word is 6, a subcommand number contained in bits 3, 4 and 5 is decoded by the system. Bits 6 and 7 of the command word are used as flags by most commands to determine which stepper motor(s) is associated with the command. When executed, all commands could effect one or both stepper motors. Bit 6 is always associated with stepper 1, bit 7 always with stepper 2.



FORMAT FOR COMMAND 6

Subcommands are numbered from 0 to 7.

• **Command 6, Subcommand 0** (initialize stepper driver) initializes digital I/O Port 3 for the stepper driver interface. Bits 6 and 7 of the command word determine the type of stepper motor (three- or four-phase). When these bits are cleared (0), the associated stepper motor is three-phase. When the bit is set (1), the associated stepper motor is four-phase.

You can initialize the interface for any combination of three- or four-phase stepper motors. Bit 6 determines the type for stepper 1, bit 7 for stepper 2. Once the stepper driver interface has been initialized, the circuit must be reset, or power must be turned off and then on again before it can be reinitialized.

Digital I/O port 3 isn't affected by command 4 (write digital output byte) or Command 5 (read digital input byte) once the stepper driver has been initialized. Once initialized, a Command 4 execution doesn't change the output states of Port 3, and Command 5 execution sends back a null (0) for the input states of Port 3. These commands function for the other ports.

Bits 0 through 3 of Port 3 are stepper 1 phases one through four, in that order, and bits 4 thru 7 of Port 3 are stepper 2 phases one through four.

Example:

```
SF2=0: SF1=1: SUBC=0: COMMAND=6
PRINT-1,CHR$((SF2*128)+(SF1*64)+(SUBC*8)+COMMAND);
```

When the system receives this command from the host PC, digital I/O Port 3 is initialized as the stepper driver interface. Stepper 1 is driven as a four-phase, stepper 2 as a three-phase motor.

• **Command 6, Subcommand 1** (change step clock rate). This command changes the step clock rate. Bits 6 and 7 aren't used by this command. A default step clock rate value of 199 (approximately 500 Hz) is selected when the stepper driver is initialized. This command allows you to fine-tune the maximum step rate of the stepper motors. Both stepper motors are directly affected by this command.

The step clock rate is a value between 0 and 255. The step clock rate value is sent immediately after the command word. If the stepper(s) is in motion when this command is executed, the step clock rate doesn't change. If the step clock rate value is increased, the step rate increases, but not linearly. You must experiment with this value to achieve optimum performance for your particular application.

Example:

```
RATE=197: SUBC=1: COMMAND=6
PRINT-1,CHR$((SUBC*8)+COMMAND);
PRINT-1,CHR$(RATE);
```

When the system receives the rate value from the host PC, the step clock rate for both stepper motors is increased to approximately 488 Hz.

• **Command 6, Subcommand 2** (change steppers speed divisor). This command allows you to change the stepper(s) speed by a given step rate division. A default speed divisor of 10 is set when the stepper driver is initialized. The stepper speed divisor can be changed for either stepper or both at the same time. When bit 6 of the command word is set (1), the speed divisor for stepper 1 is changed. When bit 7 of the command word is set (1), the speed divisor for stepper 2 is changed. Both of the stepper motor speed divisors will be changed if bits 6 and 7 are set.

The speed divisor value is sent immediately following the command word. Speed divisor(s) can be changed when the stepper(s) is in motion. The speed divisor ranges from 1 to 255. The divisor value represents the number of times the step clock rate is divided for the particular stepper(s).

Example 1:

```
DIV=10: SF2=0: SF1=1: SUBC=2: COMMAND=6
PRINT-1,CHR$((SF2*128)+(SF1*64)+(SUBC*8)+COMMAND);
PRINT-1,CHR$(DIV);
```

When the system receives the divisor value from the host PC, the speed of stepper 1 is $\frac{1}{10}$ the value of the step clock rate.

Example 2:

```
DIV=5: SF2=1: SF1=1: SUBC=2: COMMAND=6
```

Command Summary

```
PRINT-1,CHR$((SF2*128)+(SF1*64)+(SUBC*8)+COMMAND);
PRINT-1,CHR$(DIV);
```

When the system receives the divisor value from the host PC, the speed of both steppers is ⅕ the value of the step clock rate.

• **Command 6, Subcommand 3** (load step count). This command loads the step count for one or both stepper motors. Bits 6 and 7 are used with this command. If bit 6 is set (1), the step count that follows the command will be loaded into stepper 1's step count register. If bit 7 is set (1), the step count is loaded into stepper 2's step count register. If bits 6 and 7 are set, both steppers load the same step count.

