BUILD THIS

Build REACTS: THE RADIO-ELECTRONICS ADVANCED CONTROL SYSTEM

MICHAEL A. TUCKER

Part 7 IN ORDER to write programs and develop practical systems using REACTS, some type of terminal is required as part of the development system. Many end-user applications also require a terminal interface for normal operation. REACTS allows you to use any one of three devices for the terminal. The first choice is that you use a standard CRT terminal. The second alternative is to use an IBM PC as a terminal, and the third choice is to build a general-purpose CRT controller/keyboard interface module using the information supplied in this article.

The interface has 2 microprocessors; one is a general-purpose Z-80, and the other is a special-purpose CRT controller. The design is flexible enough to allow you to create a custom terminal by simply changing the on-board PROM. An added feature on the module is a Centronics-type parallel-printer interface, for an IBM PC compatible printer.

Standard CRT terminal

Over the past decade, computer terminals have come down in price (a quality terminal can be purchased for less than $400.00). A number of different terminals have been tested and found to be satisfactory. There are also a number of used terminals on the market that can be purchased for next to nothing. Just be sure that the one you buy is compatible with a standard RS-232 interface, and try not to buy any terminal that doesn't have its technical manual.

The following parameters and their settings were used with REACTS and the Qume 101 terminal. They will give some idea where to start even if you use a different monitor.

- EMULATION—QVT101C
- PARITY—OFF
- DATA BITS—8
- STOP BITS—1
- BIT8—0
- FDX/HDX (Full or Half Duplex)—FDX
- CHAR/LINE/BLCK—CHAR
- ON LINE/LOCAL—ON LINE
- BAUD RATE—9600

Your PC as a terminal

The **Radio-Electronics** bulletin board (516-293-2283, 300/1200 baud, 8N1) contains the software necessary to convert your PC into a terminal for controlling REACTS. That software is also available on a floppy disk with a manual from the source listed elsewhere in this article. The software package from that source contains a file-transfer program on a PROM, which, in addition to making your PC act like a "dumb" terminal, allows you to use your PC disk, hard or floppy, as storage devices for your REACTS. That capability is very useful when developing software and allows you to keep your investment in the system to a minimum.

The PC-terminal software is contained in the file TERMINAL.EXE. To invoke that software, just type TERMINAL at the system's prompt. The software will automatically clear the screen. The only requirements for the software is that it must be run on an IBM PC or compatible, and REACTS be configured for a baud rate of 1200. The IBM-terminal software is designed to emulate the Televideo-912 terminal, so any software designed for the Televideo terminal should work as long as the host system is set up for 1200 baud. The relatively low baud rate is due to the speed limitations of a standard IBM PC. The REACTS will accept and transmit data at rates of at least 10 times faster than the PC when the PC is operating in an emulation mode.

The PC and its keyboard act just like a standard terminal at this point. Any operation that you would perform on a standard terminal is possible using your PC. With the PROM-based file-transfer program there is

