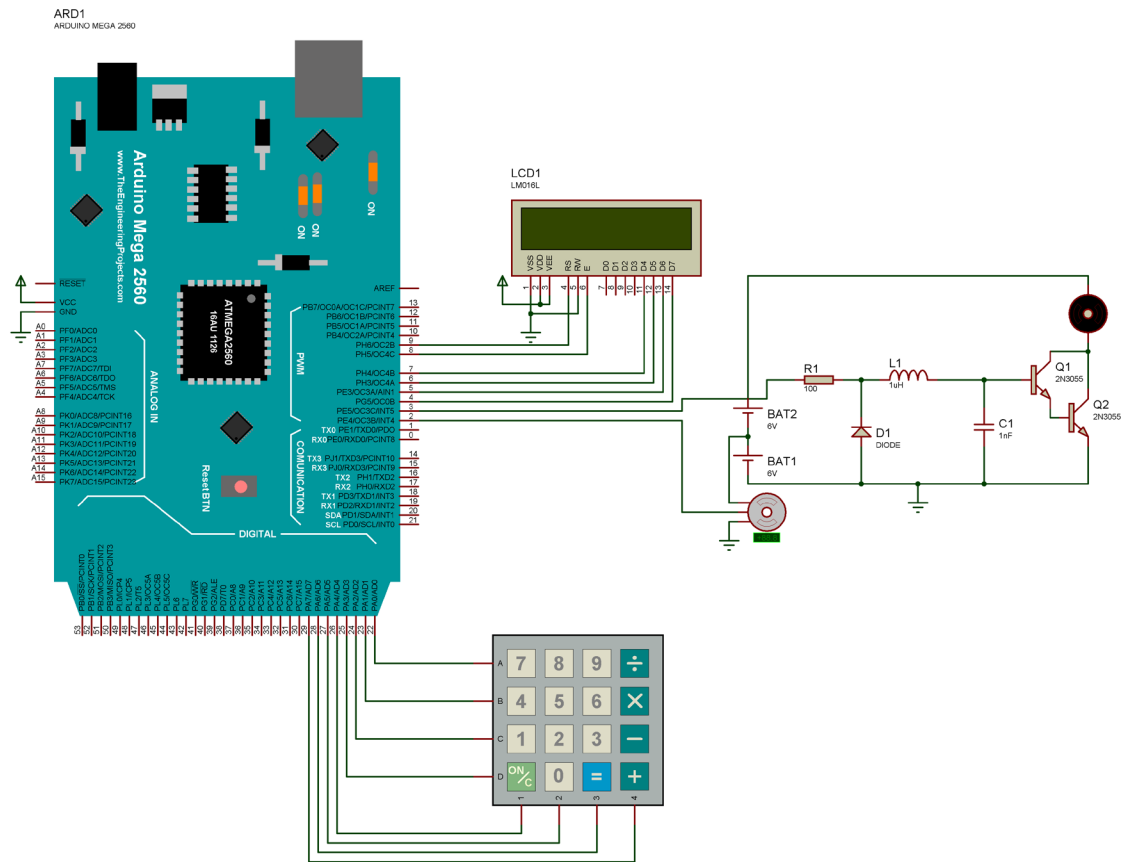


آزمایش 6: نیم-پروژه



```

1. #include <math.h>
2. #include <Servo.h>
3. #include <Keypad.h>
4. #include <LiquidCrystal.h>
5.
6. #define SPEED_PIN 3
7. #define INCLINE_PIN 2
8. #define SPD_STEP 16
9. #define INC_STEP 3
10. #define MIN_INC -6
11. #define MAX_INC 30
12.
13. Servo rampServo;
14. short spd;
15. short lastSpd;
16. float inc;
17. float lastInc;
18. bool reset;
19. bool negInc;
20.
21. const byte ROWS = 4;
22. const byte COLS = 4;
23. const char keys[ROWS][COLS] = {
24.   {'7','8','9','F'},
25.   {'4','5','6','L'},
26.   {'1','2','3','U'},
27.   {'S','0','I','D'}
28. };
29. const byte rowPins[ROWS] = {22, 23, 24, 25};
30. const byte colPins[COLS] = {26, 27, 28, 29};
31. Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
32.

