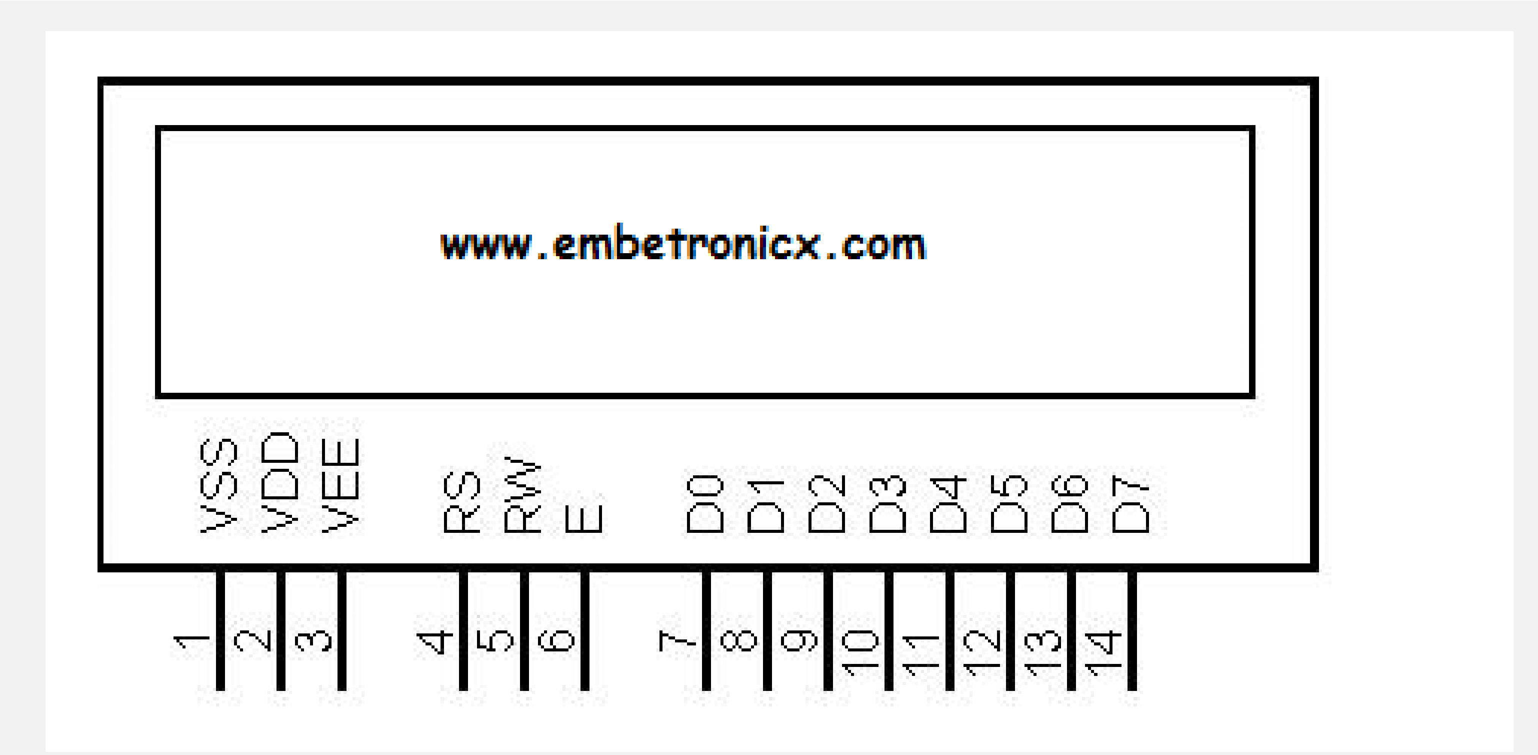
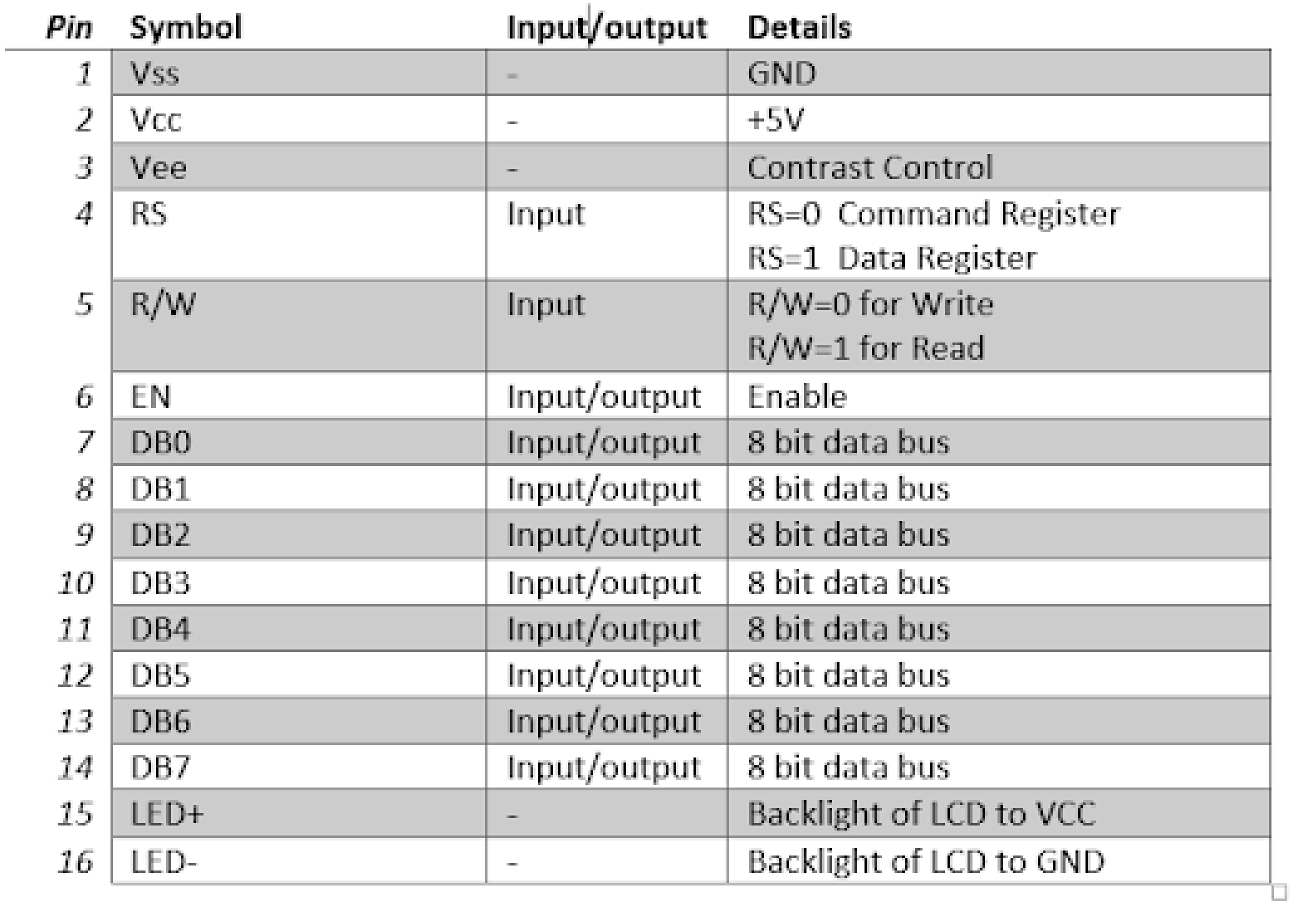
Interfacing an external LCD to the MSP430 Microcontroller and Onboard LCD :

- Engr.Shahzada Fahim Jan

- LCDs are economical
- easily programmable
- have no limitation of displaying special & even custom characters (unlike in seven segments), animations, and so on.
- A 16×2 LCD means it can display 16 characters per line and there are 2 such lines
- In this LCD each character is displayed in a 5×7 pixel matrix.
- This LCD has two registers, namely, Command and Data.

- The command register stores the command instructions given to the LCD.
- A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling the display, etc.
- The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD.


www.embetronicx.com
VSS
VDD
VEE
RS
RW
E
D0
D1
D2
D3
D4
D5
D6
D7
1
2
3
4
5
6
7
8
9
10
11
12
13
14

