

Using LCD Displays

Using the intelligent liquid crystal display in your projects.

Don Jackson

In the past few years the popularity of alphanumeric LCD displays has grown dramatically. This rapid growth can be traced to a number of characteristics which are very desirable in modern electronics. These features include: very low power requirements, small size, light weight, ease of use and availability. These devices have been around for some time but were very expensive; however, in the past few years several types have shown up on the surplus market (see side bar for the names of some suppliers of surplus units). This article is intended to show you how you can add intelligent alphanumeric displays to your projects.

General Information On LCD Displays

Photo 1 shows a 1 line x 40 character display, 1 line x 16 and a 2 line x 20 character display. These are of various manufacturers but all use the same controller IC. They all have a similar interface and use the same commands to control the display.

Photo 1

In general this type of LCD module contains a display, a driver IC and a controller IC. I will not go into the working of the LCD display itself, but would refer you to articles which have appeared in the trade journals. In a single line display each character field is ten rows x five columns and in a 2 line display the character fields are eight rows x five columns. To display ASCII data generally a format of 7 rows x 5 columns is used, as is typical for LED or

CRT displays. In the LCD display the additional rows are used to show an underscore cursor and the descending tails on some of the characters.

I will concentrate on the interface and its requirements and the commands used to talk to the controller chips. Figure 1 shows the block diagram of a LCD device. The controller chip is a Hitachi HD44780 and the display driver is a HD44100. The chip numbers are given for reference only, as they are mounted on the back of the LCD module (see photo 2). Most manufacturers produce 16 character x 1 line modules which do not include a HD 44100 (such as Sanyo LM16155 or

Hitachi LM020L). This type of device requires additional driver routines and is not supported by the software presented here.

The HD44780 is one of the newer LCD controller chips from Hitachi. It is used in several displays manufactured by OEM companies such as Optrex, Densitron Corporation, Philips, Stanley and some Epson devices as well as Hitachi's own line of units. This IC acts as the interface and LCD controller. The HD44100 is a driver chip and the number used is dependent on the number of characters in the display.

The HD44780 is a dot matrix LCD controller IC that displays the ASCII and

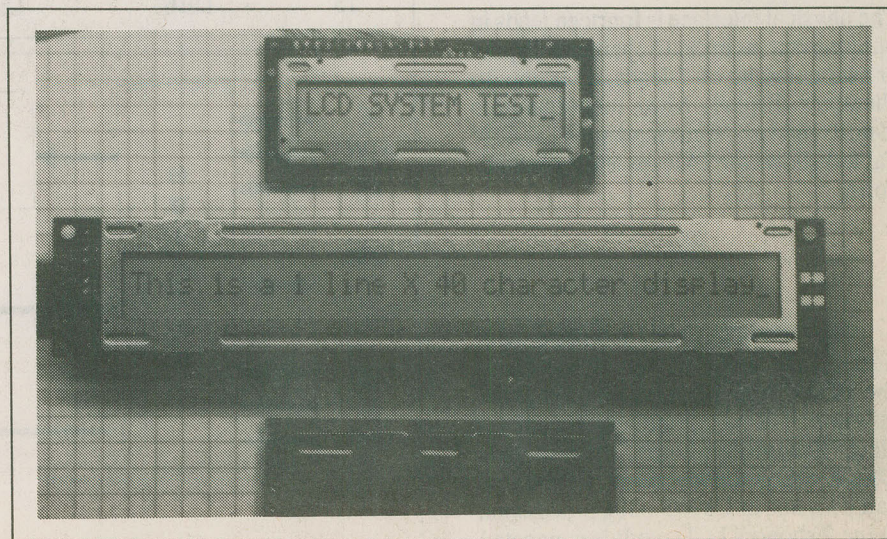


Photo 1. Three of the styles of LCDs available.

