# Microprocessor Interfacing & Programming

LECTURE 28

# LCD Interfacing

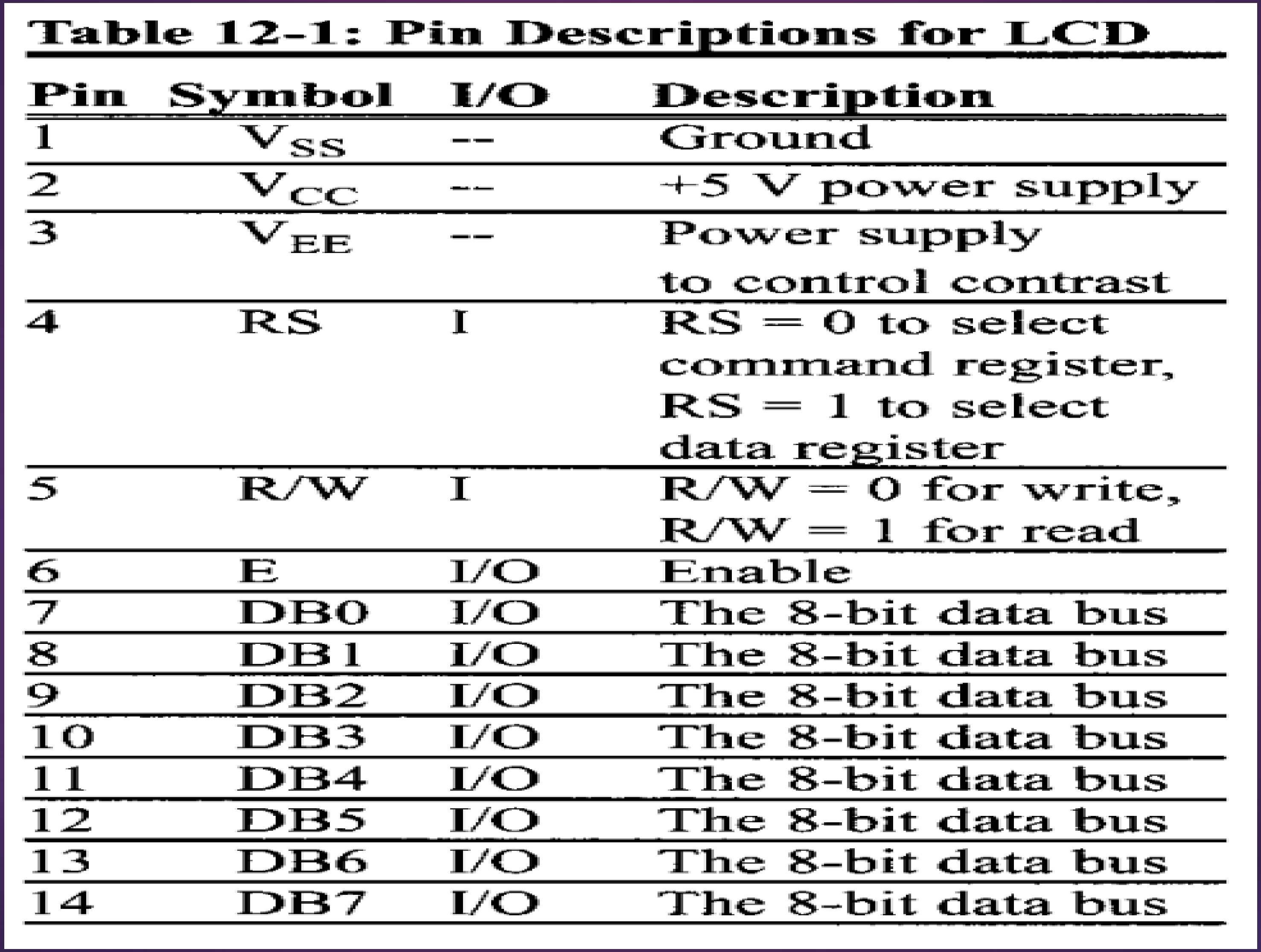


**Table 12-1: Pin Descriptions for LCD**

| Pin | Symbol | I/O | Description |
|---|---|---|---|
| 1 | $V_{SS}$ | -- | Ground |
| 2 | $V_{CC}$ | -- | +5 V power supply |
| 3 | $V_{EE}$ | -- | Power supply to control contrast |
| 4 | RS | I | RS = 0 to select command register,<br>RS = 1 to select data register |
| 5 | R/W | I | R/W = 0 for write,<br>R/W = 1 for read |
| 6 | E | I/O | Enable |
| 7 | DB0 | I/O | The 8-bit data bus |
| 8 | DB1 | I/O | The 8-bit data bus |
| 9 | DB2 | I/O | The 8-bit data bus |
| 10 | DB3 | I/O | The 8-bit data bus |
| 11 | DB4 | I/O | The 8-bit data bus |
| 12 | DB5 | I/O | The 8-bit data bus |
| 13 | DB6 | I/O | The 8-bit data bus |
| 14 | DB7 | I/O | The 8-bit data bus |

