LAB NO : 7 DATE : 19/02/2025

Title : PROGRAMS ON MULTIPLEXED SEVEN SEGMENT DISPLAY

# Solved Exercise 1: Write C program to display message on LCD.

## Code :

#include <lpc17xx.h>

void lcd\_init(void);

void write(int, int);

void delay\_lcd(unsigned int);

void lcd\_comdata(int, int);

void clear\_ports(void);

void lcd\_puts(unsigned char \*);

int main(void)

{

unsigned char Msg1[4] = {"MIT"};

unsigned char Msg2[19] = {"Department of CSE:"};

SystemInit();

SystemCoreClockUpdate();

lcd\_init();

lcd\_comdata(0x80, 0);

delay\_lcd(800);

lcd\_puts(&Msg1[0]);

lcd\_comdata(0xC0, 0);

delay\_lcd(800);

lcd\_puts(&Msg2[0]);

}

// LCD initialization

void lcd\_init()

{

/\* Ports initialized as GPIO \*/

LPC\_PINCON->PINSEL1 &= 0xFC003FFF; // P0.23 to P0.28

/\* Setting the directions as output \*/

LPC\_GPIO0->FIODIR |= 0x0F<<23 | 1<<27 | 1<<28;

clear\_ports();

delay\_lcd(3200);

lcd\_comdata(0x33, 0);

delay\_lcd(30000);

lcd\_comdata(0x32, 0);

delay\_lcd(30000);

lcd\_comdata(0x28, 0); // Function set

delay\_lcd(30000);

lcd\_comdata(0x0c, 0); // Display on, cursor off

delay\_lcd(800);

lcd\_comdata(0x06, 0); // Entry mode set, increment cursor right

delay\_lcd(800);

lcd\_comdata(0x01, 0); // Display clear

delay\_lcd(10000);

return;

}

void lcd\_comdata(int temp1, int type)

{

int temp2 = temp1 & 0xf0; // Move data (26-8+1) times : 26 - HN place, 4 - Bits

temp2 = temp2 << 19; // Data lines from 23 to 26

write(temp2, type);

temp2 = temp1 & 0x0f; // 26-4+1

temp2 = temp2 << 23;

write(temp2, type);

delay\_lcd(1000);

return;

}

void write(int temp2, int type) // Write to command/data register

{

clear\_ports();

LPC\_GPIO0->FIOPIN = temp2; // Assign the value to the data lines

if(type == 0)

LPC\_GPIO0->FIOCLR = 1 << 27; // Clear bit RS for Command

else

LPC\_GPIO0->FIOSET = 1 << 27; // Set bit RS for Data

LPC\_GPIO0->FIOSET = 1 << 28; // EN = 1

delay\_lcd(25);

LPC\_GPIO0->FIOCLR = 1 << 28; // EN = 0

return;

}

void delay\_lcd(unsigned int r1)

{

unsigned int r;

for(r = 0; r < r1; r++);

return;

}

void clear\_ports(void)

{

/\* Clearing the lines at power on \*/

LPC\_GPIO0->FIOCLR = 0x0F << 23; // Clearing data lines

LPC\_GPIO0->FIOCLR = 1 << 27; // Clearing RS line

LPC\_GPIO0->FIOCLR = 1 << 28; // Clearing Enable line

return;

}

void lcd\_puts(unsigned char \*buf1)

{

unsigned int i = 0;

unsigned int temp3;

while(buf1[i] != '\0')

{

temp3 = buf1[i];

lcd\_comdata(temp3, 1);

i++;

if(i == 16)

{

lcd\_comdata(0xc0, 0); // Move to the next line on LCD

}

}

return;

}