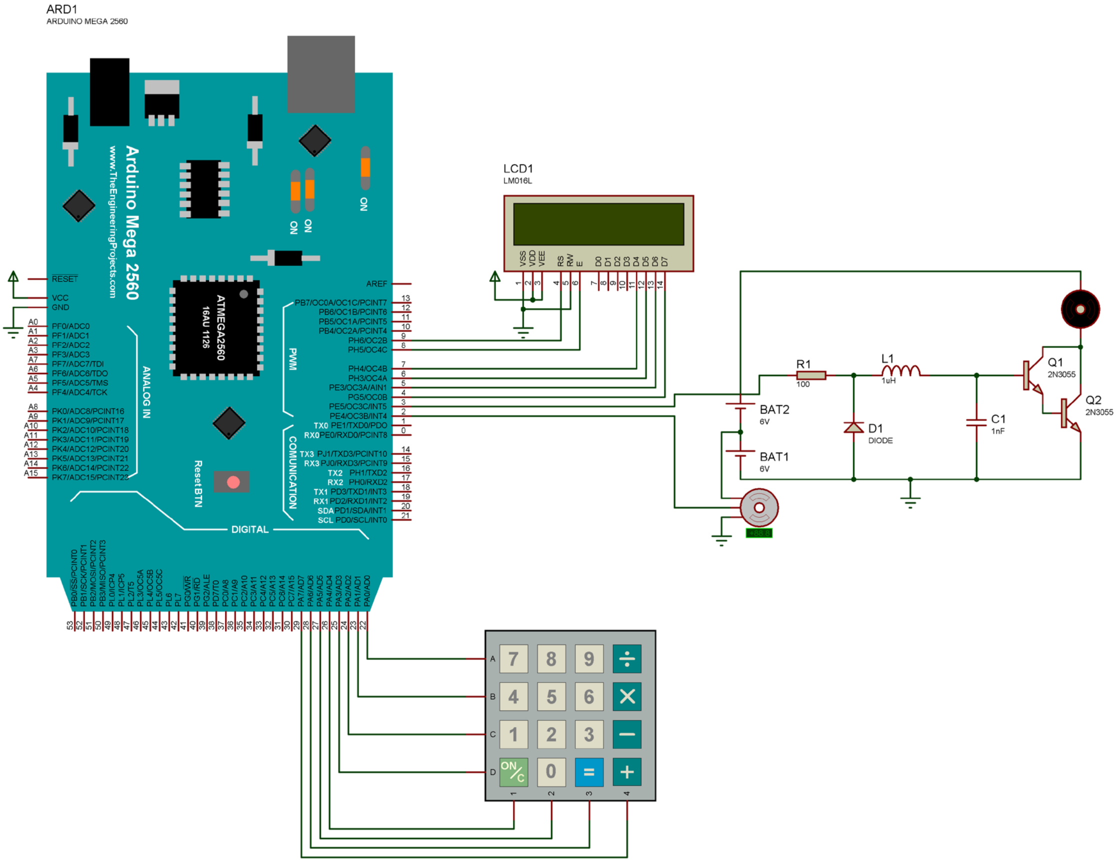
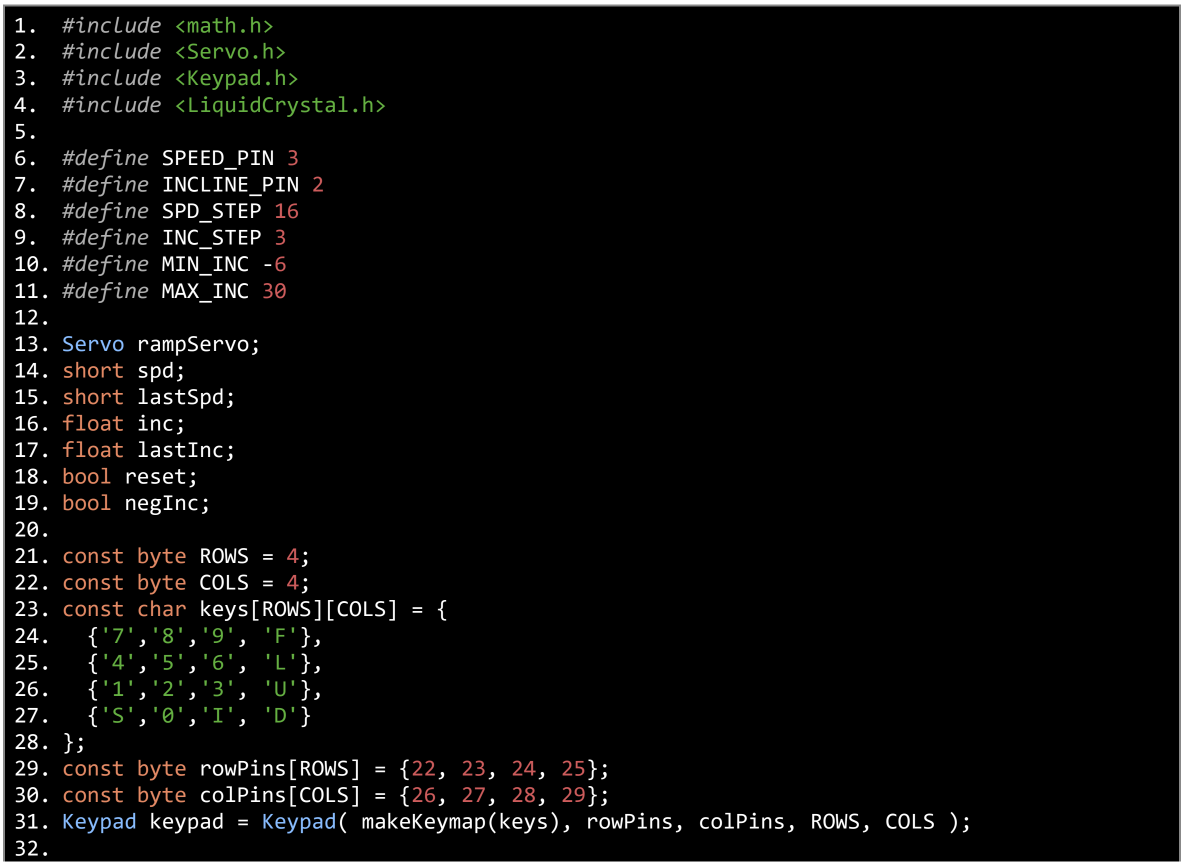
آزمایش 6: نیم-پروژه





```
#include <math.h>
#include <Servo.h>
#include <Keypad.h>
#include <LiquidCrystal.h>

#define SPEED_PIN 3
#define INCLINE_PIN 2
#define SPD_STEP 16
#define INC_STEP 3
#define MIN_INC -6
#define MAX_INC 30

Servo rampServo;
short spd;
short lastSpd;
float inc;
float lastInc;
bool reset;
bool negInc;

const byte ROWS = 4;
const byte COLS = 4;
const char keys[ROWS][COLS] = {
  {'7','8','9', 'F'},
  {'4','5','6', 'L'},
  {'1','2','3', 'U'},
  {'S','0','I', 'D'}
};
const byte rowPins[ROWS] = {22, 23, 24, 25};
const byte colPins[COLS] = {26, 27, 28, 29};
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

```

```
#define RS_PIN 9
#define EN_PIN 8
#define D4_PIN 7
#define D5_PIN 6
#define D6_PIN 5
#define D7_PIN 4
#define NUM_COL 10
LiquidCrystal lcd (RS_PIN, EN_PIN, D4_PIN, D5_PIN, D6_PIN, D7_PIN);

uint8_t MODE; //0: +/- input, 1:numeric speed, 2:numeric incline

void setup() {
  pinMode(SPEED_PIN, OUTPUT);
  spd = 0;
  lastSpd = spd;
  analogWrite(SPEED_PIN, spd);
  pinMode(INCLINE_PIN, OUTPUT);
  rampServo.attach(INCLINE_PIN, 1000, 2000);
  inc = 0;
  lastInc = inc;
  setIncline();

  lcd.begin(16,2);
  setLCD();
  MODE = 0;
}

void loop() {
  char key = keypad.getKey();

  if (key){
    if(MODE == 0) {
      switch(key) {
        case 'F':
          spd++;
          setSpeed();
          break;
         case 'L':