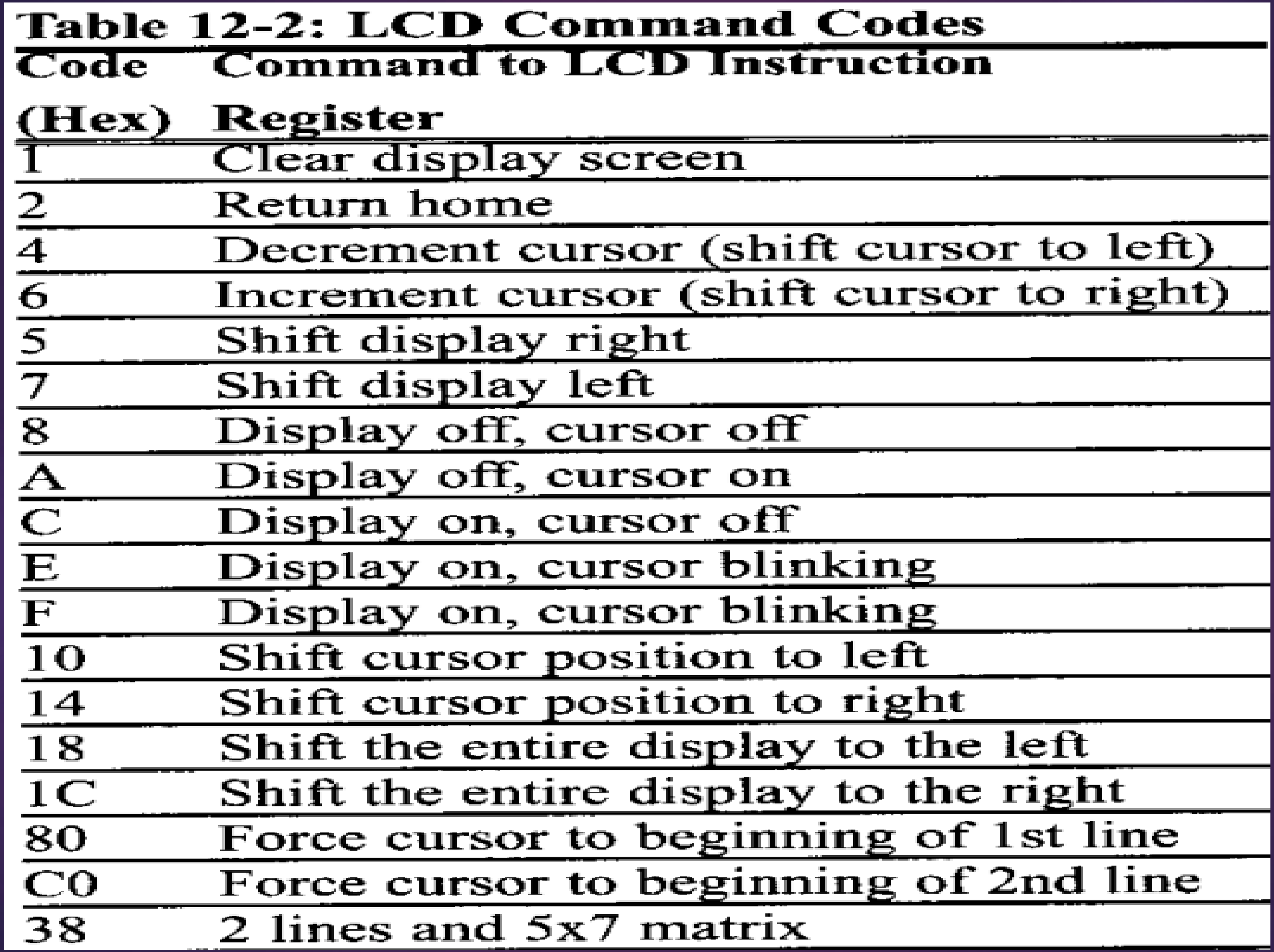


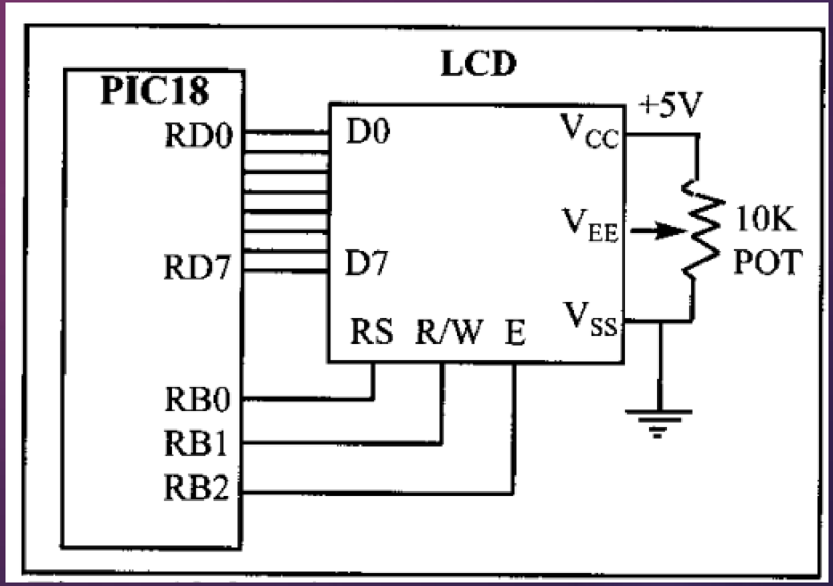
Table 12-2: LCD Command Codes

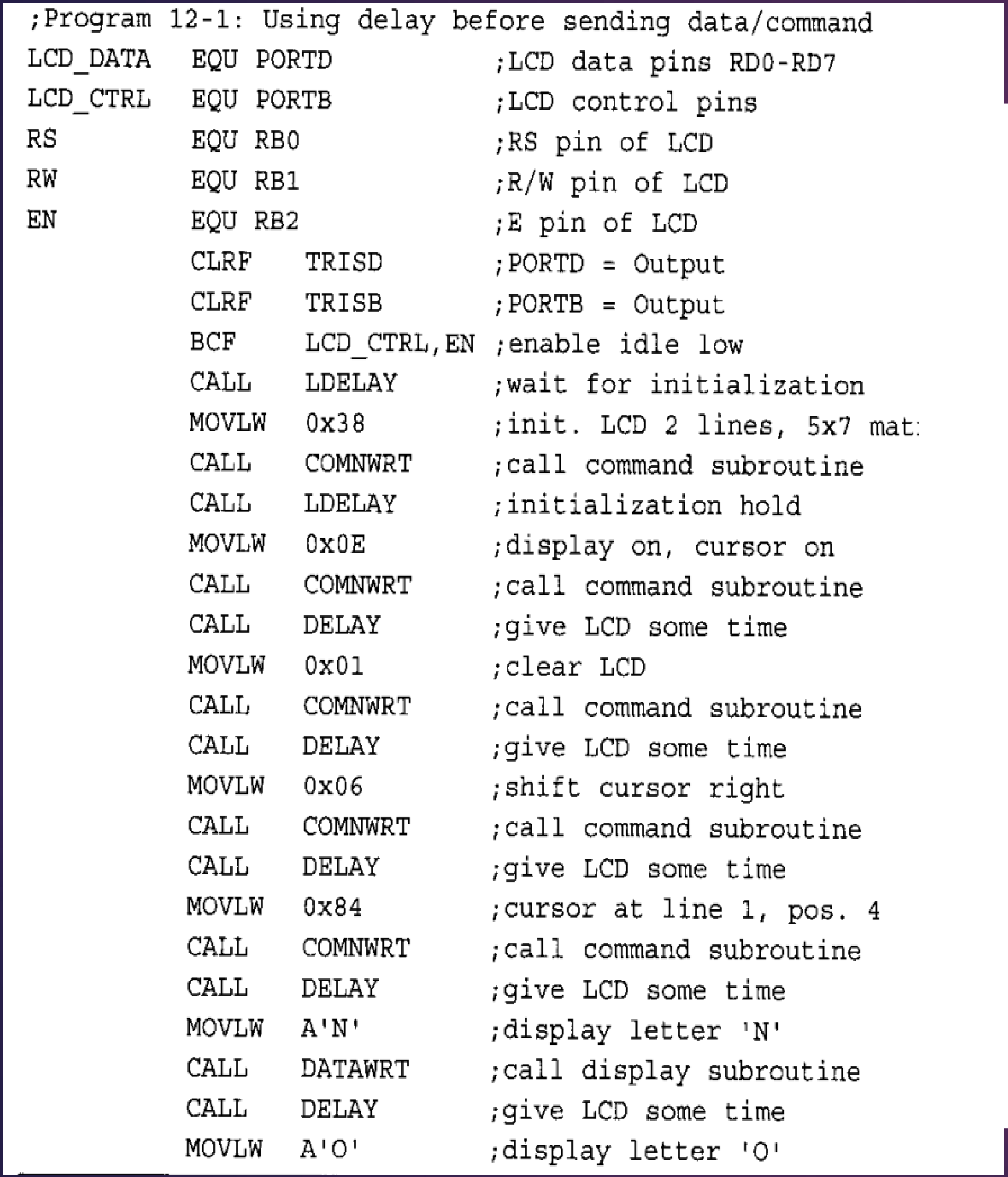
| Code (Hex) | Command to LCD Instruction Register |
|---|---|
| 1 | Clear display screen |
| 2 | Return home |
| 4 | Decrement cursor (shift cursor to left) |
| 6 | Increment cursor (shift cursor to right) |
| 5 | Shift display right |
| 7 | Shift display left |
| 8 | Display off, cursor off |
| A | Display off, cursor on |
| C | Display on, cursor off |
| E | Display on, cursor blinking |
| F | Display on, cursor blinking |
| 10 | Shift cursor position to left |
| 14 | Shift cursor position to right |
| 18 | Shift the entire display to the left |
| 1C | Shift the entire display to the right |
| 80 | Force cursor to beginning of 1st line |
| C0 | Force cursor to beginning of 2nd line |
| 38 | 2 lines and 5x7 matrix |