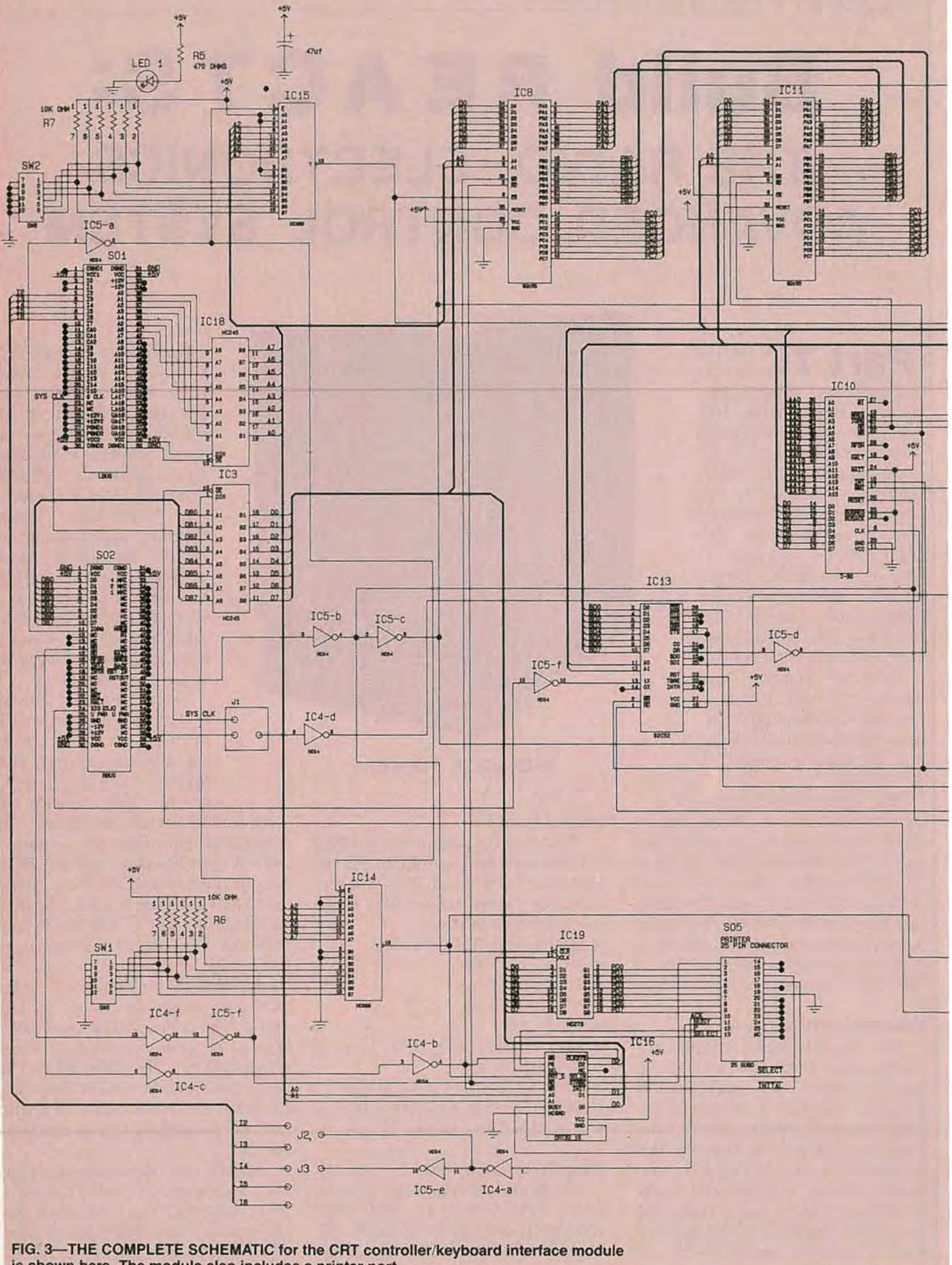
