10
Alpha numeric displays

Using an Alpha Numeric Display in a project can bring it alive. Instead of showing a number on a 7 segment display the Alpha Numeric Display could indicate 'The Temperature is 27°C'. Instructions can also be given on screen.

This section details the use of a 16 character by 2 line display, which incorporates an HITACHI HD44780 Liquid Crystal Display Controller Driver Chip. The HD44780 is an industry standard also used in displays other than Hitachi (fortunately). The chip is also used as a driver for other display configurations i.e. 16×1, 20×2, 20×4, 40×2 etc. It has an on board character generator ROM which can display 240 character patterns.

The circuit diagram connecting the Alpha Numeric Display to the 16F84 is shown in Figure 10.1. This configuration is for the HD44780 driver and can be used with any of the displays using this chip.

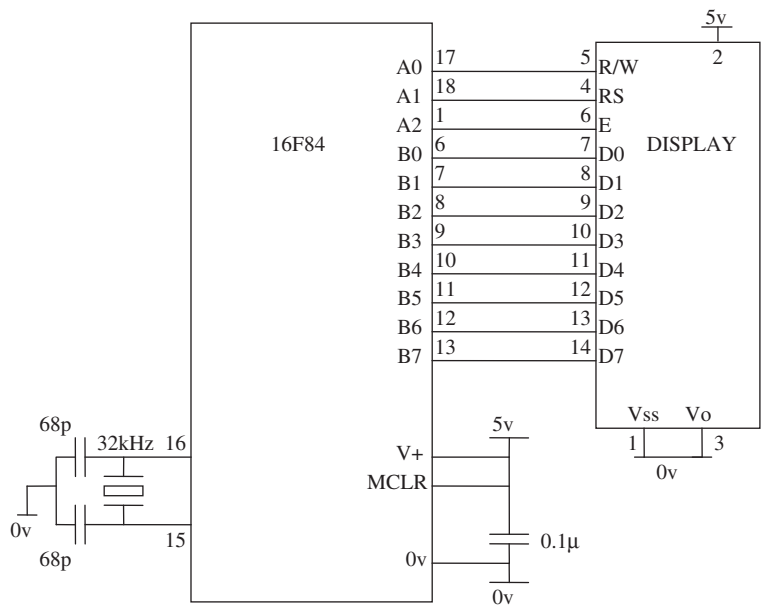

Figure 10.1 The 16F84 driving the alpha numeric display

Display pin identification

This display configuration shows 11 outputs from the Microcontroller, 3 control lines and 8 data lines connecting to the display.

R/W is the read/write control line, RS is the register select and E is the chip enable.

The R/W line tells the display to expect data to be written to it or to have data read from it. The data that is written to it is the address of the character, the code for the character or the type of command we require it to perform such as turn the cursor off.

The R/S line selects either a command to perform (R/S = 0) i.e. clear display, turn cursor on or off, or selects a data transfer (R/S = 1).

The E line enables, (E = 1) and disables, (E = 0) the display.

There is much more to this display than we are able to look at here. If you wish to know more about them you will need to consult the manufacturers data book.

If we use 11 lines to drive the display that would only leave 2 lines for the rest of our control with the 16F84. We could of course use a micro with 22 or 33 I/O. The display can however be driven with 4 data lines instead of 8, 4 bits of data are then sent twice. This complicates the program a little – but since I have done that work in the header it requires no more effort on your part.

Also the R/W line is used to write commands to the micro and read the busy line which indicates when the relatively slow display has processed the data. If we allow the micro enough time to complete its task then we do not have to read the busy line we can just write to the display. The R/W line can then be connected to 0v in a permanent write mode and we do not require a read/write line from the micro.

We will therefore only require 4 data lines and 2 control lines to drive the display leaving 7 lines available for I/O on the 16F84.

This 6 line control for the display is shown in Figure 10.2.