- There are two ways to send characters (command/data) to the LCD:

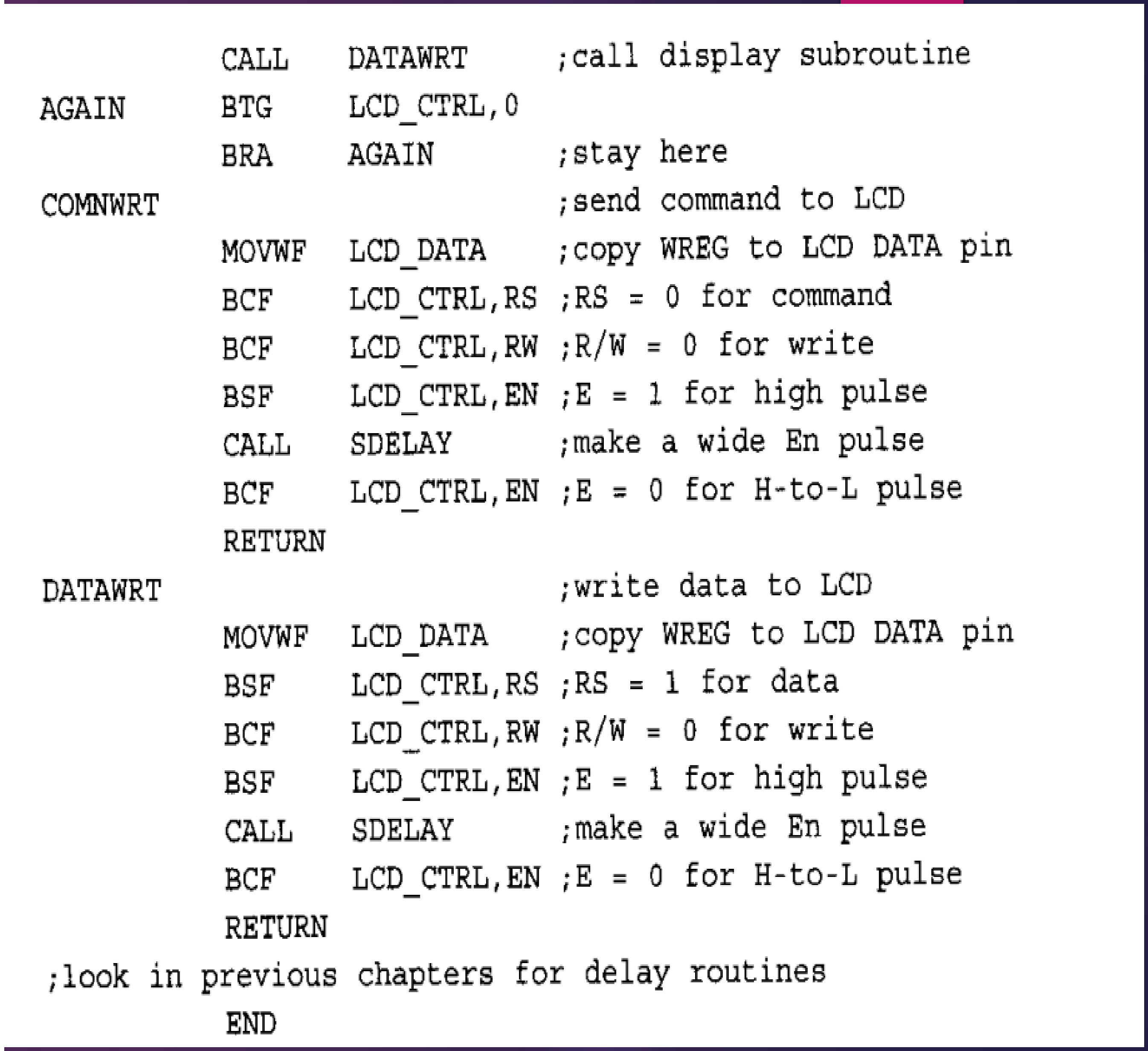
1. Use a delay before sending the next one.
2. Use the busy flag to see if the LCD is ready for the next one.