Using LCD Displays

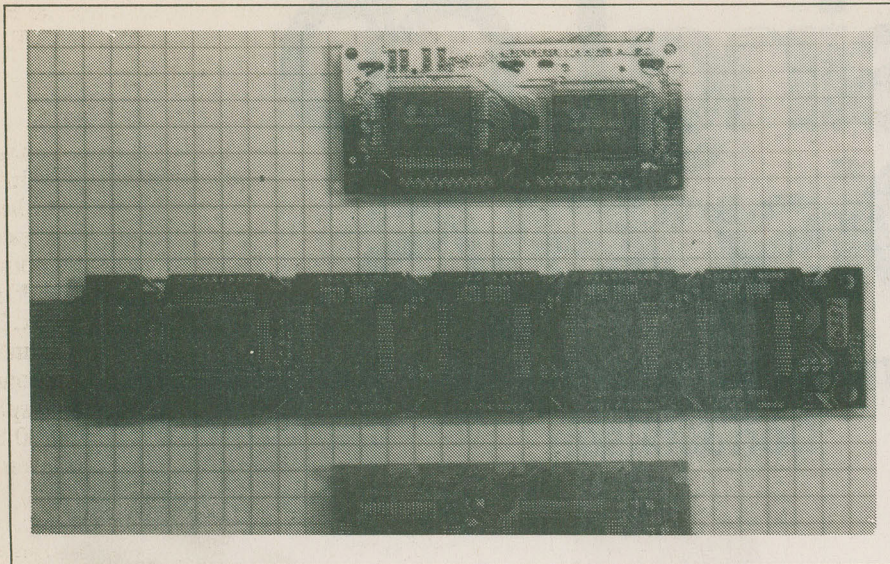


Photo 2. The rear view of the LCD displays shown in Fig. 1.

kana (Japanese) characters and symbols. The HD44780 contains a 80 byte display memory. This display memory is used to hold the data characters and the display acts as a window into this memory. In the case of a 16 character display the window shows 16 characters and the balance (64 characters or memory locations) is still available. By using commands supported in the HD44780 the display memory can shift through the window much the same way moving LED signs work. The HD44780 is a bus oriented unit and will operate in a 4 or 8 bit operation. I will only address the 8 bit operation of the unit as we will be attaching it to a 8 bit MPU.

The Interface

The physical interface is fourteen leads in one long row, as in the 16 character display or two columns of 7 leads, as in the 40 character display. Most manufacturers identify pin 1 and some mark pins 1 and 14. Table 1 shows the 14 leads and their functions.

The HD44780 will interface with most of the common microprocessors. The HD44780 is however tailored to the 680x and 650x bus structures. Hitachi has a large line of 680x chips. These include 6801/3, 6805s, 6804s, 6809, 68000 and their own line of CMOS 630x processors. Figure 2 shows a LCD module connected to a 6800 bus but the same general technique can be used for other processors.

Figure 2

Of the 14 interface leads only VO requires elaboration. This signal is used to control the viewing angle at which the display looks best. The viewing angle is important because just as in your LCD watch the intensity of the display will appear different depending on your viewing angle. To adjust the intensity of the LCD display a potentiometer is connected as shown in the schematic. A fixed resistor combination could be used but you will find that as the viewing angle of the display is changed the intensity may need to be adjusted. I have also seen an implementation where a thermistor is to control the VO to compensate for ambient temperature changes, however, I have never found this technique necessary.

The schematic shows a LCD (Hitachi type H2572) display connected to a 6800 processor. I have used this because it is a

Table 1

Pin No	Symbol	Function
1	VVS	0 Volts (GND)
2	VDD	5 Volts (less than 1 mA, typ)
3	VO	See text
4	RS	Reg select (0=Command, 1=Data)
5	R/W	Read/Write (1-Read 0=Write)
6	E	Enable (Latched on falling edge of E)
7	DB0	Data bit 0
8	DB1	Data bit 1
9	DB2	Data bit 2
10	DB3	Data bit 3
11	DB4	Data bit 4
12	DB5	Data bit 5
13	DB6	Data bit 6