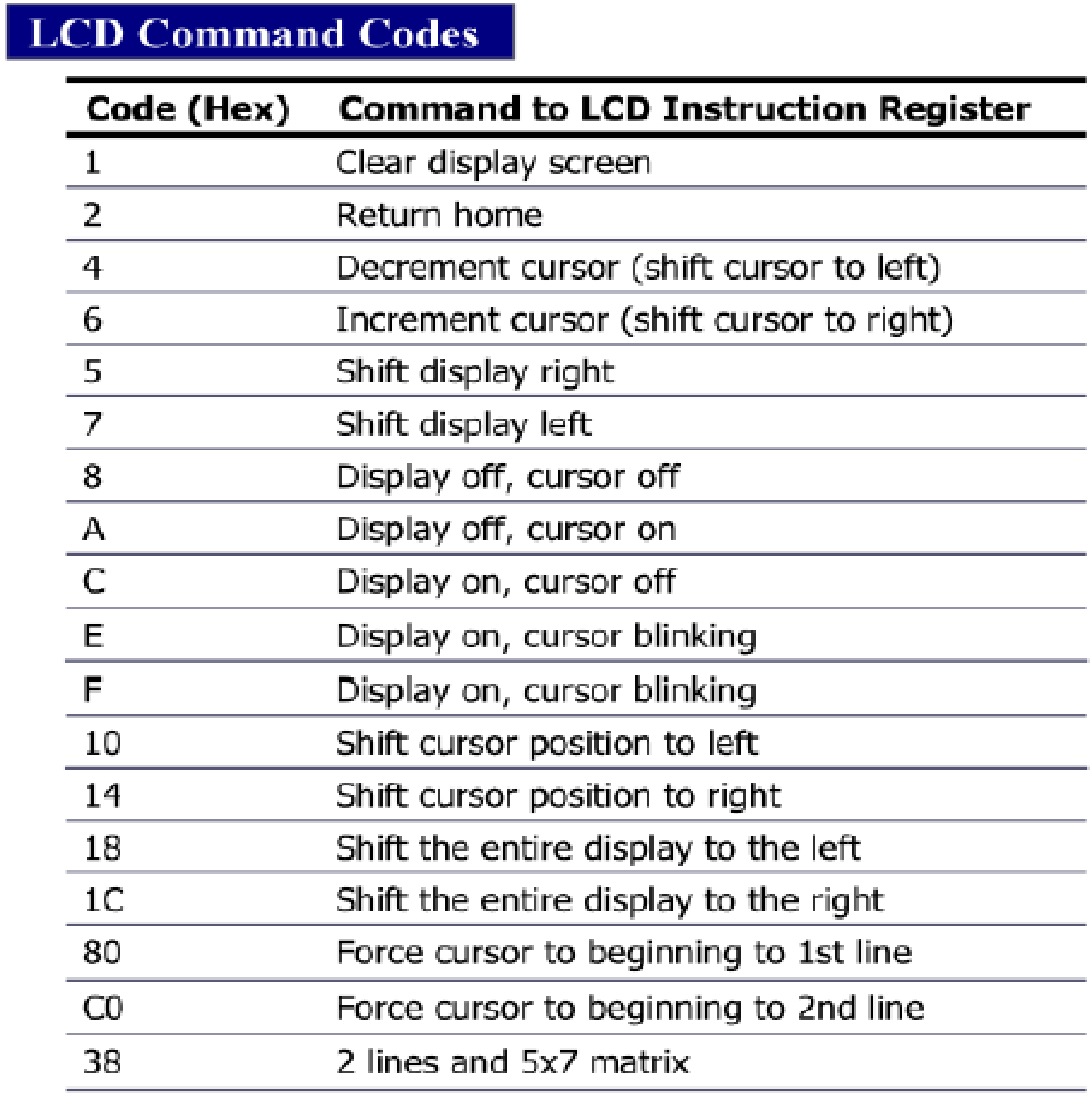


Pin	**Symbol**	**Input/output**	**Details**
1	Vss	-	GND
2	Vcc	-	+5V
3	Vee	-	Contrast Control

4	RS	Input	RS=0 Command Register
			RS=1 Data Register

5	R/W	Input	R/W=0 for Write R/W=1 for Read
6	EN	Input/output	Enable
7	DB0	Input/output	8 bit data bus
8	DB1	Input/output	8 bit data bus
9	DB2	Input/output	8 bit data bus
10	DB3	Input/output	8 bit data bus
11	DB4	Input/output	8 bit data bus
12	DB5	Input/output	8 bit data bus
13	DB6	Input/output	8 bit data bus
14	DB7	Input/output	8 bit data bus
15	LED+	-	Backlight of LCD to VCC
16	LED-	-	Backlight of LCD to GND

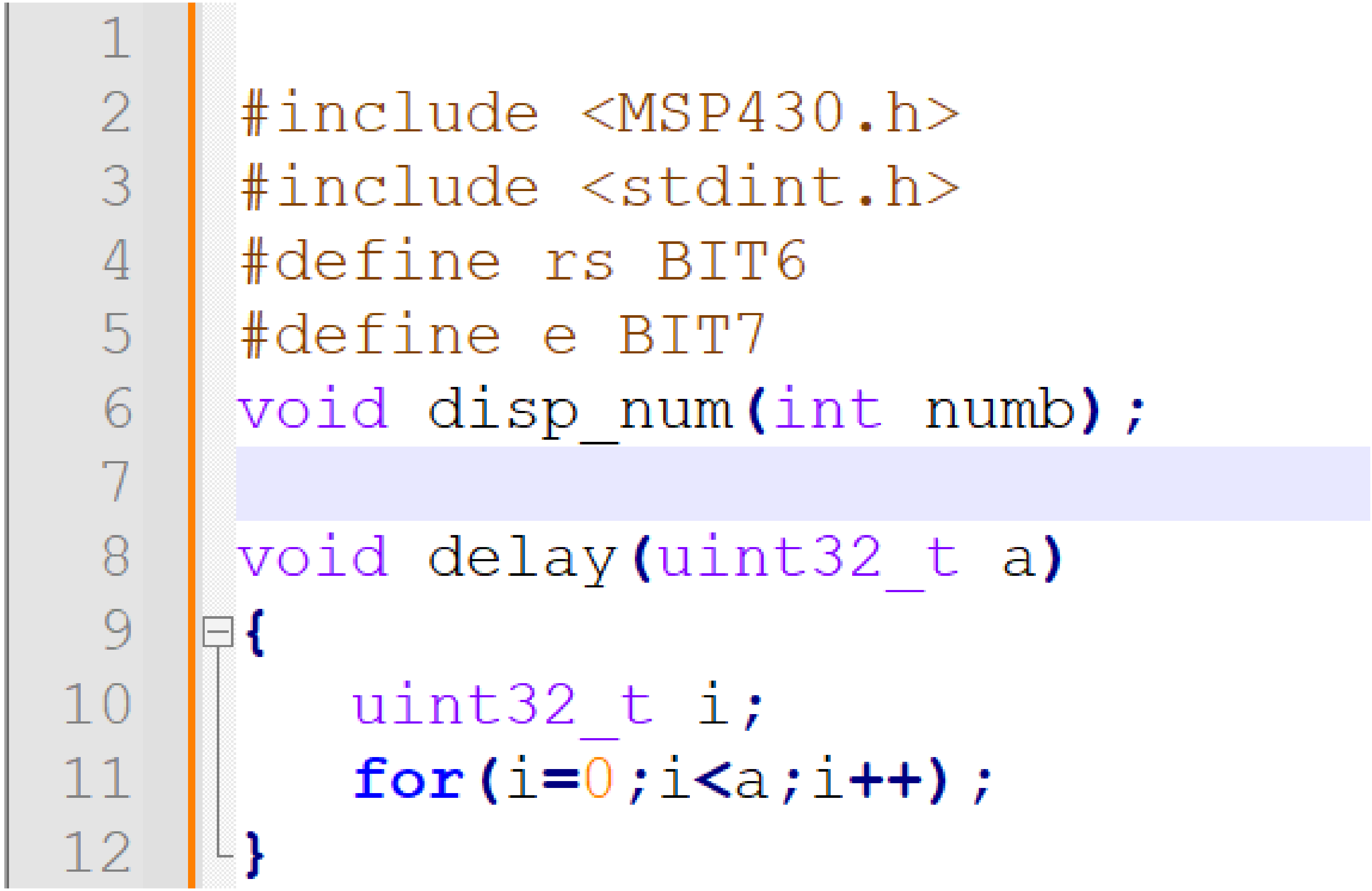
- **2) RS (Register Select):**
-
- The RS pin is used to select command code register or data register. If RS=0 the command code register is selected which allows us to send the instructions to LCD. If RS=1 the data register is selected which allows us to send data to be displayed on LCD.
- **3) RW(Read/Write):**
-
- R/W input allows the user to write information to the LCD or read info from it.
- R/W =1 when reading
- R/W =0 when writing
- **4) EN (Enable):**
- when data or command is sent to LCD a high-to-low pulse must applied to the PIN. So Enable pin is toggled by sending first 1 and then 0.



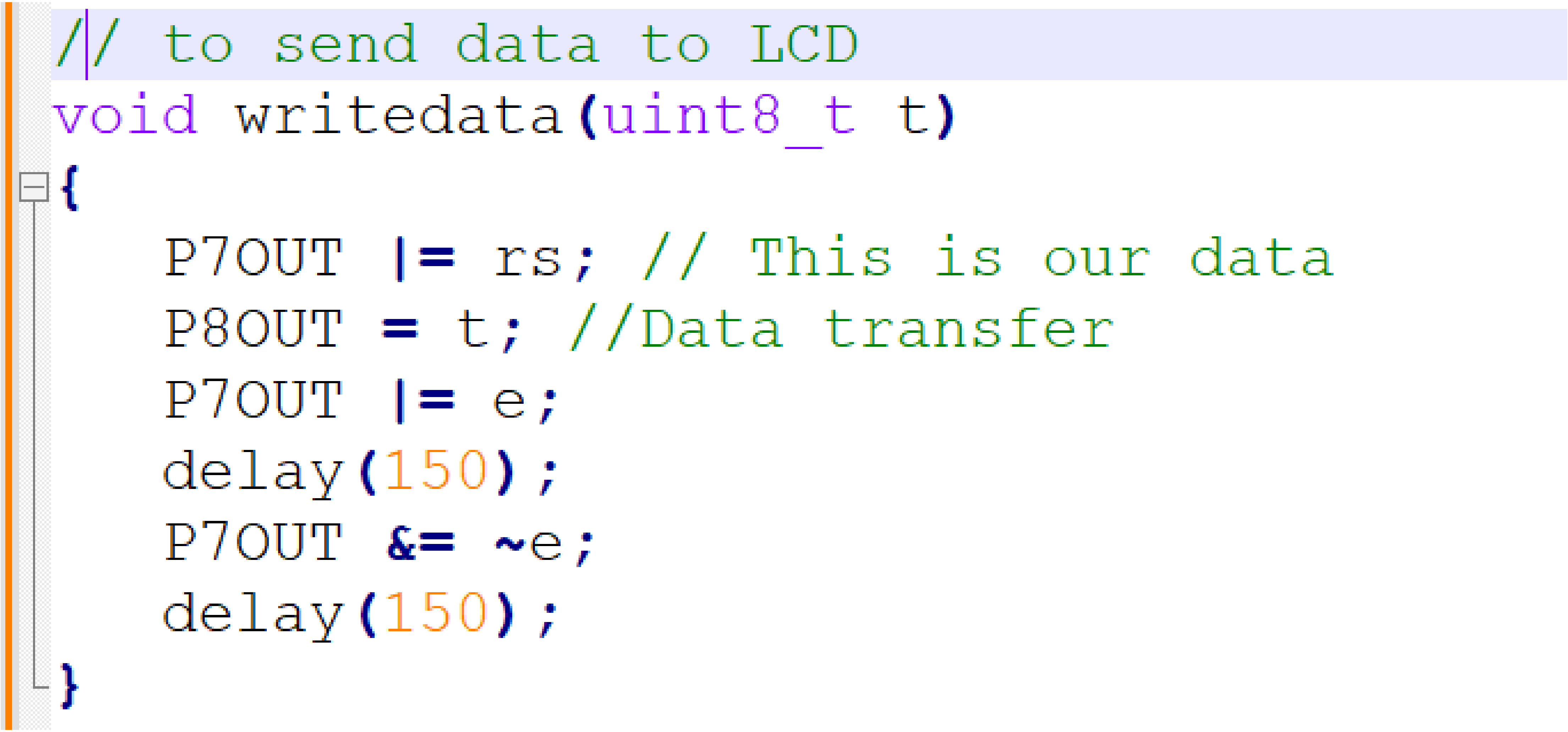
LCD Command Codes

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 to 1st line
C0	Force cursor to beginning to 2nd line
38	2 lines and 5x7 matrix

