# 5 Liquid Crystal Displays

## 5.1 Introduction

Liquid Crystal Displays (LCD) are widely used as output devices. This because they consume little current as compared to LEDs and other display devices. Also they can be made physically small and have colour capabilities.

While bare LCD displays are available, it is quite troublesome to interface to one. LCD *modules* on the other hand, incorporate a graphic processor and interface electronics. In doing so modules have the added advantages of :

a) Having relatively powerful display functions and can be easily controlled and interfaced.

b) Not requiring a need to refresh the display.

Selecting a LCD module involves two basic design decisions.

1) What size and format is required to display the desired information.

2) What optical characteristics will look best in the package and attract the userto the product. This chapter mainly deals with the alphanumeric (A/N) or character type modules.

Alphanumeric modules display characters, numerals, symbols and some limited graphics. They are normally connected to a host processor through a parallel data bus, although serial interfaces are available.

Display control features such as Character Generation, Display RAM Addressing, Cursor Scrolling, Blanking, and are all included.User programmable fonts are supported.

There are various LCD modules available on the market from many manufacturers. The LCDs come in various sizes, with 1-4 lines, 16 to 40 characters per line, and 5x7 or 5x10 dot display fonts. Character height spans 9.130" (3.31 mm) to 0.500" (12.71 mm). Most formats are available in a variety of packages to meet various mounting requirements. Multi-line models offer the best value when analyzed by a "cost per character" basis. By selecting a backlight option displays are visible both day and night. Extended temperature modules are available which operate between -20 and +70 C.

For requirements of more than 4 lines or 40 characters across, select a graphic formatted

module. Graphic modules are also used when different sized characters are needed, and when special fonts such as Chinese or Arabic are required.

### 5.1.1 LCD Fluid Types

LCDs use a fluid dye which type determines the contrast ratio, viewing angle, and temperature range. There are 3 basic types of dye: TN (Standard type), NTN (high contrast type), and STN (premium high contrast type). Many TN and NTN models are available in extended temperature range.

### 5.1.2 Character LCD Controller

Software determines what, how and where data is displayed on the LCD. Most character modules use the Hitachi HD44780 or equivalent controller IC. Some of its features are:

Built-in character generator with 192 character modified ASCII character set
Ability to program up to 8 user defined characters.
Bi-directional 8 or 4 bit bus interface
80 character RAM
Automatic reset on power up
Wide range of instruction functions including:
Display clear, ON/OFF, Cursor positioning, Display or cursor shift on data entry

## 5.2 LCD Hardware overview

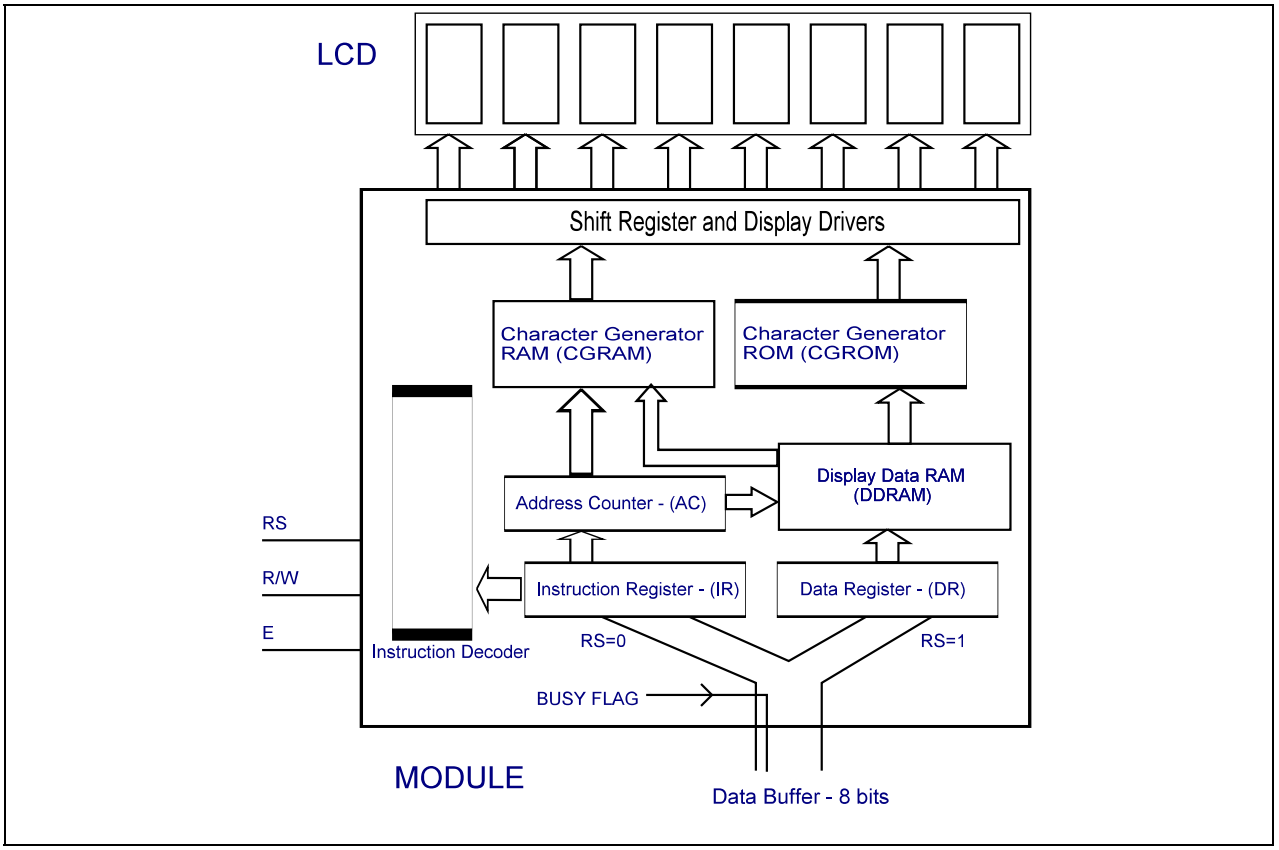



**Figure 1** Block Diagram of LCD module

General operation

The module interfaces with the host processor through a data bus. By controlling the RS pin, data goes to either of two 8-bit registers: an instruction register (IR) or a data register (DR). The IR stores instruction codes, such as display clear and cursor shift, and address information for display data RAM (DDRAM). The IR can only be written from the host processor.

