```c
#include <lpc214x.h>
#define PLOCK 0x00000400

// LED and LCD control macros
#define LED_OFF (IO0SET = 1U << 31)
#define LED_ON (IO0CLR = 1U << 31)
#define RS_ON (IO0SET = 1U << 20)
#define RS_OFF (IO0CLR = 1U << 20)
#define EN_ON (IO1SET = 1U << 25)
#define EN_OFF (IO1CLR = 1U << 25)

void SystemInit(void);
static void delay_ms(unsigned int j); // Millisecond delay
static void delay_us(unsigned int count); // Microsecond delay
static void LCD_SendCmdSignals(void);
static void LCD_SendDataSignals(void);
static void LCD_SendHigherNibble(unsigned char dataByte);
static void LCD_CmdWrite(unsigned char cmdByte);
static void LCD_DataWrite(unsigned char dataByte);
static void LCD_Reset(void);
static void LCD_Init(void);
void LCD_DisplayString(const char *ptr_string);

int main() {
    SystemInit();
    IO0DIR |= 1U << 31 | 0x00FF0000; // Set P0.16 to P0.23 as output
    IO1DIR |= 1U << 25; // Set P1.25 as output (EN)

    // Blink LED for testing
    LED_ON; delay_ms(500);
    LED_OFF; delay_ms(500);

    // Initialize LCD
    LCD_Reset();
    LCD_Init();
    delay_ms(100);

    // Display messages on the LCD
    LCD_CmdWrite(0x80); // Move to the first line
    LCD_DisplayString("RV College Of Engrng");
    LCD_CmdWrite(0xC0); // Move to the second line
    LCD_DisplayString("Computer Science");
    LCD_CmdWrite(0x94); // Move to the third line
    LCD_DisplayString("4th Semester");
    LCD_CmdWrite(0xD4); // Move to the fourth line
    LCD_DisplayString("B Section");

    while (1);
}

static void LCD_CmdWrite(unsigned char cmdByte) {
    LCD_SendHigherNibble(cmdByte);
    LCD_SendCmdSignals();
    cmdByte = cmdByte << 4; // Shift to lower nibble
    LCD_SendHigherNibble(cmdByte);
    LCD_SendCmdSignals();
}

static void LCD_DataWrite(unsigned char dataByte) {
    LCD_SendHigherNibble(dataByte);
    LCD_SendDataSignals();
    dataByte = dataByte << 4; // Shift to lower nibble
    LCD_SendHigherNibble(dataByte);
    LCD_SendDataSignals();
}

static void LCD_Reset(void) {
    /* LCD reset sequence for 4-bit mode */
    LCD_SendHigherNibble(0x30);
    LCD_SendCmdSignals();
    delay_ms(100);
    LCD_SendHigherNibble(0x30);
```

```
    LCD_SendCmdSignals();
    delay_us(200);
    LCD_SendHigherNibble(0x30);
    LCD_SendCmdSignals();
    delay_us(200);
    LCD_SendHigherNibble(0x20); // Set to 4-bit mode
    LCD_SendCmdSignals();
    delay_us(200);
}

static void LCD_SendHigherNibble(unsigned char dataByte) {
    // Send the D7,6,5,D4 (upper nibble) to P0.16 to P0.19
    IO0CLR = 0x000F0000; // Clear bits
    IO0SET = ((dataByte >> 4) & 0x0F) << 16; // Send upper nibble
}

static void LCD_SendCmdSignals(void) {
    RS_OFF; // Command mode
    EN_ON; delay_us(100); EN_OFF; // Enable pulse
}

static void LCD_SendDataSignals(void) {
    RS_ON; // Data mode
    EN_ON; delay_us(100); EN_OFF; // Enable pulse
}

static void LCD_Init(void) {
    delay_ms(100);
    LCD_Reset();
    LCD_CmdWrite(0x28); // Initialize LCD for 4-bit, 5x7 matrix display
    LCD_CmdWrite(0x0E); // Display ON, cursor ON
    LCD_CmdWrite(0x01); // Clear display
    LCD_CmdWrite(0x80); // Go to first line, first position
}

void LCD_DisplayString(const char *ptr_string) {
    // Loop through the string and display character by character
    while ((*ptr_string) != 0) {
        LCD_DataWrite(*ptr_string++);
    }
}

static void delay_us(unsigned int count) {
    unsigned int j = 0, i = 0;
    for (j = 0; j < count; j++) {
        for (i = 0; i < 10; i++);
    }
}

void SystemInit(void) {
    PLL0CON = 0x01;
    PLL0CFG = 0x24;
    PLL0FEED = 0xAA;
    PLL0FEED = 0x55;
    while (!(PLL0STAT & PLOCK)) { ; }
    PLL0CON = 0x03;
    PLL0FEED = 0xAA;
    PLL0FEED = 0x55;
    VPBDIV = 0x01; // PCLK is the same as CCLK, i.e., 60 MHz
}

void delay_ms(unsigned int j) {
    unsigned int x, i;
    for (i = 0; i < j; i++) {
        for (x = 0; x < 10000; x++);
    }
}
```