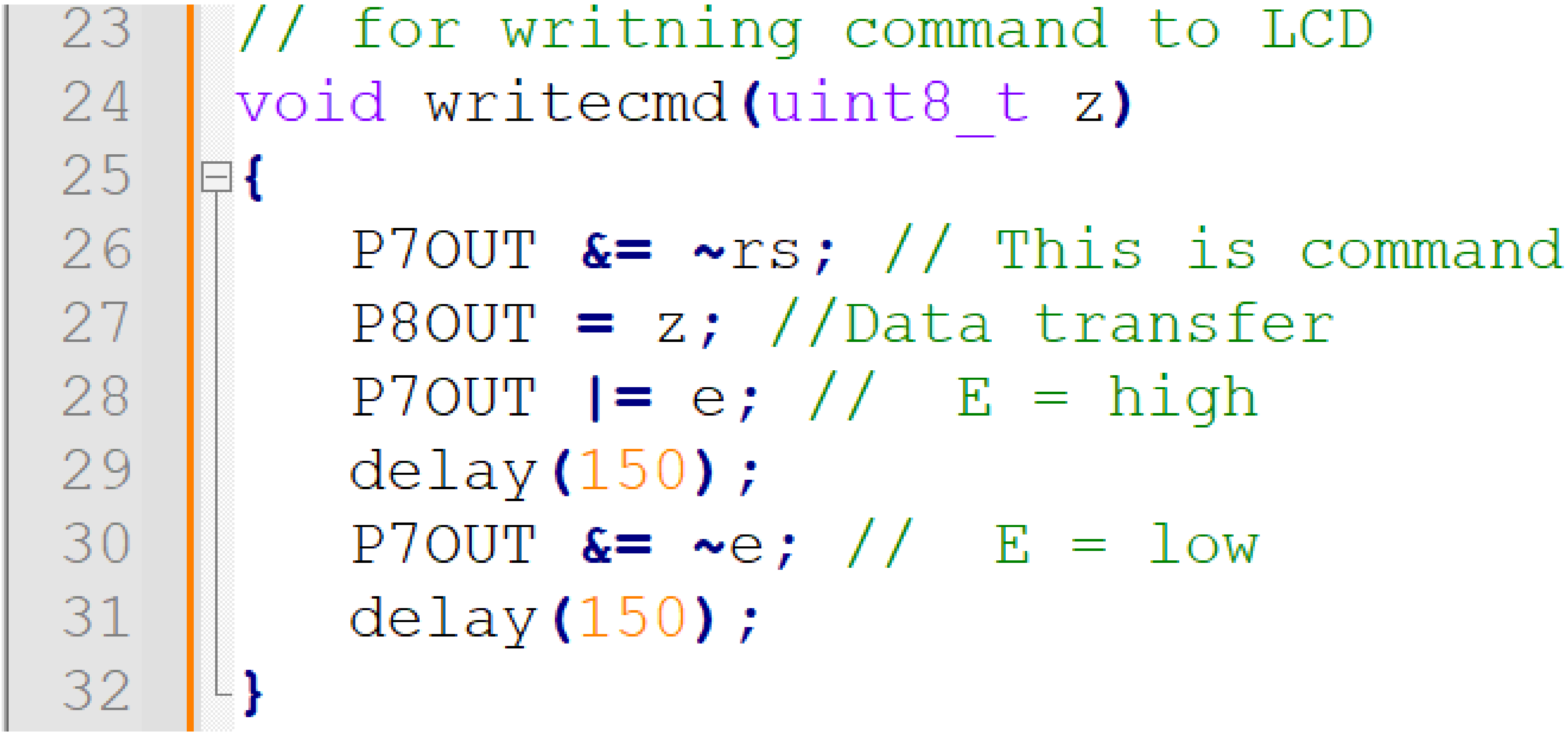
Interfacing external LCD (proteus)



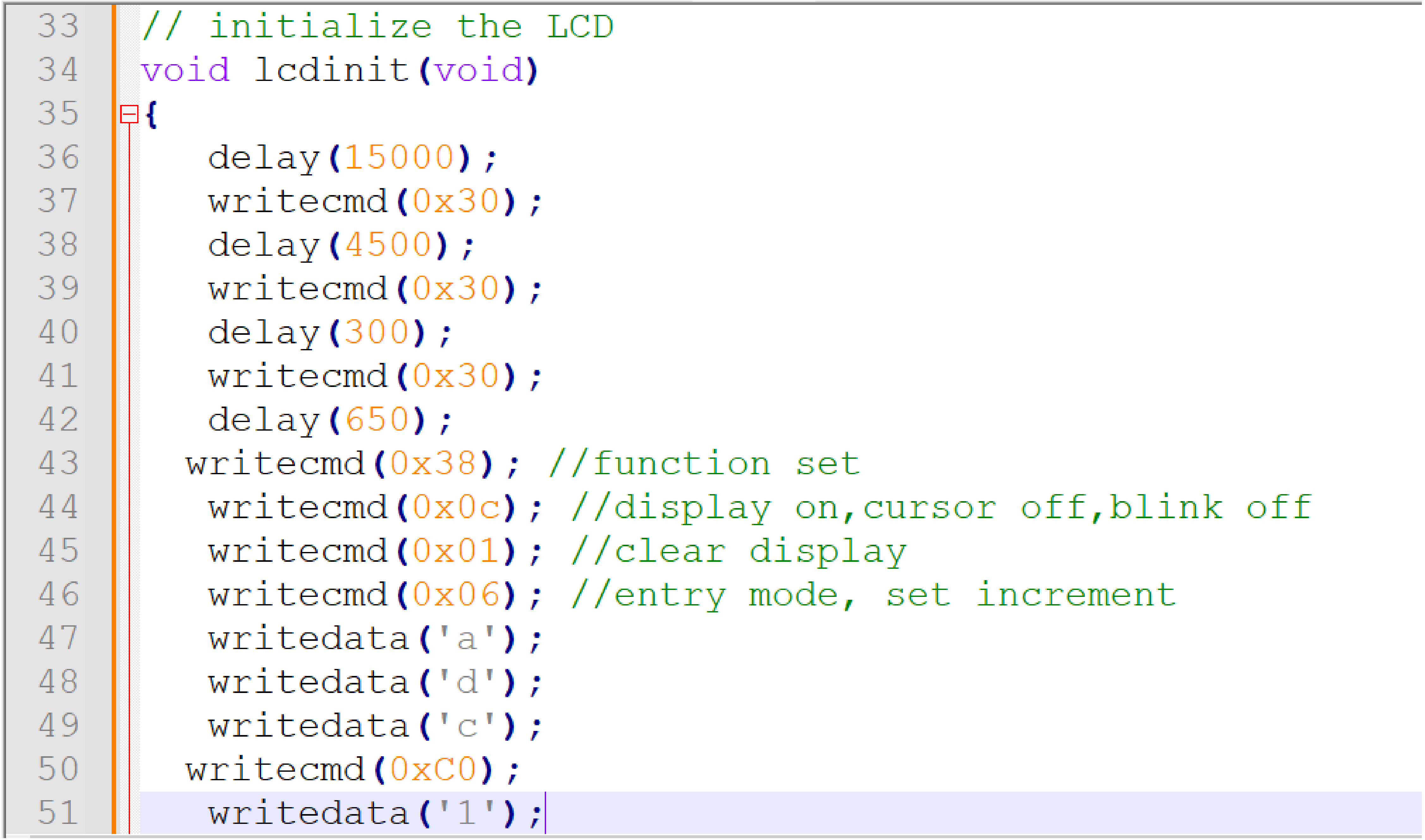
```

#include <MSP430.h>
#include <stdint.h>
#define rs BIT6
#define e BIT7
void disp_num(int numb);

void delay(uint32_t a)
{
    uint32_t i;
    for(i=0;i<a;i++);
}
```