When the host processor writes to the Data Register, this register will temporarily hold the data. After that, it will be automatically transferred to the DDRAM by an internal operation.

When address information is written into the IR, data is read and then stored into the DR from DDRAM by an internal operation. Data transfer from the module is completed when the host processor reads the DR.

When reading from the module, the DR will hold data read from DDRAM. After the host processor does a read, a special feature of the module is that, data at the next address in DDRAM is automatically read. This is sent to the DR in preparation for the next read.

Data Display RAM

As implied by the name, DDRAM is used to store data to be displayed. These are 8-bit character “modified” ASCII codes. It can handle 80 x 8 bits, or 80 characters. If the actual number of characters that can be viewed is less than this, the unused DDRAM can be used as general data RAM.

Character Generator ROM/RAM

CGROM holds the actual bit patterns to be shown on the display. In the Alphanumeric modules we are considering, these patterns are constructed using a 5 by 8 bit matrix. Internal logic in the LCD module takes data from DDRAM. It then calculates where the bit pattern for that data is stored in CGROM. These patterns are shifted out to the display drivers and there to the individual LCD display elements.

The CGROM in a standard module has 160 characters in a 5x8 bit pattern format and 32 characters in 5x10 pattern.

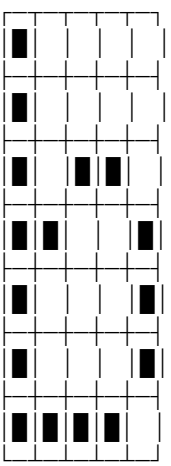
For added flexibility, users can define up to 8 special character or symbols in a 5x8 format (for 5x10, only 4 characters). These bit patterns are stored in another type of RAM known as Character Generator RAM. Once programmed, the newly formed characters may be accessed as if they were in the "normal" CGROM. It must be reprogrammed if power to the module is interrupted. Further information on how to make use of CGRAM may be found in the LCD module databook.

Relationship between DDRAM, CGROM

To illustrate, let's say the value 62H (ASCII ‘b’) is sent to the Data Register. From there it goes into the DDRAM at an address specificed by the Address Counter. The module logic will access the bit pattern stored in the CGROM. This is shifted out and displayed on the actual display as shown below. A ‘1' will turn on the pixel in a row. Note that characters are shown as a 5 by 7 dot matrix but stored as an 8 byte (8 by 8) array. A complete list of character codes and bit patterns is shown later.

```
DDRAM       CGROM/CGRAM      Actual LCD

62H         00010000
            00010000
            00010110
            00011001
            00010001
            00010001
            00011110
            00000000
```



Busy Flag (BF)
When the busy flag is 1, the module is in the internal operation mode, and the next instruction will not be accepted. In this state, fi the host processer performs a read, the busy flag is output to DB7. The next instruction must be written after ensuring that the busy flag is 0.

Address Counter (AC)
The address counter (AC) is used to access data in DDRAM or CGRAM. When a "set address" instruction is sent to the Instruction Register (IR), the address information is sent from the IR to the AC. The type of instruction determines whether DDRAM or CGRAM is addressed.

After writing to DDRAM, the AC is automatically incremented by 1. When reading from DDRAM, it is decremented by 1. The AC contents are then output to DB0 to DB6. Note that the AC register holds only 7 bits.

When addressing CGRAM, for example, to load in a user defined character, the cursor or blinking effect will appear. This effect should be ignored as the AC is pointing to a CGRAM address and thus does not affect the DDRAM.

Through the register selector (RS) signal, these two registers can be selected as shown:

| RS | R/W | Operation |
|---|---|---|
| 0 | 0 | write to IR to start an internal operation (display clear, etc.) |
| 0 | 1 | Read busy flag (DB7) and address counter (DB0 to DB6) |
| 1 | 0 | write to DR for display (DR to DDRAM) |
| 1 | 1 | read from DR the contents of DDRAM (DDRAM to DR) |

Timing Generation Circuit
The timing generation circuit generates timing signals for the operation of internal circuits such as DDRAM and CGROM. RAM read timing for display and internal operation timing by the host processor access are generated separately to avoid interfering with each other. Therefore, when writing data to DDRAM, for example, there will be no side effects, such as flickering, in areas other than the display area.

Cursor/Blink Control Circuit
The cursor/blink control circuit generates the cursor or character blinking. The cursor or the blinking will appear with the digit located at the display data RAM (DDRAM) address set in the address counter (AC). For example, when the address counter is 08H, the cursor position is displayed at DDRAM address 08H. When a character is written to the Data Register, the ASCII character will appear at the cursor position.