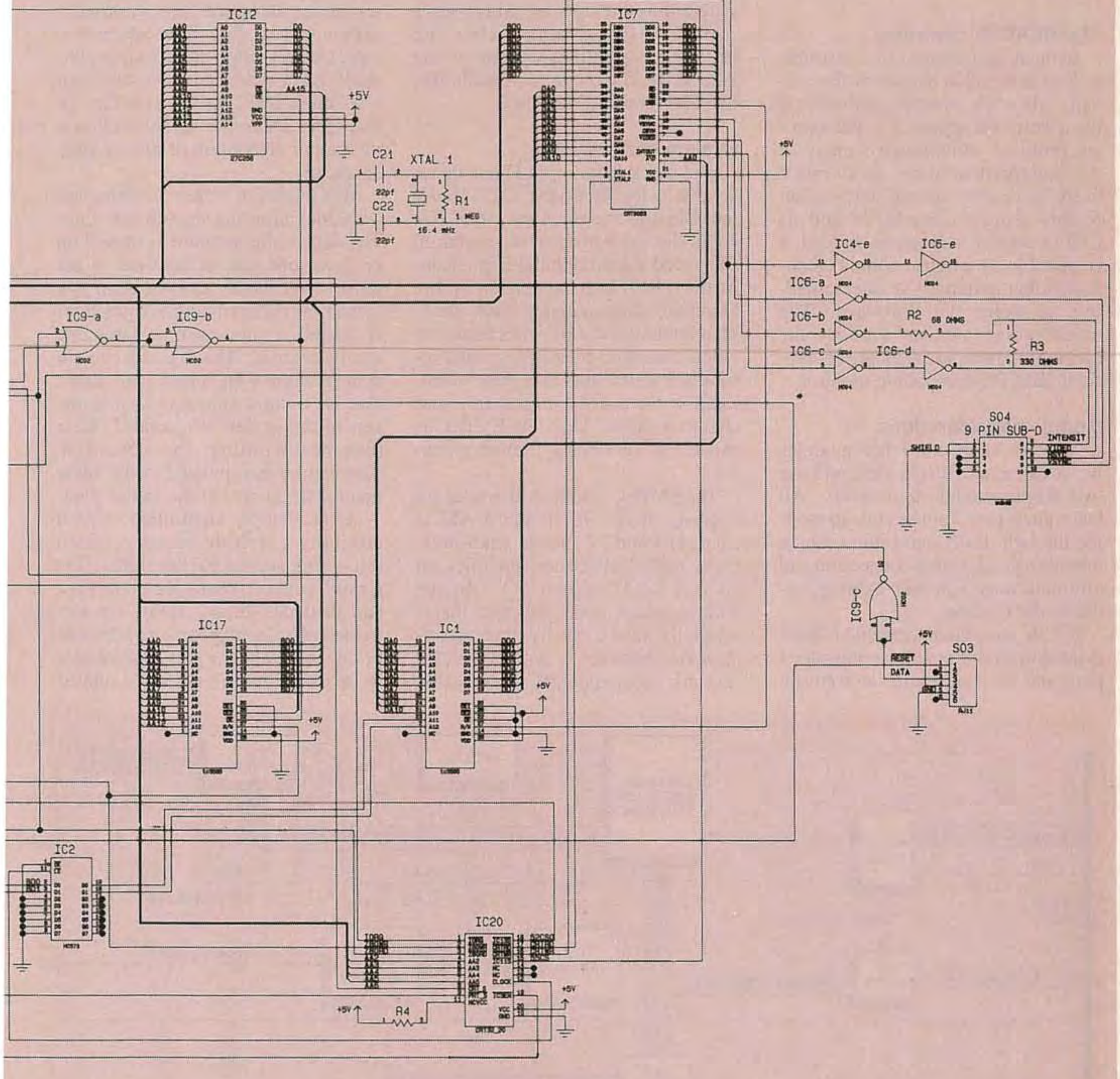