- In programming an LCD, we also need a delay for the power-up process which we called long delay (LDELAY).
- We usually take LDELAY greater than or equal to 25ms.
- There is another delay used to make the enable signal width enough for the LCD's enable input.





```
;Program 12-1: Using delay before sending data/command
LCD_DATA    EQU PORTD           ;LCD data pins RD0-RD7
LCD_CTRL    EQU PORTB           ;LCD control pins
RS          EQU RB0             ;RS pin of LCD
RW          EQU RB1             ;R/W pin of LCD
EN          EQU RB2             ;E pin of LCD
            CLRF   TRISD        ;PORTD = Output
            CLRF   TRISB        ;PORTB = Output
            BCF    LCD_CTRL,EN  ;enable idle low
            CALL   LDELAY       ;wait for initialization
            MOVLW  0x38         ;init. LCD 2 lines, 5x7 mat
            CALL   COMNWRT      ;call command subroutine
            CALL   LDELAY       ;initialization hold
            MOVLW  0x0E         ;display on, cursor on
            CALL   COMNWRT      ;call command subroutine
            CALL   DELAY        ;give LCD some time
            MOVLW  0x01         ;clear LCD
            CALL   COMNWRT      ;call command subroutine
            CALL   DELAY        ;give LCD some time
            MOVLW  0x06         ;shift cursor right
            CALL   COMNWRT      ;call command subroutine
            CALL   DELAY        ;give LCD some time
            MOVLW  0x84         ;cursor at line 1, pos. 4
            CALL   COMNWRT      ;call command subroutine
            CALL   DELAY        ;give LCD some time
            MOVLW  A'N'         ;display letter 'N'
            CALL   DATAWRT      ;call display subroutine
            CALL   DELAY        ;give LCD some time
            MOVLW  A'O'         ;display letter 'O'
```



```
            CALL    DATAWRT      ;call display subroutine
AGAIN       BTG     LCD_CTRL,0
            BRA     AGAIN        ;stay here
COMNWRT                          ;send command to LCD
            MOVWF   LCD_DATA     ;copy WREG to LCD DATA pin
            BCF     LCD_CTRL,RS  ;RS = 0 for command
            BCF     LCD_CTRL,RW  ;R/W = 0 for write
            BSF     LCD_CTRL,EN  ;E = 1 for high pulse
            CALL    SDELAY       ;make a wide En pulse
            BCF     LCD_CTRL,EN  ;E = 0 for H-to-L pulse
            RETURN
DATAWRT                          ;write data to LCD
            MOVWF   LCD_DATA     ;copy WREG to LCD DATA pin
            BSF     LCD_CTRL,RS  ;RS = 1 for data
            BCF     LCD_CTRL,RW  ;R/W = 0 for write
            BSF     LCD_CTRL,EN  ;E = 1 for high pulse
            CALL    SDELAY       ;make a wide En pulse
            BCF     LCD_CTRL,EN  ;E = 0 for H-to-L pulse
            RETURN
;look in previous chapters for delay routines
            END
```

# Delays

- LCD needs time to:

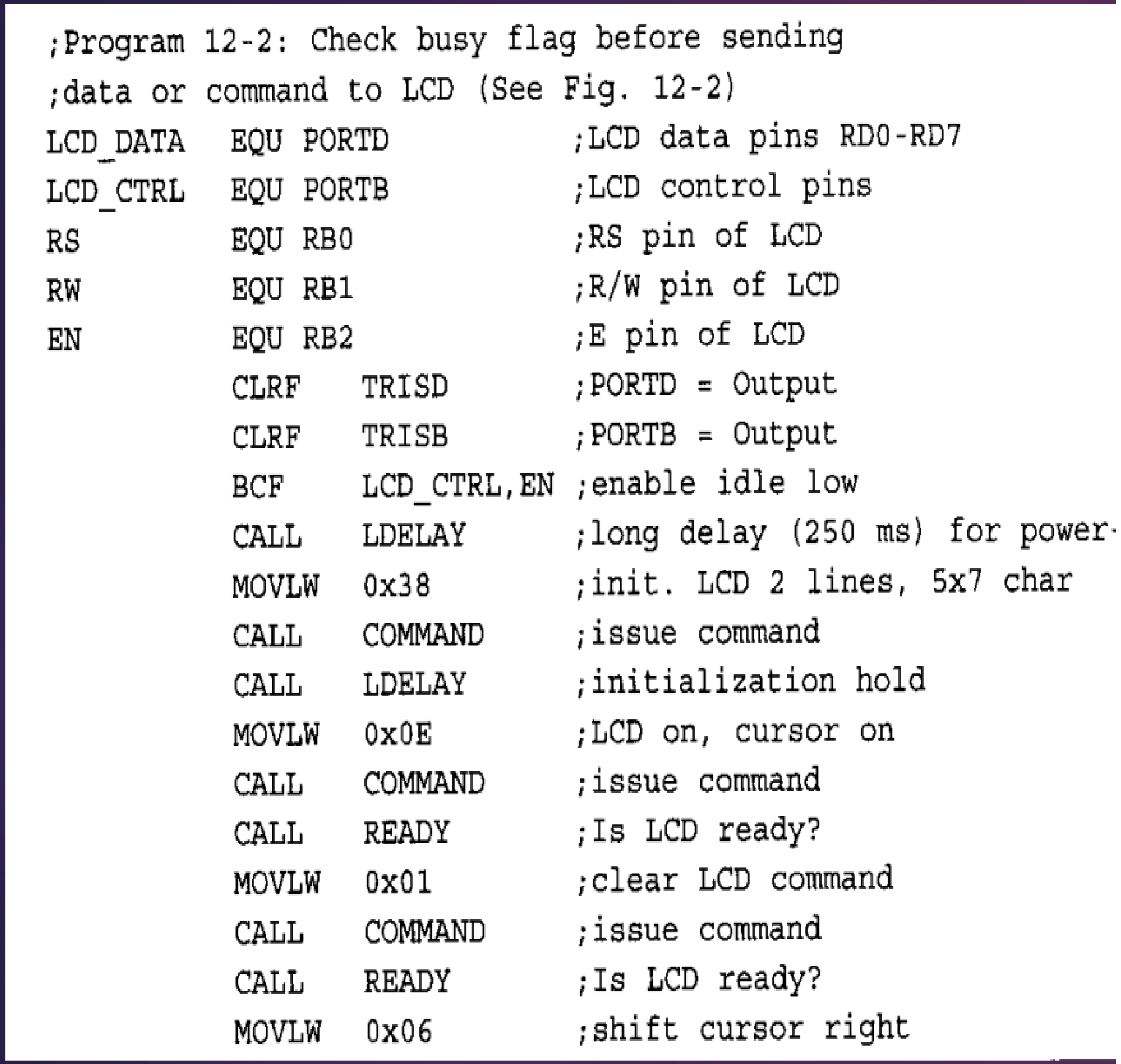
1. Power up internally
2. Reset its registers
3. Stabilize oscillator

