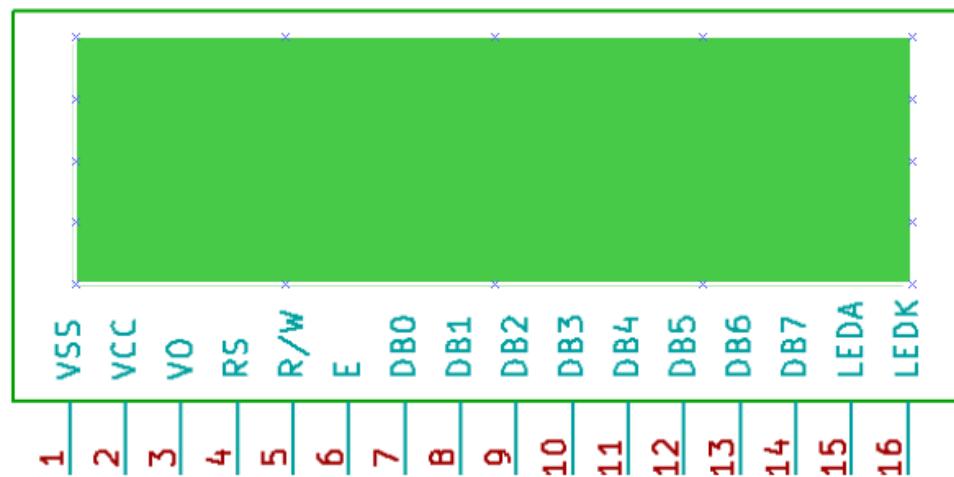
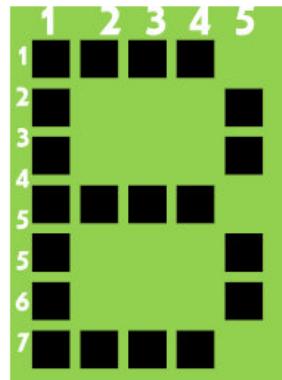


