VOL. 1 No. 7 November 1984

COMPUTER DIGEST

NEW KIND OF MAGAZINE FOR ELECTRONICS PROFESSIONALS

SUPER-HIGH DENSITY FLOPPY DISK DRIVES...

How 2.5 megabytes gets stored on a 5¼-inch floppy disk.

RS-232 INCOMPATIBILITY...

Find out why your computer won't talk to your serial printer.

COMPUTER-AIDED DESIGN...

Put your computer to work designing audio networks.

A GERNSBACK PUBLICATION

BUILD YOUR OWN ROBOT!

Send today for your 52-page (8½ × 11") booklet containing complete reprints of all eleven articles in the Build Your Own Robot series by Jim Gupton.

This all-inclusive reprint gives you all the data you need to build your own Robot.

■ TELLS EVERYTHING YOU NEED TO KNOW to build the Unicorn-1 Robot without the need for an engineering degree or special equipment.

The robot is fully mobile with minipulator arms to grasp, lift and carry.

■ MANIPULATOR ARMS and end-effectors (hands) are what enable the robot to perform useful tasks. Details of construction techniques and considerations are fully explored.

■ MOBILITY BASE is not a lunar space station. It is the drive system that permits the robot to move from here to there. Full construction details along with a discussion of power sources is included.

■ THE BODY—FRAME AND ROTATION MECHANISM. This is the part that makes Unicorn-1 look like a robot. Wood and Formica are the materials for the body. Motors and gears are what make it function.

■ COMMUNICATIONS. How you can tell your robot what to do. Preprogramming techniques....radio control ....computer control are all detailed.

■ SENSORS. How to add sensors so your robot doesn't bump into things.

Radio-Electronics. Robot Reprints
200 Park Ave. South
New York, N.Y. 10003

RE11

I want to order ______ reprints @$12.00 plus $1.00 postage and handling for U.S., Canada and Mexico. Add 96¢ sales tax for New York State residents only. U.S. Funds only.

I want to order ______ reprints @$12.00 plus $3.00 Air Postage and handling for all other countries. U.S. Funds only.

Allow 6-8 weeks for delivery.

Please print

____
(Name)

____
(Street address)

____
(City) (State) (Zip)

We do not bill, check must be enclosed.

CONTENTS

Vol. 1 No. 7 November 1984

ON THE COVER

The evoluation of the floppy disk is fascinating. We've gone from single-sided, single density to two-sided, double density, and now RANA introduces what appears to be the ultimate... A disk that can cram 2.5 megabytes of information into the same amount of space. To find out how they do it; **see page 5.**

ComputerDigest is published monthly as an insert in Radio-Electronics magazine by Gernsback Publications, Inc., 200 Park Avenue South, New York, NY 10003. Second-Class Postage Paid at New York, N.Y. and additional mailing offices. Printed in U.S.A.

COMPUTER DIGEST

Hugo Gernsback (1884-1967) founder
M. Harvey Gernsback, editor-in-chief
Larry Steckler, CET, publisher
Art Kleiman, editorial director
Byron G. Wels, editor
Brian C. Fenton, technical editor
Carl Laron, associate editor
Robert A. Young, assistant editor
Ruby M. Yee, production manager
Robert A. W. Lowndes, production associate
Dianne Osias, production assistant
Karen Tucker, production assistant
Jacqueline Weaver, circulation director
Arline R. Fishman, advertising coordinator

Gernsback Publications, Inc.
200 Park Ave. South
New York, NY 10003
Chairman of the Board: M. Harvey Gernsback
President: Larry Steckler

ADVERTISING SALES 212-777-6400
Larry Steckler
Publisher

EAST/SOUTHEAST
Stanley Levitan
Radio-Electronics
200 Park Ave. South
New York, NY 10003
212-777-6400

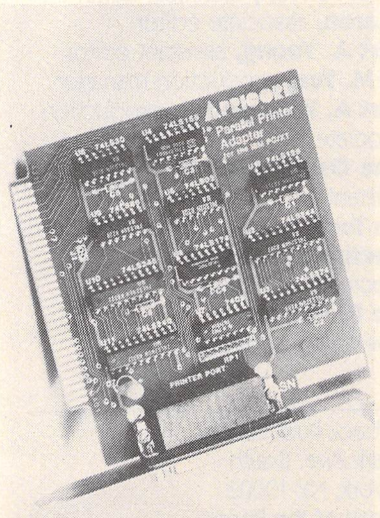
MIDWEST/Texas/Arkansas/Okla.
Ralph Bergen
Radio-Electronics
540 Frontage Road—Suite 325
Northfield, Illinois 60093
312-446-1444

PACIFIC COAST
Mountain States
Marvin Green
Radio-Electronics
15335 Morrison St., Suite 227,
Sherman Oaks, CA 91403
818-986-2001

COMPUTER PRODUCTS

For more details use the free information card inside the back cover

PLUG-IN INTERFACE ADAPTERS, the Apricorn *Parallel Interface Adapter* (shown in photo) and the Apricorn *Asynchronous Communications Adapter* are compatible with the IBM *PC/XT* and work with all operating



CIRCLE 111 ON FREE INFORMATION CARD

systems and software packages available for the IBM-PC and IBM-XT. Each card is fully burned-in and tested, and is backed by a lifetime warranty.

The *Parallel Interface Adapter* creates a parallel I/O interface accessible through a connector at the rear panel of the computer system, providing control of numerous printers with parallel (or Centronics) type interfaces.

The *Asynchronous Communications Adapter* provides a fully programmable RS-232 serial interface for communication with a variety of devices, including printers, modems, terminals, other computers, and plotters.