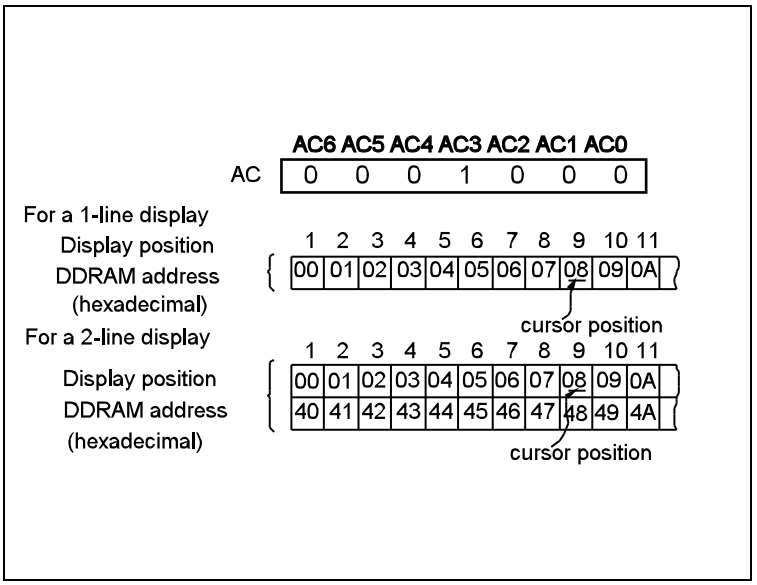



**Figure 2** Cursor position

## 5.2.1 Relationship between DDRAM and Display Characters

The DDRAM has a capacity of 80 characters. If the physical display has less than 80 characters, what is on the display is a "window" on the DDRAM. The following address diagrams on the next page show DDRAM addresses as they appear after a Clear Display or Return Home instruction, or when Entry Mode Set instruction has bit S=0.

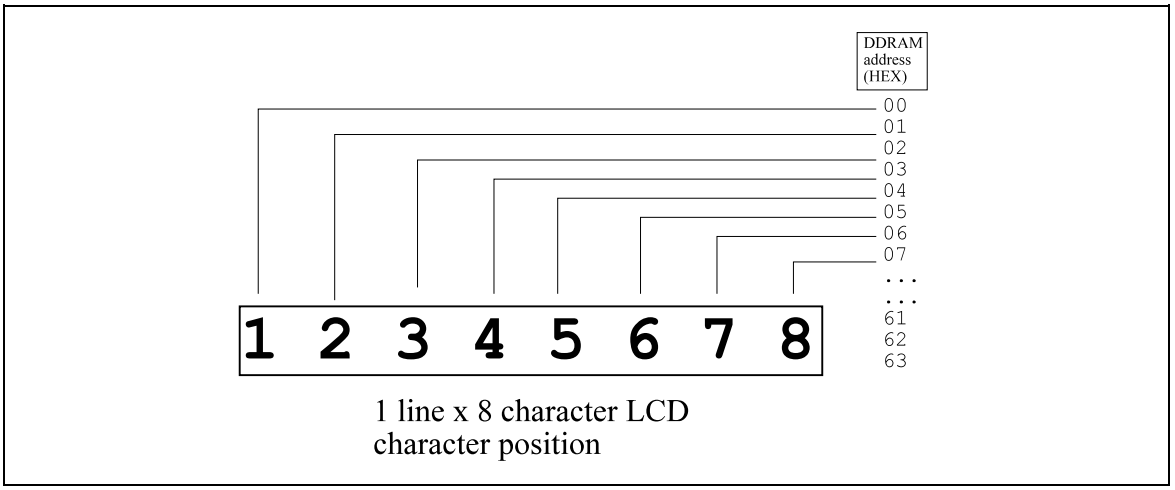



**Figure 3** One Line Display - Character Position vs. DDRAM Address

When there are fewer than 80 display characters, the display begins at the starting position. From the diagram above, we see that DDRAM memory location 00 corresponds to character position 1.
So when we write to DDRAM address 00H, it will appear at display position 1. Note that characters in DDRAM memory locations higher than 08 are not visible for this 1 line by 8

display.

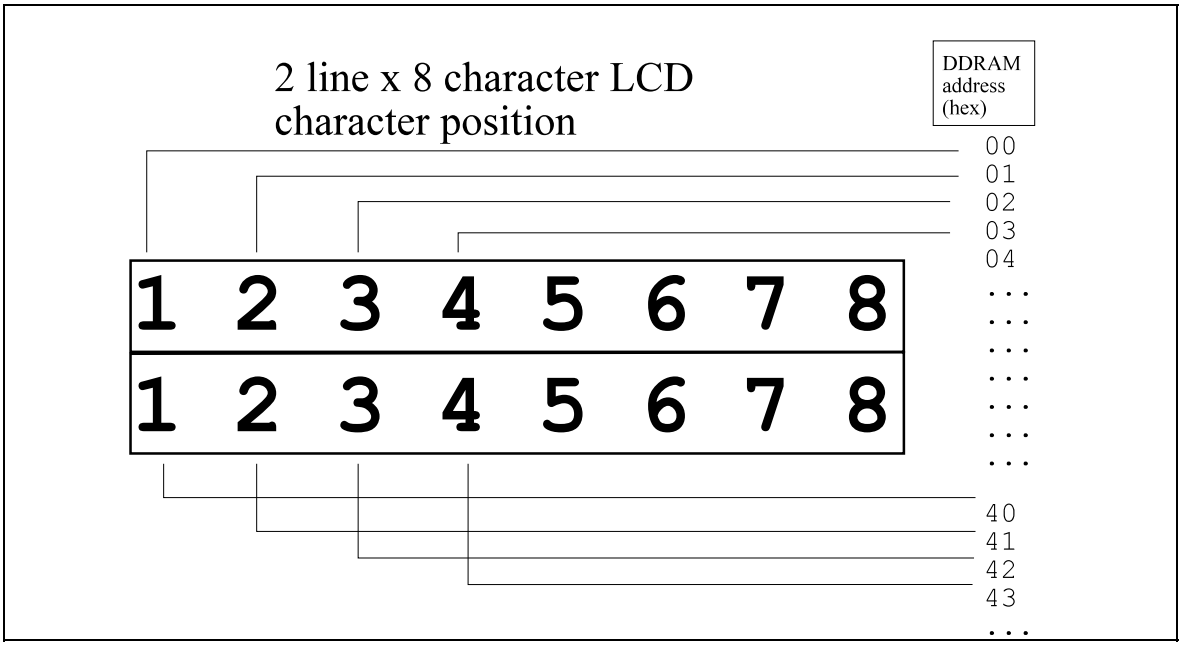



**Figure 4** Two Line Display - Character Position vs. DDRAM Address

In a two line display, the first character on the next line is *not* "one more" than the last character on the previous line. In the 2 line by 8 character module above, what we think is the 9th character (first character on line 2) is *not* found at DDRAM address 08. This first character on line 2 is actually at DDRAM address 40h. To put data on the second line, a Set DDRAM Address instruction must be sent.

