

```

1  #include <lpc214x.h>
2  #define PLOCK 0x00000400
3
4  // LED and LCD control macros
5  #define LED_OFF (IO0SET = 1U << 31)
6  #define LED_ON (IO0CLR = 1U << 31)
7  #define RS_ON (IO0SET = 1U << 20)
8  #define RS_OFF (IO0CLR = 1U << 20)
9  #define EN_ON (IO1SET = 1U << 25)
10 #define EN_OFF (IO1CLR = 1U << 25)
11
12 void SystemInit(void);
13 static void delay_ms(unsigned int j); // Millisecond delay
14 static void delay_us(unsigned int count); // Microsecond delay
15 static void LCD_SendCmdSignals(void);
16 static void LCD_SendDataSignals(void);
17 static void LCD_SendHigherNibble(unsigned char dataByte);
18 static void LCD_CmdWrite(unsigned char cmdByte);
19 static void LCD_DataWrite(unsigned char dataByte);
20 static void LCD_Reset(void);
21 static void LCD_Init(void);
22 void LCD_DisplayString(const char *ptr_string);
23
24 int main() {
25     SystemInit();
26     IO0DIR |= 1U << 31 | 0x00FF0000; // Set P0.16 to P0.23 as output
27     IO1DIR |= 1U << 25; // Set P1.25 as output (EN)
28
29     // Blink LED for testing
30     LED_ON; delay_ms(500);
31     LED_OFF; delay_ms(500);
32
33     // Initialize LCD
34     LCD_Reset();
35     LCD_Init();
36     delay_ms(100);
37
38     // Display messages on the LCD
39     LCD_CmdWrite(0x80); // Move to the first line
40     LCD_DisplayString("RV College Of Engrng");
41     LCD_CmdWrite(0xC0); // Move to the second line
42     LCD_DisplayString("Computer Science");
43     LCD_CmdWrite(0x94); // Move to the third line
44     LCD_DisplayString("4th Semester");
45     LCD_CmdWrite(0xD4); // Move to the fourth line
46     LCD_DisplayString("B Section");
47
48     while (1);
49 }
50
51 static void LCD_CmdWrite(unsigned char cmdByte) {
52     LCD_SendHigherNibble(cmdByte);
53     LCD_SendCmdSignals();
54     cmdByte = cmdByte << 4; // Shift to lower nibble
55     LCD_SendHigherNibble(cmdByte);
56     LCD_SendCmdSignals();
57 }
58
59 static void LCD_DataWrite(unsigned char dataByte) {
60     LCD_SendHigherNibble(dataByte);
61     LCD_SendDataSignals();
62     dataByte = dataByte << 4; // Shift to lower nibble
63     LCD_SendHigherNibble(dataByte);
64     LCD_SendDataSignals();
65 }
66
67 static void LCD_Reset(void) {
68     /* LCD reset sequence for 4-bit mode */
69     LCD_SendHigherNibble(0x30);
70     LCD_SendCmdSignals();
71     delay_ms(100);
72     LCD_SendHigherNibble(0x30);

```

```
73     LCD_SendCmdSignals();
74     delay_us(200);
75     LCD_SendHigherNibble(0x30);
76     LCD_SendCmdSignals();
77     delay_us(200);
78     LCD_SendHigherNibble(0x20); // Set to 4-bit mode
79     LCD_SendCmdSignals();
80     delay_us(200);
81 }
82
83 static void LCD_SendHigherNibble(unsigned char dataByte) {
84     // Send the D7,6,5,D4 (upper nibble) to P0.16 to P0.19
85     IOOCLR = 0x000F0000; // Clear bits
86     IOOSET = ((dataByte >> 4) & 0x0F) << 16; // Send upper nibble
87 }
88
89 static void LCD_SendCmdSignals(void) {
90     RS_OFF; // Command mode
91     EN_ON; delay_us(100); EN_OFF; // Enable pulse
92 }
93
94 static void LCD_SendDataSignals(void) {
95     RS_ON; // Data mode
96     EN_ON; delay_us(100); EN_OFF; // Enable pulse
97 }
98
99 static void LCD_Init(void) {
100     delay_ms(100);
101     LCD_Reset();
102     LCD_CmdWrite(0x28); // Initialize LCD for 4-bit, 5x7 matrix display
103     LCD_CmdWrite(0x0E); // Display ON, cursor ON
104     LCD_CmdWrite(0x01); // Clear display
105     LCD_CmdWrite(0x80); // Go to first line, first position
106 }
107
108 void LCD_DisplayString(const char *ptr_string) {
109     // Loop through the string and display character by character
110     while ((*ptr_string) != 0) {
111         LCD_DataWrite(*ptr_string++);
112     }
113 }
114
115 static void delay_us(unsigned int count) {
116     unsigned int j = 0, i = 0;
117     for (j = 0; j < count; j++) {
118         for (i = 0; i < 10; i++);
119     }
120 }
121
122 void SystemInit(void) {
123     PLL0CON = 0x01;
124     PLL0CFG = 0x24;
125     PLL0FEED = 0xAA;
126     PLL0FEED = 0x55;
127     while (!(PLLOSTAT & PLOCK)) { ; }
128     PLL0CON = 0x03;
129     PLL0FEED = 0xAA;
130     PLL0FEED = 0x55;
131     VPBDIV = 0x01; // PCLK is the same as CCLK, i.e., 60 MHz
132 }
133
134 void delay_ms(unsigned int j) {
135     unsigned int x, i;
136     for (i = 0; i < j; i++) {
137         for (x = 0; x < 10000; x++);
138     }
139 }
140
```