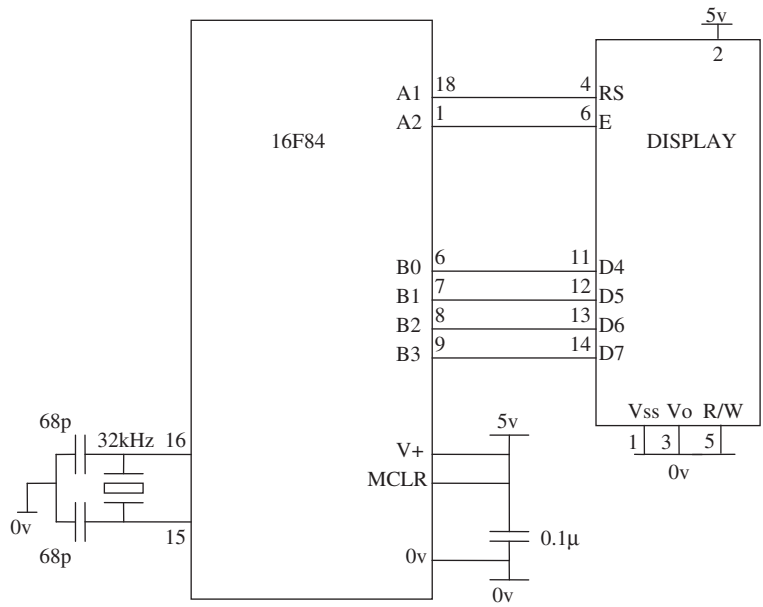

Figure 10.2 Driving the alpha numeric display with 6 control lines

Configuring the display

Before writing to the display you first of all have to configure it. That means tell it if you are:

(a) using a 4 bit or 8 bit Microcontroller,
(b) using a 1 or 2 line display,
(c) using a character font size of 5×10 or 5×7 dots,
(d) turning the display on or off,
(e) turning the cursor on or off,
(f) incrementing the cursor or not. The cursor position increments after a character has been written to the display.

In the program shown below the display has been set up in the Configuration Section with Function Set at 32H to use a 4 bit Microcontroller with a 2 line display and Font size of 5×7 dots. The Display is turned on and Cursor turned off with 0CH and the Cursor set to increment with 06H. This information was obtained from the display data sheet.

Writing to the display

- To write to the display you first of all set the address of the cursor (where you want the character to appear). The Cursor address locations are shown in Figure 10.3 Line1 address starts at 80H. Line2 address starts at C0H.

80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF

Figure 10.3 Cursor address location

- Then tell the display what the character code is, e.g. A has the code 41H, B has the code 42H, C is 43H, 0 is 30H, 1 is 31H, 2 is 32H etc.

To print an A on the screen – first enable the display, send 2 to PORTA, send the code 41H to PORTB and CLOCK this data.

These instructions have been written in the Subroutine Section so all you have to do is CALL A.

To write HELLO on the display the program would be:

```
CALL H
CALL E
CALL L
CALL L
CALL O
```

Program example

The program below is the listing to spell out MICROCONTROLLERS AT THE MMU.