Both boards carry a suggested list price of $119.00.—**Apricorn,** 7050 Convoy Court, San Diego, CA 92111.

MULTI-CHANNEL SWITCH/SPOOLER, model *SX-60* serves up to four computers and feeds two printers simultaneously. Any of the four input computers can direct its outputs to either of the printers. All switching is under software control.

Spooler buffer sizes of 8, 32, or 64K store from 2.5 to 20 pages of dense, single-spaced text. For busy installations, an expansion board can increase the spooler's buffer capacity to 384K, 120 pages. Standard configuration is four computers in, two printers out, or two printers in, four printers out. Using jumpers, the model *SX-60* may be configured for 5 in, 1 out; 3 in, 3 out; or 1 in, 5 out.

The model *SX-60* will accept serial data on each channel from 150 to 19,200 baud with each computer at a different baud rate. Printers also may be set from 150 to 19,200 baud with each printer at a different baud rate—for example: a letter-quality printer at 2400 baud and a dot-matrix printer at 1200 baud. A variety of serial-serial

CIRCLE 112 ON FREE INFORMATION CARD

and parallel-serial combinations are available. All conections are made through 25-pin D connectors at the rear. A copy function is provided for multiple copies of documents.

Prices for the model *SX-60* start at $469.00.—**Kearsarge Industries, Inc.,** Reston, VA 22091.

NEW HIGH DENSITY DISC DRIVE

Here's more storage power for your computer.

MARC STERN

■At last, a floppy-disk-drive manufacturer has broken the 2-megabyte barrier. Rana Systems has developed a floppy-disk-drive that interfaces with the IBM *Personal Computer* and IBM-*PC XT*. Called the Rana *2.5 Super Floppy Disk System*, this super high-density storage device is capable of storing 2.5-megabytes of formatted data (3.3 megabytes unformatted) on a single 5-1/4-inch floppy diskette.

The *Super Floppy*, which, according to Rana has been in development for 2½ years, takes advantage of microprocessor control and LSI (*Large-Scale-Integration*) techniques. The head-movement commands are under the control of a closed-loop servo system.

The Rana drive

Rana developed their *Super Floppy* system in conjunction with Drivetec. The read-write heads—there are two—are made of manganese zinc ferrite and are mounted on a carriage which is located on precision guide rods that insure accurate radial motion. Head positioning is accomplished on a closed-loop basis, using servo data recorded between each sector of the (pre-formatted) diskette.

A special disk-controller card, which fits into one of the IBM-PC's expansion slots, is required. It communicates with the microprocessor via the IBM system bus and replaces the standard IBM disk-

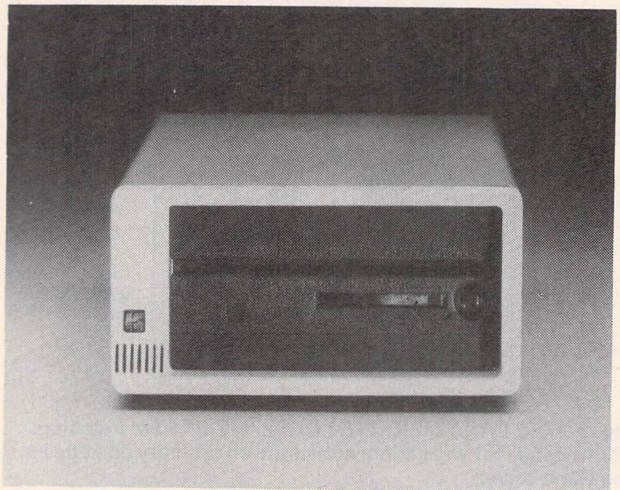


Fig. 1—THE RANA 2.5 SUPER FLOPPY-DISK SYSTEM is capable of storing 2.5 megabytes on a 5¼ inch floppy diskette. It ushers in a new era in high-density storage devices.

controller cards. This card will support up to four drives. For instance, a PC user can have two standard, 360K floppy-disk drives installed and 2 Rana *Super Floppy* drives.

The drive itself uses two stepper motors, which work in concert with the drive's controller circuitry, to correctly position the read-write heads. The first motor is a coarse stepper and finds the particular track—there are 160 per side—while the second motor is a fine stepper which brings the read-write head to the correct sector. It must maintain an accuracy of 200-microinches.

The direct DC drive motor is on continuously. The acquisition time for data loading is 35 milliseconds; track-to-track access time is 0.3 ms. The direct-DC drive motors "come up" much more quickly than AC units.

A brief walkthrough

A standalone unit, the *Super Floppy*, is a good example of the super high-density units now appearing

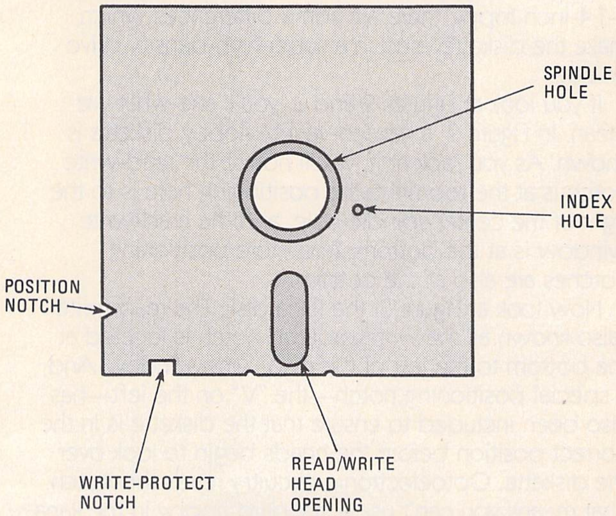


Fig. 2—AN ORDINARY DISKETTE shows the notching that we are all familiar with. Compare this with Fig. 3.

