```
#include <LiquidCrystal.h>
#include <SoftwareSerial.h>

SoftwareSerial mySerial(1, 0); // RX, TX
const int trigPin = 12; // Trigger Pin of Ultrasonic Sensor
const int echoPin = 11; // Echo Pin of Ultrasonic Sensor
const int motorInPin1 = A5; // DC Motor input1 Pin
const int motorInPin2 = A4; // DC Motor input2 Pin
const int motorEnPin = A3; // DC Motor Pin
const int rs = 2, en = 3, d4 = 4, d5 = 5, d6 = 6, d7 = 7;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

int ledpin_a = 10;
int ledpin_b = 9;
int ledpin_c = 8;
int ledpin_d = 13;

void setup() {
// Start serial communication
Serial.begin(9600); // Starting Serial Terminal
mySerial.begin(9600);

// Initialize the LCD
lcd.begin(16, 2);
lcd.print("Sensor Based Autonomous Drone");
Serial.print(" Hardware Circuit Design ");
delay(1000);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print(" Ultrasonic ");
Serial.print(" Ultrasonic Sensor ");
lcd.setCursor(0, 1);
lcd.print(" Implementation ");
Serial.print(" Distance ");
delay(1000);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Developed By : ");
delay(1000);
Serial.print("Developed By :");
lcd.setCursor(0, 0);
lcd.print(" Electrical/Electronics ");
delay(1000);
```

```
lcd.setCursor(0, 1);
lcd.print("Engineers");
delay(1000);
Serial.print(" Electrical/Electronics Engineers");
delay(2000);

// Set up pins
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
pinMode(ledpin_d, OUTPUT);
pinMode(ledpin_a, OUTPUT);
pinMode(ledpin_b, OUTPUT);
pinMode(ledpin_c, OUTPUT);
pinMode(motorInPin1, OUTPUT);
pinMode(motorInPin2, OUTPUT);
pinMode(motorEnPin, OUTPUT);
}

void loop() {

// Ultrasonic sensor logic
long duration, inches, cm, m;
pinMode(trigPin, OUTPUT);
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

pinMode(echoPin, INPUT);
duration = pulseIn(echoPin, HIGH);
inches = microsecondsToInches(duration);
cm = microsecondsToCentimeters(duration);
m = microsecondsToMeter(duration);
lcd.setCursor(0, 1);
// Print distance to Serial Monitor
Serial.print("Distance: ");
Serial.print(cm);
Serial.println("cm");
delay(100);
Serial.print("Distance: ");
Serial.print(m);
Serial.println("m");
delay(100);
```

```
// Control DC motor and display based on distance
if (cm >= 250) {
// Display condition on LCD
lcd.clear();
lcd.print("Condition: LED On");
Serial.println("Condition: LED On");
// Blink all LEDs
digitalWrite(ledpin_a, HIGH);
//delay(500);
// digitalWrite(ledpin_d, LOW);
//delay(50);
digitalWrite(ledpin_b, HIGH);
//delay(500);
// digitalWrite(ledpin_a, LOW);
// delay(50);
digitalWrite(ledpin_c, HIGH);
//delay(500);
// digitalWrite(ledpin_b, LOW);
// delay(50);
digitalWrite(ledpin_d, HIGH);
//delay(500);
// digitalWrite(ledpin_c, LOW);
// delay(50);
analogWrite(motorEnPin, 0);
} else {
//speed regulation based on distance
int motorSpeed = map(cm, 0, 25, 0,255);
// Run DC motor continuously
analogWrite(motorEnPin, motorSpeed); // Full speed
digitalWrite(motorInPin1, HIGH);
digitalWrite(motorInPin2, LOW);
Serial.println("Motor Speed.");
```

```
// Display motor speed on LCD
lcd.clear();
lcd.print("Motor Speed: ");
lcd.print(motorSpeed);  // Display full speed
Serial.print("Motor Speed: ");
Serial.println(motorSpeed);
 digitalWrite(ledpin_d, HIGH);
```

/delay(50);
digitalWrite(ledpin_d, LOW);

```
delay(50);
digitalWrite(ledpin_a, HIGH);
delay(50);
digitalWrite(ledpin_a, LOW);
delay(50);
digitalWrite(ledpin_b, HIGH);
delay(50);
digitalWrite(ledpin_b, LOW);
delay(50);
digitalWrite(ledpin_c, HIGH);
delay(50);
digitalWrite(ledpin_c, LOW);
delay(50);/
}

// Print distance to LCD
lcd.setCursor(0, 0);
lcd.print("");
delay(10);
lcd.setCursor(0, 1);
lcd.print("Distance:");
lcd.print(cm);
lcd.print("cm");
lcd.print(m);
lcd.print("m");
delay(100);

// Send distance to external serial device
mySerial.println(cm);
mySerial.println("cm");
mySerial.println();
delay(100);

}

// Function to convert microseconds to inches
long microsecondsToInches(long microseconds) {
return microseconds / 74 / 2;
}
long microsecondsToMeter(long microseconds) {
const int speedOfSound = 330.0;
return (speedOfSound microseconds) / (2 1000000.0);
}
// Function to convert microseconds to centimeters
long microsecondsToCentimeters(long microseconds) {
```

```
return microseconds / 29 / 2;
}
```