It is possible to shift the display so that the first character does not correspond to DDRAM location 0. Thus This we can make the characters rotate through the display when data is entered. It also lets small displays, for example 2x16, to have data stored in non-visible areas of the DDRAM and have them shifted in to be viewed. This is is covered in the discussion on "shift mode".

### 5.2.2 Character code and bit patterns

As mentioned before, typical LCD modules store the bit patterns in CGROM. They use a modified ASCII code. For example, the character 'A' is binary 0100 0001 or 41H. This is shown in the diagram below. Character codes E0H to FFH are 5x10 bit patterns.

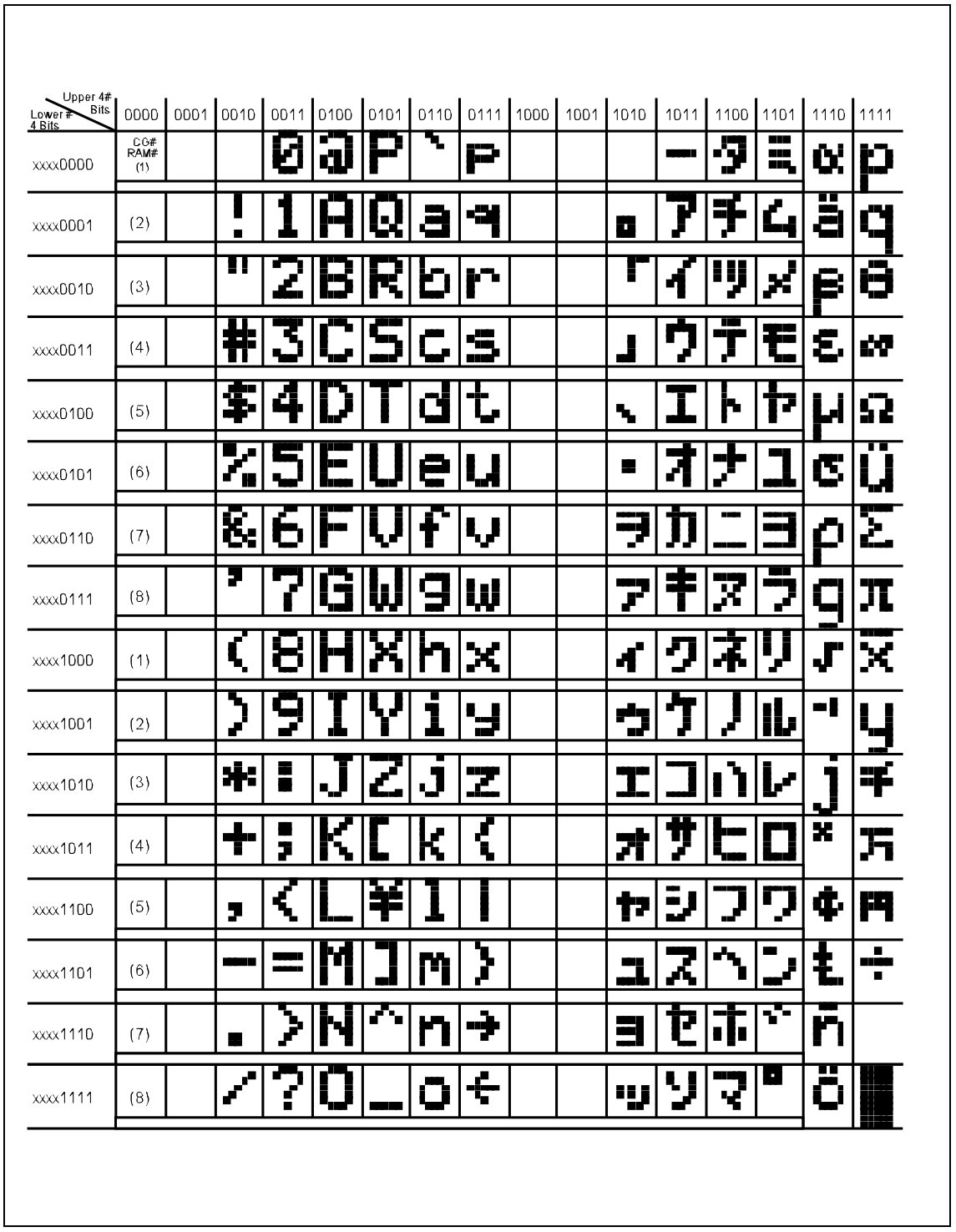



**Figure 5** Correspondence Between Character Code and Character Pattern

## 5.3 LCD Software control

Only the Instruction Register (IR) and the Data Register (DR) of the LCD module is accessible to the host processor. By writing to these registers, we control how data is to be displayed on the module.

### 5.3.1 General Considerations - timing

An LCD module takes a relatively long time to execute instructions. On the average, this is around 40 μs. When an instruction is being processed only the Busy Flag/address read instruction can be executed. The busy flag is set to 1 and will go to logic 0 when done.

Busy Flag vs standard time delay
Most of the time, data is written to the LCD module and keeping track of its state is done by the host processor. In this case, it is not necessary to read from the LCD very often, or not at all! We can dispense with having to read the Busy Flag, and use a standard time delay of 40 μs for most instructions. In many cases, this will save the use of a buffer (or needing a bidirectional port) to accommodate the read function.

There are four categories of instructions namely:

- setting the modes of operation, such as display format, data length, etc.
- set internal addresses, both for DDRAM and CGRAM
- Perform data transfer with internal RAM
- Perform miscellaneous functions

Normally, the most commonly used instructions are those that perform data transfer to DDRAM. The autoincrement (or decrement) feature of the AC register mentioned earlier, makes it easier to program the module.