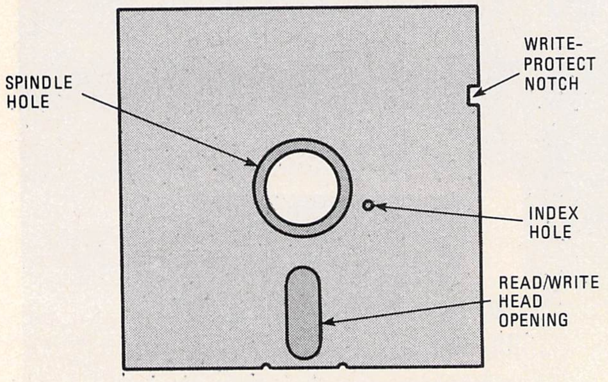


Fig. 3—THE RANA DISKETTE has entirely different notching. This means that you cannot substitute an ordinary diskette in the Rana drive.

on the market. When the power is turned on, the drive remains inactive until a diskette is inserted and a special latch at the front is closed, at which time, the drive's center spindle comes up and centers the diskette. A cone drops as the motor comes on and runs continuously at 360 rpm (the standard 8-inch floppy-disk rate). Optoelectronic circuitry reads a special notch in the diskette to ensure that it has been positioned properly (See Figure 3) and the drive further checks other position-sensing circuitry at the spindle to ensure that the diskette remains centered. A user can't remove the diskette because the latch locks the diskette into place. The latch must be opened before the diskette can be removed.

As the diskette spins, the read-write heads, which are controlled by intelligent LSI circuitry, look for Track 0 and look for bad sectors on the diskette. If the diskette is bad, it indicates that to the system board, which, in turn, gives the user an error message.

Diskette is the key

Besides the special LSI positioning circuitry and dual-stepper motors, the real key to the super high-density disk drive is the diskette itself. Although it is a 5¼-inch minifloppy and looks much like any other 5-1.4-inch foppy, there are some differences which make the diskette used in a super-high-density drive unique.

If you look at Figures 2 and 3, you'll see what we mean. In Figure 2, a garden-variety floppy diskette is shown. As you look at it, you'll notice the read-write notch is at the top right; the positioning hole is to the right of the center spindle hole, and the read-write window is at the bottom. Two more positioning notches are also at the bottom.

Now look at Figure 3, the Rana disk. The read-write (also known as the write-protect) notch is located at the bottom to the left of the read-write window. And, a special positioning notch—the "V" on the left—has also been included to ensure that the diskette is in the correct position before the heads begin to look over the diskette. Optoelectronic circuitry reads this notch. That means you can't use a standard floppy in the Rana drive. Instead, you must use one supplied by Rana, which costs about $15, at the moment. (Rana officials expect the price to drop to about $8, when more super high-density drives appear on the market.

In reality, these super high-density disks are quad-density storage diskettes. Using modified frequency modulation (MFM), as many as 160 tracks of information can be laid down per side. (Single-density disks use one recording head and 36 to 40 tracks. Double-density disks use one recording head, modified frequency modulation, and effectively double the amount of information capability, and quad-density disks use both sides and double the number of tracks per side. Double-sided, double-density disks use both sides, but have only 40 tracks per side, while quad-density disks have 80 tracks, effectively doubling the amount of information capability. The key to the ability to double the information capacity with recording technique relies on the type of modulation used. With standard FM techniques, a certain amount of timing and address data is stored along with the digital data. Each byte of data on the disk not only has digital information, but also timing and address information. With MFM, only digital and address information is stored, effectively freeing a great deal of space for further information storage. Timing is supplied by read-write head intelligence.)

Servo information

Equally as important as the disk media and recording technique is a method of storing positioning information on each super high-density disk. Rana calls this information the "servo" information and stores it in between each sector along a track. (Rana uses 17 sectors per track.)

In operation, the disk read-write head interacts with this servo information for precise head positioning. It reads the positioning data in each sector and finds the ones which hold the file of data or program the user asks for.

The servo information is preformatted on each diskette supplied by Rana and this is the reason you must buy the diskettes from them or their dealers. The servo information, which acts as header data for the read-write head, lies before the sector ID header on each track (See Fig. 4.)

In operation, the read data is decoded and the servo data is presented to two electronic devices. The first does a digital identification of the servo patterns, while the second compares the amplitude of the left and right servo signals and repositions the stepper motor toward the center line of the recorded data.

The read-write head constantly looks for servo data on the disk and constantly updates itself and repositions itself. Special circuitry also compensates for thermal and humidity-related expansion or contraction of the diskette and for other mechanical tolerances of the drive.

A little history

If you've been a personal computer user for more than a couple of years, you probably remember the first disk drive you purchased. Think about it for a couple of minutes. When you purchased that first disk drive for your first personal computer, it was probably a "state-of-the-art" storage device, capable of holding about 90K of data. It made no difference whether your system was an Apple, a CP/M machine, or one of the Radio Shack computers: The amount of storage was probably about the same, give or take a few thousand bytes.

It all seems an age-and-a-half ago, but it really wasn't. Single-sided, single-density disk drives were the standards of the industry as recently as three to four years ago. That type of disk drive—still used by many computer owners—had a storage capacity that was limited not only by its format, typically 36 to 40 tracks, but also by the method used to store data on the disk.

With a single-density disk drive, each range of frequencies that's recorded represents not only the data, but also certain timing and address information which enables the computer to find specific files or data on the disk. That manner of encoding data on the disk effectively limited the amount of data which could be stored to about 90,000 bytes of usable space.