Step count ranges from 0 to 65,535, and two count bytes must be sent after the command word. The low step count byte is sent immediately after the command, followed by the high step count byte. If the stepper(s) is in motion when this command is executed, the step count will not be excepted.

Example:

```
COUNT=100: SF2=0: SF1=1: SUBC=3: COMMAND=6
PRINT-1,CHR$((SF2*128)+(SF1*64)+(SUBC*8)+(COMMAND);
PRINT-1,CHR$(COUNT);
PRINT-1,CHR$(0);
```

When the system receives the high step count value, stepper 1 is ready to move 100 steps. The example sends a high step count of 0. In this manner, step counts up to 255 can be sent without having to separate the step count into high and low bytes.

• **Command 6, Subcommand 4** (change stepper direction). This command changes one or both of the stepper direction flags. Both stepper direction flags default to clockwise upon initialization of the stepper motor driver. Bits 6 and 7 are used with this command. Bit 6 determines the state of the direction flag for stepper 1, bit 7 for stepper 2. If the bit is cleared (0), the stepper direction sets to clockwise. If the bit is set (1), the stepper direction sets to counterclockwise. Clockwise direction is when the stepper phase outputs sequence up from phase one to three or four. Counterclockwise direction is when the stepper phase outputs sequence down from phase four or three to phase one. If the stepper(s) is in motion when this command is executed, the direction flags will not change.

Example:

```
SF2=1: SF1=0: SUBC=4: COMMAND=6
PRINT-1,CHR$((SF2*128)+(SF1*64)+(SUBC*8)+COMMAND);
```

When this command is received by the system, stepper 1 moves in the clockwise direction, stepper 2 in the counterclockwise direction when they're set in motion.

• **Command 6, Subcommand 5** (start stepper). This command sets one or both steppers in motion. Bits 6 and 7 are used with this command. Bit 6 affects stepper 1, bit 7 stepper 2. If the bit is set, the stepper(s) starts moving, provided the step count isn't 0.

Example:

```
SF2=1: SF1=0: SUBC=5: COMMAND=6
PRINT-1,CHR$(SF2*128)+(SF1*64)+(SUBC*8)+COMMAND);
```

When this command is received by the system, stepper 2 starts in motion if its step count isn't 0.

• **Command 6, Subcommand 6** (stop stepper). This command stops one or both steppers. Bits 6 and 7 are used with this command. Bit 6 affects stepper 1, bit 7 stepper 2. If the bit is set, the stepper(s) will stop if it's not already stopped. The step count is unchanged by this command.

Example:

```
SF2=1: SF1=1: SUBC=6: COMMAND=6
PRINT-1,CHR$SF2*128)+(SF1*64)+(SUBC*8)+COMMAND);
```

When this command is received by the system, both steppers stop.

• **Command 6, Subcommand 7** (get steppers status). When this command is executed, the system sends a status word to the host PC that indicates whether the stepper(s) position is complete. Bits 6 and 7 aren't used with this command, which is used to determine whether a stepper has reached it's position. The status word retrieved by this command indicates if a stepper has exhausted its step count and stopped. The status word value retrieved equals 0 when both steppers exhaust their step counts (position complete) and stopped. The status word equals 1 when stepper 2 has reached its position complete and stepper 1 has not. The status word equals 2 when stepper 1 reaches position complete and stepper 2 hasn't. The status word equals 3 when both steppers haven't reached position complete.

Example:

```
SUBC=7: COMMAND=6
PRINT-1,CHR$((SUBC*8)+COMMAND);
I=ASC(INPUT$(1,-1))
```

When this command is executed by the host PC, variable "I" equals the status word.

It's quite possible that communications may get out of synchronization during system operation. Bad connections in the system, power upsets and electrical noise are all potential causes for a failure. The system can be re-synchronized if the host PC sends four or more null (0) characters in sequence. The control system software for the embedded controller circuit keeps constant watch for a sequence of four null characters. When a sequence of four nulls is detected, the first character received that isn't a null is interpreted as a command.

Darlington-array IC. More relays could be used if their coil resistances were greater.

The power source used for the LED in circuit (A) and relay in Circuit (B) should have its ground connected to the ground connections on connector *J3*.