Then CONTACT DAVE SMITH. Together with the time delays.

```
;ANHEAD84.ASM Header for the alpha numeric display using 6 I/O

TMR0      EQU     1       ;means TMR0 is file 1.
STATUS    EQU     3       ;means STATUS is file 3.
PORTA     EQU     5       ;means PORTA is file 5.
PORTB     EQU     6       ;means PORTB is file 6.
TRISA     EQU     85H     ;TRISA (the PORTA I/O selection) is file 85H
TRISB     EQU     86H     ;TRISB (the PORTB I/O selection) is file 86H
OPTION_R  EQU     81H     ;the OPTION register is file 81H
```

```
ZEROBIT     EQU     2           ;means ZEROBIT is bit 2.
COUNT       EQU     0CH         ;COUNT is file 0C, a register to count events.
;*****************************************************

LIST        P=16F84             ;we are using the 16F84.
ORG         0                   ;the start address in memory is 0
GOTO        START               ;goto start!

;*************************************************************
; Configuration Bits

__CONFIG H'3FF0'    ;selects LP oscillator, WDT off, PUT on,
                    ;Code Protection disabled.

;*************************************************

; SUBROUTINE SECTION.

;3 SECOND DELAY
DELAY3      CLRF    TMR0            ;Start TMR0
LOOPA       MOVF    TMR0,W          ;Read TMR0 into W
            SUBLW   .96             ;TIME - W
            BTFSS   STATUS,ZEROBIT  ;Check TIME-W = 0
            GOTO    LOOPA
            RETLW   0               ;return after TMR0 = 96

;P1 SECOND DELAY
DELAYP1     CLRF    TMR0            ;Start TMR0
LOOPC       MOVF    TMR0,W          ;Read TMR0 into W
            SUBLW   .3              ;TIME - W
            BTFSS   STATUS,ZEROBIT  ;Check TIME-W = 0
            GOTO    LOOPC
            RETLW   0               ;return after TMR0 = 3

CLOCK       BSF     PORTA,2
            NOP
            BCF     PORTA,2
            NOP
            RETLW   0
;*****************************************************
A           MOVLW   2               ;enables the display
            MOVWF   PORTA
            MOVLW   4H
```

```
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   1H          ;41 is code for A
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

BB      MOVLW   2           ;enables the display
        MOVWF   PORTA
        MOVLW   4H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   2H          ;42 is code for B
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

C       MOVLW   2           ;enables the display
        MOVWF   PORTA
        MOVLW   4H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   3H
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

D       MOVLW   2           ;enables the display
        MOVWF   PORTA
        MOVLW   4H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   4H
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

E       MOVLW   2           ;enables the display
        MOVWF   PORTA
        MOVLW   4H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   5H
```

```
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

F       MOVLW   2           ;enables the display
        MOVWF   PORTA
        MOVLW   4H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   6H
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

G       MOVLW   2           ;enables the display
        MOVWF   PORTA
        MOVLW   4H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   7H
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

H       MOVLW   2           ;enables the display
        MOVWF   PORTA
        MOVLW   4H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   8H
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

I       MOVLW   2           ;enables the display
        MOVWF   PORTA
        MOVLW   4H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   9H
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0
```

```
J       MOVLW   2       ;enables the display
        MOVWF   PORTA
        MOVLW   4H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   0AH
        MOVWF   PORTB
        CALL    CLOCK   ;clock character onto display.
        RETLW   0

K       MOVLW   2       ;enables the display
        MOVWF   PORTA
        MOVLW   4H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   0BH
        MOVWF   PORTB
        CALL    CLOCK   ;clock character onto display.
        RETLW   0

L       MOVLW   2       ;enables the display
        MOVWF   PORTA
        MOVLW   4H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   0CH
        MOVWF   PORTB
        CALL    CLOCK   ;clock character onto display.
        RETLW   0

M       MOVLW   2       ;enables the display
        MOVWF   PORTA
        MOVLW   4H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   0DH
        MOVWF   PORTB
        CALL    CLOCK   ;clock character onto display.
        RETLW   0

N       MOVLW   2       ;enables the display
        MOVWF   PORTA
        MOVLW   4H
```

```
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        MOVLW   0EH
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

O       MOVLW   2           ;enables the display
        MOVWF   PORTA
        MOVLW   4H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   0FH
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

P       MOVLW   2
        MOVWF   PORTA
        MOVLW   5H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   0H
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

Q       MOVLW   2
        MOVWF   PORTA
        MOVLW   5H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   1H
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

R       MOVLW   2
        MOVWF   PORTA
        MOVLW   5H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   2H
```

```
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

S       MOVLW   2
        MOVWF   PORTA
        MOVLW   5H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   3H
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

T       MOVLW   2
        MOVWF   PORTA
        MOVLW   5H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   4H
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

U       MOVLW   2
        MOVWF   PORTA
        MOVLW   5H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   5H
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

V       MOVLW   2
        MOVWF   PORTA
        MOVLW   5H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   6H
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0
```

```
WW      MOVLW   2
        MOVWF   PORTA
        MOVLW   5H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   7H
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

X       MOVLW   2
        MOVWF   PORTA
        MOVLW   5H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   8H
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

Y       MOVLW   2
        MOVWF   PORTA
        MOVLW   5H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   9H
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

Z       MOVLW   2
        MOVWF   PORTA
        MOVLW   5H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   0AH