However, new ways to cram more data onto storage disks were found. An example is modified frequency modulation or MFM. Using that technique, which strips away the necessity of using a constant clock pulse for timing information, the amount of data that could be stored on a diskette was effectively doubled.

Still designers weren't content and they saw a vast wasteland on the unused side of the disk. So they included a second read-write head opposite the already-existing read-write head and effectively doubled the disk capacity again.

However, the number of tracks remained the same. Each side of the disk had only 40 tracks, or 80 tracks total. The reason this number remained the same is because the technology didn't exist to increase the number of tracks and add even more storage. Like the car industry's downsizing program where the first gains—cutting weight—were easy because all they had to do was substitute materials, the early gains in mass storage were fairly easy because all that had to be modified was read-write technology. When it came to further gains,more-elaborate circuitry, some of which hadn't been developed in 1981, had to be created.

Until this time, there was a fair amount of leeway in read-write head positioning because the requirements weren't as critical as might be. Potentially, a 5¼-inch floppy disk could store far more than just 360K but any increase in the amount of storage couldn't be reached

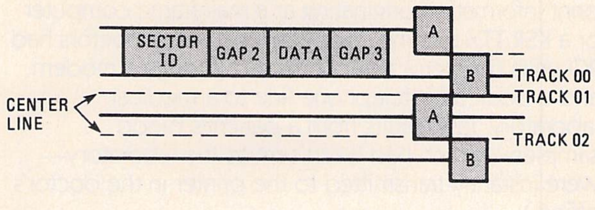


Fig. 4—SERVO INFORMATION acts as header data for the read-write head. It lies before the sector ID header on each track.

because the head positioning required was far too critical.

But, disk drive designers weren't about to let that stand in their way. They continued their work until they were able to lay down double or more the number of storage tracks on the disk. Thus, the 40-track disk became the 80 to 96-track disk and, as a result, the amount of information that could be stored increased by a factor of two and disks could store as much as 720K of data. These quad density disks were state-of-the-art until disk drive developers moved toward the 1 megabyte barrier.

Disk drives from such companies as Amlyn, Tandon and Rana use precise positioning servos and super-precise positioning circuitry to achieve information densities as high as 2.5 megabytes of storage, and have reached a new plateau in floppy disk storage technology.

What it means

The advent of the super high-density disk has several advantages for the microcomputer user. First it is a good way point in the seemingly inevitable trek to a hard disk and, second, when you reach the hard disk, it is a good backup device.

At one point or another in the life of every microcomputer user, there comes a day when ordinary floppy disk drives just aren't enough, for whatever reasons, whether business, professional or personal. Quite likely, this point is reached where several programs are used frequently and different data files are accessed in succession.

Using floppy disk drives, at this time, can become torture because you must constantly switch disks to change programs or data disks.

At this time, the small-computer user probably wishes he had a hard disk drive to accommodate at least his highly-used programs and files, especially because of the high-density storage potential and because of the speed of program or data access. Hard-disk drives spin at 3,600 rpm and information on them literally leaps onto the screen.

However, the user also probably can't justify a hard disk on the grounds of expense, so he must remain with a dual-drive system.

Super high-density disks solve this quandary because they offer prodigious amounts of storage and are less expensive than hard disk drives. So, this type of drive is a good way point in the move to the hard drive.

Unless the computer user is willing to invest in a costly streaming tape cartridge drive, then he must back up his hard disk with floppy diskettes. On the IBM *PC*, that will take about 28 standard diskettes and require the better part of an afternoon. With a super high-density disk, however, it takes only four diskettes to back up the hard disk and it's a much speedier proposition. The super high-density disk has a transfer rate of 500 kilobits per second.

So, from many viewpoints the super high-density disk makes sense. It's only a recent development, but it's likely the rest of the industry will soon follow suit.

PRINTER PROBLEMS II

Last month we examined parallel printers. Now serial printers come under the spotlight.

HERB FRIEDMAN

■More often than necessary, the best of intentions lead to even greater problems, frustrations, or outright disasters. The *RS-232C serial printer interface* is a classic example of how to make something that is inherently simple into something so complex that it is often beyond the ability of non-technical users.

How teletypewriters work

The original teletypewriters were mechanical marvels that used the interruptions in a closed circuit—a current loop—to exchange information between several machines or printers. As shown in Figure 1, the teletypewriter consisted of two completely independent devices—a keyboard and a printer—though both units were generally housed in the same cabinet. The complete teletypewriter, consisting of a keyboard and printer was called a KSR—for *Keyboard Send Receive*. If the printer was housed in its own cabinet, it was called an RO—for *Receive Only*.

The loop current—called the *mark* (standby) current—was provided by a power supply or batteries. Depressing any key on the keyboard caused the motorized keyboard machinery to produce an interruption to the current loop—called a *space* which all machines in the loop recognized as the "start" signal for the next character. The time period used for the space interruption to the current was called a "bit." (It had nothing to do with the binary code.) Following the space, the machinery produced a series of five bits consisting of marks and spaces which represented characters. Each character was represented by its own specific pattern of marks and spaces, which the printers recognized as the character to be printed.