          spd--;
          setSpeed();
          break;
         case 'U':
          inc += INC_STEP;
          setIncline();
          break;
         case 'D':
          inc -= INC_STEP;
          setIncline();
          break;
         case 'S':
          MODE = 1;
          reset = false;
          break;
         case 'I':
          MODE = 2;
          reset = false;
          break;
      }
    } else if(MODE == 1 && (key >= '0' && key <= '9' || key == 'S')) {
      if(key == 'S') {
        MODE = 0;
        setSpeed();
      } else if(reset && spd > 9) {
        spd = 256/SPD_STEP;
        setSpeed();
        MODE = 0;
      } else {
        if(!reset) {
          reset = true;
```

```
     spd = 0;
    }
    spd *= 10;
    spd += key - '0';
    lcd.setCursor(NUM_COL, 0);
    lcd.print(spd);
    lcd.print("      ");
   }

  } else if(MODE == 2 && (key >= '0' && key <= '9' || key =='U' || key == 'I')) {
   if(key == 'I') {
    MODE = 0;
    setIncline();
   }else if(reset && (inc > 9 || inc < 0)) {
    if(inc > 9)
     inc = MAX_INC;
    else
     inc = MIN_INC;
    setIncline();
    MODE = 0;
   } else {
    if(!reset) {
     reset = true;
     inc = 0;
     if(key == 'U')
      negInc = true;
     else
      negInc = false;
    }
    if(key != 'U') {
     inc *= 10;
     if(negInc)
      inc -= key - '0';
     else
      inc += key - '0';
    }
    lcd.setCursor(NUM_COL, 1);
    if(inc == 0 && negInc)
