

# KIT 134. INTRODUCTION TO LCD'S

The aim of this kit is to show how to use a 16x2 alphanumeric Liquid Crystal Display (LCD) with a PC. First we show how to connect it to the parallel port and echo and handle keyboard input. Then we show how to use the LCD to display temperature from a DS1620 Digital Thermometer/Thermostat chip and set the HI & LO triggerpoints in it. All C source code is provided.

Please obtain additional information to read with this kit from the Internet. Start with these websites:

**beyondlogic.org** (excellent, must-visit site)  
**geocities.com/ResearchTriangle/1495/ee\_lcd.html**  
**iaehv.nl/users/pouweha/lcd.htm**  
**woe.de.vu**  
**pobox.com/~lcd\_info**

Do a search on yahoo.com for 'LCD parallel' and you will get 100's of links to follow-up.

## LIQUID CRYSTAL DISPLAYS

LCDs have become very popular over recent years for information display in many 'smart' appliances. They are usually controlled by microcontrollers. They make complicated equipment easier to operate.

LCDs come in many shapes and sizes but the most common is the 16 character x 2 line display. It requires only 11 connections – eight bits for data (which can be reduced to four if necessary) and three control lines (we have only used two here). It runs off a 5V DC supply and only needs about 1mA of current. The display contrast can be varied by changing the voltage into pin 3 of the display, usually with a trimpot.

To get the display working requires eight bits of data, a register select line (RS) and a strobe line (E). These are supplied from the PC printer port (refer to schematic).

A third input, R/W, is used to read or write data to/from the LCD. In this kit the R/W line is tied low so only writes to the LCD are possible (more on this later).

The eight bits of data are supplied from the printer port data lines and two printer port control lines are used for RS ('auto') and E ('strobe').

Basically the LCD has two registers, a data register and a control register. Data is written into the control register when RS is low and into the data register when RS is high. Data is latched into the LCD register on the falling edge of 'Enable'.

The sequence for writing to the LCD is:

1. To begin, E is low
2. Select the register to write to by setting RS high (data) or low (control)
3. Write the eight bits of data to the LCD
4. Set the Enable signal high then low again.

There are certain minimum timing requirements that must be followed when writing to the LCD, such as data setup times and Enable signal pulse width. These are in the order of tens and hundreds of nanoseconds.

Extra timing delays are also introduced by the cable 'capacitance' used to connect the kit to the PC printer port. Delay routines may be needed when using fast PCs to meet these timing requirements. The software supplied include these delay routines.

LCD displays have a 'busy' flag that is set while it is executing a control command. This flag is not accessible because the R/W line has been tied low (write). This is not such a problem because all commands have a maximum execution time. We simply wait for this time to pass before accessing the LCD again, eliminating the need to check the flag and thereby saving on an I/O line!

Eg. The 'Clear Display' command has an execution time of approx. 1.6mS. After sending this command to the LCD we simply wait 2mS before continuing on. This ensures that the command has finished.

Characters to be displayed are written into the LCD's 'data' RAM memory. The amount of RAM available depends on the type of LCD. The LCD used in this kit has 80 bytes of RAM. An internal address counter holds the address of the next byte to write to.

These 80 bytes are broken up into two blocks of 40 bytes. The address range of the first block is from 00h to 27h and the second block from 40h to 67h.

After power up and initialization address 00h is the first character of the top line and address 40h is the first character of the bottom line. The address counter is set to address 00h and is automatically incremented after each byte is written.

Only the first 16 bytes of each line are visible.

If we now start writing data to the LCD it will be stored from address 00h on but only the first 16 characters will be visible. To display the rest of the characters we need to 'scroll' the display.

Scrolling simply means changing the start address of each line. If we scroll left one position address 01h becomes the first character of the top line and address 41h the first character of the bottom line. Scrolling right does the opposite – addresses 27h and 67h become the first characters of the top and bottom lines.

As you can see, each block of addresses 'wrap' around in a circular manner. The next address after 27h is 00h; the next address after 67h is 40h. If we scroll left 40 times we will bring address 00h back into view as the last character on the top line.