### 5.3.2 Default Initialisation

Before programming the LCD module, we must know its initial state when power is turned on. The following instructions are executed during the initialization period. This lasts for 15 ms after the power supply rises to 4.5V. The busy flag (BF) will be at logic one until the initialization ends when it will be at logic zero. If the rise time of the power supply meets the criteria below, the module will default to the following functions via an internal initialization routine:

1. Clear Display

| | | |
|---|---|---|
| 2. | Function Set | DL=1: 8 bits interface<br>N=0: 1 line display<br>F=0: 5x7 dot font |
| 3. | Display ON/OFF control | D=0: Display OFF<br>C=0: Cursor OFF<br>B=0: Blink OFF |
| 4. | Entry Mode Set | I/O=1: +1 increment |
| 5. | DDRAM is selected | |

If the power is not applied successfully, the initialisation routines will fail. Also, if different parameters are required such as for a 2 line display are required, the host processor has to perform the initializing. We will look at some examples later.

After initialising, we should have a clear display with a flashing cursor in the upper left position. The cursor will then increment to the right with each DDRAM write command.

### 5.3.3 INSTRUCTION TABLE

The following is a summary of the instructions that can be sent to the LCD module and their typical timings as well. Some instructions can perform several operations at one time depending on the setting of individual bits. These bits are formed into a data byte which is sent to the Instruction Register.

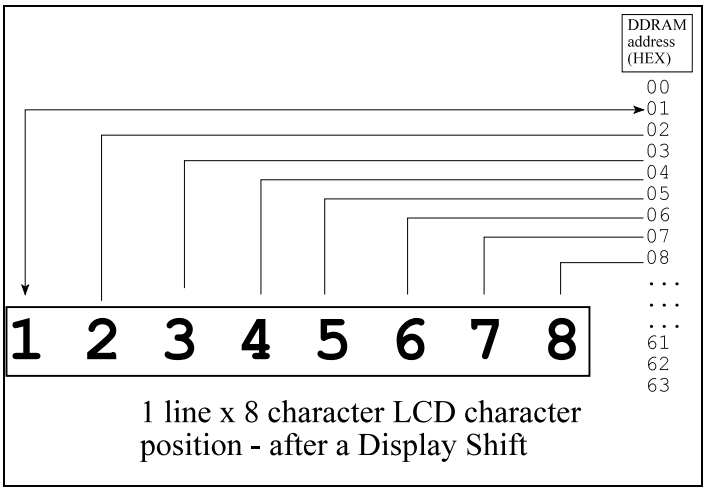
| Instruction | RS | R/W | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 | Description | Execution Time** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Clear Display | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | Clears Display and returns cursor to the Home Position (Address 00) | 80uS = 1.64mS |
| Return Home | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | * | Returns cursor to Home Position. Returns shifted display to original position. Does not clear display | 40uS = 1.6mS |
| Entry Mode Set | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | I/D | S | Sets DD RAM counter to increment or decrement (I/D) Specifies cursor or display shift during to Data Read or Write (S) | 40uS |
| Display ON/OFF Control | 0 | 0 | 0 | 0 | 0 | 0 | 1 | D | C | B | Sets Display ON/OFF (D), cursor ON/OFF (C), and blink character at cursor position | 40uS |
| Cursor or Display Shift | 0 | 0 | 0 | 0 | 0 | 1 | S/C | R/L | * | * | Moves cursor or shifts the display w/o changing DD RAM contents | 40uS |
| Function Set | 0 | 0 | 0 | 0 | 1 | DL | N | F | * | * | Sets data bus length (DL), # of display lines (N), and character font (F) | 40uS |
| Set CG RAM Address | 0 | 0 | 0 | 1 | $A_{CG}$ | | | | | | Sets CG RAM address. CG RAM data is sent and received after this instruction | 40uS |
| Set DD RAM Address | 0 | 0 | 1 | $A_{DD}$ | | | | | | | Sets DD RAM address. DD RAM data is sent and received after this instruction | 40uS |
| Read Busy Flag & Address | 0 | 1 | BF | AC | | | | | | | Reads Busy Flag (BF) and address counter contents | 1uS |
| SIZE=2>Write Data from DD or CG RAM | 1 | 0 | Write Data | | | | | | | | Writes data to DD or CG RAM and increments or decrements address counter (AC) | 40uS |
| Read Data from DD or CGRAM | 1 | 1 | Read Data | | | | | | | | Reads data from DD or CG RAM and increments or decrements address counter (AC) | 40uS |
| I/D=1: Increment<br>S=1: Display Shift on data entry<br>S/C=1: Display Shift (RAM unchanged)<br>R/L=1: Shift to the Right<br>DL=1: 8 bits<br>N=1: 2 Lines<br>F=1: 5x10 Dot Font<br>D=1: Display ON<br>C=1: Cursor ON<br>B=1: Blink ON<br>BF=1: Cannot accept instruction | | | I/D=0: Decrements<br>S=0: Cursor Shift on data entry<br>S/C=0: Cursor Shift (RAM unchanged)<br>R/L=0: Shift to the Left<br>DL=0: 4 bits<br>N=0: 1 Line<br>F=0: 5x7 Dot Font<br>D=0: Display OFF<br>C=0: Cursor OFF<br>B=0: Blink OFF<br>BF=0: Can accept instruction | | | | | | | | Definitions:<br>DD RAM: Display data RAM<br>CG RAM: Character generator RAM<br>$A_{CG}$: CG RAM Address<br>$A_{DD}$: DD RAM Address(Cursor Address)<br>AC: Address Counter used for both DD and CG RAM Address | Execution Time changes when Frequency changes per the following example:<br>If $F_{CP}$ or $f_{osc}$ is 27 KHz<br>40uS x 250/270 = 37uS |

* Don't Care **(when $F_{cp}$ or $f_{osc}$ is 250KHz)

## 5.3.4 Special operation modes

Four bit operation

We have been describing how data can be sent as 8 bit bytes. LCD modules are able to receive this byte as two 4 bit nybbles. In many cases this can save an extra 8 bit latch, not to mention decoding resources. This slight increase in complexity can be hidden in the host processor's output routine. The main user program then calls this routine with the standard 8 bit byte, leaving the two 4 bit transfer to be handled by the routine. We also need to initialise the module into 4 bit mode.