     lcd.print('-');
    lcd.print((int)inc);
    lcd.print("%      ");
   }

  }
 }

}

void setIncline() {
 if(inc > MAX_INC)
  inc = MAX_INC;
 else if(inc < MIN_INC)
  inc = MIN_INC;
 float finalInc = inc;
 int sgn = 1;
 if(lastInc > inc)
  sgn = -1;
 for(inc = lastInc; fabs(inc - finalInc) >= 3; inc += sgn) {
  rampServo.write(map(atan(inc/100)*180/PI*8.94, -30, 150, 0, 180));
  if((int)(inc-finalInc) % 5 == 0)
   setLCD();
  delay(50);
 }
 inc = finalInc;
 rampServo.write(map(atan(inc/100)*180/PI*8.94, -30, 150, 0, 180));
 setLCD();
 lastInc = inc;
}

```

```
uint8_t safeMult(short a, short b) {
  if(a < 0 || b < 0)
    return 0;
  if(a*b > 255)
    return 255;
  return a*b;
}

void setSpeed() {
  int finalSpd = spd;
  int sgn = 1;
  if(finalSpd < lastSpd)
    sgn = -1;
  for(spd = lastSpd; spd >= 0 && spd <= 16 && spd != finalSpd; spd += sgn) {
    analogWrite(SPEED_PIN, safeMult(spd, SPD_STEP));
    if((int)(spd-finalSpd) % 4 == 0)
      setLCD();
    delay(100);
  }
  spd = finalSpd;
  if(spd*SPD_STEP > 255) {
    spd = 256/SPD_STEP;
  } else if(spd < 0) {
    spd = 0;
  }
  analogWrite(SPEED_PIN, safeMult(spd, SPD_STEP));
  lastSpd = spd;
  setLCD();
}

void setLCD() {
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Speed:");
  lcd.setCursor(NUM_COL, 0);
  lcd.print(spd);
  lcd.setCursor(0, 1);
  lcd.print("Incline:");
  lcd.setCursor(NUM_COL, 1);
  lcd.print((int)inc);
  lcd.write('%');
}

```