## KIT 134. INTRODUCTION TO LCD'S

As you can see the address ranges of each block are not continuous. There is a gap of 24 bytes between the end of the first block (27h) and the start of the second (40h). So, if we keep writing characters to the display the first 40 will be stored in addresses 00h to 27h, the next 24 will be 'lost' and the next 40 will be stored at addresses 40h to 67h.

The internal address counter is directly accessible so we can set the next address to write to. This is how the characters are written to the second line of the LCD. For example, we can send 5 characters to the first line, set the address counter to 40h and then start writing characters to the second line.

### DS1620

The DS1620 is a Digital Thermometer/Thermostat IC from Dallas Semiconductor. The chip measures temperatures from  $-55^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$  ( $-67^{\circ}\text{F}$  to  $+257^{\circ}\text{F}$ ) in  $0.5^{\circ}\text{C}$  increments and has three "alarm" outputs. It requires no calibration or external components. You may get the data sheet from [dalsemi.com](http://dalsemi.com) or from my website software download page.

The three alarm outputs, designated  $T_{\text{HIGH}}$ ,  $T_{\text{LOW}}$  and  $T_{\text{COM}}$  function as follows:

- $T_{\text{HIGH}}$  goes high when the measured temperature is greater than or equal to a user-defined upper limit (TH).
- $T_{\text{LOW}}$  goes high when the measured temperature is less than or equal to a user-defined lower limit (TL).
- $T_{\text{COM}}$  goes high when the temperature exceeds the upper limit and stays high until the temperature falls below the lower limit.

These three outputs can be used to directly control heating and cooling appliances via suitable drive circuitry (eg. Relays and optocouplers).

Data is read from and written to the DS1620 via a 3-wire serial interface (DQ, CLK and RST). The user-defined upper and lower trip points are stored in the IC in non-volatile memory. This means that the trip points are 'remembered' even when the power is removed.

The DS1620 can be used in two modes – Three-Wire mode for interface to a computer or Stand-Alone mode. This last mode does not require a computer interface. With RST and CLK low, the chip will continuously monitor the temperature and set the alarm outputs accordingly.

This means that once you have programmed the upper and lower trip points the kit can be disconnected from the PC and you can use the alarm outputs for monitoring or control functions. You can even remove the chip from its socket and put it in your own temperature controller, using the kit simply as a 'DS1620 programmer'.

Temperature registers in the DS1620 are nine bits long and can be positive or negative values. Each reading has a resolution of  $0.5^{\circ}\text{C}$ . For positive temperatures, the most significant bit (bit 8) = 0 and bits 1-7 hold the integer value. Bit 0, is set, adds  $0.5^{\circ}$ .

For negative temperatures, bit 8 = 1 and bits 7-0 represent the two's complement of the temperature. To get the actual value, invert bits 7-0 then add 1.

The DS1620 is controlled via a set of 8-bit instructions. Depending on the instruction an 8 or 9 bit data value is either read from or written to the chip. All instructions and data are transferred serially starting with the least significant bit D0.

### To read or write a value to the DS1620:

1. To begin, RST is low and CLK high.
2. RST high
3. CLK low
4. Set DQ equal to Bit 0 of instruction
5. CLK high
6. Repeat steps 3, 4 and 5 for remaining instruction bits

### To write to the chip:

7. Repeat steps 3, 4 and 5 for Bits 0-7 or 0-8 of the data to be written
8. RST low for at least 5ms

### To read from the chip do Steps 1 through 6 above then:

7. Set the AUTO signal high so that data can be read from the DS1620
8. CLK low. DQ now outputs the data to be read (LSB first)
9. Read and store DQ
10. CLK high
11. Repeat steps 8, 9 and 10 for the remaining bits of data to be read
12. RST low

As with the LCD, certain minimum timing requirements must be observed between each of these steps. In particular, RST must remain low for at least 5ms after writing data to the DS1620 (step 8). This gives the chip's internal EEPROM time to store the data. Also, the DS1620 needs 1 second to execute a 'start-convert' instruction. So, after starting a temperature conversion, you must wait 1 second before reading the result.