- That’s why LDELAY is used in the beginning of the program, during initialization.
- DELAY is used for normal commands because each command required different execution times.
- SDELAY is required for the Enable signal.
- The two LDELAYs are used because the LCD requires:

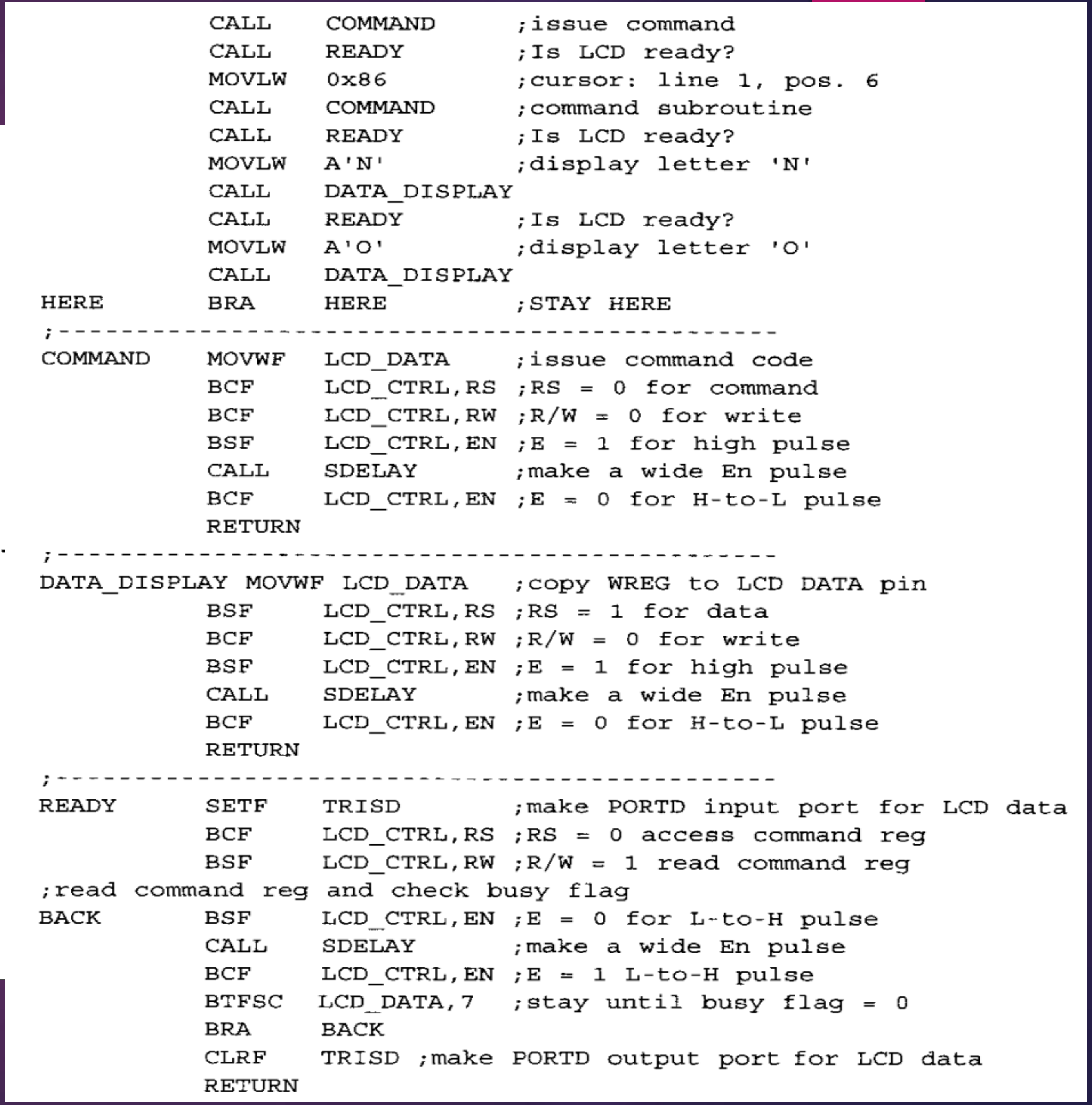
1. A long delay after power-on.
2. Another long delay after the first function set command.

# Sending command and data to the LCD using busy flag

- We use RS=0 to read the busy flag bit to see if the LCD is ready to receive information.
- The busy flag is D7, and can be read when R/W =1 and RS =0.
- When busy flag (D7=1), the LCD is busy taking care of internal operations and will not accept any new information.
- When D7=0, the LCD is ready to receive new information.