Circuit (C) illustrates use of a solid-state ac relay driven by a digital output from connector *J2*. Such relays can control high-power ac appliances with good isolation between control circuit and device being controlled. Circuits (D) through (F) are analog input interfacing examples. Circuit (D) can be used to monitor the mechanical position of a moving machine part. A potentiometer could be mechanically coupled to a machine shaft, or a knob could be affixed to it for operator control.

Circuit (E) illustrates how a potential that exceeds +5 volts can be monitored. If the resistances of *R1* and *R2* in this circuit are equal, a voltage that varies from 0 to 10 volts

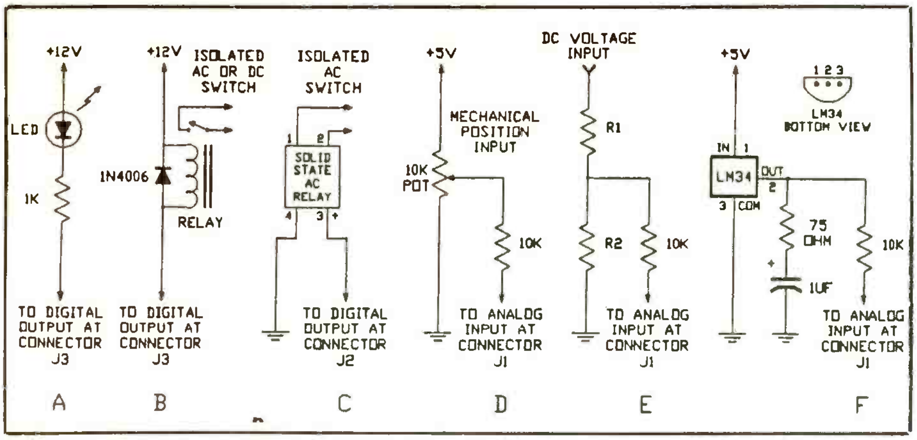


Fig. 6. Details of some simple hardware interfacing circuits.

can be monitored. Select the ratio between *R1* and *R2* so that the maximum input voltage is divided to produce 5 volts at the junction of of the two resistors. The 10,000-ohm resistors in series with the analog inputs help prevent damage to the ADC0817 inputs if a potential greater than 5 volts is applied to them. Do *not* intentionally allow the input voltage on any of the analog inputs to exceed +5 volts with respect to ground.

Circuit (F) illustrates use of an LM34 for temperature measurement. This sensor comes in various temperature ranges and can be purchased from Digi-Key Corp., among others. The LM34DZ has a range of +32°to +212° F and costs about $2. The 75-ohm resistor and 1-μF capacitor on pin 2 of the LM34 filter electrical noise from the output.

The simplest digital input devices are switch contacts connected to the digital I/O lines of connector *J2* to ground. These 16 digital I/O lines have pull-up resistors. You can also install optional pull-up resistors *RN4* and *RN5* for the analog inputs at connector *J1* and use switch contacts on these inputs also if you wish.

If you're interested in developing your own 8051 assembly-language applications software for this project, a valuable tool is *8-bit Embedded Controller Handbook* available from Intel Literature Sales (P.O. Box 58130, Santa Clara, CA 95052-8130; tel.: 1-800-548-4725). There's also a diskette available (see the Note at the end of the Parts List) that has the fully commented 8031 assembly source code for the control system described here. Many of the routines on this disk can be adapted to other applications. The disk also contains a binary file if you wish to program your own 2764 EPROM for the control system.

Many available cross assemblers will allow you to develop programs on an IBM PC or compatible. The 8051 cross assembler used to develop this project was purchased from Comtronics, Inc. (1447 Parker Rd., Conyers, GA, 30207; telephone: 404-922-0603).

Letters *(from page 3)*

as mentioned in the software review, makes DR DOS the outstanding choice as an operating system. Incidentally, DR DOS' editor is a stand-alone program. MS-DOS' editor is a QBASIC program (this explains its fancy pull-down menus). MS-DOS users have found this out the hard way by deleting QBASIC and then trying to run the ASCII editor. If you use *Windows*, you're better off with Notepad or Write. If you use MS-DOS, you're better off using *Norton Commander*. The question of whose built-in editor is better isn't a serious one and shouldn't enter into a discussion of which OS platform is better.