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

NUM0    MOVLW   2           ;enables the display
        MOVWF   PORTA
        MOVLW   3H
        MOVWF   PORTB
```

```
        CALL    CLOCK
        MOVLW   0H
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

NUM1    MOVLW   2           ;enables the display
        MOVWF   PORTA
        MOVLW   3H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   1H
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

NUM2    MOVLW   2           ;enables the display
        MOVWF   PORTA
        MOVLW   3H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   2H
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

NUM3    MOVLW   2           ;enables the display
        MOVWF   PORTA
        MOVLW   3H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   3H
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

NUM4    MOVLW   2           ;enables the display
        MOVWF   PORTA
        MOVLW   3H
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        MOVLW   4H
```

```
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

NUM5    MOVLW   2           ;enables the display
        MOVWF   PORTA
        MOVLW   3H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   5H
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

NUM6    MOVLW   2           ;enables the display
        MOVWF   PORTA
        MOVLW   3H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   6H
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

NUM7    MOVLW   2           ;enables the display
        MOVWF   PORTA
        MOVLW   3H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   7H
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

NUM8    MOVLW   2           ;enables the display
        MOVWF   PORTA
        MOVLW   3H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   8H
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0
```

```
NUM9    MOVLW   2           ;enables the display
        MOVWF   PORTA
        MOVLW   3H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   9H
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

GAP     MOVLW   2
        MOVWF   PORTA
        MOVLW   2H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   0H
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

DOT     MOVLW   2
        MOVWF   PORTA
        MOVLW   2H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   0EH
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        RETLW   0

CLRDISP CLRF    PORTA
        MOVLW   0H
        MOVWF   PORTB
        CALL    CLOCK       ;clock character onto display.
        MOVLW   1
        MOVWF   PORTB
        CALL    CLOCK
        CALL    DELAYP1
        RETLW   0

;*****************************************************
; CONFIGURATION SECTION.

START   BSF     STATUS,5    ;Turns to Bank1.
        MOVLW   B'00000000' ;PORTA is O/P
        MOVWF   TRISA
```

```
                MOVLW   B'00000000'
                MOVWF   TRISB           ;PORTB is OUTPUT

                MOVLW   B'00000111'     ;Prescaler is /256
                MOVWF   OPTION_R        ;TIMER is 1/32 secs.

                BCF     STATUS,5        ;Return to Bank0.
                CLRF    PORTA           ;Clears PortA.
                CLRF    PORTB           ;Clears PortB.

;Display Configuration
                MOVLW   03H             ;FUNCTION SET
                MOVWF   PORTB           ;8bit data (default)
                CALL    CLOCK

                CALL    DELAYP1         ;wait for display

                MOVLW   02H             ;FUNCTION SET
                MOVWF   PORTB           ;change to 4bit
                CALL    CLOCK           ;clock in data

                CALL    DELAYP1         ;wait for display
                MOVLW   02H             ;FUNCTION SET
                MOVWF   PORTB           ;must repeat command
                CALL    CLOCK           ;clock in data

                CALL    DELAYP1         ;wait for display
                MOVLW   08H             ;4 bit micro
                MOVWF   PORTB           ;using 2 line display.
                CALL    CLOCK           ;clock in data

                CALL    DELAYP1
                MOVLW   0H              ;Display on, cursor off
                MOVWF   PORTB           ;0CH
                CALL    CLOCK
                MOVLW   0CH
                MOVWF   PORTB
                CALL    CLOCK

                CALL    DELAYP1
                MOVLW   0H              ;Increment cursor, 06H
                MOVWF   PORTB
                CALL    CLOCK
                MOVLW   6H
                MOVWF   PORTB
                CALL    CLOCK
```

```
;*****************************************************
;Program starts now.

BEGIN   CALL    CLRDISP
        CLRF    PORTA
        MOVLW   8H          ;Cursor at top left, 80H
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   0H
        MOVWF   PORTB
        CALL    CLOCK

        CALL    M           ;display M
        CALL    DELAYP1     ;wait 0.1 seconds
        CALL    I           ;display I
        CALL    DELAYP1     ;wait 0.1 seconds
        CALL    C           ;Etc.
        CALL    DELAYP1
        CALL    R
        CALL    DELAYP1
        CALL    O
        CALL    DELAYP1
        CALL    C
        CALL    DELAYP1
        CALL    O
        CALL    DELAYP1
        CALL    N
        CALL    DELAYP1
        CALL    T
        CALL    DELAYP1
        CALL    R
        CALL    DELAYP1
        CALL    O
        CALL    DELAYP1
        CALL    L
        CALL    DELAYP1
        CALL    L
        CALL    DELAYP1
        CALL    E
        CALL    DELAYP1
        CALL    R
        CALL    DELAYP1
        CALL    S
        CALL    DELAYP1
```

```
CLRF    PORTA
MOVLW   0CH     ;Cursor on second line, C3
MOVWF   PORTB
CALL    CLOCK
MOVLW   3H
MOVWF   PORTB
CALL    CLOCK

CALL    A
CALL    DELAYP1
CALL    T
CALL    DELAYP1
CALL    GAP
CALL    T
CALL    DELAYP1
CALL    H
CALL    DELAYP1
CALL    E
CALL    DELAYP1
CALL    GAP
CALL    M
CALL    DELAYP1
CALL    M
CALL    DELAYP1
CALL    U
CALL    DELAYP1

CALL    DELAY3  ;wait 3 seconds

CALL    CLRDISP

MOVLW   8H      ;Cursor at top left, 80H
MOVWF   PORTB
CALL    CLOCK
MOVLW   0H
MOVWF   PORTB
CALL    CLOCK

CALL    C
CALL    DELAYP1
CALL    O
CALL    DELAYP1
CALL    N
CALL    DELAYP1
CALL    T
```

```
        CALL    DELAYP1
        CALL    A
        CALL    DELAYP1
        CALL    C
        CALL    DELAYP1
        CALL    T
        CALL    DELAYP1
        CLRF    PORTA
        MOVLW   0CH             ;Cursor on 2nd line
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   3H
        MOVWF   PORTB
        CALL    CLOCK

        CALL    D
        CALL    DELAYP1
        CALL    A
        CALL    DELAYP1
        CALL    V
        CALL    DELAYP1
        CALL    E
        CALL    DELAYP1
        CALL    GAP
        CALL    DELAYP1
        CALL    S
        CALL    DELAYP1
        CALL    M
        CALL    DELAYP1
        CALL    I
        CALL    DELAYP1
        CALL    T
        CALL    DELAYP1
        CALL    H

        CALL    DELAY3          ;wait 3 seconds

        GOTO    BEGIN
END
```