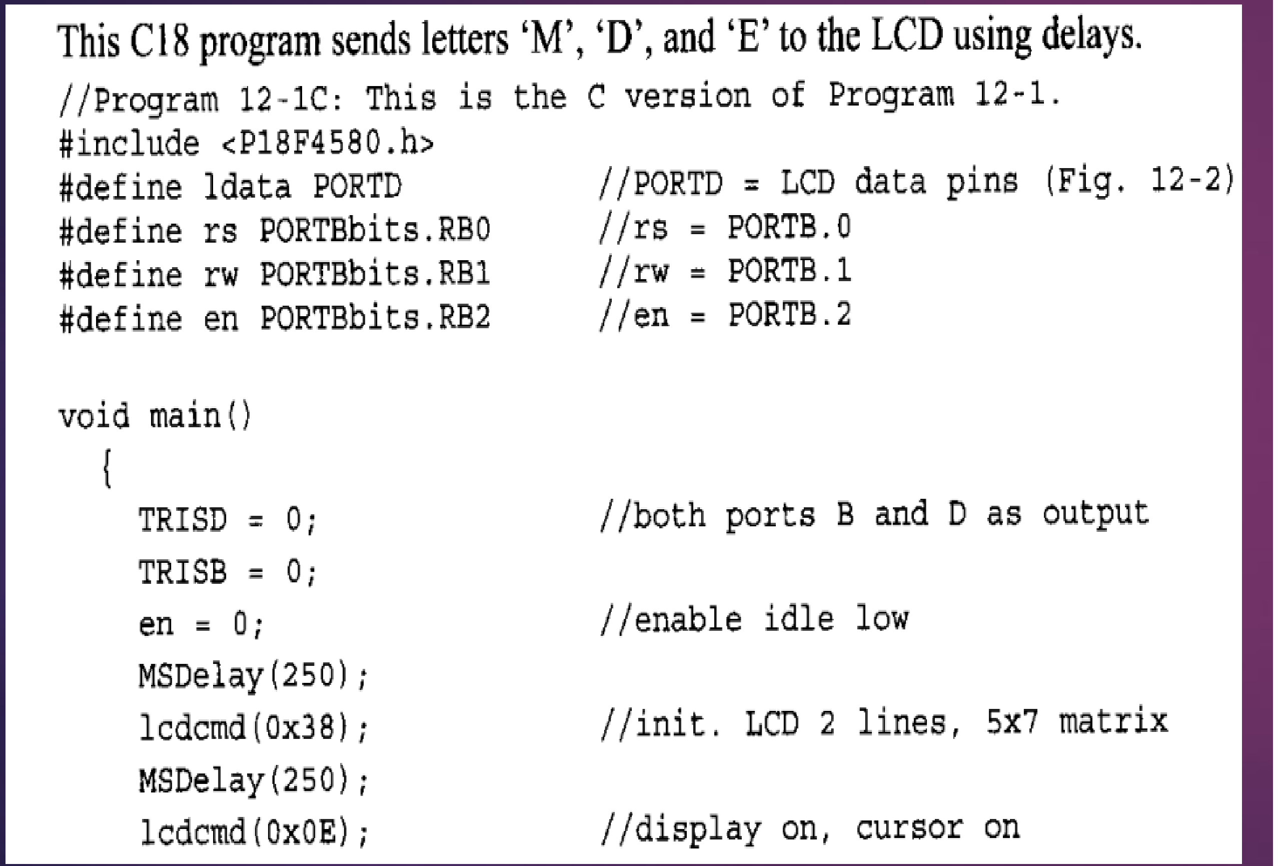


```
;Program 12-2: Check busy flag before sending
;data or command to LCD (See Fig. 12-2)
LCD_DATA  EQU PORTD             ;LCD data pins RD0-RD7
LCD_CTRL  EQU PORTB             ;LCD control pins
RS        EQU RB0               ;RS pin of LCD
RW        EQU RB1               ;R/W pin of LCD
EN        EQU RB2               ;E pin of LCD
          CLRF    TRISD         ;PORTD = Output
          CLRF    TRISB         ;PORTB = Output
          BCF     LCD_CTRL,EN   ;enable idle low
          CALL    LDELAY        ;long delay (250 ms) for power-
          MOVLW   0x38          ;init. LCD 2 lines, 5x7 char
          CALL    COMMAND       ;issue command
          CALL    LDELAY        ;initialization hold
          MOVLW   0x0E          ;LCD on, cursor on
          CALL    COMMAND       ;issue command
          CALL    READY         ;Is LCD ready?
          MOVLW   0x01          ;clear LCD command
          CALL    COMMAND       ;issue command
          CALL    READY         ;Is LCD ready?
          MOVLW   0x06          ;shift cursor right
```



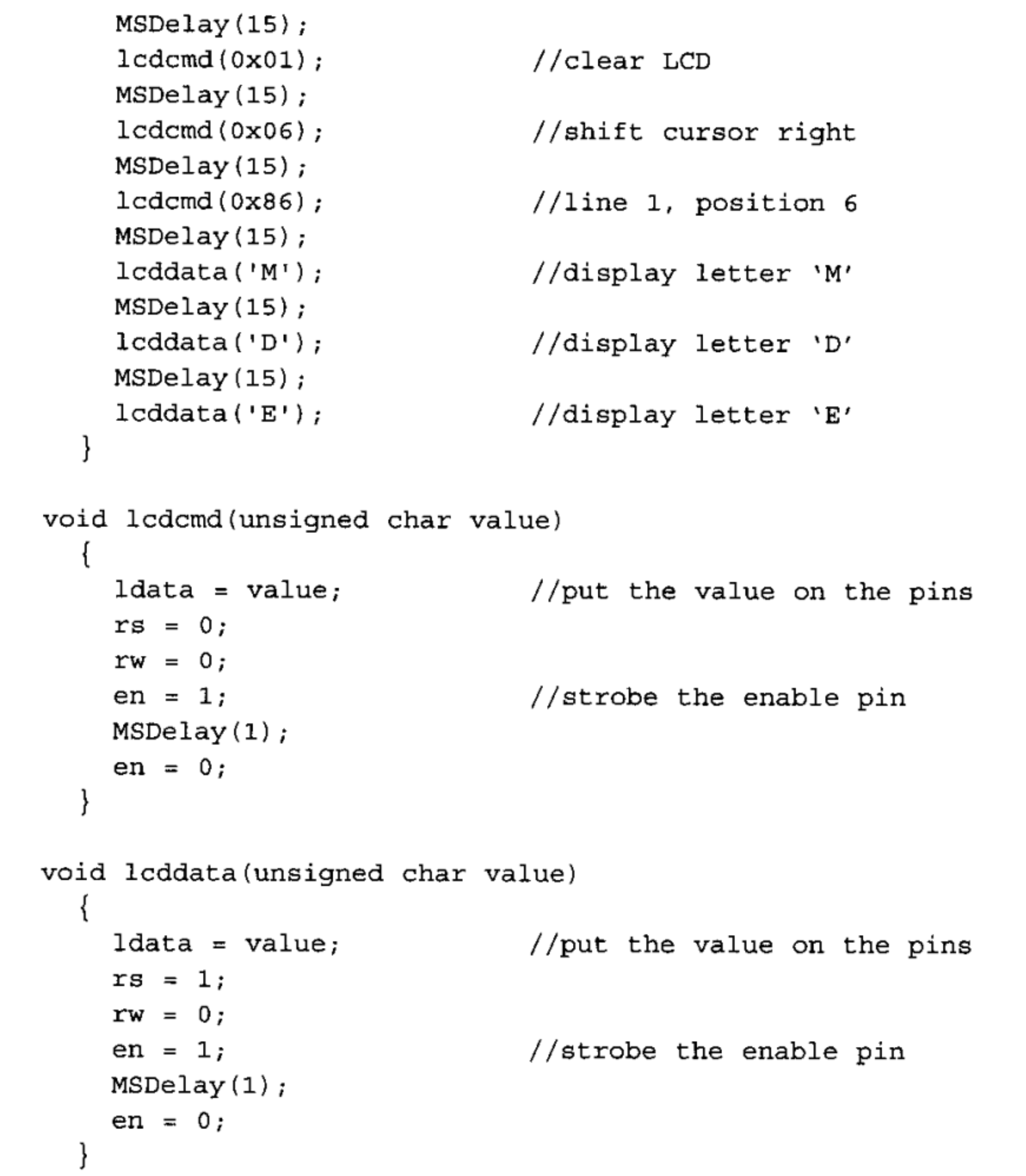
```
             CALL    COMMAND       ;issue command
             CALL    READY         ;Is LCD ready?
             MOVLW   0x86          ;cursor: line 1, pos. 6
             CALL    COMMAND       ;command subroutine
             CALL    READY         ;Is LCD ready?
             MOVLW   A'N'          ;display letter 'N'
             CALL    DATA_DISPLAY
             CALL    READY         ;Is LCD ready?
             MOVLW   A'O'          ;display letter 'O'
             CALL    DATA_DISPLAY
HERE         BRA     HERE          ;STAY HERE
;--------------------------------------------------
COMMAND      MOVWF   LCD_DATA      ;issue command code
             BCF     LCD_CTRL,RS   ;RS = 0 for command
             BCF     LCD_CTRL,RW   ;R/W = 0 for write
             BSF     LCD_CTRL,EN   ;E = 1 for high pulse
             CALL    SDELAY        ;make a wide En pulse
             BCF     LCD_CTRL,EN   ;E = 0 for H-to-L pulse
             RETURN
;--------------------------------------------------
DATA_DISPLAY MOVWF LCD_DATA        ;copy WREG to LCD DATA pin
             BSF     LCD_CTRL,RS   ;RS = 1 for data
             BCF     LCD_CTRL,RW   ;R/W = 0 for write
             BSF     LCD_CTRL,EN   ;E = 1 for high pulse
             CALL    SDELAY        ;make a wide En pulse
             BCF     LCD_CTRL,EN   ;E = 0 for H-to-L pulse
             RETURN
;--------------------------------------------------
READY        SETF    TRISD         ;make PORTD input port for LCD data
             BCF     LCD_CTRL,RS   ;RS = 0 access command reg
             BSF     LCD_CTRL,RW   ;R/W = 1 read command reg
;read command reg and check busy flag
BACK         BSF     LCD_CTRL,EN   ;E = 0 for L-to-H pulse
             CALL    SDELAY        ;make a wide En pulse
             BCF     LCD_CTRL,EN   ;E = 1 L-to-H pulse
             BTFSC   LCD_DATA,7    ;stay until busy flag = 0
             BRA     BACK
             CLRF    TRISD ;make PORTD output port for LCD data
             RETURN
```