Alfred B. Rodney, Pres.
Advanced Research Communications
Brooklyn, NY

Thanks for your insights. There are certainly lots of people who share your enthusiasm since DR DOS's first-quarter fiscal '92 exceeded a record $200-million. Novell's Desktop Systems Group (formerly Digital Research, Inc.) just sent us a DR DOS 6.0 update for compatibility with Windows *3.1. Terrific automatic, free service for registered owners. Users can also download the driver from CompuServe (GO DRFORUM) or by calling the company's BBS at 408-649-3443.*

The company points out that its disk cache is far superior to MS-DOS's Smart-Drive and that both Windows *and* Windows applications run faster with it.—Ed.

Another Power Saver

• I really enjoyed Jan Axelson's excellent article in the January issue of *ComputerCraft* on "Power Saving Tips." One trick I've used that Jan didn't mention is using high-brightness LEDs at reduced drive current. At a normal LED load of 20 mA, these LEDs are extremely bright. But what many people may not realize is that at a reduced load of only 3 mA, these LEDs are as bright as a normal LED. Thus, you can get standard LED brightness at a small fraction of the current drain.

Jake Mendelssohn
W. Hartford, CT

fertile males and females? How much should the ratio of nursers to workers be changed? What is the colony's overall health? So many decisions, but they're all important if you want to win.

A *SimAnt* game can last a long time if you play a regular full game. Although players can sit for as long they wish, it still takes much real time to work out everything. Just defeating the red colony might take several hours of human time. The time goes quickly, though. *SimAnt* is so much fun and so easy to operate that one gets easily and thoroughly involved.

Documentation for *SimAnt* is very good. It's easy to understand and refuses to get overly technical. Most of the manual is like this. The technical stuff is saved for the manual's technical section, which is interesting reading for those who want the real story about ants. But you don't have to know anything about real ants to play the game. The game offers an unusual look at strategy, but it's fun and educational.

Social Strategy

Civilization is the name of a new game from MicroProse. Players have the opportunity to rule an entire civilization through many generations. Make the decisions and guide your people from ancient society of 6,000 years ago to space travel and planetary colonization.

Perhaps the most difficult part of acting as ruler is to make wise decisions. Some decisions that appear wise at the moment turn out to be disastrous in the long run. These are the kinds of decisions you make in *Civilization*. How will a specific decision today affect things years from now?

Game players begin lording over a prehistoric wandering tribe. The immediate task is to build one city, organize it and make it prosper. The people farm the outlying land and bring in food, forming the basis for survival. In time, the city grows and expands.

Other cities arise. Eventually, trade and commerce begin. Resources and trade beget better materials, perhaps for schools or weapons of war. As things get more complex, better technology is needed. So are armies and diplomats. *Civilization* evolves, and you determine the speed, direction and quality of its evolution.

Clearly, the most outstanding feature of *Civilization* is its strong educational content. Playing out a scenario teaches the complexities and compromises involved with organizing and directing a group of people. Making it more difficult are the threats of starvation, social unrest, oppression from other groups of people and even outright war. After playing this game for a short while, one can begin to know that playing the ruler of a country is a lot of work.

MicroProse has done its usual good job on game design, player interface and documentation. Granted, the concept presented by this game is complex. But handled so well are the complexities that the player doesn't become overwhelmed. Taking game play a little at a time creates an atmosphere of intelligent involvement and a sense of purpose. Successes are thrilling, and it's satisfying to see your civilization make significant peaceful strides. Can you build an empire to stand the test of time? Now you have the chance.

Military Strategy

Military strategy is one of the more common themes for computer games. Although not often regarded as military in nature, chess is the quintessential military game. "Chess, like music, like love, has the power to make people happy" is quote taken from the user manual for *Chessmaster 3000*. It's from Dr. Siegbert Tarrasch, a noted Chessmaster.

Chessmasters do have ratings like 3000 but I don't know if this chess game from The Software Toolworks has earned its rating. *Chessmaster 3000* is an upgrade from *Chessmaster 2100* that set up its pieces in the computer game market some years ago. The two games are much alike in appearance and operation, save for the fact my evaluation version of *Chessmaster 3000* is written for Microsoft *Windows*.

Features of *Chessmaster 3000* include various board layouts and

Bird's Eye View

SimAnt, $59.95
Maxis
2 Theatre Sq., Ste. 230
Orinda, CA 94563-3346
Tel.: 510-254-9700

Requirements

Memory	640K
Graphics	VGA, MCGA, EGA, Hercules, Tandy
Sound	Ad Lib, Sound Blaster, Sound Master, Roland, Tandy
Controllers	Keyboard, mouse

Evaluation

Documentation	Excellent
Graphics	Good
Learning Curve	Short
Complexity	Medium
Playability	Excellent

In Brief: Unusual strategy game of ant life. Educational and fun.