# KIT 134. INTRODUCTION TO LCD'S

## THE PC PARALLEL PORT

There are three common address ranges used for parallel port interfaces, as follows:

|             |  |
|-------------|--|
| 3BCh - 3BFh | Parallel interface on monochrome or TTL video card |
| 378h - 37Fh | Parallel interface 1                               |
| 278h - 27Fh | Parallel interface 2                               |

Table 1

The standard PC parallel port consists of three registers, each referred to by their functional name. These registers occupy the first three addresses in the range. The first address in the range is referred to as the **base address**.

|                         |                  |            |
|-------------------------|------------------|------------|
| <b>Data register</b>    | Base address     | Read/Write |
| <b>Status register</b>  | Base address + 1 | Read only  |
| <b>Control register</b> | Base address + 2 | Read/Write |

For example, for parallel interface 1, the data register would be at address 378h, status register at address 379h and control register at address 37Ah.

**Note** that the data and control registers can be written to and read from. This does not mean that they are bidirectional. It means that the outputs of these registers can be read back to check their status.

During the system boot-up sequence, the BIOS tests for the presence of parallel ports in order, according to Table 1. The first port found becomes LPT1, the second LPT2, and so on. The BIOS stores the base address of each port found in a table in the BIOS variable segment of memory, as follows:

| Address    | Contents             |
|------------|----------------------|
| 0040:0008H | Base address of LPT1 |
| 0040:000AH | Base address of LPT2 |
| 0040:000CH | Base address of LPT3 |
| 0040:000EH | Base address of LPT4 |

(If address=0 then there is no port for that LPT number.)

Every signal line on the port, whether it is used for read/input or write/output, is allocated one particular bit at one of the three addresses Base, Base+1 or Base+2. The logic state of the bit indicates the state of the wire (0V or 5V.) The following table shows each printer port signal used by the kit and the register, address and bit position associated with it. The last three signal names are preceded by a minus sign. This means that the signal is active low. Writing a high to the bit causes the signal to go low.

| Signal  | Reg.    | Addr.  | Bit   | Direction |
|---------|---------|--------|-------|-----------|
| D0 - D7 | Data    | Base   | 0 - 7 | Output    |
| ACK     | Status  | Base+1 | 6     | Input     |
| -STROBE | Control | Base+2 | 0     | Output    |
| -AUTO   | Control | Base+2 | 1     | Output    |
| INIT    | Control | Base+2 | 2     | Output    |
| -SLCTIN | Control | Base+2 | 3     | Output    |

## ASSEMBLY INSTRUCTIONS

Use the component overlay on the PCB to place the components, in the following order:

- Resistors and diodes
- Wire link. Use one of the resistor offcuts
- Monobloc capacitor C3 and IC socket
- Trimpot, voltage regulator and 'X3' header pins
- Electrolytic capacitors. Make sure the electrolytics are inserted the right way around
- LCD socket, power jack and D25 connector.

Fit and solder the 14-way male header strip to the underside of the LCD, keeping it at right angles to the LCD. The header strip is inserted in the leftmost 14 holes (looking from front) on the LCD.

The LCD is held in place by the socket pins on the top and two screws on the bottom. Fit the two screws to the bottom mounting holes of the LCD and secure with nuts. Add a second nut to each screw about half way up. See the jpg photos on the floppy disk.

Insert the LCD into its socket on the PCB with the screws protruding through the matching mounting holes. Adjust the nuts so that the LCD is parallel to the PCB. Fit and tighten the last two nuts to secure the LCD to the PCB.

## CIRCUIT DESCRIPTION

The circuit is quite straightforward. IC2 is a 5V regulator the powers the kit. Diode D2 protects the kit against reverse polarity on the DC input jack. The trimpot is used to adjust the contrast and viewing angle of the LCD.

Resistors R2 and R3 hold the DS1620 CLK and RST inputs low when the kit is disconnected from the PC. This allows the chip to work in "stand-alone" mode.

Resistor R1 and diode D1 convert the separate data input and output lines from the PC to the single bidirectional data line on the DS1620. When writing to the chip, R1 pulls DQ high when AUTO is high. DQ is pulled low via D1 when AUTO is low. When reading from the DS1620, AUTO must be set high first. This high is blocked by D1 so that the DQ output can drive ACK.