Program operation

- PORTA and PORTB are configured as outputs in the CONFIGURATION SECTION.

Display configuration

- In the Display Configuration Section, the Register Select (R/S) line, A1on the microcontroller, is set low by CLRF PORTA in the Configuration Section.
- R/S = 0 ensures that the data to the display will change the registers. Later R/S = 1 writes the characters to the display.
- The display is expecting its data to arrive via 8 lines, but to save I/O lines we will use 4 and write them twice. The code to do this and also tell the driver chip the display is a two line display is:

```
MOVLW   03H     ;FUNCTION SET
MOVWF   PORTB   ;8bit data (default)
CALL    CLOCK

CALL    DELAYP1 ;wait for display

MOVLW   02H     ;FUNCTION SET
MOVWF   PORTB   ;change to 4bit
CALL    CLOCK   ;clock in data

CALL    DELAYP1 ;wait for display
MOVLW   02H     ;FUNCTION SET
MOVWF   PORTB   ;must repeat command
CALL    CLOCK   ;clock in data

CALL    DELAYP1 ;wait for display
MOVLW   08H     ;4 bit micro
MOVWF   PORTB   ;using 2 line display.
CALL    CLOCK   ;clock in data
```

The data is set up on PORTB using B0,1,2 and 3. As in

```
MOVLW   03H     ;FUNCTION SET
MOVWF   PORTB
```

This data is then clocked into the display by pulsing the Enable line, (E, A2 on the micro) high and then low with:

```
CLOCK   BSF     PORTA,2
        NOP
        BCF     PORTA,2
        NOP
        RETLW   0
```

CALL DELAYP1 , waits for 0.1 seconds to give the display time to activate before continuing.

When the display has been configured to: Turn on, switch the cursor off, and increment the cursor after every character write. We are then ready to write to the display.

Writing to the display

- The display is cleared if required with:

```
CALL CLRDISP
```

- The address of the character is first written to the display, say, the 80H position (top left hand corner).

```
CLRF    PORTA
MOVLW   8H          ;Cursor at top left, 80H
MOVWF   PORTB
CALL    CLOCK
MOVLW   0H
MOVWF   PORTB
CALL    CLOCK
```

Notice the 8 is sent first followed by the 0.

To write to the position mid-way along the top line the address would be 88H. So the 80H in the code above would be replaced by 88H.