Display Shift

DDRAM can be used for displays such as for advertising when combined with the display shift operation. When instruction S=1, the display is shifted. What is displayed depends on the Entry Mode Set instruction. The display shift operation changes only the display position with DDRAM contents unchanged. After a Display Shift Right command, we can see that DDRAM address 01 now corresponds to display character 01 and the character at DDRAM address 08 is now visible on this 1 line by 8 display. Similarly, a Display Shift Left will bring the last character of DDRAM into view. For a two line display, both lines will be shifted in the same way.

## 5.4 Further initialisation considerations

With a proper understanding of the instructions available, we can look in greater detail at the initialisation routines. Remember, this is only necessary when the default initialisation is not sufficient. These are placed in the chapter appendix.

Sample display program

As a summary, the following shows the commands sent to set up a 2 line LCD module with 5x7 font, 8 bit data transfer, display address increment and shifting the cursor to the right when writing. This will also display a simple message.

| Step # | Instruction | | | | | | | | | Display | Operation |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | RS | R/_W | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 DB0 | | |
| 1 | Power supply on (the HD44780 is initialized by the internal reset circuit) | | | | | | | | | | Initialized, no display |
| 2 | Function set | | 0 | 0 | 1 | 1 | 1 | 0 | x x | | Sets to 8-bit operation and select 2-line, 5x7 dot font. (after this, no. of lines and character fonts cannot be changed.) |
| 3 | Display on off control | | 0 | 0 | 0 | 0 | 1 | 1 | 1 0 | _ | Turns on display and cursor. Entire display is in space mode because of initialization. |
| 4 | Entry mode set | | 0 | 0 | 0 | 0 | 0 | 1 | 1 0 | _ | Sets mode to increment the address by one and to shift the cursor to right at the time of write to the DD/CGRAM. |
| 5 | Set 00H to DDRAM address | | 1 | 0 | 0 | 0 | 0 | 0 | 0 0 | _ | Start from the first character of the first line |
| 6 | Write data to DDRAM | | 0 | 1 | 0 | 0 | 1 | 0 | 0 0 | H_ | Writes “H”. The cursor is incremented by one and shifted to the right |
| 7 | Write data | | 0 | 1 | 0 | 0 | 1 | 0 | 0 1 | HI_ | Writes “I”. |
| 8 | Set DDRAM address to 2nd line | | 1 | 1 | 0 | 0 | 0 | 0 | 0 0 | HI _ | Cursor goes to second line |
| 9 | Writes data | | 0 | 1 | 0 | 0 | 1 | 1 | 1 1 | HI O_ | Writes “O” |
| 10 | Write data | | 0 | 1 | 0 | 0 | 1 | 0 | 1 1 | HI OK_ | Writes “K” |
| 11 | Write data | | 0 | 0 | 1 | 0 | 0 | 0 | 0 0 | HI OK _ | Writes space |

**LCD 8 Bit Operation, 8-Digit x 2-line Display Example**

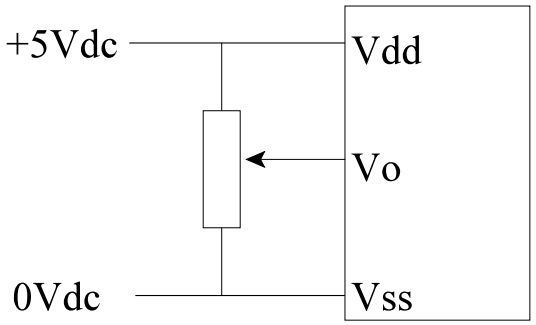
## 5.5 LCD Hardware Design

Here we consider the power supply, lighting control and the logic signals.

### 5.5.1 POWER SUPPLY REQUIREMENTS

Power Supply to Module Connection
Modules require +5V at 1 to 10 milliamps. Extended temperature and some high contrast modules require -5V, also at low current. Inexpensive ICs convert +5V to -5V efficiently. If the display has backlighting, required power must also be budgeted.

A module's logic circuits have 3 connections to the power supply: VDD (+5VDC); VSS (Ground); and $V_O$, viewing angle adjustment, sometimes called contrast or bias control. The diagrams show typical connections. Contrast can also be controlled digitally with a digital potentiometer or a Digital Analog Converter.