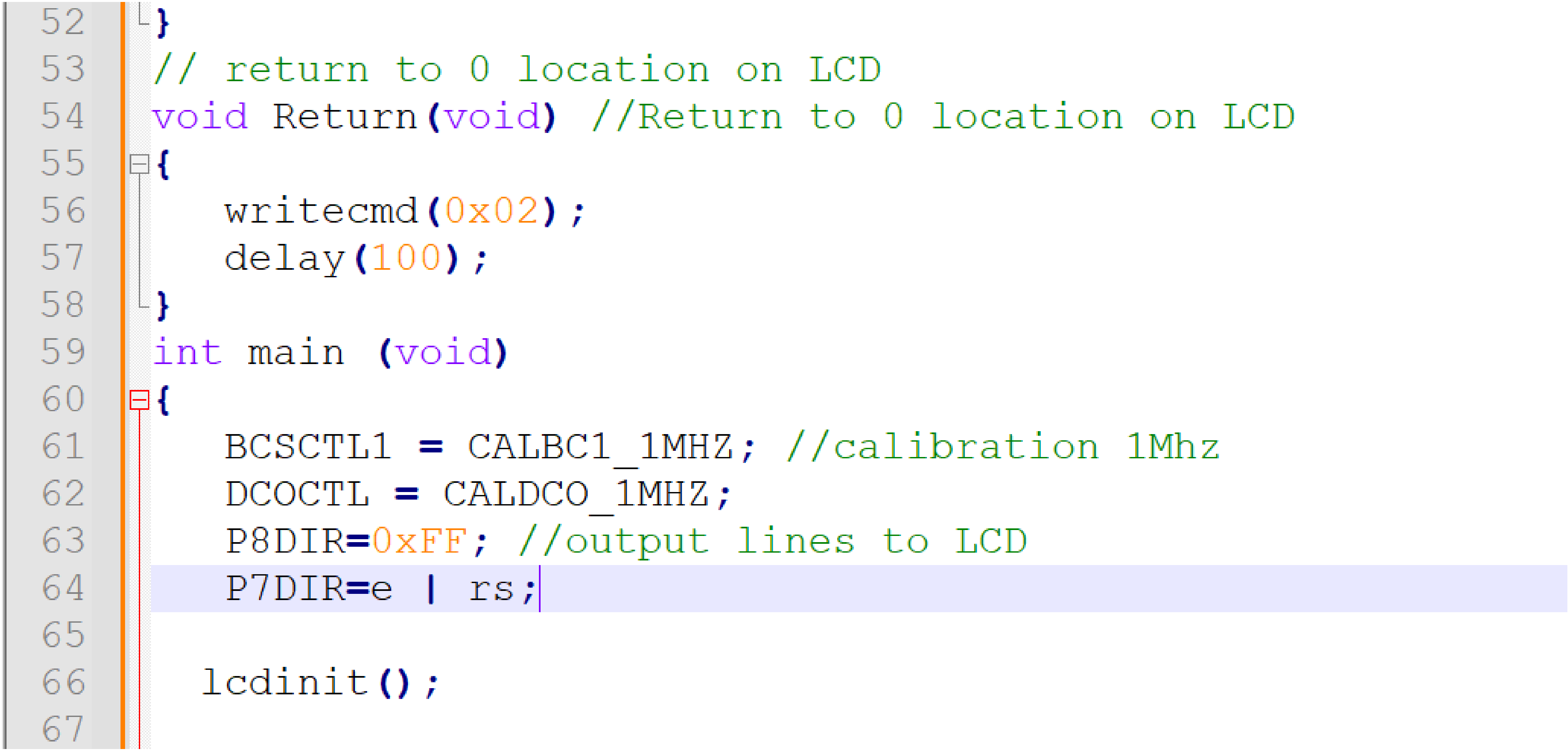
```c
// to send data to LCD
void writedata(uint8_t t)
{
    P7OUT |= rs; // This is our data
    P8OUT = t; //Data transfer
    P7OUT |= e;
    delay(150);
    P7OUT &= ~e;
    delay(150);
}
```



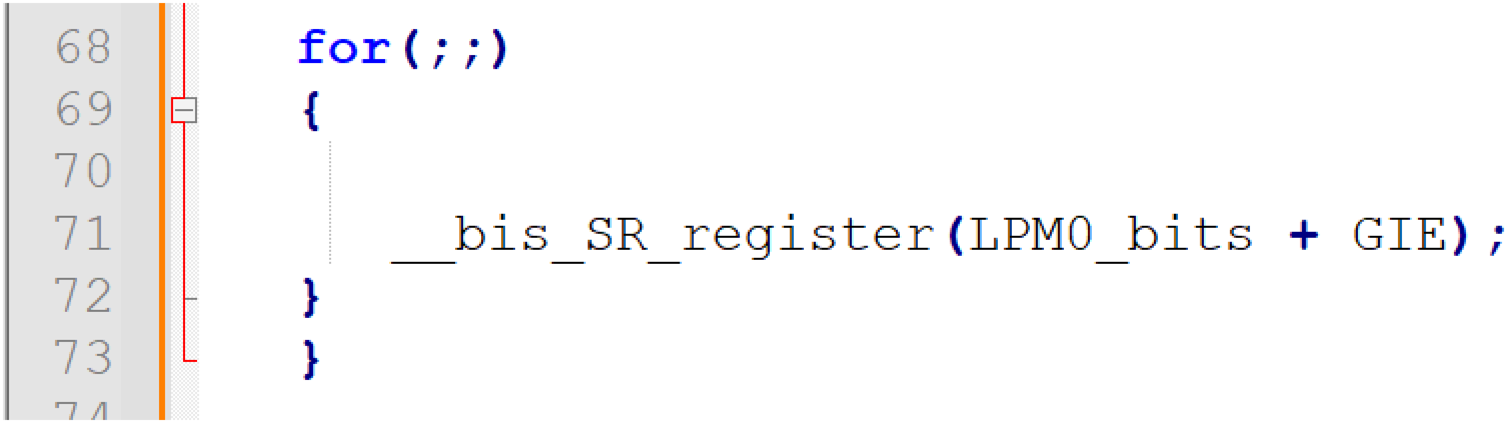
```
// for writning command to LCD
void writecmd(uint8_t z)
{
    P7OUT &= ~rs; // This is command
    P8OUT = z; //Data transfer
    P7OUT |= e; //  E = high
    delay(150);
    P7OUT &= ~e; //  E = low
    delay(150);
}
```



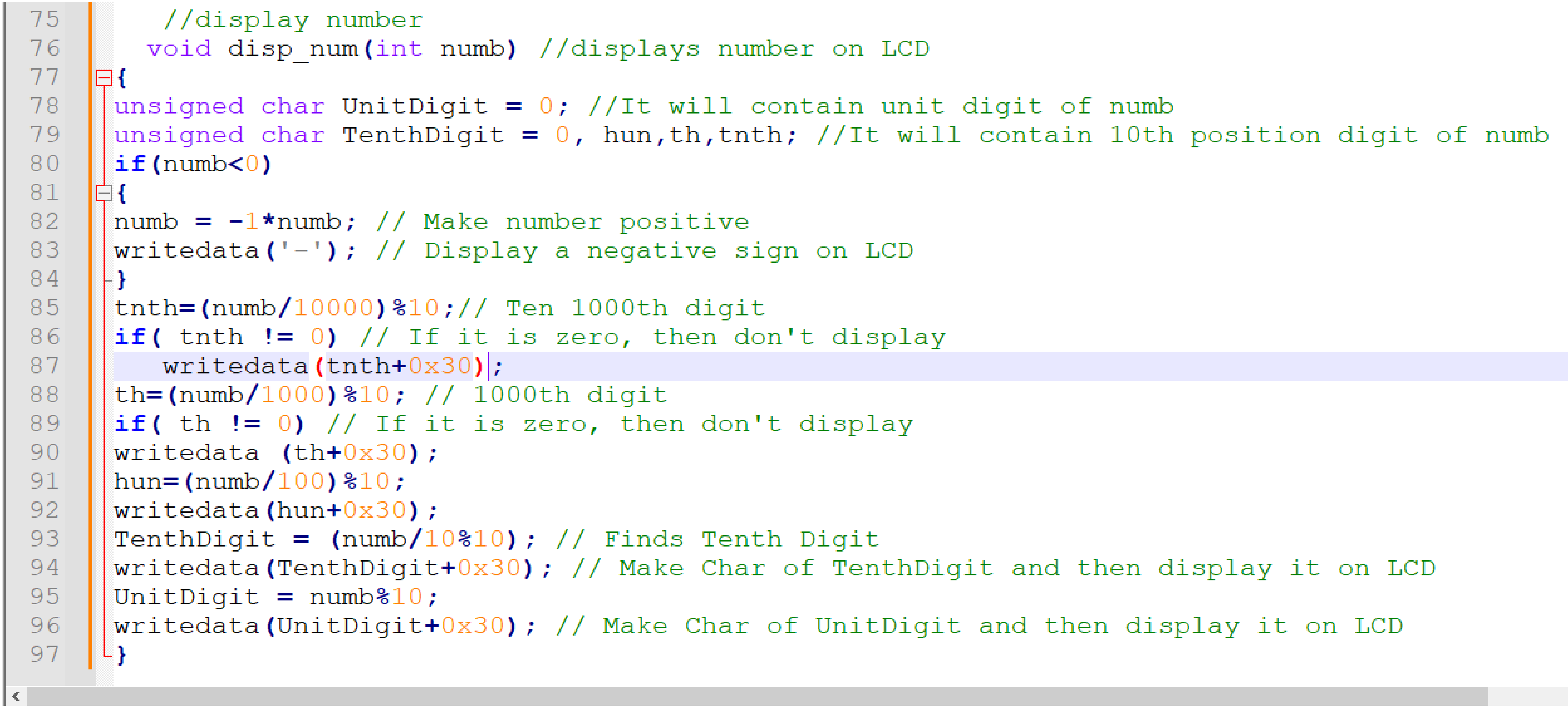
```c
// initialize the LCD
void lcdinit(void)
{
   delay(15000);
   writecmd(0x30);
   delay(4500);
   writecmd(0x30);
   delay(300);
   writecmd(0x30);
   delay(650);
  writecmd(0x38); //function set
   writecmd(0x0c); //display on,cursor off,blink off
   writecmd(0x01); //clear display
   writecmd(0x06); //entry mode, set increment
   writedata('a');
   writedata('d');
   writedata('c');
  writecmd(0xC0);
   writedata('1');
```