CIRCLE NO. 165 ON FREE INFORMATION CARD

Bird's Eye View

Civilization, $69.95
MicroProse
180 Lakefront Dr.
Hunt Valley, MD 21030
Tel:. 301-771-1151

Requirements

Memory	640K
Graphics	VGA, MCGA, EGA, Tandy
Sound	IBM, Roland, Ad Lib, Sound Blaster, Tandy
Controllers	Keyboard, mouse

Evaluation

Documentation	**Excellent**
Graphics	Good
Learning Curve	Long
Complexity	High
Playability	Good

In Brief: Educational and enthralling social strategy. Teaches decision-making and planning.

CIRCLE NO. 166 ON FREE INFORMATION CARD

Bird's Eye View

Chessmaster 3000 (Windows), $59.95
The Software Toolworks
60 Leveroni Ct.
Novato, CA 94949
Tel:. 415-883-3000

Requirements

Memory	2M of RAM, Microsoft Windows
Graphics	Windows-dependent
Sound	Computer speaker
Controllers	Keyboard, mouse

Evaluation

Documentation	Fair
Graphics	Windows-dependent
Learning Curve	Short
Complexity	Easy
Playability	Good

In Brief: Upgrade to a long-standing chess game. New features and improved play. Mouse and hard disk required.

CIRCLE NO. 168 ON FREE INFORMATION CARD

views. The 3D view is always interesting, if a somewhat difficult stance from which to play. It seems that no matter how good the picture, it never mimics the real thing. The flat top-down view is more suited to serious play.

A game feature new to *Chessmaster* is its support for ASCII files. Chess databases are fairly abundant, if one knows where to look. An ASCII move list can encode an entire game in a simple format that doesn't take much to import into a chess game. *Chessmaster 3000* supports ASCII move lists and can import or export them. This exchange feature makes it possible to share information between different kinds of chess programs.

Another feature of the game is good for players who may feel intimidated by a computer opponent. The computer can be handicapped by using any number of functions. When challenging *Chessmaster* to a game, you can make the computer behave more humanly by having it miss key moves, ignore its library of moves or have it avoid piece exchanges in optional situations. Furthermore, response controls can place a limit on how long *Chessmaster* thinks before making a move and how much depth it uses when looking ahead. Use of such tools can balance the scales for the novice until he learns more about chess strategy.

For experienced players, especially those looking for a real challenge, try playing blindfold chess. This means that you imagine the board layout and play from memory. If you get lost, you can activate the pieces for a look-see and then try again. It's a good way to develop chess skills and sharpen your memory.

Other features of *Chessmaster 3000* include its analysis mode, chess rating system and teaching mode that aids new players in understanding rules and moves. Since I'm not a chessmaster and not on my way to being such, it's bewildering to evaluate a chess game that may or may not truly rate 3000. But I have friends who are better than I am and they gave *Chessmaster 3000* a tough time. They aren't chess masters, either. I suppose it doesn't much matter. Like Tarrasch said, "chess has the power to make people happy." These words describe the chessmaster when playing and the ordinary player when winning.

ComputerCraft Mart

Classified Commercial Rates: $1 per word, 15-word minimum ($15.00) prepaid. (Word count includes name and address; ZIP code and abbreviation each count as one word; P.O. Box number and telephone number count as two words each.) Indicate free category heading (**Computers, Software, Electronics, Video** or **Miscellaneous**). A special heading is available for a $6 surcharge. First word only is set boldface caps at no charge. Add 25¢ for each additional boldface word.

Non-Commercial Rates: FREE to subscribers, as space permits; maximum 15 words. A recent **COMPUTERCRAFT** mailing label must accompany ad.

(All advertisers with P.O. Box addresses must supply permanent address and telephone number. Copy is subject to publisher approval.)

Mailing Information: Copy must be received by the publisher by the 25th of the third month preceding the cover date. Send Advertising material with check or money order or credit card information (Visa or MasterCard only) with number and expiration date to: **COMPUTERCRAFT**, Classified Department, 76 N. Broadway, Hicksville, NY 11801.

COMPUTERS

IBM PC AND LAPTOPS VIDEO DIGITIZER, 640 BY 480 RESOLUTION. 256 GRAY LEVELS $89.98. DEMO DISK $3, INFORMATION $1. COLORBURST, BOX 3091 NASHUA, NH 03061. PHONE (603) 891-1588.