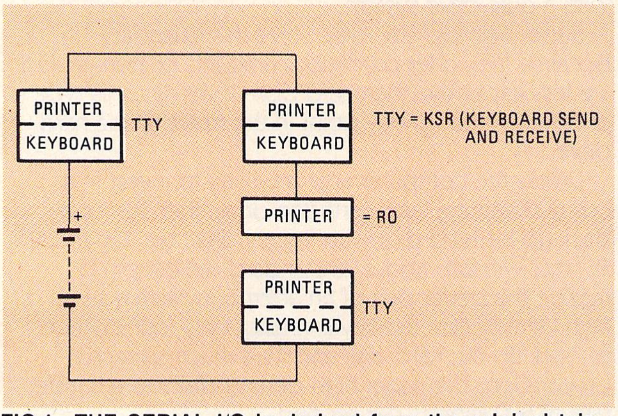


FIG.1—THE SERIAL I/O is derived from the original teletypewriter system, in which all elements were connected in series and responded to serial loop current interruptions that represented characters and control functions.

The bit timing is extremely critical. If the space and character bits weren't transmitted within a specific time period, the printing mechanisms got out of synchronization and produced random characters—"garbage." To insure that all machines started and ended together, after the five *character bits* were transmitted, the keyboard returned the current loop to the marking state (current flow), which the printers recognized as the "end of character" and "standby" for the next character. Because of "slop" (tolerances) in the mechanical system, the minimum length of time for marking current before the next character was started was standardized at the equivalent of 1.5 or 2.0 times the bit length. This was enough time for the machine to get ready to receive the next character. The time required for all the bits necessary to transmit one character—the start, character and stop bits—is called the *frame*.

Reliability

In order to transmit both upper and lower case characters and to provide for a quick-and-dirty way to test the reliability of the transmission, the number of character bits were eventually increased to eight: seven bits to produce the ASCII character set and control codes, and one bit—called the *parity bit* —for testing the reliability of the communications path. (Later, instead of the eighth bit being used to indicate parity, it became part of the character code for transmitting or printing graphic symbols.)

The mainframe computer and teletypewriter terminals that connected to a computer via modems were essentially a *communications system*. The same is true for the printers driven by modems which could print information originating at a mainframe computer or a KSR TTY terminal. (For example, many doctors had RO units (printers) that connected through a modem and a dedicated telephone line to a medical laboratory. Test results from a patient's blood samples—which had been sent to the laboratory—were instantly transmitted to the printer in the doctor's office.)

Notice that in every instance, except when the terminal is connected directly to the computer, there is a modem before or after the TTY device—the whole system is meant for communications.

The RS-232C I/O is born.

While effective, the current loop was unweildy and lacked hardware handshaking, and so an "industry group" came up with the RS-232C I/O, an electronic replacement for the current loop. RS-232C substituted variations in a DC voltage level for the current loop and also provided special DC voltages for hardware handshaking, a means whereby one item of equiupment knows if another is turned on or ready to accept data, or capable of transmitting.

But the RS-232C standardization was meant for the mainframe—or dinosaur—age of computing, and so RS-232 was designed for communications. Its connections—shown in Table 1—are entirely communications-oriented.

TABLE 1

TERMINAL	NAME	FUNCTION
1	HGND1	Hardware Ground (often not used)
2	TX	Transmit. Data fed into modem or DCE device from computer or terminal.
3	RX	Receive. Data received by modem from the telephone line (to printer or terminal).
4	RTS	Request to Send (from DTE)
5	CTS	Clear to Send (DCE device ready)
6	DSR	Data Set Ready (DCE device ready)
7	GND	Common signal ground
8	CD	Carrier detect (Modem received carrier)
20	DTR	Data terminal or DTE ready
22	RD	Ring detector. (For auto-modems)

Even for communications, there is a serious problem. When using a current loop, anything connected into the loop will work as long as the *framing* is correct. Every printer in the loop will simultaneously print whatever is entered in a keyboard, and any keyboard can enter a character into the loop. But an electronic circuit using variations in DC voltage level must have seperate send and receive circuits. For example, if RS-232C terminal #2 is input, it's input for every piece of equipment, while terminal #3 is output for every piece of equipment. If terminal #2 of the keyboard is connected to terminal #2 of the printer, absolutely nothing will happen. In order to print, terminal #2 of the keyboard or modem (the output) must connect to terminal #3 (the input) of the printer.

Standards again.

This was understandable by everybody, but there are always people who believe "standards" should be standard even if they don't work. Since it was impossible to have a single set of connections serve as a "standard," the group created two sets of RS-232C connections: One called DCE for *D*ata *C*communications *E*quipment, the other DTE for *D*ata *T*erminal *E*quipment. And the DTE connections mated with the DCE. Anything remotely resembling terminal equipment was considered DTE; anything that originated or passed data into or from a telephone line was considered DCE. A modem, being DCE, could be connected directly to a printer, which was DTE. Unfortunately, none of the connections allowed for the personal computer, and it is this lack of foresight that creates some of the worst interfacing problems with personal computers.

Consider this situation: Your computer has one serial I/O which is used for both the printer and the modem. Since the modem is DCE, the computer's serial must be DTE. Then what is the printer? DTE or DCE? By convention, a printer is DTE because it is meant to be fed by the modem. But we feed printers from personal computers. If the computer's output is DTE in order to drive the modem, it can't drive the printer, which is also DTE.

The hardware handshaking of the DTE computer is also intended for the modem; it expects to see a positive handshake voltage on terminals #4, #5, #6 or #8, or a combination of terminals, when the modem is ready to accept data. This is the exact opposite of what the printer needs because the printer was originally intended to work with the modem in place of the computer. The printer does not need a hardware handshake because it originates the handshake for the modem, or the printer provides a software handshake through pin #2. But if both the computer and printer are providing a handshake on the same terminals neither will know what's going on.

All this—the result of poor compromises and some manufacturers who seemingly go out of their way to make serial connections as difficult as possible—is the reason why what should be a simple printer-to-computer connection becomes a week-long nightmare.

Get the handshake right.