CONTENT

- ✓ **Embedded C Programming:
LCD Interfacing**

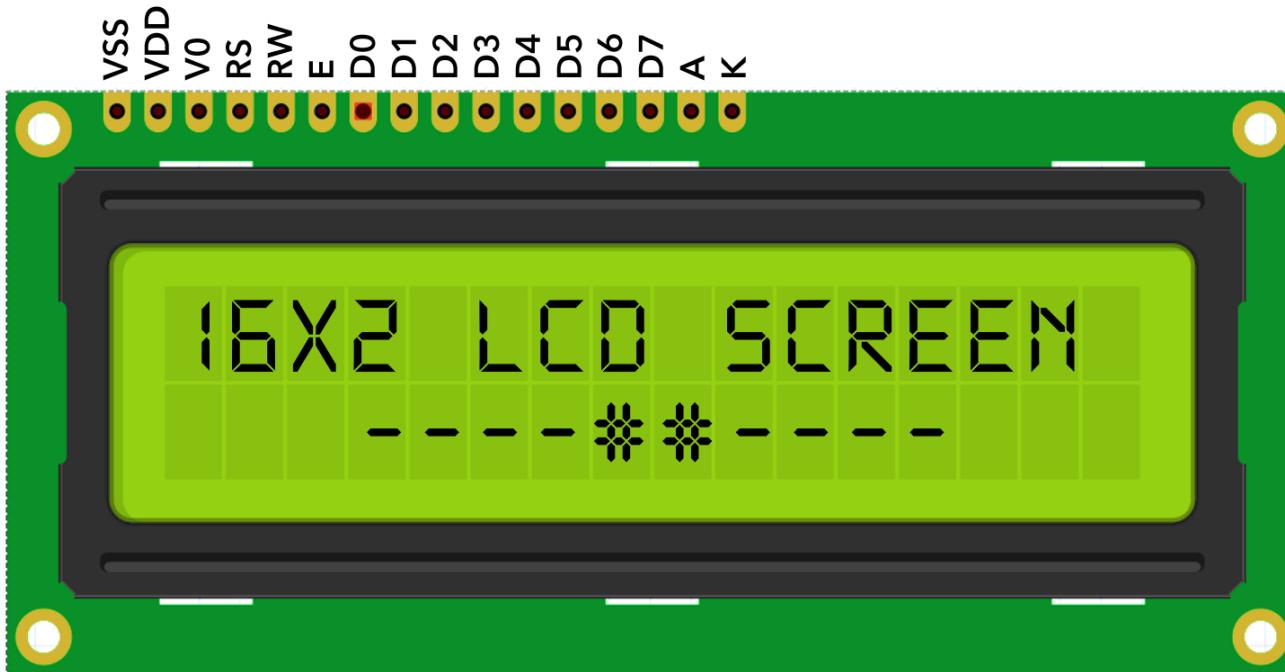


LCD INTERFACING



No	Symbol	Function
1	VSS	Ground
2	VDD	5V +
3	VO	Contrast
4	RS	Register
5	RW	Read/Write
6	E	Enable
7	D0	Data bus
8	D1	Data bus
9	D2	Data bus
10	D3	Data bus
11	D4	Data bus
12	D5	Data bus
13	D6	Data bus
14	D7	Data bus
15	A	Anode (5V+)
16	K	Cathode (GND)

LCD INTERFACING



No	Symbol	Function
1	VSS	Ground
2	VDD	5V +
3	VO	Contrast
4	RS	Register
5	RW	Read/Write
6	E	Enable
7	D0	Data bus
8	D1	Data bus
9	D2	Data bus
10	D3	Data bus
11	D4	Data bus
12	D5	Data bus
13	D6	Data bus
14	D7	Data bus
15	A	Anode (5V+)
16	K	Cathode (GND)

4	RS	Register Select. 0:Command, 1: Data
5	R/W	Read/Write, R/W=0: Write & R/W=1: Read
6	EN	Enable. Falling edge triggered

LCD COMMAND

Code (hex)	Command to LCD Instruction Register
1	Clear display screen
2	Return home
4	Shift cursor to left
5	Shift display right
6	Shift cursor to 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

LCD PROGRAMMING

- 1. Main** program
- 2. Command** sub program
- 3. Data** sub program
- 4. Send byte** sub program
- 5. Delay** sub program

COMMAND SUB-PROGRAM

Steps for Sending Command:

- step1: Send the I/P command to LCD.
- step2: Select the Control Register by making RS low.
- step3: Select Write operation making RW low.
- step4: Send a High-to-Low pulse on Enable PIN with some delay_us.

COMMAND SUB-PROGRAM

```
/* Function to send the command to LCD */
void Lcd_CmdWrite(char cmd)
{
    sendByte(cmd);                                //Send the command
    LcdControlBusPort &= ~(1<<LCD_RS); // Send LOW pulse on RS pin for selecting Command register
    LcdControlBusPort &= ~(1<<LCD_RW); // Send LOW pulse on RW pin for Write operation
    LcdControlBusPort |= (1<<LCD_EN); // Generate a High-to-low pulse on EN pin
    delay(1000);
    LcdControlBusPort &= ~(1<<LCD_EN);

    delay(10000);
}
```

DATA SUB-PROGRAM

Steps for Sending Data:

- step1: Send the character to LCD.
- step2: Select the Data Register by making RS high.
- step3: Select Write operation making RW low.
- step4: Send a High-to-Low pulse on Enable PIN with some delay_us.

The timings are similar as above only change is that **RS** is made high for selecting Data register.

DATA SUB-PROGRAM

```
/* Function to send the data to LCD */
void Lcd_DataWrite(char dat)
{
    sendByte(dat);                      //Send the data
    LcdControlBusPort |= (1<<LCD_RS);   // Send HIGH pulse on RS pin for selecting data register
    LcdControlBusPort &= ~(1<<LCD_RW); // Send LOW pulse on RW pin for Write operation
    LcdControlBusPort |= (1<<LCD_EN);   // Generate a High-to-low pulse on EN pin
    delay(1000);
    LcdControlBusPort &= ~(1<<LCD_EN);

    delay(10000);
}
```

