```
#include<reg51.h>
#include<string.h>

//Micro
#define lcdport P2

//LCD
sbit rs = P3^4;
sbit rw = P3^5;
sbit en = P3^6;

//Motor Pins
sbit m1 = P3^0;
sbit m2 = P3^1;

//KeyRows
sbit r1 = P1^0;
sbit r2 = P1^1;
sbit r3 = P1^2;
sbit r4 = P1^3;

//KeyCol
sbit c1 = P1^4;
sbit c2 = P1^5;
sbit c3 = P1^6;


// buzzPin
sbit buzzer = P1^7;


// Pass
char uid[] = "54321";
char id[5];



// Bike ignition and lock flags
bit bike_on_flag = 0;
bit bike_locked_flag = 0;


int wrong_attempts = 0;

// Function
void lcdint();
void lcddis(char *);
void delay(int);
void lcdcmd(char);
void lcddata(char);
char lcdkey();
char scan_key();
void ignite_on();
void ignite_off();
void sounder();


void main() {

  int n;
  char key;


  P2 = 0x00;
  P1 = 0x00;
  buzzer = 0;

  lcdint();

  if (bike_on_flag == 1) {
    lcddis("Odometer");
    while (1);
  }


  if (bike_locked_flag == 1) {
    lcddis("Bike Locked");
```

```
78        while (1);
79      }
80
81
82      // Initial prompt
83      lcddis("Ignition Locked");
84      delay(100);
85      lcdcmd(0x01);
86      lcdcmd(0x02);
87
88
89    retry_password:
90      lcddis("Enter Password:");
91      lcdcmd(0xc0);
92
93      n = 0;
94      while (n < 5) {
95        key = scan_key();
96        id[n] = key;
97        delay(70);
98        n++;
99      }
100
101     lcdcmd(0x01);
102     lcdcmd(0x02);
103
104
105
106     if (strcmp(uid, id) == 0) {
107       lcddis("Bike ON");
108       delay(200);
109       ignite_on();
110       bike_on_flag = 1;
111       delay(200);
112
113       lcdcmd(0x01);
114       lcdcmd(0x02);
115       lcddis("Enjoy The Ride");
116       delay(500);
117
118       lcdcmd(0x01);              // Clear display
119       lcdcmd(0x80);              // Move to line 1
120       lcddis("Speed : 000 km/h");
121
122       lcdcmd(0xC0);              // Move to line 2
123       lcddis("Trip A: 0762 Km");
124       while (1);
125     }
126
127
128     else {
129       wrong_attempts++;
130       lcddis("Access Denied");
131       if(wrong_attempts == 1){
132         lcdcmd(0xC0);
133       lcddis("Only 2 attempt left");
134       }
135       else if(wrong_attempts == 2){
136         lcdcmd(0xC0);
137       lcddis("Only 1 attempt left");
138       }
139
140       sounder();
141       delay(150);
142       lcdcmd(0x01);
143       lcdcmd(0x02);
144
145         if (wrong_attempts >= 3) {
146           int i;
147           lcdcmd(0x01);
148           lcddis("Warning!");
149           delay(200);
150
151           for (i = 3; i > 0; i--) {
152               lcdcmd(0x01);
153               lcdcmd(0x80);
154               lcddis("Locking in ");
```

```
155                 lcddata(i + '0');  // Convert int to char
156                 lcddata('.');
157                 lcddata('.');
158                 lcddata('.');
159                 delay(100);
160             }
161
162             lcdcmd(0x01);
163             lcddis("Bike Locked");
164             bike_locked_flag = 1;
165
166             while (1);  // Lock the system permanently
167         }
168
169
170         lcddis("Try Again");
171         delay(200);
172         lcdcmd(0x01);
173         lcdcmd(0x02);
174         goto retry_password;
175       }
176     }
177
178     // LCDFun
179     void lcdint() {
180
181       lcdcmd(0x38);
182       delay(2);
183
184       lcdcmd(0x01);
185       delay(2);
186
187       lcdcmd(0x80);
188       delay(2);
189
190       lcdcmd(0x0e);
191       delay(2);
192     }
193
194     void delay(int x) {
195       int i, j;
196       for (i = 0; i < x; i++)
197         for (j = 0; j < 1275; j++);
198     }
199
200     void lcdcmd(char A) {
201       lcdport = A;
202       rs = 0;
203       rw = 0;
204       en = 1;
205       delay(1);
206       en = 0;
207     }
208
209     void lcddis(char *p) {
210       while (*p != '\0') {
211         lcddata(*p);
212         delay(10);
213         p++;
214       }
215     }
216
217     void lcddata(char value) {
218       lcdport = value;
219       rs = 1;
220       rw = 0;
221       en = 1;
222       delay(1);
223       en = 0;
224     }
225
226     // Buzzer
227     void sounder() {
228       int i;
229       for (i = 0; i < 5; i++) {
230         buzzer = 1;
231         delay(200);
```

```
232       buzzer = 0;
233       delay(200);
234     }
235   }
236
237   // Ignition control
238   void ignite_on() {
239     m1 = 1;
240     m2 = 0;
241   }
242
243
244   void ignite_off() {
245     m1 = 0;
246     m2 = 0;
247     delay(20);
248     m1 = 0;
249     m2 = 1;
250     delay(500);
251     m1 = 0;
252     m2 = 0;
253   }
254
255   // Keypad
256   char scan_key() {
257     char b = 'a';
258     while (b == 'a') {
259       b = lcdkey();
260     }
261     return b;
262   }
263
264   char lcdkey() {
265     c1 = c2 = c3 = 1;
266     r1 = r2 = r3 = r4 = 0;
267
268     r1 = 0; r2 = r3 = r4 = 1;
269     if (c1 == 0) { lcddata('*'); delay(2); return '1'; }
270     if (c2 == 0) { lcddata('*'); delay(2); return '2'; }
271     if (c3 == 0) { lcddata('*'); delay(2); return '3'; }
272
273     r2 = 0;
274     r1 = r3 = r4 = 1;
275
276     if (c1 == 0) { lcddata('*'); delay(2); return '4'; }
277     if (c2 == 0) { lcddata('*'); delay(2); return '5'; }
278     if (c3 == 0) { lcddata('*'); delay(2); return '6'; }
279
280     r3 = 0;
281     r1 = r2 = r4 = 1;
282     if (c1 == 0) { lcddata('*'); delay(2); return '7'; }
283     if (c2 == 0) { lcddata('*'); delay(2); return '8'; }
284     if (c3 == 0) { lcddata('*'); delay(2); return '9'; }
285
286     r4 = 0;
287     r1 = r2 = r3 = 1;
288     if (c1 == 0) { lcddata('*'); delay(2); return '*'; }
289     if (c2 == 0) { lcddata('*'); delay(2); return '0'; }
290     if (c3 == 0) { lcddata('*'); delay(2); return '#'; }
291
292     return 'a';
293   }
294
```