MEASUREMENT-Microprocessor kits. Catalog $1.00. Temperature Datalogger plans, $5.00. TENSOR, Box 2543-C, Thunder Bay, Ontario, P7B 5G1.

COMPUFACTS The professional's computer troubleshooting manual. Nothing like it on the market today. "Tricks of the Trade" from working technicians. Thoroughly illustrated. Electronic students MUST read this before job hunting. BBS access included. 30 day guarantee. $39.95 Computers Etcetera, PO Box 8090, Berkeley, CA 94707-9991. 510-841-5201.

BARCODES BARCODES BARCODES they are everywhere you look! Now you can get in on an industry that is growing by 20% each year with **THE BARCODE COURSE.** This course comes complete with a detailed barcode textbook, portable barcode reader, RS232 interface and barcode printing software for only $149.95 + $7.00 S&H. OBR Inc., RR3 Box 47-7, Carbondale, IL 62901. (618) 549-7100.

16 bit 2H/2F AT controller, 1:1, MFM. $30 ppd. Dan, (301) 498-8915 eves.

80C52-BASIC microcontroller board. BASIC interpreter, 32K RAM, 16K EPROM, EPROM programmer, RS232, expansion connector. Bare board with manual, schematics $19.95 +$3 P&H. 80C52-BASIC microprocessor chip $24.95 +$1 P&H. Assembled and tested $119.95 +$5 P&H. MD residents include sales tax. SASE for info. Prologic Designs, P.O. Box 19026, Baltimore, MD 21204.

SOFTWARE

FREE IBM PD and Shareware disk catalog. Low prices since 1988! ASP Approved Vendor. VISA/MC Accepted. Finto Software, Dept CC, Route 2 Box 44, Rosebud, TX 76570.

FREE PROGRAM!! FREE DISKS!! Save $15.00 & more! Most shareware and other programs require an additional investment in viewing, or decompression utilities before running. You get 4 of the newest and most widely used picture and decompression programs available. All on 360K disk! Free!! Plus, if you mention CC Special, we will send you another 360K disk chock full of our huge rental, adult, & shareware listings absolutely free!! Just send $4.00 for shipping (refundable) or $4.50 (refundable) for 720K disk. **BEACH RADIO**, 1148 Center Street #269, Newton Center, MA 02159, Dept CC.

MINI-LANGUAGE demonstration language for beginners. Repeat until, while wend, if then, case input subscripts. Program and variable table displayed. IBM PC 3½" $29.95. Robert Ryan, 177 Idaho Rd., Austintown, Ohio, 44915.

QUALITY SHAREWARE from PUBLIC DOMAIN USERS GROUP. Huge selection for IBM, Commodore. Reasonably Priced. Great rental plan too! For catalog send SASE to: Kathleen Lewis, c/o PDUG, Box 720396-A, San Diego, CA 92172-0396.

FREE PRINTED 122 PAGE CATALOG 3200+ IBM Prgms. ASP Vendor Sunshine Software, 6492 South St., Suite 470, Lakewood, CA 90713.

2400+IBM SHAREWARE PROGRAMS. Free disk catalog. Specify disk size. GSEL ShareWare, 3870 LaSierra Ave., #134C, RiverSide, CA, 92505.

ELECTRONICS

VIDEOCIPHER II/Satellite/Scanner/Cable/Amateur/Cellular/Repair Manuals, Modification Books & Software. Catalog - $3.00. **TELECODE** P.O. Box 6426-CC, YUMA, AZ 85366-6426.

KENWOOD & ICOM Service Bulletins. 175+ pages covering all models. $39.95 Catalog - $3.00. **CODs** 602-782-2316/ **FAX** 602-343-2141. **TELECODE** Box 6426-CC, Yuma, AZ 85366-6426.

CABLE TEST CHIPS. Scientific-Atlanta 8550-321, 8500-310, 311, 320, 321 (specify) - $33.95. 8580-338 - $69.95, 8570/8590 - $79.95. ZENITH ZF-1 - $33.95. STARCOM 6 - $33.95. STARCOM 7 - $49.95. TOCOM 5503/07 VIP - $33.95. **TELECODE** P.O. BOX 6426-CC, YUMA, AZ 85366-6426.