SEND BYTE SUB-PROGRAM

```
/* Function send a complete byte on the Data bus */

void sendByte(char byte)
{
    LcdDataBusPort&=~(LCD_dataBusMask);                                // Clear previous data
    LcdDataBusPort|= (((byte >>0x00) & 0x01) << LCD_D0);
    LcdDataBusPort|= (((byte >>0x01) & 0x01) << LCD_D1);
    LcdDataBusPort|= (((byte >>0x02) & 0x01) << LCD_D2);
    LcdDataBusPort|= (((byte >>0x03) & 0x01) << LCD_D3);
    LcdDataBusPort|= (((byte >>0x04) & 0x01) << LCD_D4);
    LcdDataBusPort|= (((byte >>0x05) & 0x01) << LCD_D5);
    LcdDataBusPort|= (((byte >>0x06) & 0x01) << LCD_D6);
    LcdDataBusPort|= (((byte >>0x07) & 0x01) << LCD_D7);
}
```

EXAMPLE - 1



Control pins connected to P1.0 to P1.2 (RS, R/W, En)

Data pins are connected to P2.0 to P2.7 (D0 to D7)

EXAMPLE - 1

```
#include "LPC17xx.h"
#include<stdio.h>
void delay(unsigned int x);
void LCD_CmdWrite(char cmd);
void LCD_DataWrite(char data);
void sendByte(char byte);
int main(void)
{
    // char i, a[]={“Hello World!”};
    SystemInit();                                // Clock and PLL configuration

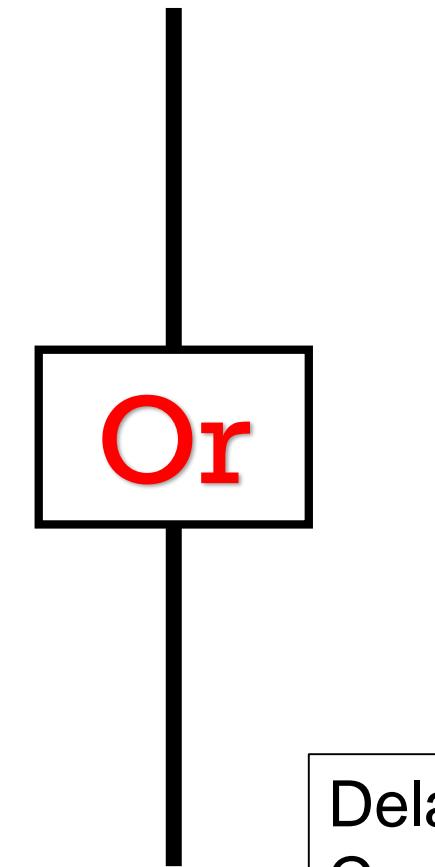
    LPC_GPIO2→FIODIR |= 0x000000FF;              // Configure all the LCD pins as output
    LPC_GPIO1→FIODIR |= 0x00000007;              // RS,R/W, Enable as output pins

    Lcd_CmdWrite(0x38);                          // enable 5x7 mode for chars
    Lcd_CmdWrite(0x01);                          // Clear Display
    Lcd_CmdWrite(0x0E);                          // Display ON, Cursor blinking
    Lcd_CmdWrite(0x80);                          // Move the cursor to beginning of first line
```

EXAMPLE - 1

```
Lcd_DataWrite('H');
Lcd_DataWrite('e');
Lcd_DataWrite('l');
Lcd_DataWrite('l');
Lcd_DataWrite('o');
Lcd_DataWrite(' ');
Lcd_DataWrite('w');
Lcd_DataWrite('o');
Lcd_DataWrite('r');
Lcd_DataWrite('l');
Lcd_DataWrite('d');

}
```



```
for(i=0;a[i]!=0;i++)
{
    Lcd_DataWrite(a[i]);
}
```

Delay Program
Commend program
Data program
Send byte program

EXAMPLE - 1

```
/* Function to send the command to LCD */  
  
void Lcd_CmdWrite(char cmd)  
{  
    sendByte(cmd);                      //Send the command  
  
    LPC_GPIO1→FIOPIN &= ~(1<<0);      // Send LOW pulse on RS pin for selecting Command register  
  
    LPC_GPIO1→FIOPIN &= ~(1<<1);      // Send LOW pulse on RW pin for Write operation  
  
    LPC_GPIO1→FIOPIN |= (1<<2);        // Generate a High-to-low pulse on EN pin  
  
    delay(1000);  
  
    LPC_GPIO1→FIOPIN &= ~(1<<2);  
  
    delay(10000);  
}
```