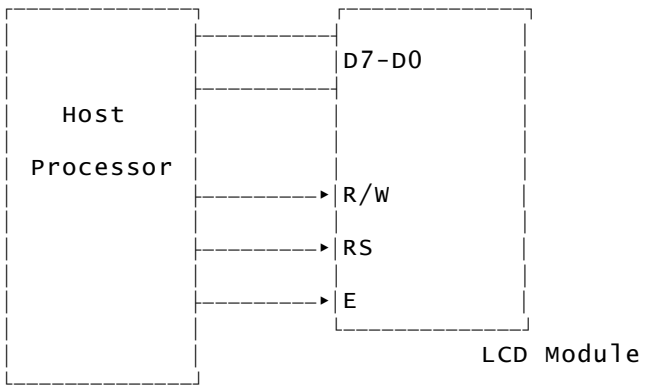



Power connections to LCD module

### 5.5.2 Logic connections

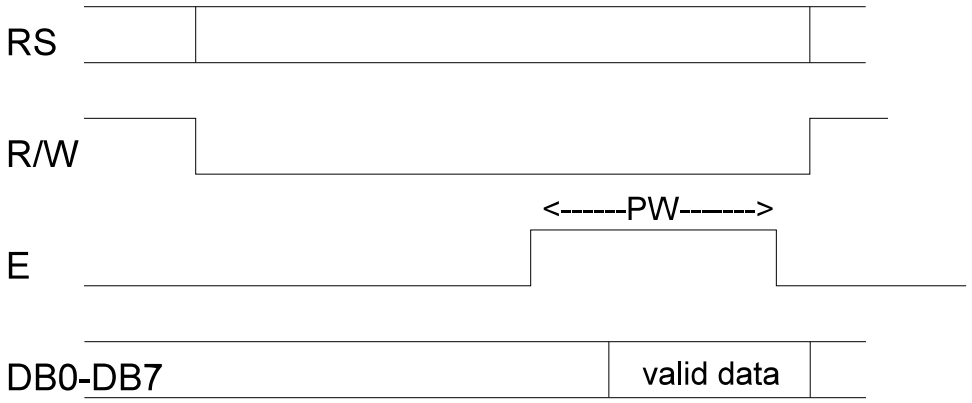
As we have seen, the LCD module appears as two 8 bit registers to a microprocessor system. Additionally, there are three control pins, making a total of 11 pins or two registers. Generally, interfacing the module to an existing system involves:

a) applying the proper viewing angle voltage to the display's V0 pin.
b) connecting the module to the host processor - directly to a port, or through buffers and latches
c) developing a strobe signal for the "E" signal
d) applying the appropriate "RS" and "R/W" signals to modules




LCD 8-bit interfacing diagram

The 8 bit mode is more straightforward and the software control is simpler. It needs 8 bits for data and 3 control signals. The timing is shown below. We need two 8 bit devices to interface to it.




Timing diagram for LCD

Timing Aspects of Alphanumeric LCDs

They are classified as slow peripherals. The Enable ("E") signal is the key signal line. This signal clocks the data and control signals into the LCD module. The E signal must be a clean, positive going digital strobe, which is active while data and control information are stable and true.

The two control lines, RS and R/W, must be set up before the activation, or rise, of E. These signals must remain stable for 10ns after the fall of "E".

A likely choice for host connections are: RD and WR strobes (or the R/W line) and to connect the E pin as a “chip select” line. But the E strobe must be 450ns wide (PW) minimum. Most modern processors operate at higher speeds than this and it is not practical to slow down the host for this purpose. Often we latch both the data and control information, toggling the E signal in this way. Of course if a port is free, this may be used as well.

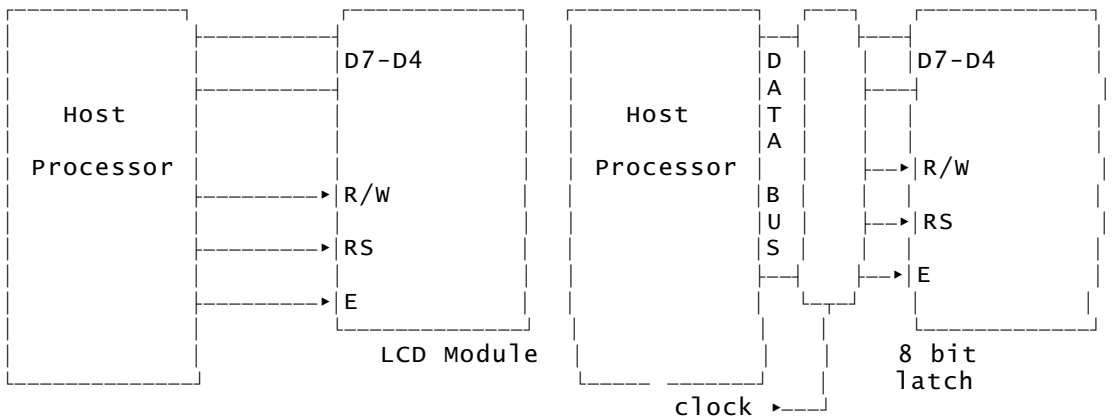
When a parallel port supplies RS, R/W and E, all these lines should not change together. This would result if a single instruction was employed and would surely violate the set-up requirement. Instead a second instruction must independently set the E bit high, after the other lines are set.

Another option is to tie the R/$\overline{W}$ and RS pins to the host's address lines which is set up earlier in the machine cycle. The data bus must set-up 195nS prior to the fall of "E". These lines must hold for at least 10nS after E falls. Most host strobes should meet these requirements without difficulty.

Normally E strobes would be approximately 40 microseconds apart - which is the maximum display throughput. (See Instruction Table for complete list of execution times).