FIG. 3—THE COMPLETE SCHEMATIC for the CRT controller/keyboard interface module is shown here. The module also includes a printer port.


IC12
27C256
AA15
+5V
IC7
C21
22pf
C22
22pf
XTAL 1
16.4 mHz
R1
1 MEG
IC4-e
IC6-e
IC6-a
IC6-b
R2
68 OHMS
R3
330 OHMS
IC6-c
IC6-d
S04
9 PIN SUB-D
SHIELD
INTENSIT
VIDEO
IC9-a
IC9-b
IC9-c
RESET
DATA
S03
IC17
IC1
IC2
IC20
R4
CLOCK

the added capability of the PC's mass-storage potential; the terminal software can transfer your ASCII files from REACTS to the PC and vice versa.

The REACTS controller

In many applications it is desirable to have a terminal as part of the system. Also, the integral controller is the lowest-cost approach to the terminal problem. Additionally, many of the entrepreneur types will find it useful to develop special-purpose terminals using the REACTS and its CRT controller as the basis. Indeed, it is possible to emulate a multi-thousand dollar terminal—if you are willing to write the software. The monochrome monitor used by the controller can be any standard off-the-shelf IBM PC-compatible monitor.

Module considerations

As with all of the other modules we've discussed, this module will use two 60-pin module connectors. All bus signals pass from module to module through those connectors, which provide a good, sound connection and eliminate any external cabling between the modules.

A 32K ultra-violet erasable PROM is used to store the video controller's program. You can rewrite the software on the PROM and create your own custom terminal or emulate some other terminal. You would use REACTS itself as the development system to write the custom software. The program would then be burned onto a 32K × 8-bit PROM using the PROM-programmer option of the PROM/RAM module, covered in **Radio-Electronics**, May 1988.

System components

The block diagram in Fig. 1 shows the basic layout of the CRT/printer interface-module circuitry. The CRT-controller portion centers around an *E*nhanced *Vdeo T*erminal *L*ogic *C*ontroller (EVTLC), specifically the Standard Microsystem's CRT 9053. That integrated circuit will enable our video controller to display visual attribute features such as reverse video, intensity control, underline, and character blink. Also, the EVTLC is capable of producing limited graphics.

The EVTLC's built-in character set consists of the 96 standard ASCII characters and 32 special characters. Each ASCII character occupies an area of 9 × 12 screen dots. In the wide-graphics mode, the space that is normally taken up by one alphanumeric character is instead divided into six independently addressable segments. On a screen set up for 25 rows and 80 columns, the graphics mode would allow the independent addressing of twelve-thousand segments (80 × 25 × 6). In the thin graphics mode, the space that is normally taken up by one alphanumeric character is divided into four independently addressable segments layed out in a "cross-hair" fashion (see Fig. 2). The video controller will also allow a mixture of alphanumeric and graphic characters.

Two modes of screen scrolling are available: jump and smooth scrolling. The data on the monitor is moved up or down one row at the time in the jump-scroll mode, and one scan line at the time in the smooth-scroll mode. If desired, a non-scrolling status line can be enabled. The line will remain at the bottom of the screen at all times; that is, it stays stationary while the rest of the screen is scrolled. Other than non-scrolling, the status-line data can be manipulated in the same manner as the rest of the screen data.

As previously mentioned, screen attributes include reverse video (changing screen backgrounds), intensity control, character underline, and character blink. There are two modes of selecting the characters to be enhanced and the type of attributes to be used. In the first mode, called