EXAMPLE - 1

```
/* Function to send the data to LCD */  
void Lcd_DataWrite(char dat)  
{  
    sendByte(dat);                      //Send the command  
    LPC_GPIO1→FIOPIN |= (1<<0);        // Send High pulse on RS pin for selecting data register  
    LPC_GPIO1→FIOPIN &= ~(1<<1);       // Send LOW pulse on RW pin for Write operation  
    LPC_GPIO1→FIOPIN |= (1<<2);        // Generate a High-to-low pulse on EN pin  
    delay(1000);  
    LPC_GPIO1→FIOPIN &= ~(1<<2);  
    delay(10000);  
}
```

EXAMPLE - 1

```
/* Function send a complete byte on the Data bus */  
void sendByte(char byte)  
{  
    LPC_GPIO2→FIOPIN &= ~(0xFF);                                // Clear previous data  
    LPC_GPIO2→FIOPIN |= (((byte >>0x00) & 0x01) << 0);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x01) & 0x01) << 1);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x02) & 0x01) << 2);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x03) & 0x01) << 3);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x04) & 0x01) << 4);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x05) & 0x01) << 5);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x06) & 0x01) << 6);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x07) & 0x01) << 7);  
}
```

EXAMPLE - 1

```
void delay(unsigned int ms)
{
    unsigned int i, j;
    for(i=0; i<ms; i++)
        for(j=0; j<20000; j++);
}
```

EXAMPLE - 2

Microcontroller
F1 D2 slots

Data pins are connected to P2.0 to P2.7

Control pin are connected to P1.0 to P1.2 (RS, R/W, Enable)

EXAMPLE - 2

```
#include "LPC17xx.h"
#include<stdio.h>
void delay(unsigned int x);
void LCD_CmdWrite(char cmd);
void LCD_DataWrite(char data);
void sendByte(char byte);
int main(void)
{
    // char i, a[]={“Microcontroller”}, b[]={“F1 D2 Slots”}; //for using for loop
    SystemInit();                                            // Clock and PLL configuration

    LPC_GPIO2→FIODIR |= 0x000000FF;                         // Configure all the LCD pins as output
    LPC_GPIO1→FIODIR |= 0x00000003;                         // RS,R/W, Enable as output pins

    Lcd_CmdWrite(0x38);                                     // enable 5x7 mode for chars
    Lcd_CmdWrite(0x01);                                     // Clear Display
    Lcd_CmdWrite(0x0E);                                     // Display ON, Cursor blinking
    Lcd_CmdWrite(0x80);                                     // Move the cursor to beginning of first line
```

EXAMPLE - 2

```
Lcd_DataWrite('M');
Lcd_DataWrite('I');
Lcd_DataWrite('C');
Lcd_DataWrite('R');
Lcd_DataWrite('O');
Lcd_DataWrite('C');
Lcd_DataWrite('O');
Lcd_DataWrite('N');
Lcd_DataWrite('T');
Lcd_DataWrite('R');
Lcd_DataWrite('O');
Lcd_DataWrite('L');
Lcd_DataWrite('L');
Lcd_DataWrite('E');
Lcd_DataWrite('R');
```

```
}
```

```
Lcd_CmdWrite(0xC0); // Move the cursor to beginning of second line

Lcd_DataWrite('F');
Lcd_DataWrite('1');
Lcd_DataWrite(' ');
Lcd_DataWrite('D');
Lcd_DataWrite('2');
Lcd_DataWrite(' ');
Lcd_DataWrite('S');
Lcd_DataWrite('L');
Lcd_DataWrite('O');
Lcd_DataWrite('T');
```

EXAMPLE - 2 (Alternative Approach)

```
Lcd_CmdWrite(0x80); // Move the cursor to beginning of first line  
for(i=0;a[i]!=0;i++)  
{  
    Lcd_DataWrite(a[i]);  
}  
Lcd_CmdWrite(0xC0); // Move the cursor to beginning of second line  
for(i=0;b[i]!=0;i++)  
{  
    Lcd_DataWrite(b[i]);  
}  
}
```