Four bit connection

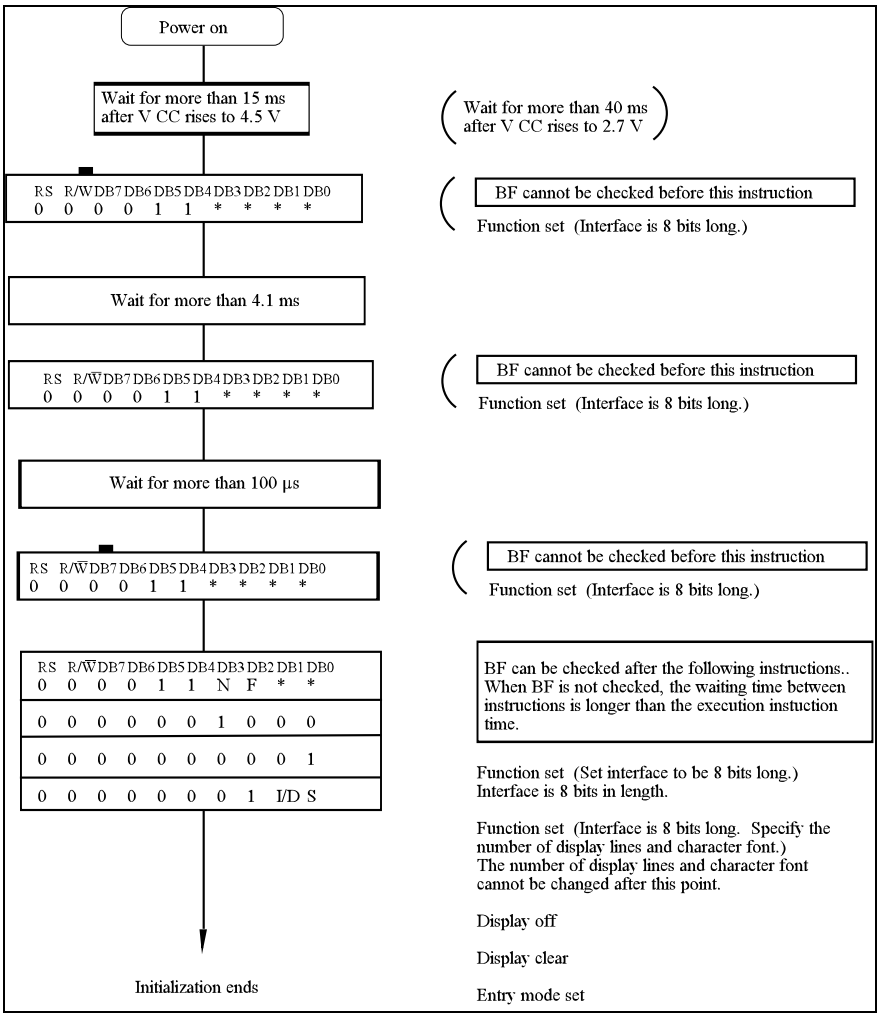
An important advantage to using a four bit transfer mode is that it only needs 4 data lines and 3 control lines. So one 8 bit port will be enough. This reduces the number of 8 bit devices used. There is only a slight increase in software complexity. The following shows two ways of connecting up. One is directly to a port, the other through a latch.




Four bit data transfer Port/Latch Interface

The data transfer between the module and the host processor is completed after the 4-bit data has been transferred twice. The four high order bits (DB4 to DB7) are transferred first, then the four low order bits (operation, DB0 to DB3). If the busy flag is used, it must be checked after the 4-bit data has been transferred twice. This being one instruction. Two more 4-bit operations then transfer the busy flag and address counter data.

## 8 bit initialisation




In summary, there should be a 15ms delay after power up before initialization begins. Then there are 4.1ms and a 100 µs delays where the "3X" codes are sent. The following are some sample initialisation codes for various configurations (all in hexadecimal)

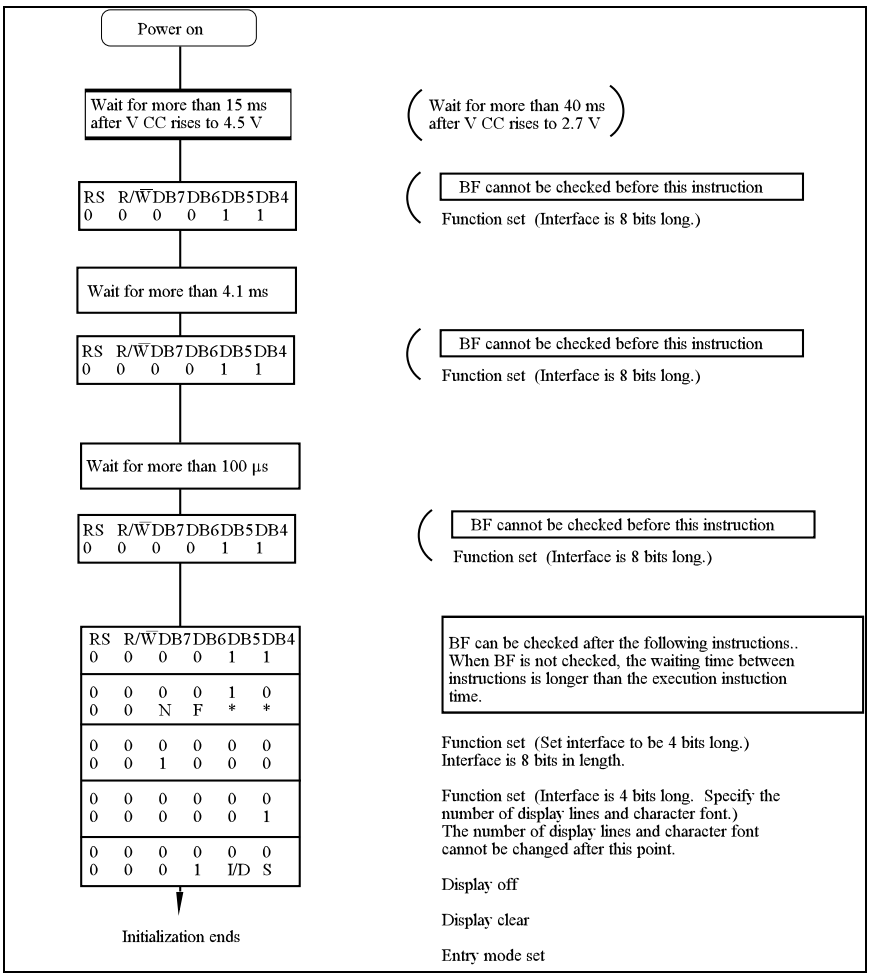
1 line display with 5x7 font 30, 30, 30, 30, 08, 01, 06

1 line display with 5x10 font 34, 34, 34, 34, 08 ,01, 06

2 line display with 5x7 font: 38, 38, 38, 38, 08, 01, 06

Of course, a 0E code should be sent to turn on the display!

Initialization for 4-bit operation




The modules will operate from a 4-bit wide data bus. Data is transferred over data lines D7-D4. D3-D0 may float. An 8-bit hex code is sent one nybble at a time, the higher nybble first. The function set in the initialization routine must change to accommodate this mode. A recommended routine follows

2 line display with 5x7 font  28, 28, 06, 0E, 01