Connection to the DS1620 alarm outputs is via connector X3. The DC input voltage also appears on

# KIT 134. INTRODUCTION TO LCD'S

this connector as does ground. This is useful for powering devices driven by the alarm outputs directly from the kit.

## TESTING

**Do not insert the LCD or the DS1620 IC.** Connect a DC supply to the power jack. A 12V DC plug pack will do nicely. Measure the 5V output from the regulator. The easiest place to measure is between pins 4 and 8 on the IC socket. If all is well, remove power and insert the IC and LCD.

Apply power again. There should be a series of 'black squares' on the LCD. If not then adjust the trimpot until visible. You are now ready to connect to a PC and run the software.

## SOFTWARE DESCRIPTION

The programs are supplied on the floppy disk but may also be downloaded from our software download webpage. Both programs are written using Borland Turbo C for DOS. They will run quite happily in a DOS box under Windows 95/98.

The parallel port defaults to LPT1 but other ports may be chosen by a command line switch. See the readme.

### 1. K134LCD.EXE

A simple program that allows the user to enter a message to be displayed on the LCD. The message can be up to 40 characters long. The message can be scrolled left or right and the scrolling rate can be varied.

After running the program a menu is displayed on the PC from which you choose the required function.

The message is only displayed on the top line of the display. If scrolling and the message is less than 40 characters long then a number of spaces will be displayed before the message starts again.

### 2. K134TEMP.EXE

A program that displays the current temperature as well as the upper (TH) and lower (TL) temperature 'trip' points. The value of the trip points can be changed. The user can choose between Celsius or Fahrenheit display.

The DS1620 is configured to perform continuous temperature conversions. Current temperature and trip point data is continually read from the DS1620 and compared with the previous reading. If any of the three values have changed then the display is updated.

## IF IT DOES NOT WORK

Poor soldering ("dry joints") is the most common reason for the circuit not working. Check all soldered joints carefully under a good light. Re-solder any that look suspicious. Check that all components are in their correct position. Are the electrolytic capacitors and diodes the right way round? Did you add the link?

## REFERENCES

The Internet is the place today to get information. But here are some magazine References on LCD's:

Rick Walters, "A liquid crystal display driven from a PC printer port", *Silicon Chip March 1998*.

Mark Roberts, "A PC-controlled thermometer/thermostat", *Silicon Chip June 1997*.

Jan Axelson, "Temperature Monitoring With a Synchronous Serial Link", *MicroComputer Journal March/April 1995*.

## Web Address & Email

You can email us at peter@kitsrus.com if you have any problems or requests. Information on other kits in the range is available from our Web page at:

**kitsrus.com**

## PARTS LIST - KIT 134

### Resistors (0.25W carbon)

10K ..... R1-4 ..... 4

10K trimpot ..... VR1 ..... 1

### Capacitors

100nF monobloc ..... C3 ..... 1

10uF 25V electrolytic ..... C2 ..... 1

100uF 25V electrolytic ..... C1 ..... 1

### Semiconductors

1N4004 ..... D2 ..... 1

1N4148 ..... D1 ..... 1

78L05 ..... IC2 ..... 1

+5V regulator, TO-92 package

DS1620 ..... IC1 ..... 1

Digital Thermometer and Thermostat

Liquid Crystal Display ..... LCD 16 x 2 characters ..... 1

### Miscellaneous

2.5mm DC jack ..... X2 ..... 1

PCB mounting

D25 connector ..... X1 ..... 1

PCB mounting, right-angle, male

5-pin SIL header ..... X3 ..... 1

8-pin IC socket ..... for IC1 ..... 1

14-pin SIL socket ..... for LCD ..... 1

14-pin SIL header ..... for LCD ..... 1

Screw, 2.6mm x 18mm long... 2

Nut, 2.6mm ..... 6

PCB, K134 ..... 1

Floppy disk ..... 1

# K134. PC CONTROLLED LCD & THERMOSTAT