EXAMPLE - 2

```
/* Function to send the command to LCD */  
  
void Lcd_CmdWrite(char cmd)  
{  
    sendByte(cmd);                      //Send the command  
  
    LPC_GPIO1→FIOPIN &= ~(1<<0);      // Send LOW pulse on RS pin for selecting Command register  
  
    LPC_GPIO1→FIOPIN &= ~(1<<1);      // Send LOW pulse on RW pin for Write operation  
  
    LPC_GPIO1→FIOPIN |= (1<<2);        // Generate a High-to-low pulse on EN pin  
  
    delay(1000);  
  
    LPC_GPIO1→FIOPIN &= ~(1<<2);  
  
    delay(10000);  
}
```

EXAMPLE - 2

```
/* Function to send the data to LCD */  
  
void Lcd_DataWrite(char cmd)  
{  
    sendByte(cmd);                                //Send the command  
  
    LPC_GPIO1→FIOPIN |= (1<<0);                // Send High pulse on RS pin for selecting data register  
  
    LPC_GPIO1→FIOPIN &= ~(1<<1);               // Send LOW pulse on RW pin for Write operation  
  
    LPC_GPIO1→FIOPIN |= (1<<2);                // Generate a High-to-low pulse on EN pin  
  
    delay(1000);  
  
    LPC_GPIO1→FIOPIN &= ~(1<<2);  
  
    delay(10000);  
}
```

EXAMPLE - 2

```
/* Function send a complete byte on the Data bus */  
void sendByte(char byte)  
{  
    LPC_GPIO2→FIOPIN &= ~(0xFF);                                // Clear previous data  
    LPC_GPIO2→FIOPIN |= (((byte >>0x00) & 0x01) << 0);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x01) & 0x01) << 1);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x02) & 0x01) << 2);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x03) & 0x01) << 3);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x04) & 0x01) << 4);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x05) & 0x01) << 5);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x06) & 0x01) << 6);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x07) & 0x01) << 7);  
}
```

EXAMPLE - 2

```
void delay(unsigned int ms)
{
    unsigned int i, j;
    for(i=0; i<ms; i++)
        for(j=0; j<20000; j++);
}
```

PRACTICE QUESTION

1. Display your first name in 1st line of LCD and last name in 2nd line of LCD. Data pins are connected to P3.0 to P3.7 and control pins are connected to P4.26 to P4.28 (RS, R/W, Enable)
2. Data pin are connected to P2.0 to P2.7 and Control pin are connected to P1.0 to P1.2 (RS, R/W, Enable). Write a program to display the following



EXAMPLE - 2

Microcontroller
F1 D2 slots

Data pins are connected to P2.0 to P2.7

Control pin are connected to P1.0 to P1.2 (RS, R/W, Enable)

EXAMPLE - 2

```
#include "LPC17xx.h"
#include<stdio.h>
void delay(unsigned int x);
void Lcd_CmdWrite(char cmd);
void Lcd_DataWrite(char data);
void sendByte(char byte);
int main(void)
{
    // char i, a[]={“Microcontroller”}, b[]={“F1 D2 Slots”}; //for using for loop
    SystemInit();                                            // Clock and PLL configuration

    LPC_GPIO2→FIODIR |= 0x000000FF;                         // Configure all the LCD pins as output
    LPC_GPIO1→FIODIR |= 0x00000007;                         // RS,R/W, Enable as output pins

    Lcd_CmdWrite(0x38);                                     // enable 5x7 mode for chars
    Lcd_CmdWrite(0x01);                                     // Clear Display
    Lcd_CmdWrite(0x0E);                                     // Display ON, Cursor blinking
    Lcd_CmdWrite(0x80);                                     // Move the cursor to beginning of first line
```