On the other hand, there are a few manufacturers, such as Smith-Corona (printers) and Kaypro and Radio Shack (computers) who go out of their way to make serial interfacing as uncomplicated as possible. For example, Smith-Corona's serial connections use the #20/#4 hardware handshake, meaning, that if you connect the printer's #20 or #4 terminal to the #20 or #4 terminal of any computer's serial I/O *intended for printers* it's all going to work. The serial printer I/O of Radio Shack's *Color Computer* is even easier. There is one single terminal that will work if connected to any printer handshake. Then there is IBM *PC* and its clones, which expect to see a modem connected to the serial I/O. If you use a printer, you must prewire the connections so the computer thinks it's connected to a modem. (Regardless what kind of connections are made between the computer and the printer, keep in mind that handshake polarity must be observed. Some printers provide for either a positive or negative handshake. Know what your computer needs and what the printer delivers before you buy.)

Table 2 shows some of the unusual wiring problems you may run across. All three computers, the IBM *PC*, the Kaypro *II* and the Texas Instruments *Professional* have RS-232C I/O intended for a modem, yet note how all three connect differently to a standard printer's

TABLE 2

IBM PC TERMINAL NO.	SERIAL PRINTER TERMINAL NO.
2	3
3	2
7	7
5	20
6, 8, 20	6
KAYPRO II TERMINAL NO.	**SERIAL PRINTER TERMINAL NO.**
2	3
3	2
5	20
6	4
7	7
20	6
T.I. PROFESSIONAL TERMINAL NO.	**SERIAL PRINTER TERMINAL NO.**
2	3
3	2
4-5 (short)	NC (No Connection)
6	20
7	7
8	4
20	6

RS-232C input (Diablo *620*). Essentially, the connections trick the computer into thinking it's connected to a modem rather than a printer.

Getting it all together.

Any one mistake in serial interfacing, from the number of characters, stop or total bits, to the signal or handshake connections, can stop your printer cold—or cause it to print "garbage." But if you initially set up the printer in a specific order, giving attention to what appears to be the most insignificant details, you should be able to have it running at the first try. Just keep this one fact in mind at all times—regardless of the variations in connections, labelling or whatever, if *the printer* employs a DB-25 connector terminal #3 is always the data input. Do not get confused by DCE, DTE or any other nomenclature. If the printer was intended to work with a personal computer, it's looking for data on terminal #3. If it has a hardware handshake it's either #4, #20 or both, with the user selecting the one to use. If it has a software handshake (ETX/ACK or X-on/Xoff) the handshake comes from terminal #2. (There might or might not be other connections.) Inexpensive serial printers can generally work with only a three-wire connection; signal, ground and handshake.

First things first.

Serial printers have switches or a patch panel (small jumpers) that set the various parameters and functions. You must set them for the printer's baud rate and character bits. Baud rate is no problem: the computer's manual specifies the output baud rate or tells you how to program the computer for the desired baud rate. The number of character bits is something else. If the computer's serial output is programmed for eight bits, even if it is only using seven bits per character the printer must account for all eight bits. In some printers, if seven character bits are programmed, the eighth bit is always a parity bit: either on or off, but there is either a mark or space bit to fill the void. on the other hand, some printers don't recognize the eighth bit if the parity isn't specified as *even* or *odd*: if not specifically programmed for the eighth bit, the printer looks only for the seven bits between the start and stop bits. This throws the framing off and after the first or second character the printing is "garbage." You can never go wrong if you account for the parity bit. If the computer is specified as "seven bits no-parity" you must set the printer for 7 bits + parity off (8 bits total). You cannot add 7 + "no parity" and assume it equals 8 bits. Parity off on a 7 bit printer is a forced space or mark, the printer doesn't care which as long as the stop bit(s) follow at the correct time.

If you think this is confusing, some of the serial interfacing for a very popular computer as given in the set-up manual of a famous, highly-rated printer has been wrong in every printing. What generally happens is that after many futile hours of effort, the user telephones that printer manufacturer's service department and finally gets the correct switch settings over the phone. (No, I will not embarass them by mentioning names. The parity bit problem has happened to the best and worst of printer manufacturers.)

Part of the framing includes the stop bit(s). Normally, above 110 baud only one stop bit is used. However, there are computers that use two stop bits (the Radio Shack *Color Computer* is one of the more popular). Normally, the modern printer will function with only one stop bit because it frames only the first stop bit and accepts the voltage transition that takes place with the following start bit (the space) as the beginning of the next character. On the other hand, a few printers frame to the second stop bit, and you *must* set the computer to provide two stop bits.

The hardware handshake.

Except for an unbelievable blunder by one of the major foreign manufacturers, the serial hardware handshake for a printer with a DB-25 connector is always printer terminal #20 and/or #4: Usually, it's only #20. Software handshaking—the ETX/ACK or DC1/DC2 (which is also called X-on/X-off)—is always through printer terminal #2.

If your computer's RS-232C is specifically wired as a printer port (DCE) you simply bring the signal and handshake wires straight across, #2 to #2, #7 to #7, #20 to #20, etc. If the computer has an RS-232C I/O which is wired for DTE for connection to a modem, but which is software programmed for a printer, you have big trouble because the wiring at the computer's connector can vary from model to model.

Depending on the particular printer and computer being used, some of the connections might not be required or even exist. That's when the going gets sticky. Unless you have the detailed connections for a particular combination of printer and computer, you might never get the printer working if the computer does not have a true RS-232C printer I/O. Contact the manufacturer of the printer if you have any doubts. They usually can give you the connections over the phone.

COMPUTER-DESIGNED AUDIO NETWORKS

This program eliminates the hard work in designing impedance-matching networks and attenuators for audio systems.