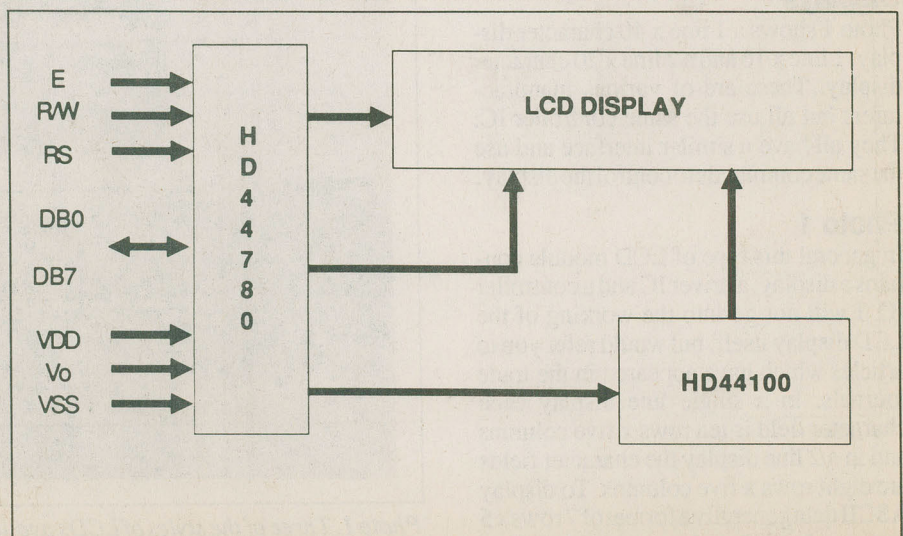


Figure 1. The basic connections for the logic.

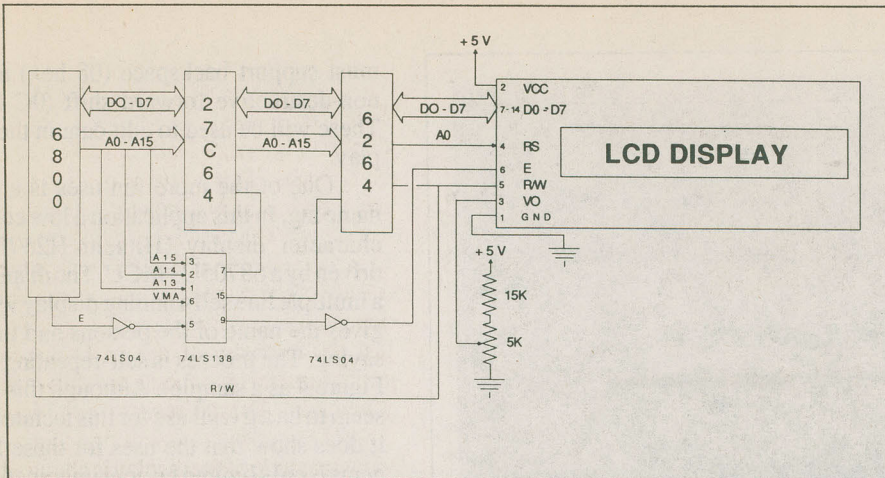


Figure 2. The interconnections to a 6800 CPU.

Table 2

Command Function

- 01h Clear display
- 02h Move cursor to home
- 04h Cursor moves left, display remains in position
- 05h Cursor remains in position, display move right
- 06h Cursor moves right, display remains in position
- 07h Cursor remains in position, display moves left
- 08h Display and cursor are turned off
- 0Ch Display is on, cursor is off
- 0Eh Display is on, underline cursor is on
- 0Fh Display is on, box cursor blinks
- 10h Shifts cursor one position left *
- 14h Shifts cursor one position right *
- 18h Shifts display and cursor to the left *

fairly simple implementation. As this is a memory mapped processor system the display is addressed at C000 hex. Other components in the system are the RAM memory at address 0000 to 1FFF, the EPROM at E000 to FFFF.

