

```
1: // LCD Pin Configuration
2: sbit LCD_RS at RB5_bit;
3: sbit LCD_En at RB6_bit;
4: sbit LCD_D4 at RB0_bit;
5: sbit LCD_D5 at RB1_bit;
6: sbit LCD_D6 at RB2_bit;
7: sbit LCD_D7 at RB3_bit;
8:
9: sbit LCD_RS_Direction at TRISB2_bit;
10: sbit LCD_EN_Direction at TRISB6_bit;
11: sbit LCD_D4_Direction at TRISB0_bit;
12: sbit LCD_D5_Direction at TRISB1_bit;
13: sbit LCD_D6_Direction at TRISB2_bit;
14: sbit LCD_D7_Direction at TRISB3_bit;
15:
16: // Timer0 setup for 10ms timing
17: unsigned cnt;
18:
19: void InitTimer0() {
20:     OPTION_REG = 0x86;    // Prescaler 1:128
21:     TMR0 = 100;           // Timer preload value
22:     INTCON = 0xA0;        // Enable timer0 interrupt
23: }
24:
25: void Interrupt() {
26:     if (TMR0IF_bit) {
27:         cnt++;
28:         TMR0IF_bit = 0;
29:         TMR0 = 100;
30:     }
31: }
32:
33: // Global variables
34: unsigned short kp, dl;
35: char keypadPort at PORTD;
36: int t, a, b, q, x;
37:
38: // Check for mount/dismount input anytime
39: void CheckMountDismount() {
40:     if (PORTC.B5 == 1 && PORTC.B7 == 0) {
41:         PORTC.B3 = 1;
42:         PORTC.B4 = 0;
43:         PWM1_Start();
44:         PWM1_Set_Duty(180);
45:         Delay_ms(500);
46:         PORTC.B3 = 0;
47:         PWM1_Stop();
48:     }
49:
50:     if (PORTC.B7 == 1 && PORTC.B5 == 0) {
51:         PORTC.B3 = 0;
52:         PORTC.B4 = 1;
53:         PWM1_Start();
54:         PWM1_Set_Duty(180);
55:         Delay_ms(500);
56:         PORTC.B4 = 0;
57:         PWM1_Stop();
58:     }
59: }
60:
61: void main() {
62:     // TRIS Setup
```

```
63:  TRISA0_bit = 1; // Unused
64:  TRISA1_bit = 1; // Unused
65:  TRISA2_bit = 0; // Red LED
66:  TRISA3_bit = 0; // Green LED
67:  TRISA4_bit = 1;
68:  TRISA5_bit = 1;
69:
70:  TRISB = 0x00; // All outputs for LCD
71:  TRISC0_bit = 0; // Lock motor
72:  TRISC1_bit = 0; // Unlock motor
73:  TRISC2_bit = 0; // PWM shared
74:  TRISC3_bit = 0; // Mount right
75:  TRISC4_bit = 0; // Mount left
76:  TRISC5_bit = 1; // Mount switch
77:  TRISC6_bit = 0; // Buzzer
78:  TRISC7_bit = 1; // Dismount switch
79:
80:  TRISD0_bit = 1; // Keypad
81:  TRISD1_bit = 1;
82:  TRISD2_bit = 1;
83:  TRISD3_bit = 0;
84:  TRISD4_bit = 1;
85:  TRISD5_bit = 1;
86:  TRISD6_bit = 1;
87:  TRISD7_bit = 1;
88:
89:  PORTC = 0;
90:  PORTA = 0;
91:
92:  lcd_init();
93:  keypad_init();
94:  PWM1_init(1000);
95:
96:  Delay_ms(600);
97:  lcd_cmd(_LCD_CURSOR_OFF);
98:  lcd_cmd(_LCD_CLEAR);
99:  lcd_out(1,6,"START");
100:  lcd_out(2,3,"MISSU Locking");
101:  Delay_ms(2000);
102:
103:  PWM1_Start();
104:  PORTC.B0 = 1;
105:  PWM1_Set_Duty(120);
106:  Delay_ms(500);
107:  PORTC.B0 = 0;
108:  PWM1_Stop();
109:
110:  lcd_cmd(_LCD_CLEAR);
111:  lcd_out(2,6,"LOCKED");
112:  PORTA.B2 = 1;
113:  PORTC.B6 = 1;
114:  Delay_ms(1000);
115:  PORTA.B2 = 0;
116:  PORTC.B6 = 0;
117:
118:  while (1) {
119: point1:
120:     t = a = b = q = x = 0;
121:     ADCON1 = 0x0F;
122:     lcd_cmd(_LCD_CLEAR);
123:     lcd_out(1,4,"Please Input Key");