# LCD Interfacing in C

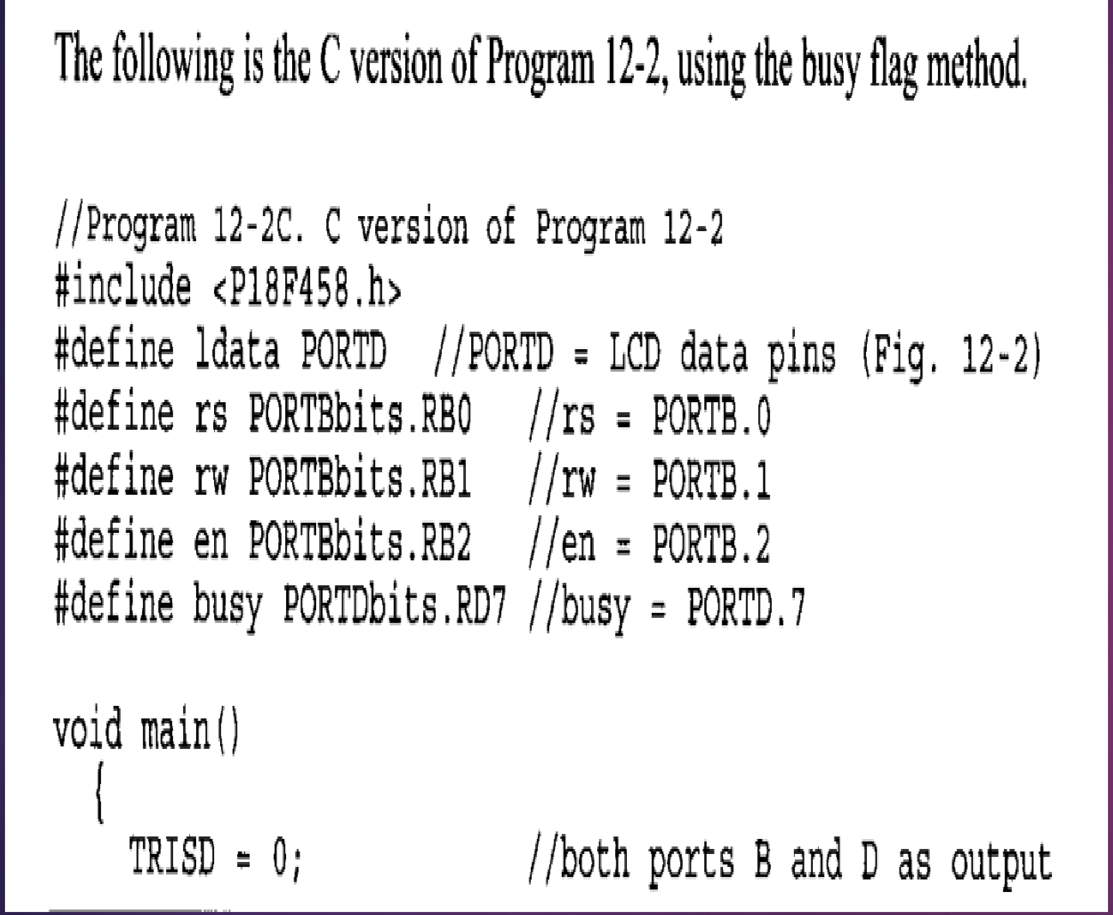


This C18 program sends letters 'M', 'D', and 'E' to the LCD using delays.

```
//Program 12-1C: This is the C version of Program 12-1.
#include <P18F4580.h>
#define ldata PORTD           //PORTD = LCD data pins (Fig. 12-2)
#define rs PORTBbits.RB0      //rs = PORTB.0
#define rw PORTBbits.RB1      //rw = PORTB.1
#define en PORTBbits.RB2      //en = PORTB.2

void main()
  {
    TRISD = 0;                //both ports B and D as output
    TRISB = 0;
    en = 0;                   //enable idle low
    MSDelay(250);
    lcdcmd(0x38);             //init. LCD 2 lines, 5x7 matrix
    MSDelay(250);
    lcdcmd(0x0E);             //display on, cursor on
```