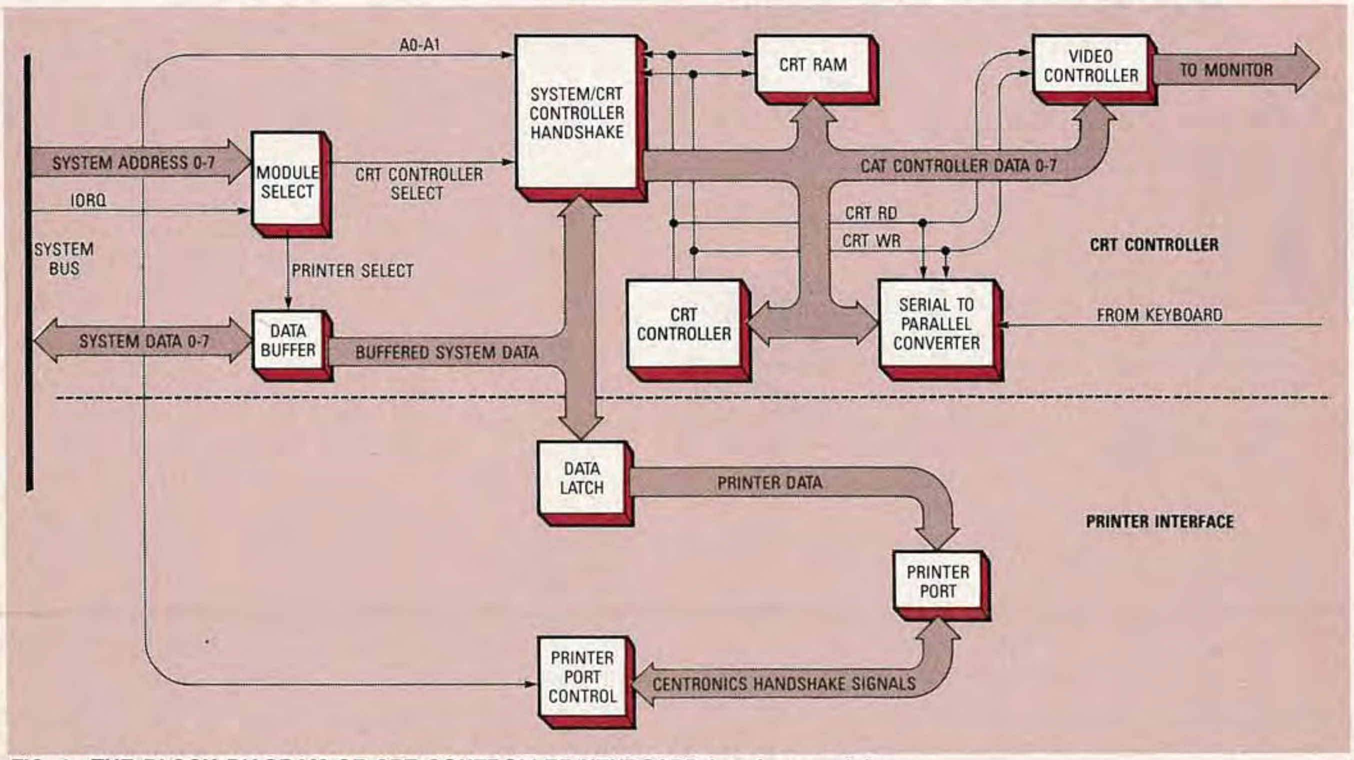


FIG. 1—THE BLOCK DIAGRAM OF CRT CONTROLLER/KEYBOARD interface module. This module allows you to interact with REACTS, and also allows REACTS to display messages on a CRT.

the 9 × 28 mode, each character to be enhanced is "tagged" with a tag bit, which is actually the most significant bit of the character byte. In that mode, only one attribute style can be enabled per screen. That is, all the "tagged" characters on any specific screen will be reverse video, intensified, underlined, or blinked. In the 9 × 56 mode, multiple attributes per screen and character are available. That is accomplished by sending an attribute byte before the character byte that will be enhanced with the desired attribute.

The fill-screen feature allows the entire screen to be filled with a given character without having to write to each display's memory address (that is ideal for quick clearing of the screen).

In addition to the EVTLC, other main components of the video controller include a Z80 microprocessor, two 8K × 8 CMOS-RAM IC's, two 82C55 programmable peripheral interfaces (PPI), an 82C52 universal asynchronous receiver/transmitter (UART), and an already mentioned 32K × 8 UVPROM.

The Z80 is the control center for the CRT controller. It manages the incoming and outgoing data between the 82C52 UART, the EVTLC, and the 82C55 PPI's. That is, it enables and/or disables the correct IC's at the correct time to maintain a smooth data flow. The Z80's clock input comes from the 8-MHz clock signal that is generated in the REACTS CPU and routed to one of the pins of the two 60-pin connectors. One of the 8K × 8 RAM chips is used by the Z80 as a scratch pad or buffer. The other provides the display memory for the EVTLC. As already mentioned, the PROM contains the video controller's control program. The 82C52 UART receives the incoming keyboard data in serial form and converts it to parallel form before transferring it to the EVTLC and/or PPI's. The PPI's act as the interface between the video controller and the CPU; that is, all data being passed between the CPU and the video controller pass through those IC's. By checking certain handshaking signals of those IC's, the CPU determines whether the video controller is ready to send data to it. Likewise, the CRT controller can determine when the CPU is sending data to it.