124:     lcd_out(2,3,"");
```

```
125:
126:     cnt = 0;
127:     InitTimer0();
128:
129:     while (1) {
130:         CheckMountDismount(); // Always check switches
131:
132:         do {
133:             CheckMountDismount();
134:             if (cnt >= 1500) {
135:                 lcd_cmd(_LCD_CLEAR);
136:                 goto point1;
137:             }
138:             kp = Keypad_Key_Click();
139:         } while (!kp);
140:
141:         switch (kp) {
142:             case 1: kp = 51; q += b * 3; break;
143:             case 2: kp = 50; b += a + 2; break;
144:             case 3: kp = 49; a += 1; break;
145:             case 5: kp = 54; break;
146:             case 7: kp = 52; x += q - 4; break;
147:             case 6: kp = 53; break;
148:             case 9: kp = 57; break;
149:             case 10: kp = 56; break;
150:             case 11: kp = 55; break;
151:             case 13: kp = 35; break;
152:             case 14: kp = 48; break;
153:             case 15: kp = 42; break;
154:         }
155:
156:         Lcd_Chrcp(kp);
157:         t++;
158:
159:         if (t >= 4) {
160:             if (x == 5) {
161:                 Delay_ms(500);
162:                 lcd_cmd(_LCD_CLEAR);
163:                 lcd_cmd(_LCD_CURSOR_OFF);
164:                 Lcd_out(1, 6, "DOOR");
165:                 PORTA.B3 = 1;
166:                 Lcd_out(2, 5, "UNLOCKED");
167:
168:                 PORTC.B1 = 1;
169:                 PWM1_Set_Duty(200);
170:                 Delay_ms(300);
171:                 PORTC.B1 = 0;
172:                 PWM1_Stop();
173:                 Delay_ms(500);
174:                 PORTA.B3 = 0;
175:
176: point2:
177:                 lcd_cmd(_LCD_CLEAR);
178:                 Lcd_out(1, 1, "Door open. Lock?");
179:                 Lcd_out(2, 1, "Press: #");
180:
181:                 do {
182:                     CheckMountDismount();
183:                     dl = Keypad_Key_Click();
184:                 } while (!dl);
185:
186:                 if (dl == 13) {
```

```
187:         Delay_ms(400);
188:         lcd_cmd(_LCD_CLEAR);
189:         Lcd_out(1,3,"DOOR LOCKED");
190:         PORTA.B3 = 0;
191:
192:         PWM1_Start();
193:         PORTC.B0 = 1;
194:         PWM1_Set_Duty(200);
195:         Delay_ms(300);
196:         PORTC.B0 = 0;
197:         PWM1_Stop();
198:
199:         PORTC.B6 = 1;
200:         PORTA = 0x0C;
201:         Lcd_out(2,1,"SECURITY ACTIVE");
202:         Delay_ms(1300);
203:         PORTA = 0x00;
204:         PORTC.B6 = 0;
205:         lcd_cmd(_LCD_CLEAR);
206:         goto point1;
207:     }
208: } else {
209:     PORTA.B2 = 0; Delay_ms(150);
210:     PORTA.B2 = 1; Delay_ms(120);
211:     PORTA.B2 = 0; Delay_ms(150);
212:     PORTA.B2 = 1; Delay_ms(120);
213:
214:     lcd_cmd(_LCD_CLEAR);
215:     Lcd_out(1,3,"Access denied");
216:     Lcd_out(2,3,"Try again..");
217:     Delay_ms(1000);
218:     PORTA.B2 = 0;
219:     goto point1;
220: }
221: }
222: }
223: }
224: }
```