Commands And Software

The HD44780 is a controller IC and is organized as two memory locations. In this schematic because the RS lead is on the A0 of the address bus the HD44780 responds to two consecutive addresses. The address selection is a function of the 74LS138 and the display is at C000 and C001 hex. The first address accesses the command register of the HD44780 and the second is the data register. To control the display, commands are sent to address C000 hex and data that is to be displayed is sent to C001 hex.

The HD44780 responds to several commands to control the interface and the display. These include commands to clear the display, position the cursor and change the font type. See Table 2 for a basic list of commands used to control the HD44780. The HD44780 also supports advanced functions such as user-definable characters. Space limitations do not permit exploring these now, however, you should obtain a copy of the manufacturer's data sheet for the device you are using. These commands are sent to the HD44780 on the MPU's write cycle. Each command has an execution time (see data sheet). During the execution of a command the HD44780 signals a busy state with a Busy Flag. This flag is the high order bit in the byte returned when the MPU does a read on the command register. If the eighth bit is high a busy state exists in the HD44780. If bit 8 is low the HD44780 is ready for the next command. The other seven data bits are the address at which the cursor is positioned in the 80 byte display memory.

An Application

Now that I have covered the operation of the HD44780, let's put the LCD display to work on an application. The application is a single line 40 character display on a 6800 based data display station. The schematic is the same as in Figure *. The key element is the software required to support the application. Before the code can be written it is necessary to define how the application will operate. I have chosen the following: the display will be initialized with the cursor at the righthand end of the display (display position 40 or 17h). As data is entered the display will shift to the left. The code

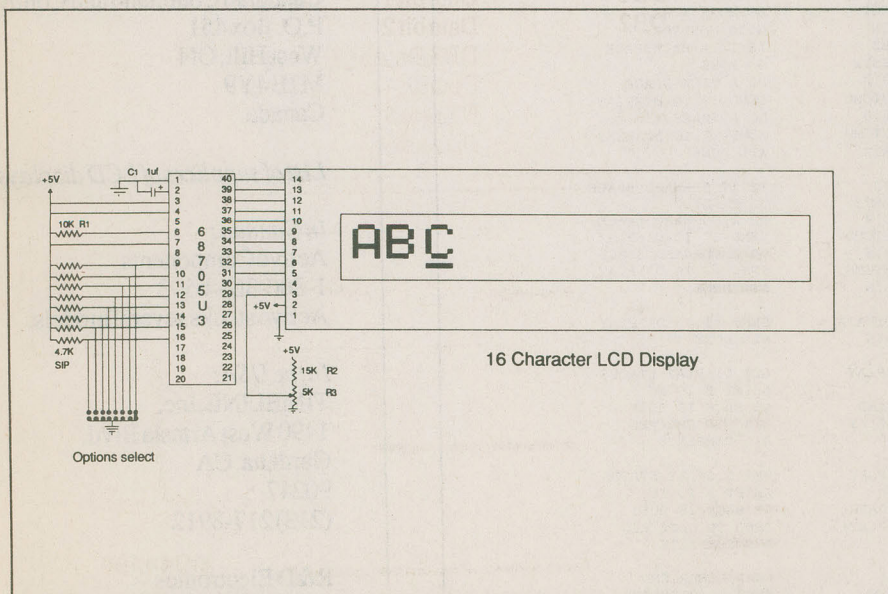


Figure 3. Pin connections.