Sources

The following items are available from DataBlocks, Inc
579 Snowhill Road, Glenwood, GA 30428. Or call (800) 652-1336; in Georgia call (912) 568-7101

- TERM-PROM: Software to enable the use of an IBM compatible as terminal for REACTS. Includes the software on a 360K floppy disk with instructions; $18.00.
- DP-CRT/PRT: Design package of schematics and instructions; $10.00.
- PC-CRT/PRT: PC board for CRT/printer interface, includes design package; $36.00.
- CRT-PROM: A control program for the CRT controller installed on a 32K UV-erasable PROM, includes documentation; $17.00.
- REC-CRT: Complete set of parts including the PC board, IC's, connectors, and PROM; $124.00.
- Keyboard for REC CRT controller; $85.00.

Please include $5.00 postage for any order of $37.00 or less and $10.00 postage for any order over $37.00. Georgia residents must add appropriate sales tax.

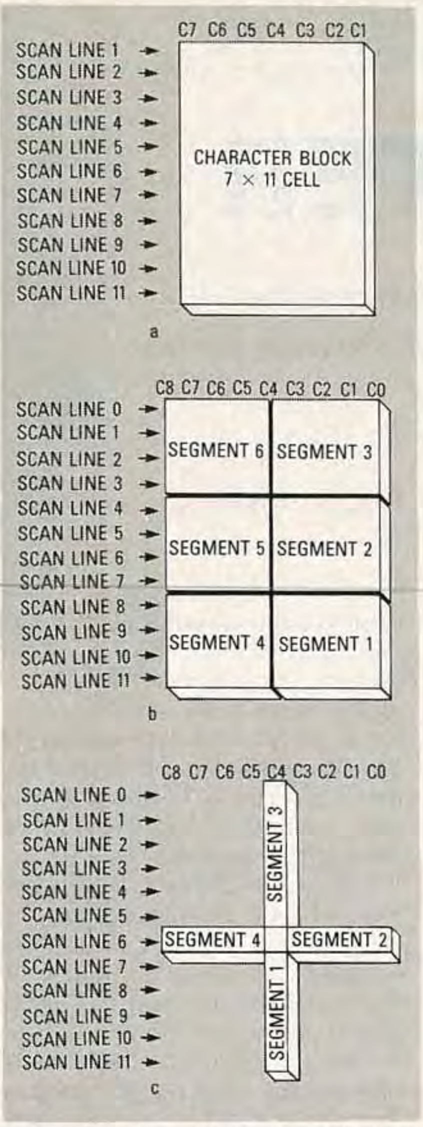


FIG. 2—The EVTLC's BUILT-IN CHARACTER SET consists of the 96 standard ASCII characters and 32 special characters. Depending on what graphics mode is being used, the screen space is divided as shown in (*a*) for the alphanumeric mode, (*b*) for the wide-graphics mode, and (*c*) for the thin-graphics mode.

Controlling the EVTLC

The EVTLC is connected to two data buses; one passes data between the Z80 and the EVTLC, and the other between the display memory and the EVTLC. That way we know that all the data we send or receive from the display memory passes through the EVTLC. The EVTLC contains several 8-bit software-programmable data registers which select the desired screen attributes, move and keep track of the screen cursor, transfer data, and select operating modes.

The EVTLC's programmable data registers are selected indirectly by the address register, which is selected when the input on the A/D pin of the EVTLC is high (see Fig. 3) and a write occurs. Data is sent to and/or read from the data registers when the input on the A/D pin is low. That is, first the desired register is selected by raising the A/D input and writing the correct register address in binary form, then the data is written to or read from the register with the A/D input lowered. When the A/D input is raised and a read is done, the status register's contents can be obtained. Bit seven of the status register is used to synchronize data transfers between the Z80 and the EVTLC. When bit seven is low, the EVTLC is busy and will not receive or send any data. Only when bit seven is high, is the EVTLC ready to send or receive data.

The data registers that are indirectly selected via the address register are: the top-of-screen address, cursor-low, cursor-high, fill-address, screen-attribute data, mode-1, mode-2, and character registers. We will give the correct address of each register and briefly discuss their functions. Unfortunately, that will have to wait unitl next month. We'll continue then with a discussion of the registers and then go on to building and programming the terminal interface. **R-E**