EXAMPLE - 2

```
Lcd_DataWrite('M');
Lcd_DataWrite('I');
Lcd_DataWrite('C');
Lcd_DataWrite('R');
Lcd_DataWrite('O');
Lcd_DataWrite('C');
Lcd_DataWrite('O');
Lcd_DataWrite('N');
Lcd_DataWrite('T');
Lcd_DataWrite('R');
Lcd_DataWrite('O');
Lcd_DataWrite('L');
Lcd_DataWrite('L');
Lcd_DataWrite('E');
Lcd_DataWrite('R');
```

```
}
```

```
Lcd_CmdWrite(0xC0); // Move the cursor to beginning of second line

Lcd_DataWrite('F');
Lcd_DataWrite('1');
Lcd_DataWrite(' ');
Lcd_DataWrite('D');
Lcd_DataWrite('2');
Lcd_DataWrite(' ');
Lcd_DataWrite('S');
Lcd_DataWrite('L');
Lcd_DataWrite('O');
Lcd_DataWrite('T');
Lcd_DataWrite('S');
```

EXAMPLE - 2 (Alternative Approach)

```
Lcd_CmdWrite(0x80); // Move the cursor to beginning of first line  
for(i=0;a[i]!=0;i++)  
{  
    Lcd_DataWrite(a[i]);  
}  
Lcd_CmdWrite(0xC0); // Move the cursor to beginning of second line  
for(i=0;b[i]!=0;i++)  
{  
    Lcd_DataWrite(b[i]);  
}  
}
```

EXAMPLE - 2

```
/* Function to send the command to LCD */  
  
void Lcd_CmdWrite(char cmd)  
{  
    sendByte(cmd);                      //Send the command  
  
    LPC_GPIO1→FIOPIN &= ~(1<<0);      // Send LOW pulse on RS pin for selecting Command register  
  
    LPC_GPIO1→FIOPIN &= ~(1<<1);      // Send LOW pulse on RW pin for Write operation  
  
    LPC_GPIO1→FIOPIN |= (1<<2);        // Generate a High-to-low pulse on EN pin  
  
    delay(1000);  
  
    LPC_GPIO1→FIOPIN &= ~(1<<2);  
  
    delay(10000);  
}
```

EXAMPLE - 2

```
/* Function to send the data to LCD */  
  
void Lcd_DataWrite(char cmd)  
{  
    sendByte(cmd);                                //Send the command  
  
    LPC_GPIO1→FIOPIN |= (1<<0);                // Send High pulse on RS pin for selecting data register  
  
    LPC_GPIO1→FIOPIN &= ~(1<<1);              // Send LOW pulse on RW pin for Write operation  
  
    LPC_GPIO1→FIOPIN |= (1<<2);                // Generate a High-to-low pulse on EN pin  
  
    delay(1000);  
  
    LPC_GPIO1→FIOPIN &= ~(1<<2);  
  
    delay(10000);  
}
```

EXAMPLE - 2

```
/* Function send a complete byte on the Data bus */  
void sendByte(char byte)  
{  
    LPC_GPIO2→FIOPIN &= ~(0xFF);                                // Clear previous data  
    LPC_GPIO2→FIOPIN |= (((byte >>0x00) & 0x01) << 0);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x01) & 0x01) << 1);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x02) & 0x01) << 2);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x03) & 0x01) << 3);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x04) & 0x01) << 4);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x05) & 0x01) << 5);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x06) & 0x01) << 6);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x07) & 0x01) << 7);  
}
```