Using LCD Displays

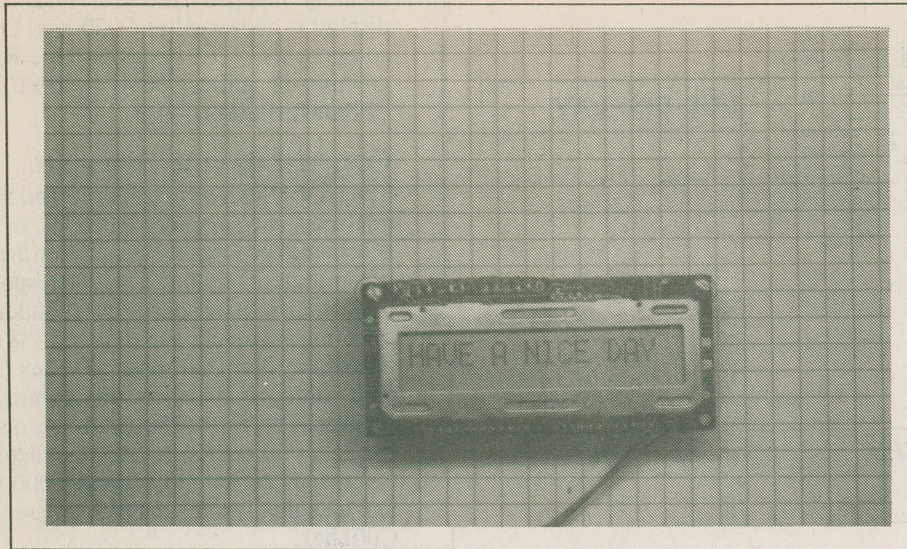


Photo 3. Programming the LCD for a message display.

```

*****
*
* This is a 6800 program to display data on a
* 40 character by one line LCD Display.
*
*****
*
*          ORG $1000          START OF RAM
*
* LCDD EQU $C000          LCD DATA ADDR
* LCDC EQU $C001          LCD CNTL ADDR
* END EQU $05            END CHARACTER
* BS EQU $08            BACK SPACE CHARACTER
* FS EQU $0C            FORWARD CHARACTER
*
* INCHAR EQU $E1AC          IN CHARACTER
* MON EQU $E0E3          MONITOR START
* DISPLAY EQU $A400          DISPLAY ADDRESS
*
*****
*
* START JMP INIT          INIT DISPLAY
*
* MAIN JSR INCHAR          GET A CHARACTER
*      CMPA #END          EXIT?
*      BNE TEST1          DO MORE WORK
*      JMP MON            GOTO MONITOR
*
* TEST1 CMPA #BS          IS IT A BACKSPACE
*       BNE TEST2          DO MORE
*       LDAA #S1C          DO A BACK SPACE
*       JSR OUTCMD          SEND IT TO DISPLAY
*       LDAA #S10          DO A SPACE
*       JSR OUTCMD          SEND IT TO DISPLAY
*       JMP MAIN          ALL DONE
*
* TEST2 CMPA #FS          IS IT FORWARD SPACE
*       BNE TEST3          DO MORE
*       LDAA #S14          DO A FORWARD SPACE
*       JSR OUTCMD          SEND IT TO DISPLAY
*       LDAA #S18          DO A FORWARD SPACE
*       JSR OUTCMD          SEND IT TO DISPLAY
*       JMP MAIN          ALL DONE
*
* TEST3 JSR OUTDATA          SEND IT TO DISPLAY
*       JMP MAIN          ALL DONE
*
* OUTCMD LDAB DISPLAY          GET DISPLAY STATUS
*        ASLB                SHIFT B TO CC
*        BCS OUTCMD          GO BACK IF BUSY
*        STAA DISPLAY        SEND TO CMD REG.
*        RTS                ALL COMPLETE
*
* OUTDATA LDAB DISPLAY          GET DISPLAY STATUS
*         ASLB                SHIFT B TO CC
*         BCS OUTDATA        GO BACK IF BUSY
*         STAA DISPLAY+1      SEND TO DATA REG.
*         RTS                ALL COMPLETE
*
* INIT LDAA #S01          LOAD CLEAR CMD
*      JSR OUTCMD          SEND TO DISPLAY
*      LDAA #S0E          LOAD DISPLAY CMD
*      JSR OUTCMD          SEND TO DISPLAY
*      LDAA #SA7          LOAD POSITION CMD
*      JSR OUTCMD          SEND TO DISPLAY
*      LDAA #S07          LOAD CURSOR CMD
*      JSR OUTCMD          SEND TO DISPLAY
*      JMP MAIN          ALL DONE

```

Listing 1. The source code.

must support backspace (08 hex) and a non-destructive forward shift (0C hex). These will be used to edit data in the display.