FRANK GALDES

■Ever since the early days of the telephone, attenuator networks have been used to control sound levels and to match impedances. Technicians, hobbyists, and experimenters who work with audio equipment and circuitry often find that a particular piece of equipment has an input (or output) impedance or line level that does not meet their requirements. In many of those instances, a commercially manufactured attenuator network is not readily available to solve the problem. Of course, you don't need a commercial network—you can put one together yourself.

Only four networks—the T, H, L, and U—will be considered. The T network consists of three resistors connected in the form of a "T," as shown in Fig. 1-a. It is an unbalanced attenuator. When used between circuits of unequal impedance, it is often called a taper pad. The H-type attenuator is a balanced T pad. It consists of 5 resistors connected in the form of an "H," as shown in Fig. 1-b.

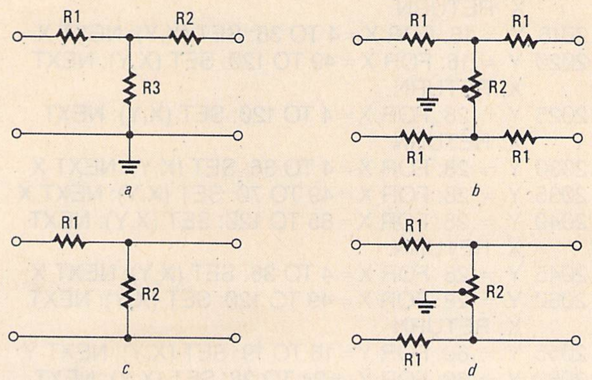


FIG. 1—THE FOUR ATTENUATOR TYPES that the program will help you design. Shown in *a* is a T attenuator, in *b* is an H-type, in *c* is an L-type, and in *d* is a U-type attenuator. Note that the tap on R2 (in *b* and *d*) is at the exact center of resistance.

An L-type attenuator, shown in Fig. 1-c, is perhaps the simplest form of attenuator, consisting of two resistive elements configured in the form of an "L." The L pad does not reflect the same impedance in both directions.

A U-type attenuator, shown in Fig. 1-d, is most often used matching a high impedance to a low impedance.

Before the era of personal computers or hand-held calculators, much time was spent using network formulas, dB charts, and possibly a slide rule to calculate the resistor values for one of those networks. Those calculations were time-consuming and tedious, especially for those not mathematically inclined.

The computer program shown in Table 1 will do all the lengthy calculations in seconds, after you choose one of the 4 networks, enter the line impedance, and the loss (in dB) desired (if applicable). It will then draw the circuit diagram showing the resistor values, the input and output impedances, and where the circuit may be grounded if necessary.

The program is written for the *TRS-80 Model III* with 16K, but it should be adaptable to other computers as well. Note that in the following program a bracket ([) indicates an exponent.