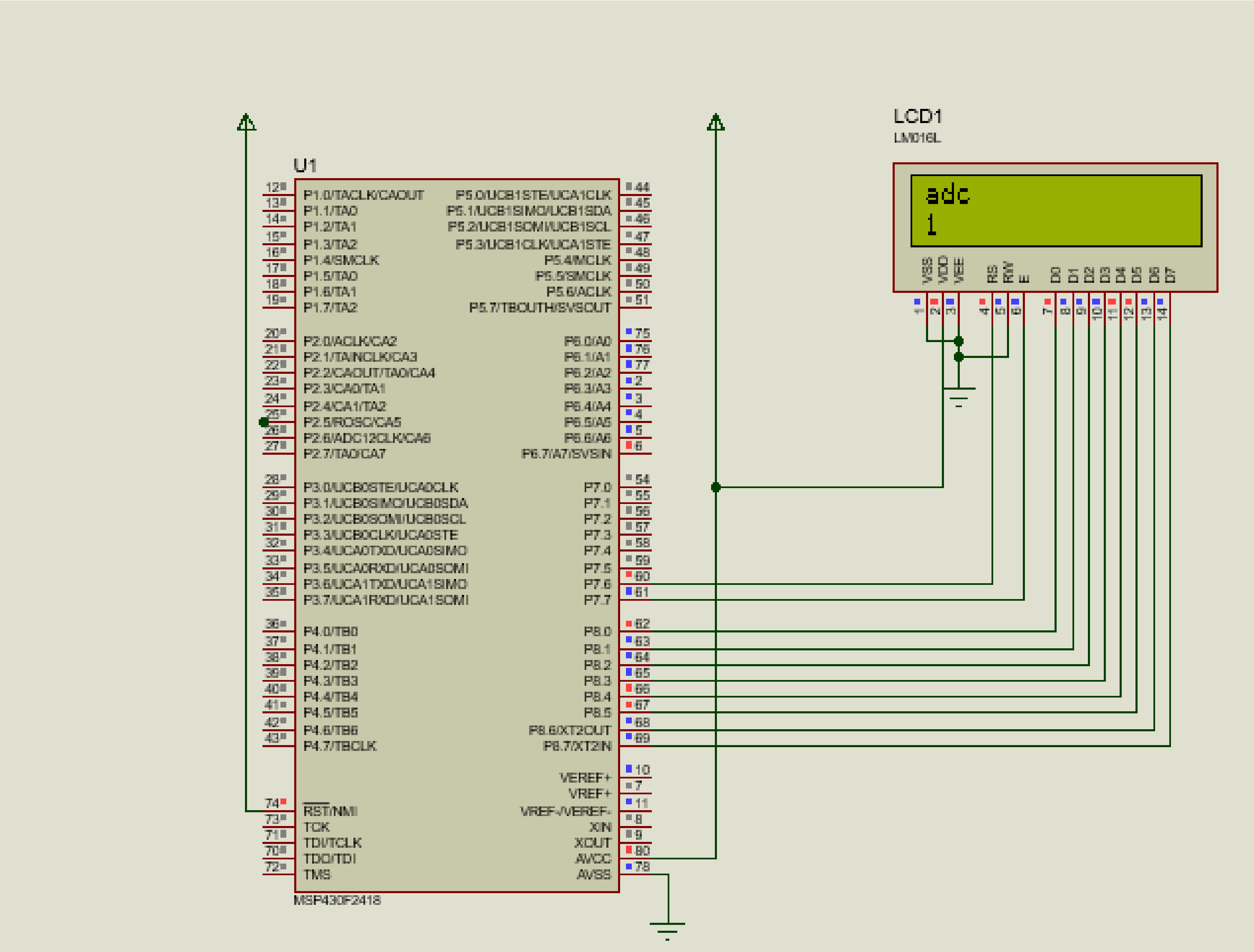
```
}
// return to 0 location on LCD
void Return(void) //Return to 0 location on LCD
{
   writecmd(0x02);
   delay(100);
}
int main (void)
{
   BCSCTL1 = CALBC1_1MHZ; //calibration 1Mhz
   DCOCTL = CALDCO_1MHZ;
   P8DIR=0xFF; //output lines to LCD
   P7DIR=e | rs;

  lcdinit();

```

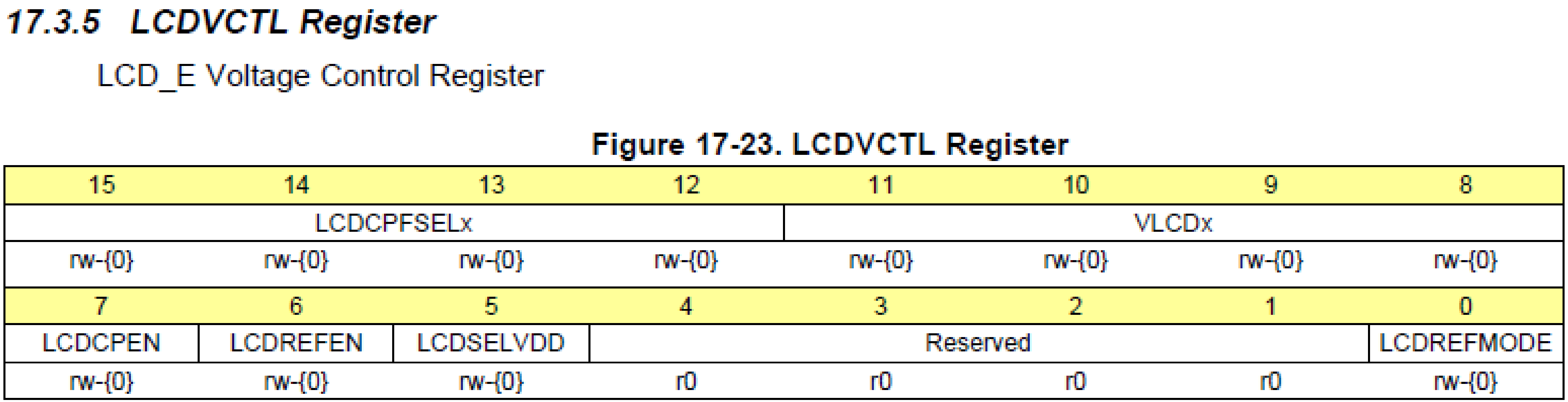


```
for(;;)
{

   __bis_SR_register(LPM0_bits + GIE);
}
}
```



```
   //display number
  void disp_num(int numb) //displays number on LCD
{
unsigned char UnitDigit = 0; //It will contain unit digit of numb
unsigned char TenthDigit = 0, hun,th,tnth; //It will contain 10th position digit of numb
if(numb<0)
{
numb = -1*numb; // Make number positive
writedata('-'); // Display a negative sign on LCD
}
tnth=(numb/10000)%10;// Ten 1000th digit
if( tnth != 0) // If it is zero, then don't display
   writedata(tnth+0x30);
th=(numb/1000)%10; // 1000th digit
if( th != 0) // If it is zero, then don't display
writedata (th+0x30);
hun=(numb/100)%10;
writedata(hun+0x30);
TenthDigit = (numb/10%10); // Finds Tenth Digit
writedata(TenthDigit+0x30); // Make Char of TenthDigit and then display it on LCD
UnitDigit = numb%10;
writedata(UnitDigit+0x30); // Make Char of UnitDigit and then display it on LCD
}
```


U1
MSP430F2418
LCD1
LM016L
adc
1
P1.0/TACLK/CAOUT
P1.1/TA0
P1.2/TA1
P1.3/TA2
P1.4/SMCLK
P1.5/TA0
P1.6/TA1
P1.7/TA2
P2.0/ACLK/CA2
P2.1/TAINCLK/CA3
P2.2/CAOUT/TA0/CA4
P2.3/CA0/TA1
P2.4/CA1/TA2
P2.5/ROSC/CA5
P2.6/ADC12CLK/CA6
P2.7/TA0/CA7
P3.0/UCB0STE/UCA0CLK
P3.1/UCB0SIMO/UCB0SDA
P3.2/UCB0SOMI/UCB0SCL
P3.3/UCB0CLK/UCA0STE
P3.4/UCA0TXD/UCA0SIMO
P3.5/UCA0RXD/UCA0SOMI
P3.6/UCA1TXD/UCA1SIMO
P3.7/UCA1RXD/UCA1SOMI
P4.0/TB0
P4.1/TB1
P4.2/TB2
P4.3/TB3
P4.4/TB4
P4.5/TB5
P4.6/TB6
P4.7/TBCLK
RST/NMI
TCK
TDI/TCLK
TDO/TDI
TMS
P5.0/UCB1STE/UCA1CLK
P5.1/UCB1SIMO/UCB1SDA
P5.2/UCB1SOMI/UCB1SCL
P5.3/UCB1CLK/UCA1STE
P5.4/MCLK
P5.5/SMCLK
P5.6/ACLK
P5.7/TBOUTH/SVSOUT
P6.0/A0
P6.1/A1
P6.2/A2
P6.3/A3
P6.4/A4
P6.5/A5
P6.6/A6
P6.7/A7/SVSIN
P7.0
P7.1
P7.2
P7.3
P7.4
P7.5
P7.6
P7.7
P8.0
P8.1
P8.2
P8.3
P8.4
P8.5
P8.6/XT2OUT
P8.7/XT2IN
VEREF+
VREF+
VREF-/VEREF-
XIN
XOUT
AVCC
AVSS
12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27
28 29 30 31 32 33 34 35
36 37 38 39 40 41 42 43
74 73 71 70 72
44 45 46 47 48 49 50 51
75 76 77 2 3 4 5 6
54 55 56 57 58 59 60 61
62 63 64 65 66 67 68 69
10 7 11 8 9 80 78
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
1 2 3 4 5 6 7 8 9 10 11 12 13 14



17.3.5 LCDVCTL Register

LCD_E Voltage Control Register

Figure 17-23. LCDVCTL Register

15	14	13	12	11	10	9	8
LCDCPFSELx				VLCDx			
rw-{0}	rw-{0}	rw-{0}	rw-{0}	rw-{0}	rw-{0}	rw-{0}	rw-{0}
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
LCDCPEN	LCDREFEN	LCDSELVDD	Reserved				LCDREFMODE
rw-{0}	rw-{0}	rw-{0}	r0	r0	r0	r0	rw-{0}