LOW COST LOGIC ANALYZER - A must for companies and serious hobbyists: 8 digital inputs, sampling rate from 50 nsec to 1 msec. Connects to PC or compatible via RS-232. Only $250.00 US. For info: **OPENTEK**, P.O. Box 71, Sorel, Quebec, Canada, J3P 2T6.

FREE! Catalog of fantastic electronic kits. Write to LNS TECHNOLOGIES, 20993 Foothill Blvd., Suite 307C, Hayward, CA 94541-1511.

PCB LAYOUT, electronics, & THOUSANDS OF IBM SHAREWARE PROGRAMS. $3.00 S&H for 2 disks full plus diskalog. American Software, P.O. Box 509, Suite M8, Roseville, MI 48066-0509.

KITS, sound, light and high tech. Send for free catalog. **JM ELECTRONICS,** Box 150454, Dept. C, Altamonte Springs, FL 32715-0454.

CIRCUIT boards etched and drilled for as low as .70 cents per square inch. **JM ELECTRONICS,** Dept. C, Box 150454, Altamonte Springs, FL 32715-0454 or (407) 767-8196.

VIDEO

VHS-VCR Repair Solutions Sets I, II, III, IV, V, VI, VII. Each contains 150 symptoms and cures, updated cross reference chart, free assistance, $11.95 each all seven $69.95. Schematics available. VISA/MC. Eagle Electronics Box A, 52053 Locks Lane, Granger, IN 46530.

TV NOTCH FILTERS PHONE RECORDING EQUIPMENT, BROCHURE $1.00. MICRO THinc. BOX 63/6025, MARGATE, FL 33063. 1-305-752-9202.

CABLE

ACTIVATE your Jerrold, Tocom, or Zenith Box (Fully) in just minutes. Test chips only $50.00. No CA. Sales. Orders Call 1-800-452-7090. Information 310-867-0081.

JERROLD Test chips for Starcom 6 and 7. Installs in just minutes only $50.00. No CA. Sales. Orders 1-800-452-7090. Information 310-867-0081.

BUSINESS OPPORTUNITIES

REACH 33 MILLION customers in 45 states, 4,000 newspapers, under $1 per paper some states, (708) 971-4625 24 hours.

MISCELLANEOUS

MORSE Code? No Problem. You can increase your speed, no matter how many times you've failed before. **Results guaranteed** when you follow the instructions. PASS Publishing's **CW Mental-Block Buster** program helps you **explode mental blocks** that hold you back. Based on 40 years of research, the **CW Mental-Block Buster** uses guided meditation, dynamic visualizations, and powerful affirmations to blast through mental blocks. **You can do code!** That means new bands, more contacts, more fun! (This is not a CW practice tape.) The **CW Mental-Block Buster** audio cassette and practice booklet are only $24.95 ppd. in the US (NY residents add $2.00 sales tax). Info: 516-584-8164. PASS Publishing, P.O. Box 570, Stony Brook, NY 11780.

ASIAN LADIES want correspondence for friendship, marriage. SUNSHINE INTERNATIONAL-Dept. TW, Box 5500, Kailua-Kona, Hawaii 96745. (808) 325-7707.

ATTRACTIVE ORIENTAL LADIES seeking correspondence, marriage. Professional introductions since 1984. Asian Experience, Box 1214TZ, Novato, CA 94948. (415) 897-2742.

$975.00 Daily possible with PC? YES! From home, easy, low cost! Free report! C.N.E.C., 153 Princeton Ave., Dover, NJ 07801.

ESP, BIOELECTRICITY, MICROWAVE HEARING. Introduction, comprehensive booklist . . . $1.00. RESEARCH ASSOCIATES, P.O. Box 3583, Boynton Beach, FL 33424-3583.

THAILAND-Marriageable women waiting! Photograph brochure $3. Video $25. Tours. "Thai" Box 4417(m) Miami, FL 33114.

WE'LL PAY YOU TO TYPE NAMES AND ADDRESSES FROM HOME. $500.00 PER 1000. Call 1-900-896-1666 ($1.49 min/18yrs.+) or Write: PASSE-XDG892, 161 S. Lincolnway, N. Aurora, IL 60542.

LOVELY Pacific Island ladies seek correspondence! FREE color photo brochure! P.I.C. P.O. Box 461873 L.A. CA 90046. (213) 650-1994.