```

```

33. #define RS_PIN 9
34. #define EN_PIN 8
35. #define D4_PIN 7
36. #define D5_PIN 6
37. #define D6_PIN 5
38. #define D7_PIN 4
39. #define NUM_COL 10
40. LiquidCrystal lcd (RS_PIN, EN_PIN, D4_PIN, D5_PIN, D6_PIN, D7_PIN);
41.
42. uint8_t MODE; //0: +/- input, 1:numeric speed, 2:numeric incline
43.
44. void setup() {
45.     pinMode(SPEED_PIN, OUTPUT);
46.     spd = 0;
47.     lastSpd = spd;
48.     analogWrite(SPEED_PIN, spd);
49.     pinMode(INCLINE_PIN, OUTPUT);
50.     rampServo.attach(INCLINE_PIN, 1000, 2000);
51.     inc = 0;
52.     lastInc = inc;
53.     setIncline();
54.
55.     lcd.begin(16,2);
56.     setLCD();
57.     MODE = 0;
58. }
59.
60. void loop() {
61.     char key = keypad.getKey();
62.
63.     if (key){
64.         if(MODE == 0) {
65.             switch(key) {
66.                 case 'F':
67.                     spd++;
68.                     setSpeed();
69.                     break;
70.                 case 'L':
71.                     spd--;
72.                     setSpeed();
73.                     break;
74.                 case 'U':
75.                     inc += INC_STEP;
76.                     setIncline();
77.                     break;
78.                 case 'D':
79.                     inc -= INC_STEP;
80.                     setIncline();
81.                     break;
82.                 case 'S':
83.                     MODE = 1;
84.                     reset = false;
85.                     break;
86.                 case 'I':
87.                     MODE = 2;
88.                     reset = false;
89.                     break;
90.             }
91.         } else if(MODE == 1 && (key >= '0' && key <= '9' || key == 'S')) {
92.             if(key == 'S') {
93.                 MODE = 0;
94.                 setSpeed();
95.             } else if(reset && spd > 9) {
96.                 spd = 256/SPD_STEP;
97.                 setSpeed();
98.                 MODE = 0;
99.             } else {
100.                 if(!reset) {
101.                     reset = true;

```

```

102.         spd = 0;
103.     }
104.     spd *= 10;
105.     spd += key - '0';
106.     lcd.setCursor(NUM_COL, 0);
107.     lcd.print(spd);
108.     lcd.print("    ");
109. }
110.
111. } else if(MODE == 2 && (key >= '0' && key <= '9' || key == 'U' || key == 'I')) {
112.     if(key == 'I') {
113.         MODE = 0;
114.         setIncline();
115.     } else if(reset && (inc > 9 || inc < 0)) {
116.         if(inc > 9)
117.             inc = MAX_INC;
118.         else
119.             inc = MIN_INC;
120.         setIncline();
121.         MODE = 0;
122.     } else {
123.         if(!reset) {
124.             reset = true;
125.             inc = 0;
126.             if(key == 'U')
127.                 negInc = true;
128.             else
129.                 negInc = false;
130.         }
131.         if(key != 'U') {
132.             inc *= 10;
133.             if(negInc)
134.                 inc -= key - '0';
135.             else
136.                 inc += key - '0';
137.         }
138.         lcd.setCursor(NUM_COL, 1);
139.         if(inc == 0 && negInc)
140.             lcd.print('-');
141.         lcd.print((int)inc);
142.         lcd.print("%    ");
143.     }
144. }
145. }
146. }
147.
148. }
149.
150. void setIncline() {
151.     if(inc > MAX_INC)
152.         inc = MAX_INC;
153.     else if(inc < MIN_INC)
154.         inc = MIN_INC;
155.     float finalInc = inc;
156.     int sgn = 1;
157.     if(lastInc > inc)
158.         sgn = -1;
159.     for(inc = lastInc; fabs(inc - finalInc) >= 3; inc += sgn) {
160.         rampServo.write(map(atan(inc/100)*180/PI*8.94, -30, 150, 0, 180));
161.         if((int)(inc-finalInc) % 5 == 0)
162.             setLCD();
163.         delay(50);
164.     }
165.     inc = finalInc;
166.     rampServo.write(map(atan(inc/100)*180/PI*8.94, -30, 150, 0, 180));
167.     setLCD();
168.     lastInc = inc;
169. }
170.

```

```

171. uint8_t safeMult(short a, short b) {
172.     if(a < 0 || b < 0)
173.         return 0;
174.     if(a*b > 255)
175.         return 255;
176.     return a*b;
177. }
178.
179. void setSpeed() {
180.     int finalSpd = spd;
181.     int sgn = 1;
182.     if(finalSpd < lastSpd)
183.         sgn = -1;
184.     for(spd = lastSpd; spd >= 0 && spd <= 16 && spd != finalSpd; spd += sgn) {
185.         analogWrite(SPEED_PIN, safeMult(spd, SPD_STEP));
186.         if((int)(spd-finalSpd) % 4 == 0)
187.             setLCD();
188.         delay(100);
189.     }
190.     spd = finalSpd;
191.     if(spd*SPD_STEP > 255) {
192.         spd = 256/SPD_STEP;
193.     } else if(spd < 0) {
194.         spd = 0;
195.     }
196.     analogWrite(SPEED_PIN, safeMult(spd, SPD_STEP));
197.     lastSpd = spd;
198.     setLCD();
199. }
200.
201. void setLCD() {
202.     lcd.clear();
203.     lcd.setCursor(0,0);
204.     lcd.print("Speed:");
205.     lcd.setCursor(NUM_COL, 0);
206.     lcd.print(spd);
207.     lcd.setCursor(0, 1);
208.     lcd.print("Incline:");
209.     lcd.setCursor(NUM_COL, 1);
210.     lcd.print((int)inc);
211.     lcd.write('%');
212. }
213.

```