EXAMPLE - 2

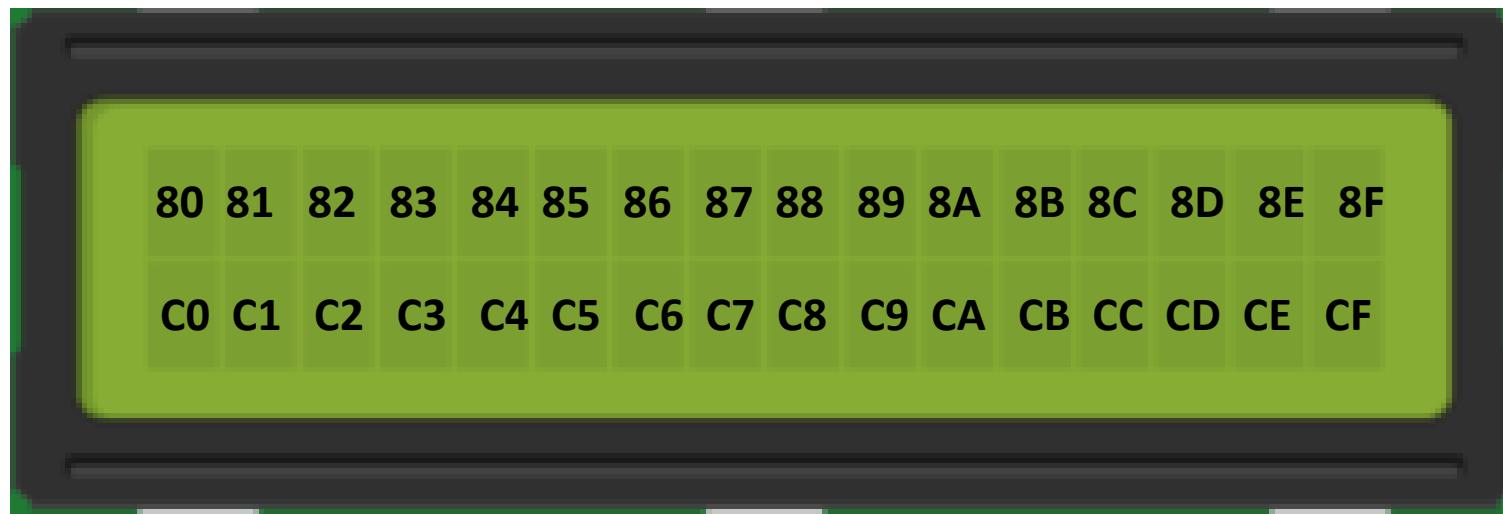
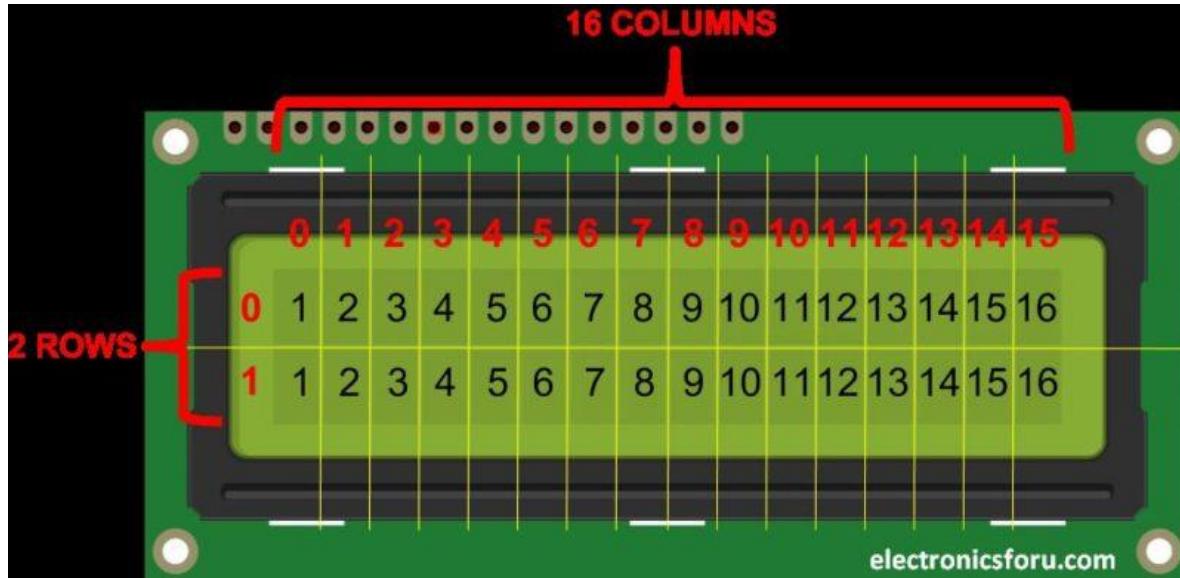
```
void delay(unsigned int ms)
{
    unsigned int i, j;
    for(i=0; i<ms; i++)
        for(j=0; j<20000; j++);
}
```