TABLE 1—AUDIO-NETWORK DESIGN

```
 5 CLS; PRINT
10 PRINT "DESIGN AN AUDIO LINE IMPEDANCE
   MATCHING NETWORK OR ATTENUATOR":
   PRINT: PRINT
15 PRINT " KIND", "Z IN/OUT",
   "CONFIGURATION", "SELECT ONE:":PRINT
20 PRINT " T", "EQUAL/TAPER",
   "UNBALANCED","1"
25 PRINT " H", "EQUAL/TAPER",
   "BALANCED","2"
```

```
30 PRINT " L", "TAPER", "UNBALANCED", "3"
35 PRINT " U", "TAPER", "BALANCED","4"
40 PRINT
45 INPUT"ENTER NUMBER FOR CIRCUIT
   DESIRED";S
50 ON S GOTO 100, 200, 300, 400
70 PRINT "ERROR -- DO OVER": GOTO 45
100 CLS: PRINT: PRINT: PRINT " INPUT AND
    OUTPUT IMPEDANCES ARE EQUAL,
    SYMMETRICAL:....1"
105 PRINT: PRINT " INPUT AND OUTPUT
    IMPEDANCES ARE NOT EQUAL, TAPER:....2"
110 PRINT: PRINT: INPUT "ENTER NUMBER FOR
    CIRCUIT DESIRED" ;N
115 ON N GOTO 120, 150: PRINT "ERROR = DO
    OVER": GOTO 110
120 CLS: GOSUB 510: CLS
125 RA=INT (Z*( (K-1)/K+1))):
    RC=INT(Z+K)/(K[2-1))):A=Z: B=Z
130 RA=RA+1: IF S = 2 THEN 205:CLS
135 T$="T": B$="UNBALANCED":
    I$="SYMMETRICAL": U$="MATCH
    IMPEDANCE AND REDUCE POWER LEVEL.
140 GOSUB 1000: GOSUB 1020: GOSUB: 1055
145 GOSUB 2000: GOSUB 2025: GOSUB 2025:
    GOTO 3000
150 CLS:GOSUB 500:GOSUB 530:GOSUB
    525:CLS
155 KA=(K-1)/(K+1):KB=(K[2-1)/(2*K):
    RC=INT((A+B)/(2*KB))
160 RA=INT(((((A+B)*KA)+(A-B))/2)
165 RB=INT((((A+B)*KA)-(A-B))/2)
170 RC=RC+1:IF S = 2 THEN 250
175 T$="T": B$="UNBALANCED";I$="TAPER":
    U$=MATCH IMPEDANCE AND REDUCE
    POWER LEVEL"
180 GOSUB 1000:GOSUB 1025:GOSUB 1055
185 GOSUB 2000:GOSUB 2025:GOSUB 2055:
    GOTO 3000
200 GOTO 100
205 RA=INT (RA/2):RB=INT(RB/2)
218 CLS
215 T$="H": B$="BALANCED":
    I$="SYMMETRICAL":U$="MATCH
    IMPEDANCE AND REDUCE POWER LEVEL"
220 GOSUB 1000: GOSUB 1050
225 GOSUB 2000: GOSUB 2030: GOSUB 2055:
    GOTO 3000
250 RA=INT (RA/2): RB=INT(RB/2)
255 T$="H": B$="BALANCED":
    I$="TAPER":U$="MATCH UNEQUAL
    IMPEDANCES"
260 GOSUB 1000:GOSUB 1035:GOSUB1050
265 GOSUB 2000:GOSUB 2030:
    GOSUB 2055:GOTO 3000
300 CLS
305 GOSUB 500:GOSUB 530: DB=DB+1:
    GOSUB 525
310 RA=INT((A/SQR(A/B))*(((K*SQR(A/B))-1/K))
315 RC=INT((A/SQR(A/B))*(1/(K-SQR(A/B))))
320 T$="L":B$="UNBALANCED"; I$="TAPER":
    U$="MATCH UNEQUAL IMPEDANCE"
325 GOSUB 1000:GOSUB 1040:GOSUB 1055
330 GOSUB 2015:GOSUB 2025:GOSUB 2055:
    GOTO 3000
400 CLS
405 GOSUB 500: GOSUB 530: CLS:RO=B/A
410 RA=INT(A*((SQR(1-RO))/2))
415 RC=INT(A*(RO/SQR(1-RO))):RC=RC+1
420 T$="U": B$="BALANCED": I$= TAPER":
    U$="MATCH UNEQUAL IMPEDANCES"
425 GOSUB 1000:GOSUB 1045
430 GOSUB 2015:GOSUB 2045:GOSUB 2055:
    GOTO 3000
500 PRINT:PRINT:INPUT "ENTER LARGER OF
    TWO IMPEDANCES" ;A
505 INPUT "ENTER SMALLER OF TWO
    IMPEDANCES";B:RETURN
510 PRINT:PRINT INPUT"ENTER THE
    ATTENUATOR IMPEDANCE";Z
515 PRINT "ENTER THE REQUIRED LOSS IN DB"
520 INPUT"DB LOSS CAN BE 0.5 TO 30+ IN
    SOME CASES";DB
525 K=EXP((DB/20)/(LOG(2.71828)/LOG(10)))
    :RETURN
530 DB=CINT(ABS(20*(LOG(SQR(1/(A/B
    ((/(1+SQR(1-(1/(A/B)))))/LOG(10)))+1)
535 RETURN
1000 PRINT @ 83, "TYPE: "T$: PRINT @ 100,B$
1005 PRINT @ 164, "LOSS IN DB: "DB
1010 PRINT @ 147,I$
1015 PRINT @ 211, "USE: "U$:RETURN
1020 PRINT @ 338, RA: PRINT @ 355,RA: PRINT
     @ 450,A: PRINT @ 476,RC: PRINT
     @ 505,A:RETURN
1025 PRINT @ 338, RA: PRINT @ 355, RB: PRINT
     @ 450, A: PRINT @ 476, RC: PRINT @ 505,B:
     RETURN
1030 PRINT @ 338, RA: PRINT @ 355, RA: PRINT
     @ 450, A: PRINT @ 476, RC: PRINT @ 505, A;
     PRINT @ 594, RA: PRINT @ 611, RA:
     RETURN
1035 PRINT @ 338, RA: PRINT @ 355, RB: PRINT
     @ 450, A: PRINT @ 476, RC: PRINT @ 505, B:
     PRINT @ 594, RA: PRINT @ 611, RB:
     RETURN
1040 PRINT @ 338, RA: PRINT @ 450, A:PRINT @
     476, RC: PRINT @ 505, B: RETURN
1045 PRINT @ 338, RA: PRINT @ 450, A: PRINT @
     476, RC: PRINT @ 505, B: PRINT @ 594, RA:
     RETURN
1050 PRINT @ 770, "IF NECESSARY GROUND AT
     CENTER OF "RC" OHMS RESISTOR.":
     RETURN
1055 PRINT @ 770, IF NECESSARY LOWER LINE
     MAY BE GROUNDED": RETURN
2000 Y = 16: FOR X=4 TO 36: SET (X,Y): NEXT X
2005 Y = 16: FOR X=49 TO 70: SET (X,Y): NEXT X
2010 Y = 16: FOR X=85 TO 120: SET (X,Y): NEXT
     X: RETURN
2015 Y = 16: FOR X=4 TO 36: SET (X,Y): NEXT X
2020 Y = 16: FOR X=49 TO 120: SET (X,Y): NEXT
     X, RETURN
2025 Y = 28: FOR X=4 TO 120: SET (X,Y): NEXT
     X: RETURN
2030 Y = 28: FOR X=4 TO 36: SET (X,Y): NEXT X
2035 Y = 28: FOR X=49 TO 70: SET (X,Y): NEXT X
2040 Y = 28: FOR X=85 TO 120: SET (X,Y): NEXT
     X: RETURN
2045 Y = 28: FOR X=4 TO 36: SET (X,Y): NEXT X
2050 Y = 28: FOR X=49 TO 120: SET (X,Y): NEXT
     X: RETURN
2055 Y = 60: FOR Y=16 TO 19: SET (X,Y): NEXT Y
2060 Y = 60: FOR Y =24 TO 28: SET (X,Y): NEXT
     Y: RETURN
3000 PRINT: PRINT: INPUT "PRESS <ENTER> TO
     START NEW CALCULATION.";:GOTO 5
3010 END
```