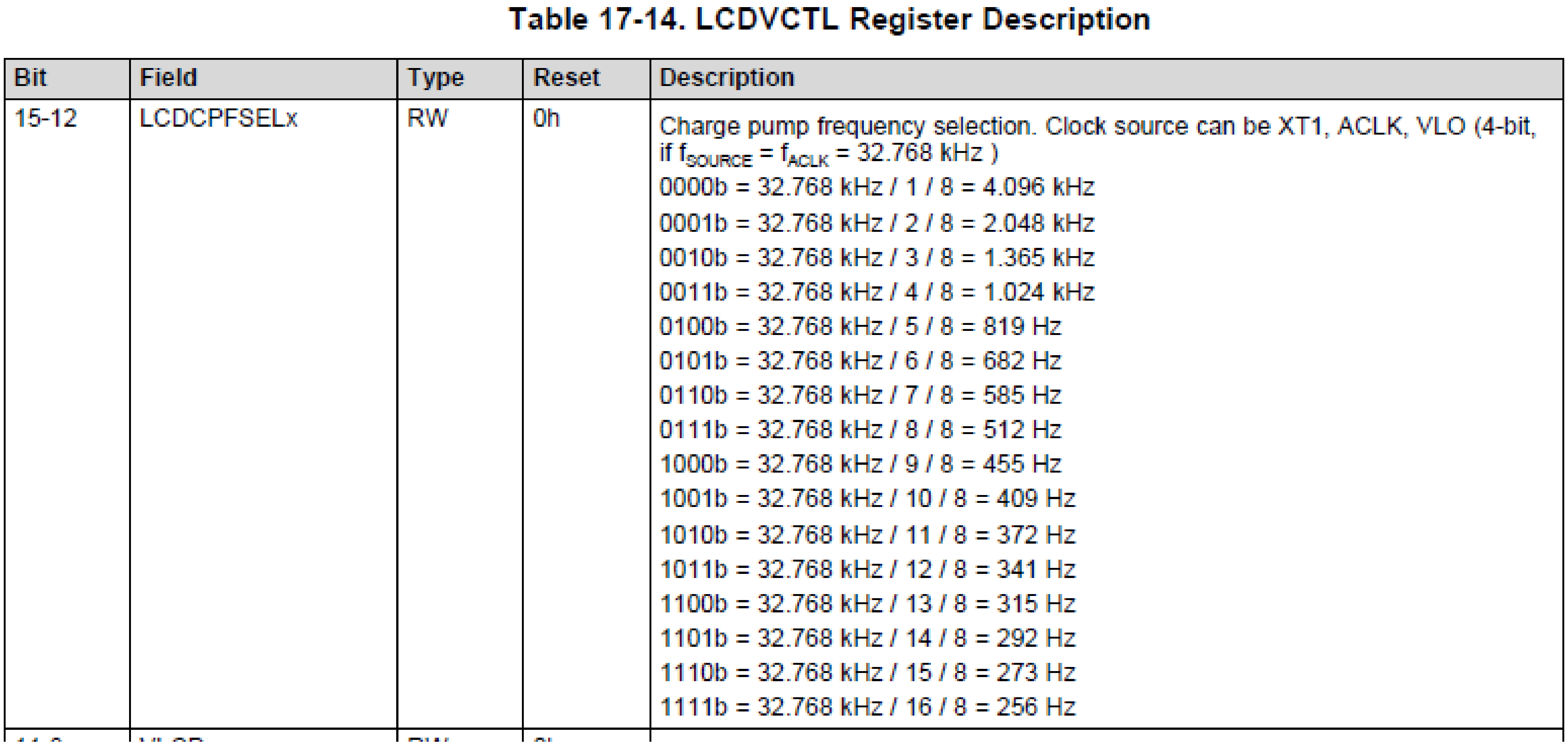
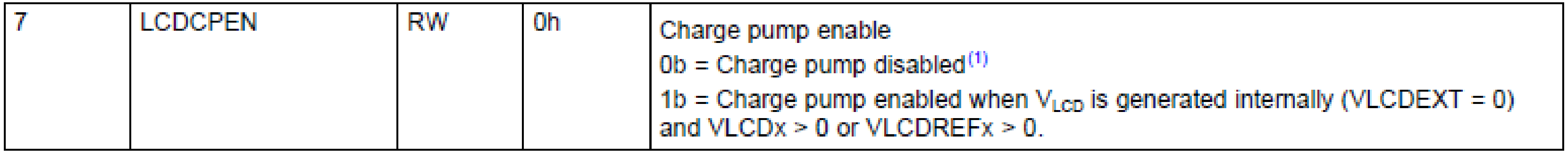


Table 17-14. LCDVCTL Register Description

Bit	Field	Type	Reset	Description
15-12	LCDCPFSELx	RW	0h	Charge pump frequency selection. Clock source can be XT1, ACLK, VLO (4-bit, if f_{SOURCE} = f_{ACLK} = 32.768 kHz) 0000b = 32.768 kHz / 1 / 8 = 4.096 kHz 0001b = 32.768 kHz / 2 / 8 = 2.048 kHz 0010b = 32.768 kHz / 3 / 8 = 1.365 kHz 0011b = 32.768 kHz / 4 / 8 = 1.024 kHz 0100b = 32.768 kHz / 5 / 8 = 819 Hz 0101b = 32.768 kHz / 6 / 8 = 682 Hz 0110b = 32.768 kHz / 7 / 8 = 585 Hz 0111b = 32.768 kHz / 8 / 8 = 512 Hz 1000b = 32.768 kHz / 9 / 8 = 455 Hz 1001b = 32.768 kHz / 10 / 8 = 409 Hz 1010b = 32.768 kHz / 11 / 8 = 372 Hz 1011b = 32.768 kHz / 12 / 8 = 341 Hz 1100b = 32.768 kHz / 13 / 8 = 315 Hz 1101b = 32.768 kHz / 14 / 8 = 292 Hz 1110b = 32.768 kHz / 15 / 8 = 273 Hz 1111b = 32.768 kHz / 16 / 8 = 256 Hz

11-8	VLCDx	RW	0h	Internal reference voltage select on R13. Only valuable when LCDCPEN = 1 and LCDREFEN = 1. 0000b = 2.60 V 0001b = 2.66 V 0010b = 2.72 V 0011b = 2.78 V 0100b = 2.84 V 0101b = 2.90 V 0110b = 2.96 V 0111b = 3.02 V 1000b = 3.08 V 1001b = 3.14 V 1010b = 3.20 V 1011b = 3.26 V 1100b = 3.32 V 1101b = 3.38 V 1110b = 3.44 V 1111b = 3.50 V



7	LCDCPEN	RW	0h	Charge pump enable 0b = Charge pump disabled[1] 1b = Charge pump enabled when V_{LCD} is generated internally (VLCDEXT = 0) and VLCDx > 0 or VLCDREFx > 0.

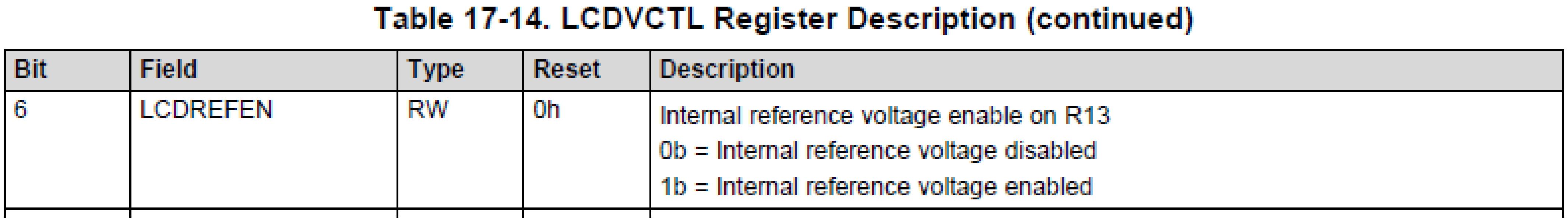
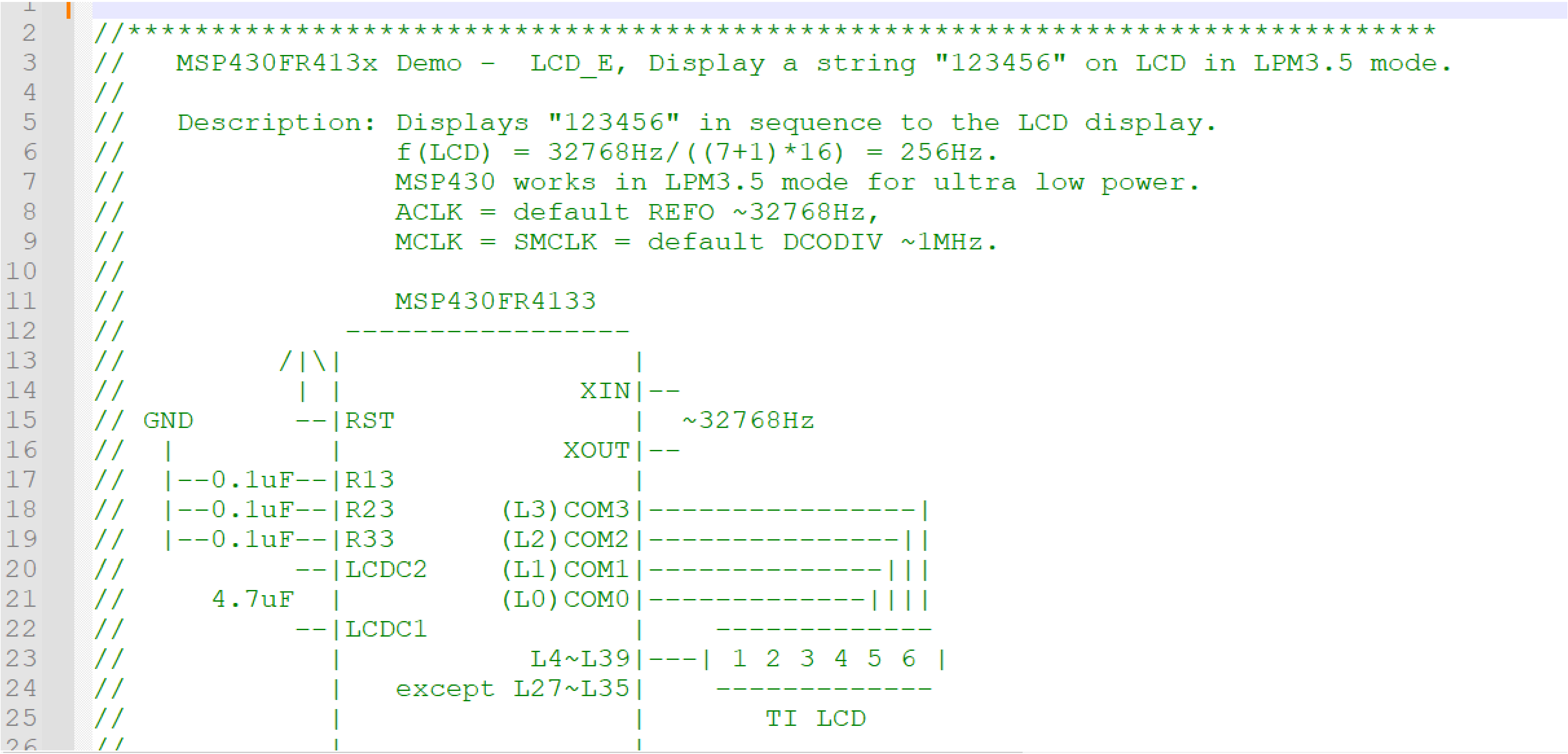