EXAMPLE - 3

Data pin are connected to P2.0 to P2.7 and Control pin are connected to P1.0 to P1.2 (RS, R/W, Enable). Write a program to display the following



EXAMPLE - 3



EXAMPLE - 3

```
#include "LPC17xx.h"
#include<stdio.h>
void delay(unsigned int x);
void Lcd_CmdWrite(char cmd);
void Lcd_DataWrite(char data);
void sendByte(char byte);
int main(void)
{
    SystemInit();                                // Clock and PLL configuration
    LPC_GPIO2→FIODIR |= 0x000000FF;              // Configure all the LCD pins as output
    LPC_GPIO1→FIODIR |= 0x00000007;              // RS,R/W, Enable as output pins
    Lcd_CmdWrite(0x38);                          // enable 5x7 mode for chars
    Lcd_CmdWrite(0x01);                          // Clear Display
    Lcd_CmdWrite(0x0E);                          // Display ON, Cursor blinking
```

EXAMPLE - 3

```
Lcd_CmdWrite(0x80);          // Move the cursor to beginning of first line
Lcd_DataWrite('F');

Lcd_CmdWrite(0x8E);
Lcd_DataWrite('F');
Lcd_DataWrite('L');

Lcd_CmdWrite(0xC0);          // Move the cursor to beginning of 2nd line
Lcd_DataWrite('S');

Lcd_CmdWrite(0xCE);
Lcd_DataWrite('S');
Lcd_DataWrite('L');

}
```

EXAMPLE - 3

```
/* Function to send the command to LCD */  
  
void Lcd_CmdWrite(char cmd)  
{  
    sendByte(cmd);                      //Send the command  
  
    LPC_GPIO1→FIOPIN &= ~(1<<0);      // Send LOW pulse on RS pin for selecting Command register  
  
    LPC_GPIO1→FIOPIN &= ~(1<<1);      // Send LOW pulse on RW pin for Write operation  
  
    LPC_GPIO1→FIOPIN |= (1<<2);        // Generate a High-to-low pulse on EN pin  
  
    delay(1000);  
  
    LPC_GPIO1→FIOPIN &= ~(1<<2);  
  
    delay(10000);  
}
```

EXAMPLE - 3

```
/* Function to send the data to LCD */  
  
void Lcd_DataWrite(char cmd)  
{  
    sendByte(cmd);                                //Send the command  
  
    LPC_GPIO1→FIOPIN |= (1<<0);                // Send High pulse on RS pin for selecting data register  
  
    LPC_GPIO1→FIOPIN &= ~(1<<1);              // Send LOW pulse on RW pin for Write operation  
  
    LPC_GPIO1→FIOPIN |= (1<<2);                // Generate a High-to-low pulse on EN pin  
  
    delay(1000);  
  
    LPC_GPIO1→FIOPIN &= ~(1<<2);  
  
    delay(10000);  
}
```

EXAMPLE - 3

```
/* Function send a complete byte on the Data bus */  
void sendByte(char byte)  
{  
    LPC_GPIO2→FIOPIN &= ~(0xFF);                                // Clear previous data  
    LPC_GPIO2→FIOPIN |= (((byte >>0x00) & 0x01) << 0);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x01) & 0x01) << 1);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x02) & 0x01) << 2);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x03) & 0x01) << 3);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x04) & 0x01) << 4);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x05) & 0x01) << 5);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x06) & 0x01) << 6);  
    LPC_GPIO2→FIOPIN |= (((byte >>0x07) & 0x01) << 7);  
}
```

EXAMPLE - 3

```
void delay(unsigned int ms)
{
    unsigned int i, j;
    for(i=0; i<ms; i++)
        for(j=0; j<20000; j++);
}
```

PRACTICE QUESTION

1. Display your first name in 1st line of LCD and last name in 2nd line of LCD. Data pins are connected to P3.0 to P3.7 and control pins are connected to P4.26 to P4.28 (RS, R/W, Enable)
2. Data pins are connected to P3.0 to P3.7 and control pins are connected to P4.26 to P4.28 (RS, R/W, Enable). Display the following

First
Second