

Class 4 Class Program 2: SET SPD: rpm

```
22 #define _XTAL_FREQ 20000000 // Define the crystal oscillator frequency as 20MHz (for delay functions)
23 void Lcdinit(void);          // Function prototype for LCD initialization
24 void LcdCommand(uint8_t i);  // Function prototype for sending commands to the LCD
25 void LcdData(uint8_t i);     // Function prototype for sending data to the LCD
26 void LcdOutput(uint16_t i);
27 void keyScan(void);
28
29 uint8_t Array[15] = {"SET SPD:   rpm"};
30 uint8_t x, m, n, value;
31 uint16_t j = 0;
32
33 void main(void){
34     Lcdinit(); // Initialize the LCD
35
36     while(1)
37     {
38         keyScan();
39         __delay_ms(100);
40     }
41 }
```

```
43 // Function to initialize the LCD
44 void Lcdinit(void)
45 {
46     TRISC = 0x00; // Set PORTC as output (for control signals)
47     TRISD = 0x00; // Set PORTD as output (for data signals)
48     TRISB = 0xF0; // Set R4 to R7 as Input
49
50     OPTION_REG &= ~(0x1U << 7); // Enable Pull Up
51     __delay_ms(100); // Wait for LCD to stabilize
52
53     // LCD initialization sequence as per HD44780 LCD datasheet
54     LcdCommand(0x30); // Send function set command (8-bit mode)
55     __delay_ms(100);
56     LcdCommand(0x30); // Repeat function set command
57     __delay_ms(100);
58     LcdCommand(0x30); // Repeat function set command again
59     __delay_ms(100);
60     LcdCommand(0x38); // Set LCD for 8-bit mode, 2-line display, 5x8 font
61     __delay_ms(100);
62     LcdCommand(0x0C); // Turn on display, cursor off
63     __delay_ms(100);
64     LcdCommand(0x01); // Clear the display
65     __delay_ms(100);
66     LcdCommand(0x06);
67     __delay_ms(100);
68 }
```

```

70 void keyScan(void)
71 {
72     value = PORTB & 0xF0; /* Mask lower 4 bits of PORTB, keeping only the upper 4 bits */
73
74     switch(value)
75     {
76         case 0xE0: /* Case when PORTB = 1110 0000 */
77             LcdCommand(0x80); /* Move cursor to the beginning of the first line on LCD */
78
79             for(int i = 0; i < 15; i++) /* Loop to print the first 15 characters from Array */
80             {
81                 LcdData(Array[i]); /* Display character from Array on LCD */
82             }
83
84             for(int i = 0; i < 4; i++) /* Loop to display four '0' characters on LCD */
85             {
86                 LcdCommand((uint8_t)0x88 + (uint8_t)i); /* Move cursor to position (0x88 + i) */
87                 LcdData(0x30 + 0); /* Display ASCII '0' on LCD */
88             }
89             break;
90
91         case 0xD0: /* Case when PORTB = 1101 0000 */
92             if (j < 5000) /* Ensure that j does not exceed 5000 */
93                 j++; /* Increment j */
94
95             LcdCommand(0x88); /* Move cursor to position 0x88 */
96             LcdOutput(j); /* Display the updated value of j on LCD */
97             break;
98
99         case 0xB0: /* Case when PORTB = 1011 0000 */
100             if (j > 1) /* Ensure that j does not go below 1 */
101                 j--; /* Decrement j */
102
103             LcdCommand(0x88); /* Move cursor to position 0x88 */
104             LcdOutput(j); /* Display the updated value of j on LCD */
105             break;
106
107         case 0x70: /* Case when PORTB = 0111 0000 */
108             LcdCommand(0x88); /* Move cursor to position 0x88 */
109             LcdOutput(0); /* Display '0' on LCD */
110             break;
111
112         default:
113             /* Handle unexpected values (if needed) */
114             break;
115     }
116 }
117
118 void LcdOutput(uint16_t i)
119 {
120     uint8_t d1, d2, d3, d4; /* Creating local var to reduce memory consumption */
121     d4 = (uint8_t)(i / 1000); /* Extract thousands place */
122     d3 = (uint8_t)((i % 1000) / 100); /* Extract hundreds place */
123     d2 = (uint8_t)((i % 100) / 10); /* Extract tens place */
124     d1 = (uint8_t)(i % 10); /* Extract ones place */
125
126     LcdCommand(0x88);
127     LcdData(0x30 + d4);
128     LcdData(0x30 + d3);
129     LcdData(0x30 + d2);
130     LcdData(0x30 + d1);
131 }
132

```

```

133 // Function to send data (characters) to the LCD
134 void LcdData(uint8_t i)
135 {
136     PORTC |= (0x1 << 3); // Set RS (RC3) = 1 (indicates data mode)
137     PORTD = i;           // Place data on PORTD
138     PORTC |= (0x1 << 0); // Set EN (RC0) = 1 (enable pulse start)
139     __delay_ms(100);     // Small delay for command execution
140     PORTC &= ~(0x1 << 0); // Set EN (RC0) = 0 (enable pulse end)
141 }
142
143 // Function to send commands to the LCD
144 void LcdCommand(uint8_t i)
145 {
146     PORTC &= ~(0x1 << 3); // Set RS (RC3) = 0 (indicates command mode)
147     PORTD = i;           // Place command on PORTD
148     PORTC |= (0x1 << 0); // Set EN (RC0) = 1 (enable pulse start)
149     __delay_ms(100);     // Small delay for command execution
150     PORTC &= ~(0x1 << 0); // Set EN (RC0) = 0 (enable pulse end)
151 }

```