PC-LOTTOpro 3.00, complete lottoprocessing package from wheeling to Bet Slip Printing (MS-DOS + Hard Drive), BBS support, 50 databases, KENO games, $99 Visa/MC, 604-479-8536, FAX 479-8227, BBS 479-7189-Lottoware, 870 Gladiola Ave., Victoria, B.C. Canada, V8Z 2T6, $65 without Bet Slip Printing.

IDEAS WANTED

INVENTORS! Can you patent and profit from your idea? Call AMERICAN INVENTORS CORP. for free information. Serving inventors since 1975. 1-800-338-5656.

Advertisers' Index

We'd like to see your company listed here too. Contact Margaret Milanese at 516-681-2922 to work out an advertising program tailored to suit your needs.

Variations in Strategy

According to one dictionary, strategy is the art of employing plans toward a goal. A goal is achieved by some orderly execution of strategy. Orderly execution or movement is called tactics. Thus, we have a relationship between strategy and tactics, a common theme of computer games. With this in mind, let's look at some rather exotic implementations of computer-game strategy.

Insect Strategy

Some people hate ants; others love them. Many of us fit somewhere between these two extremes. No matter what your disposition concerning ants, you'll probably like *SimAnt*, a new insect simulation from Maxis.

The life of an ant is simple, busy but simple. In *SimAnt*, you're born a black ant—actually yellow, but that's just so you can tell yourself apart from the other black ants. At any rate, you're the leader of the black-ant colony. Your goals are simple: lead your ants to food, defend your nest and queen, destroy the opposing red-ant colony. Fulfilling these tasks requires stamina, lots of food, endurance and dying several times. To an ant, dying is nothing as long as the colony survives. You as the yellow-ant and leader, though, are reborn to continue playing the game.

SimAnt is a strategy game of the highest order. Yet, it's simple to control. Since a top priority is foraging for food, this takes up much attention. Besides this, you have to keep watch on how fast the colony is growing. Too many mandibles to feed can exhaust the food supply, which can kill your colony. This is really bad. Fortunately, finding food isn't a difficult thing since ants eat almost anything.

One source of food, the spider, is plentiful and dangerous at the same time. Wandering spiders don't mind chomping on ant flesh. The crunching noise can be heard halfway across the backyard. When spiders become a nuisance, you rally soldier ants to your side and go for the attack. A spider can decimate a couple of bands of soldiers before putting to flight or being killed. If killed, a spider is worth its weight in food.

Sometimes the red ants spy your food and lay a scent trail for it. A battle usually ensues, and the victors get the food. At other times, food isn't so costly to obtain. A meandering caterpillar moves slowly and doesn't fight back. A few crack troops can take down the caterpillar and hack it up in short order.

Food fights can be fun and can keep a good ant away from the nest for too long. The nest gets crowded and you have go back and dig more tunnels for expansion. Worker ants then have more space for food storage.

There comes a time in the life of every colony when it needs to expand into another location. A click of your computer mouse brings up a strategic view of the backyard. The red-ant colony seems to be thriving. Maybe a carefully planned guerrilla strike would slow them down. A human is mowing the lawn with a machine that can kill the strongest ant without warning. Caution is advised. Then there's the human's house, the real prize. Driving out the two-legs would mean an unlimited supply of food.

As colony leader, you make plans. When is the best time to breed

(continued on page 84)

Intelligence arises on primal earth (Civilization).

Tactical overview of backyard patch (SimAnt).

Traditional chess with a different look (Chessmaster 3000).

DON'T JUST BUY AMERICAN... BUILD AMERICAN!

Our CYANCE LAB kits let you learn while you build!

We learn by doing. Often, building a project from a kit can give us the head start we need in a new area of interest. The entire line of CYANCE LAB kits is based on our desire to help you construct high technology products at home. If America is to maintain her leading-edge advantage in high technology science, it will be up to individuals like you. As a technologist, your role in re-inventing America will rely almost entirely upon your personal knowledge; the "under your finger-nails" knowledge that can only be had by doing! Enhance your "hands-on" education and ability by building projects that interest you. Join the many who are, quite literally, taking America's future into their own hands... Build a CYANCE LAB kit today!

U.S. CYBERLAB, INC.

CALL FOR NEW 1992 CATALOG! (501) 839-8293, RT 2 Box 284, Cyber Rd., West Fork, AR 72774

* SUPER EXPEDITE SERVICE NOW AVAILABLE ON DEMAND! One and two day kit delivery service!

Watch for progress report from Nick Goss on our Cold Fusion Cell Kits!

CIRCLE NO. 82 ON FREE INFORMATION CARD