- In order to write the letter 'M' in the display at the position defined. We CALL M and use the code 4DH, NB. Send the 4 first followed by the D. The Register Select Line, R/S, A1 on the micro, is set to 1 for the character write option. The code is:

```
M       MOVLW   2           ;enables the display
        MOVWF   PORTA       ;sets A1=1
        MOVLW   4H          ;send data 4
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   0DH         ;send data D
        MOVWF   PORTB
        CALL    CLOCK       ;clock character 'M' onto display.
        RETLW   0
```

In this way any one of the 240 characters available can be shown on the display.

The program continues by printing out the rest of the message. A delay of 0.1 seconds is maintained after printing each character to give the effect of the message being typed out.

All the Capital Letters and numbers 0 to 9 have been included in the header so you can easily enter your own message.

The complete character set for the display showing all 240 characters is illustrated in Figure 10.4.

Displaying a number

Suppose we wish to display a number thrown by a dice, for example a 4. We could use the instruction CALL NUM4, but we would not have known previously that the number was going to be a 4. The throw of the dice would be stored in a user file called, say, THROW and THROW would then have 4 in it.

Now the code for 0 is 30H
The code for 1 is 31H
The code for 2 is 32H
Etc.
If we wanted to display the number 4 the code is:

```
NUM4    MOVLW   2       ;enables the display
        MOVWF   PORTA
        MOVLW   3H      ;34H is the code for 4
        MOVWF   PORTB
        CALL    CLOCK
        MOVLW   4H
        MOVWF   PORTB
        CALL    CLOCK   ;clock character onto display.
        RETLW   0
```

If the 4 is in the file THROW, we can display this with the code:

```
        MOVLW   2       ;enables the display
        MOVWF   PORTA
        MOVLW   3H
        MOVWF   PORTB
```

```
CALL    CLOCK
MOVF    THROW,W     ;number comes from the file
MOVWF   PORTB
CALL    CLOCK       ;clock character onto display.
RETLW   0
```

Notice how the value of the number now has come from the file.

This code would then display any number in the file THROW.

If you measured a temperature as 27°C, you would probably store the 2 in a file TEMPTENS (tens of degrees) and the 7 in a file TEMPUNIT (units of degrees).

You can then modify the code above to display:

THE TEMPERATURE
IS 27°C.

The 'I' would be located at address C5H on the display. The temperature would then be written at locations C8H and C9H. There would be no need to rewrite the message just rewrite the temperature as it changed, after first moving the cursor to address C8H.

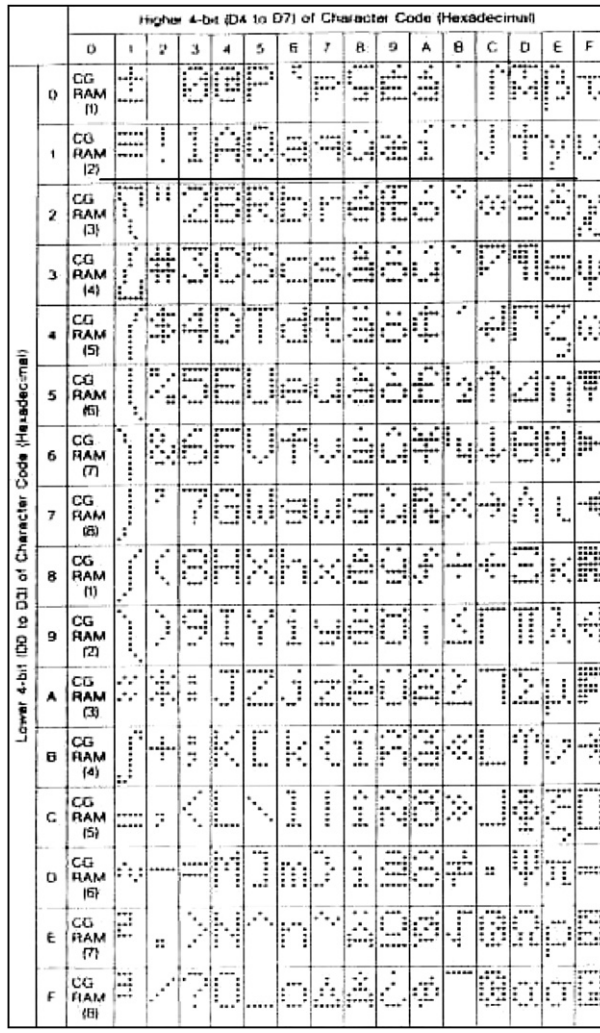

Figure 10.4 Alpha numeric display character set