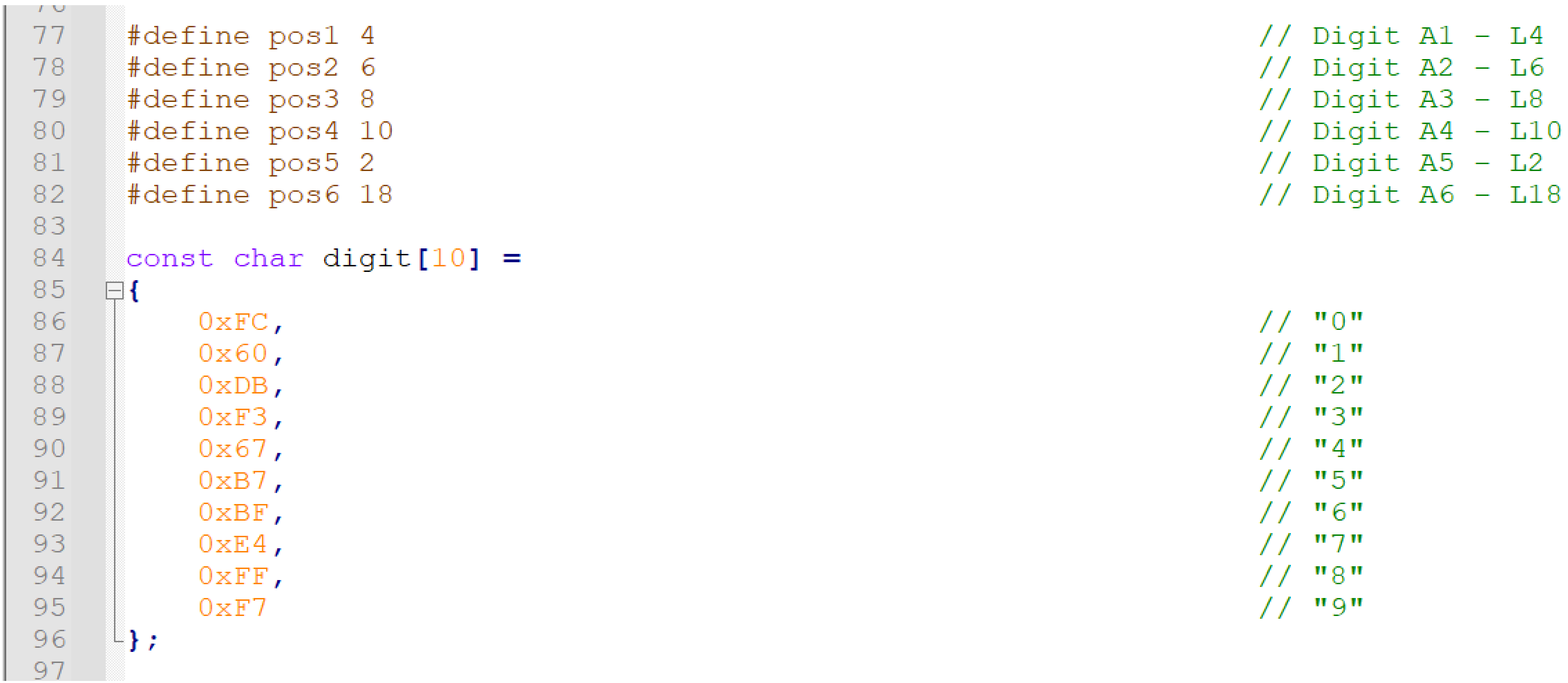
Table 17-14. LCDVCTL Register Description (continued)

Bit	Field	Type	Reset	Description
6	LCDREFEN	RW	0h	Internal reference voltage enable on R13 0b = Internal reference voltage disabled 1b = Internal reference voltage enabled