One of the more fun uses is a desk name tag. In this application a low cost 16 character display (Hitachi H2570) is driven by a 68705U3 MCU. The display is a multiple line self-running display which gives the name of the persons and then a saying. The process is self-repeating (see Figure 4 as a sample). Although this may seem to be a trivial use for this technology it does show that the uses for these little gems is only limited by your imagination.

This is the desk of John Dow Display

1

Have a nice day Display 2

I am on vacation if you need help please see my manager Display x

The schematic is very straightforward.. The 8 data bits if the LCD interface are driven by port A of the MCU. The control leads (RS, R/W and E) are driven by bits 5, 6 and 7 of Port B. Port C is a 8 bit input port used to select which sayings are to be displayed.

The MC68705 MCU is available preprogrammed with the buyer's name from:

Canadian Command and Control
P.O. Box 451
West Hill, ON
M1E4Y9
Canada

List of suppliers of LCD displays:

In Canada:
Active Components
1-800-363-6593
Active stocks several models.

In the USA:
TIMELINE, Inc.
1490 West Artesia Blvd.
Gardena, CA
90247
(213) 217-8912

R&D Electronics
1224 Prospect Ave.
Cleveland, OH
44115
(216) 621-1121

Figure 3

Due to space limitations the software can not be presented here; however, a copy (for personal use only) can be obtained from the author (write Don Jackson c/o E&TT, 1300 Don Mills Rd., Toronto, Ontario M3B 3M8). Programmed MPUs customized to the buyer's requirements are also available. The MPU is programmed with the buyer's name and multiple sayings. The 8 bit input switch selects which sayings are to be displayed.

If you plan to write your own code to drive an LCD display from an I/O port it is important to ensure that the port and the LCD are never both programmed for output at the same time as this could damage one or the other.

Conclusion

I have covered the interface requirements to attach the LCD module to a MPU system, the commands used to control the LCD display and a typical software example. Other uses for this type of display include output I/O on MPU controlled test equipment, displaying information on environment (i.e. temperature and time), remote display stations on a Home Information Net just to mention a few. These little gems are a super advent of the electronics available to the hobbyist today and I'm sure you will find many uses for these in your projects. ■