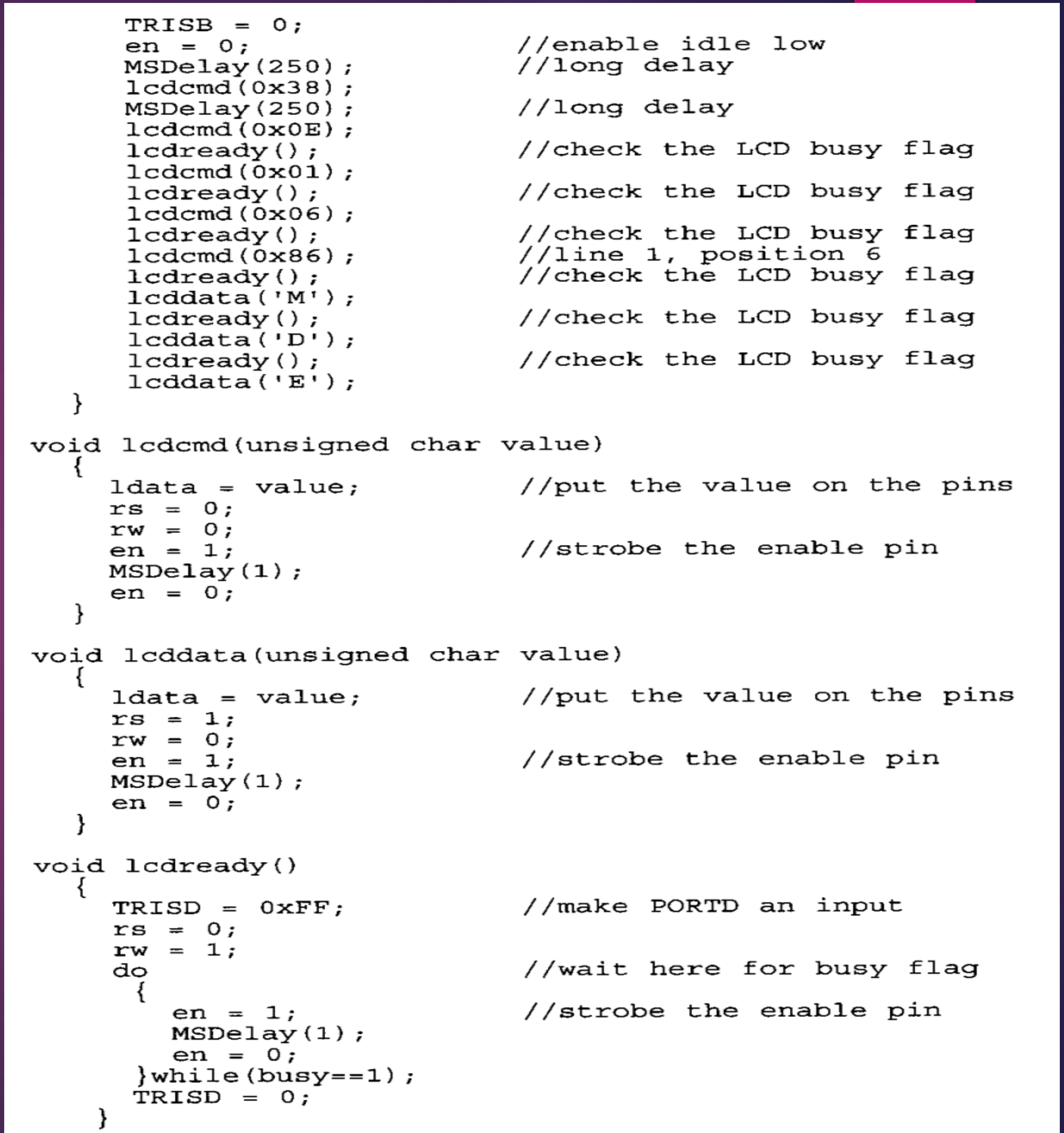


```
    MSDelay(15);
    lcdcmd(0x01);             //clear LCD
    MSDelay(15);
    lcdcmd(0x06);             //shift cursor right
    MSDelay(15);
    lcdcmd(0x86);             //line 1, position 6
    MSDelay(15);
    lcddata('M');             //display letter 'M'
    MSDelay(15);
    lcddata('D');             //display letter 'D'
    MSDelay(15);
    lcddata('E');             //display letter 'E'
  }

void lcdcmd(unsigned char value)
  {
    ldata = value;            //put the value on the pins
    rs = 0;
    rw = 0;
    en = 1;                   //strobe the enable pin
    MSDelay(1);
    en = 0;
  }

void lcddata(unsigned char value)
  {
    ldata = value;            //put the value on the pins
    rs = 1;
    rw = 0;
    en = 1;                   //strobe the enable pin
    MSDelay(1);
    en = 0;
  }
```



The following is the C version of Program 12-2, using the busy flag method.

```
//Program 12-2C. C version of Program 12-2
#include <P18F458.h>
#define ldata PORTD   //PORTD = LCD data pins (Fig. 12-2)
#define rs PORTBbits.RB0    //rs = PORTB.0
#define rw PORTBbits.RB1    //rw = PORTB.1
#define en PORTBbits.RB2    //en = PORTB.2
#define busy PORTDbits.RD7 //busy = PORTD.7

void main()
  {
    TRISD = 0;             //both ports B and D as output
```



```
     TRISB = 0;
     en = 0;                          //enable idle low
     MSDelay(250);                    //long delay
     lcdcmd(0x38);
     MSDelay(250);                    //long delay
     lcdcmd(0x0E);
     lcdready();                      //check the LCD busy flag
     lcdcmd(0x01);
     lcdready();                      //check the LCD busy flag
     lcdcmd(0x06);
     lcdready();                      //check the LCD busy flag
     lcdcmd(0x86);                    //line 1, position 6
     lcdready();                      //check the LCD busy flag
     lcddata('M');
     lcdready();                      //check the LCD busy flag
     lcddata('D');
     lcdready();                      //check the LCD busy flag
     lcddata('E');
  }

void lcdcmd(unsigned char value)
  {
    ldata = value;                    //put the value on the pins
    rs = 0;
    rw = 0;
    en = 1;                           //strobe the enable pin
    MSDelay(1);
    en = 0;
  }

void lcddata(unsigned char value)
  {
    ldata = value;                    //put the value on the pins
    rs = 1;
    rw = 0;
    en = 1;                           //strobe the enable pin
    MSDelay(1);
    en = 0;
  }

void lcdready()
  {
    TRISD = 0xFF;                     //make PORTD an input
    rs = 0;
    rw = 1;
    do                                //wait here for busy flag
      {
        en = 1;                       //strobe the enable pin
        MSDelay(1);
        en = 0;
      }while(busy==1);
     TRISD = 0;
   }
```