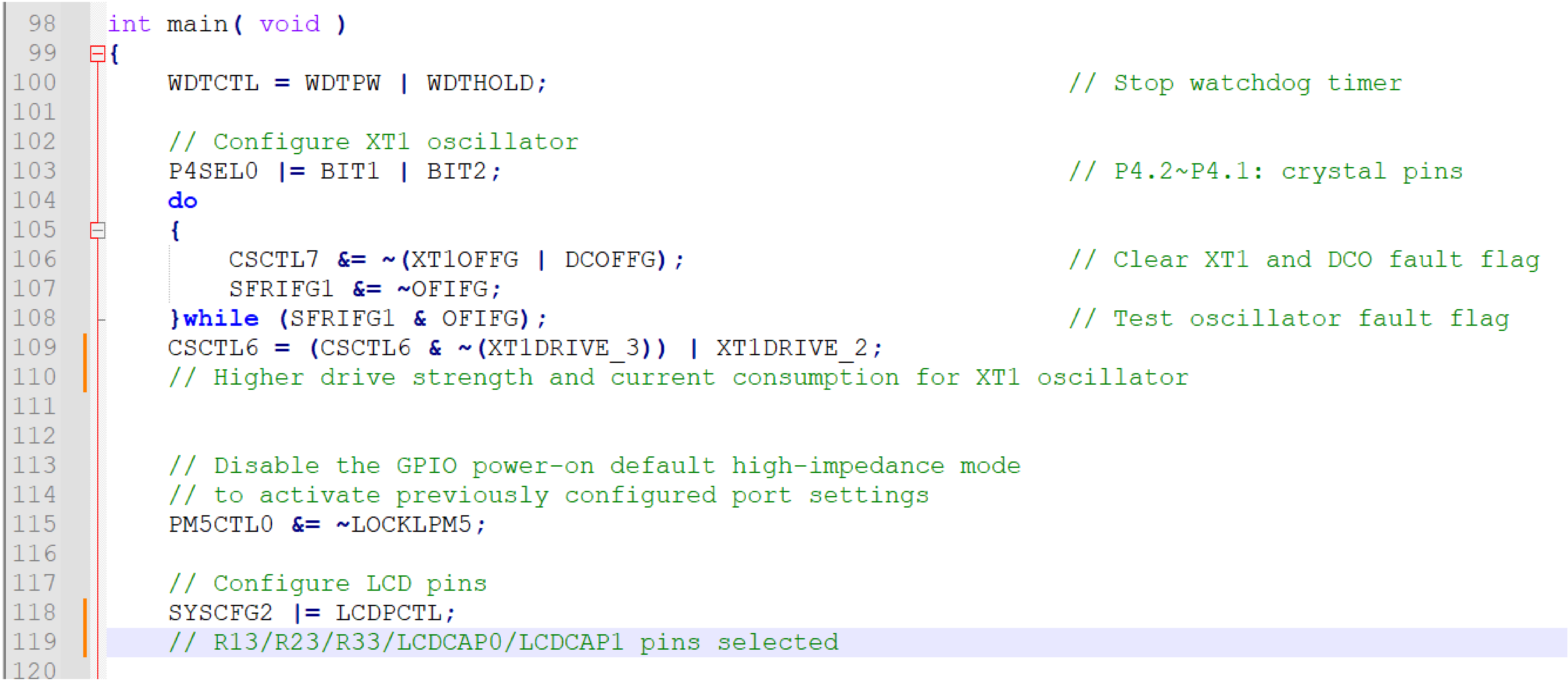
Onboard LCD of MSP430FR4133



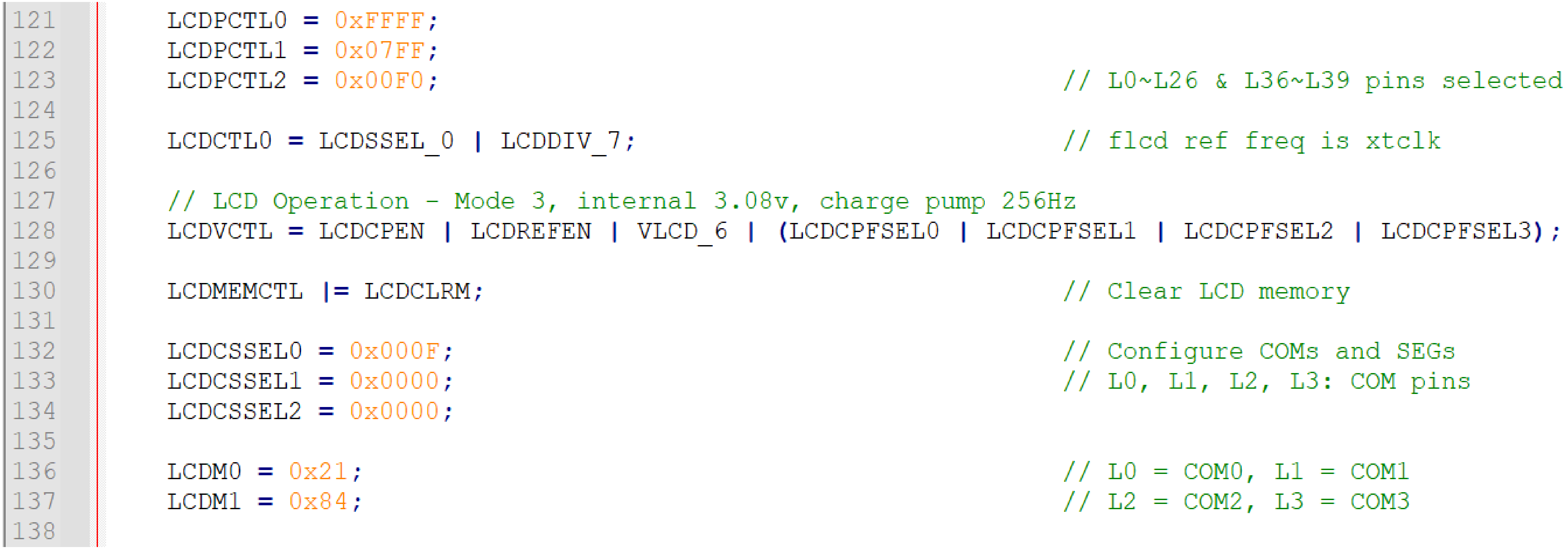
```

//******************************************************************************
//   MSP430FR413x Demo -  LCD_E, Display a string "123456" on LCD in LPM3.5 mode.
//
//   Description: Displays "123456" in sequence to the LCD display.
//                f(LCD) = 32768Hz/((7+1)*16) = 256Hz.
//                MSP430 works in LPM3.5 mode for ultra low power.
//                ACLK = default REFO ~32768Hz,
//                MCLK = SMCLK = default DCODIV ~1MHz.
//
//                MSP430FR4133
//             -----------------
//         /|\|                 |
//          | |              XIN|--
// GND      --|RST              |  ~32768Hz
//  |         |             XOUT|--
//  |--0.1uF--|R13              |
//  |--0.1uF--|R23      (L3)COM3|----------------|
//  |--0.1uF--|R33      (L2)COM2|---------------||
//          --|LCDC2    (L1)COM1|--------------|||
//     4.7uF  |         (L0)COM0|-------------||||
//          --|LCDC1            |    -------------
//            |           L4~L39|---| 1 2 3 4 5 6 |
//            |   except L27~L35|    -------------
//            |                 |       TI LCD
//            |                 |
```



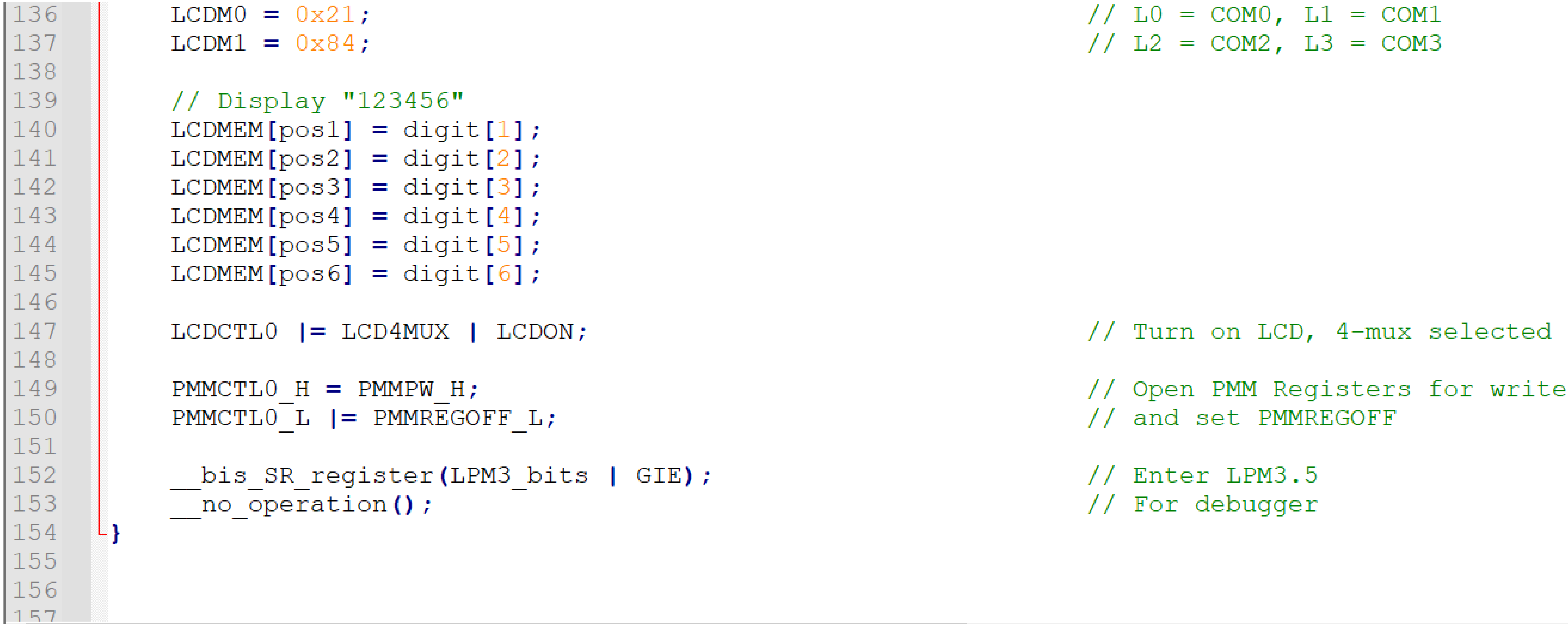
```
#define pos1 4                                                 // Digit A1 - L4
#define pos2 6                                                 // Digit A2 - L6
#define pos3 8                                                 // Digit A3 - L8
#define pos4 10                                                // Digit A4 - L10
#define pos5 2                                                 // Digit A5 - L2
#define pos6 18                                                // Digit A6 - L18

const char digit[10] =
{
    0xFC,                                                      // "0"
    0x60,                                                      // "1"
    0xDB,                                                      // "2"
    0xF3,                                                      // "3"
    0x67,                                                      // "4"
    0xB7,                                                      // "5"
    0xBF,                                                      // "6"
    0xE4,                                                      // "7"
    0xFF,                                                      // "8"
    0xF7                                                       // "9"
};
```



```c
int main( void )
{
    WDTCTL = WDTPW | WDTHOLD;                                   // Stop watchdog timer

    // Configure XT1 oscillator
    P4SEL0 |= BIT1 | BIT2;                                      // P4.2~P4.1: crystal pins
    do
    {
        CSCTL7 &= ~(XT1OFFG | DCOFFG);                          // Clear XT1 and DCO fault flag
        SFRIFG1 &= ~OFIFG;
    }while (SFRIFG1 & OFIFG);                                   // Test oscillator fault flag
    CSCTL6 = (CSCTL6 & ~(XT1DRIVE_3)) | XT1DRIVE_2;
    // Higher drive strength and current consumption for XT1 oscillator


    // Disable the GPIO power-on default high-impedance mode
    // to activate previously configured port settings
    PM5CTL0 &= ~LOCKLPM5;

    // Configure LCD pins
    SYSCFG2 |= LCDPCTL;
    // R13/R23/R33/LCDCAP0/LCDCAP1 pins selected

```



```
    LCDPCTL0 = 0xFFFF;
    LCDPCTL1 = 0x07FF;
    LCDPCTL2 = 0x00F0;                                         // L0~L26 & L36~L39 pins selected

    LCDCTL0 = LCDSSEL_0 | LCDDIV_7;                            // flcd ref freq is xtclk

    // LCD Operation - Mode 3, internal 3.08v, charge pump 256Hz
    LCDVCTL = LCDCPEN | LCDREFEN | VLCD_6 | (LCDCPFSEL0 | LCDCPFSEL1 | LCDCPFSEL2 | LCDCPFSEL3);

    LCDMEMCTL |= LCDCLRM;                                      // Clear LCD memory

    LCDCSSEL0 = 0x000F;                                        // Configure COMs and SEGs
    LCDCSSEL1 = 0x0000;                                        // L0, L1, L2, L3: COM pins
    LCDCSSEL2 = 0x0000;

    LCDM0 = 0x21;                                              // L0 = COM0, L1 = COM1
    LCDM1 = 0x84;                                              // L2 = COM2, L3 = COM3
```



```
    LCDM0 = 0x21;                                              // L0 = COM0, L1 = COM1
    LCDM1 = 0x84;                                              // L2 = COM2, L3 = COM3

    // Display "123456"
    LCDMEM[pos1] = digit[1];
    LCDMEM[pos2] = digit[2];
    LCDMEM[pos3] = digit[3];
    LCDMEM[pos4] = digit[4];
    LCDMEM[pos5] = digit[5];
    LCDMEM[pos6] = digit[6];

    LCDCTL0 |= LCD4MUX | LCDON;                                // Turn on LCD, 4-mux selected

    PMMCTL0_H = PMMPW_H;                                       // Open PMM Registers for write
    PMMCTL0_L |= PMMREGOFF_L;                                  // and set PMMREGOFF

    __bis_SR_register(LPM3_bits | GIE);                        // Enter LPM3.5
    __no_operation();                                          // For debugger
}
```

TASKS:

- 1) Write Your Name on First Line and registration Number on second line of LCD (use proteus or attach external LCD with MSP).
- 2) Write last four digit of your registration Number On the Onboard LCD of msp430fr4133.