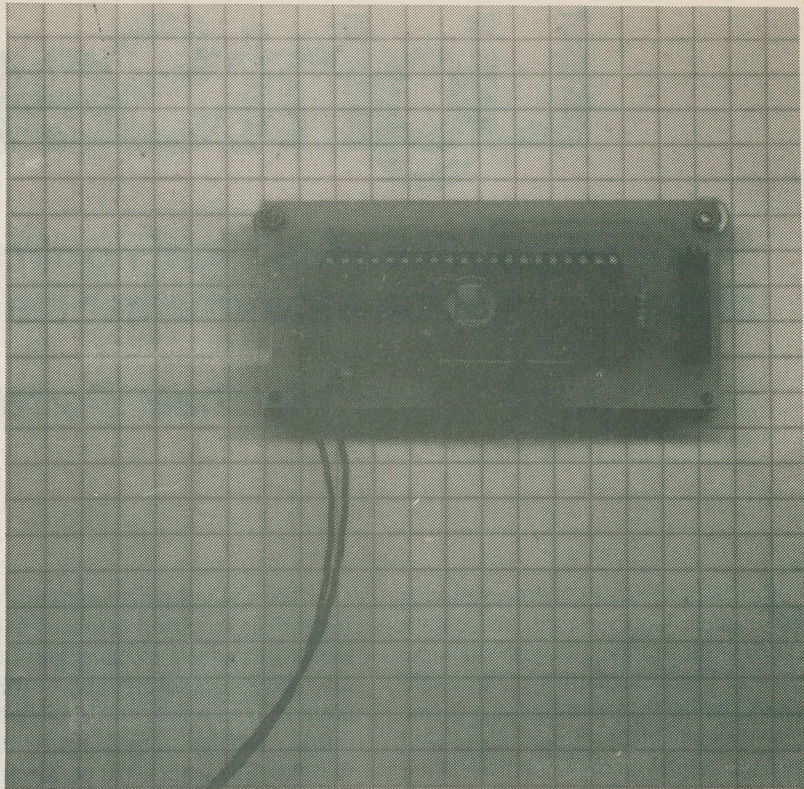
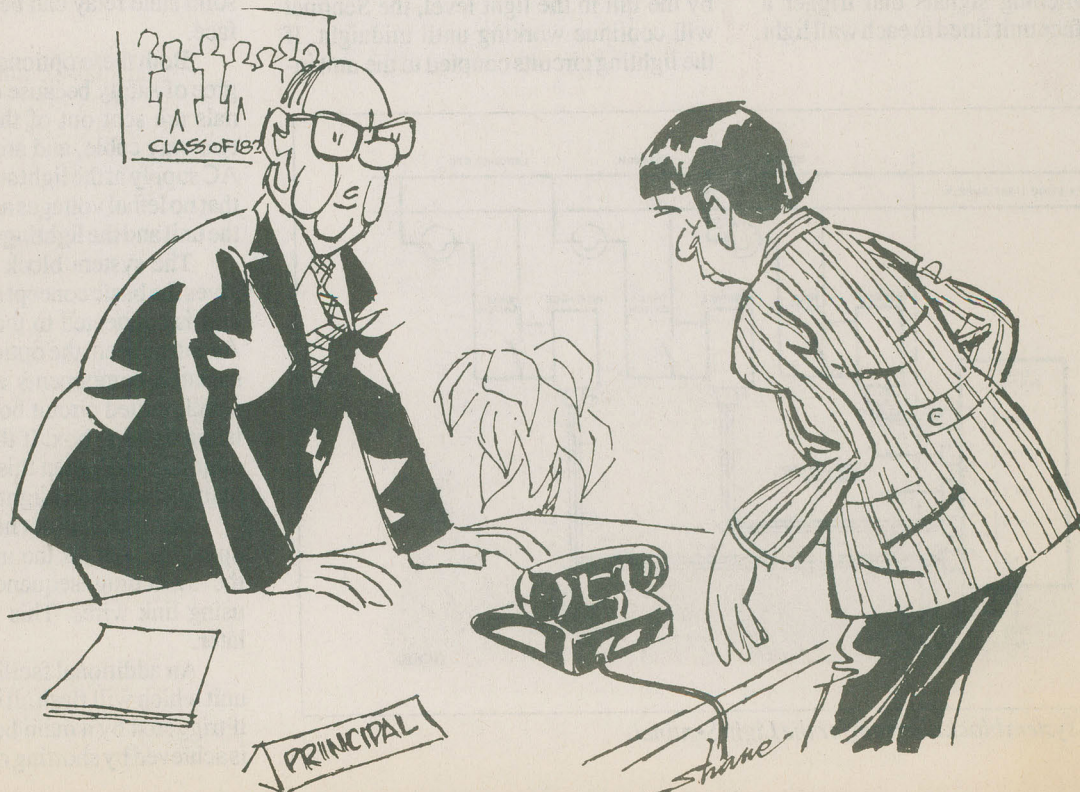


Photo 4. The rear of the display in Photo 3.



"My phone has been bugged....I want you to find the guilty party and put him in charge of the Data Base Computer."