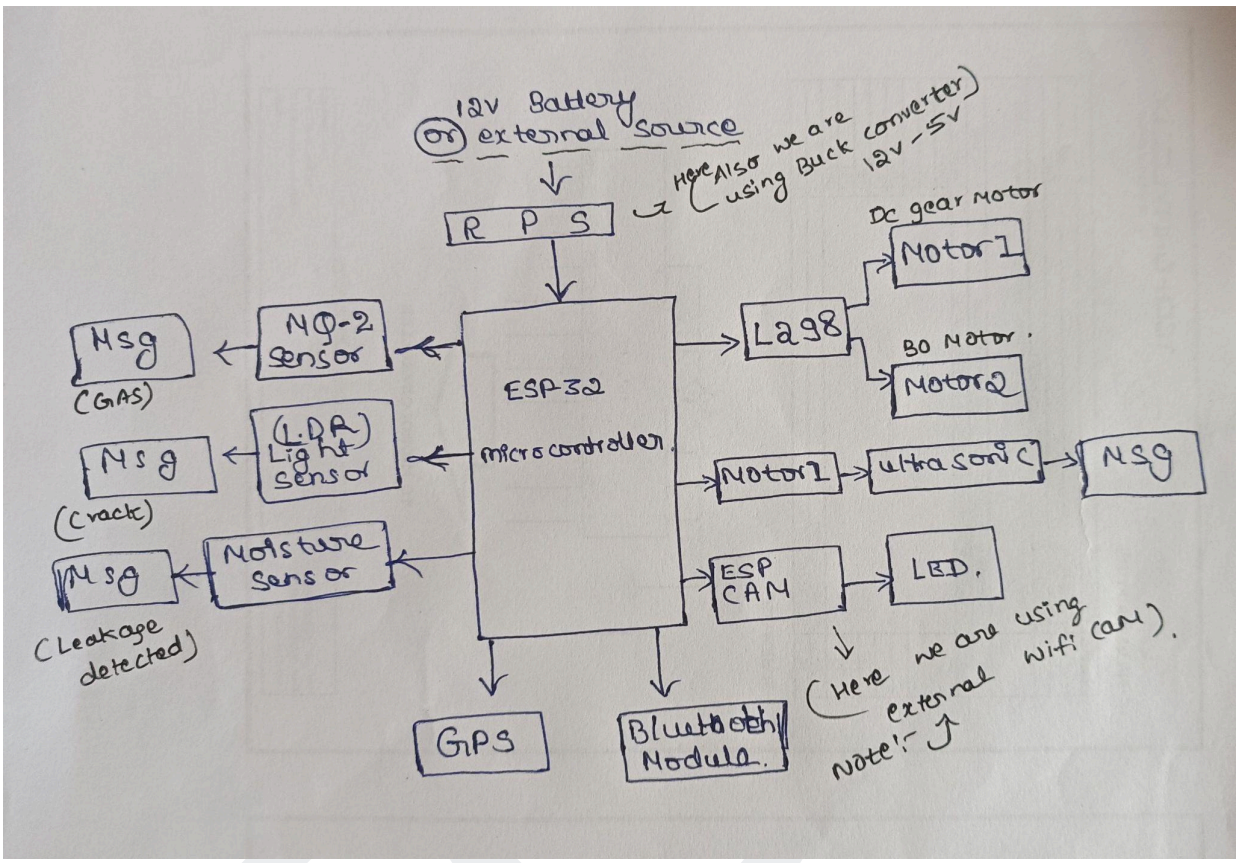
Block diagram:-



Code:-

```
#define IN1  26
#define IN2  27

#define IN3  25
#define IN4  33

#define TRIG_PIN  13
#define ECHO_PIN  15

#define RXD2 16
#define TXD2 17

#define MQ2_PIN  34

#define LIGHT_PIN  35
```

```
#define MOISTURE_PIN  32

#define GPS_RX 4
#define GPS_TX 2
#define GPS_BAUD 9600

#include <TinyGPS.h>

HardwareSerial gpsSerial(1);
HardwareSerial bluetoothSerial(2);

void setup() {
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);

  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(MQ2_PIN, INPUT);
  pinMode(LIGHT_PIN, INPUT);
  pinMode(MOISTURE_PIN, INPUT);

  Serial.begin(115200);

  bluetoothSerial.begin(9600, SERIAL_8N1, RXD2, TXD2);

  gpsSerial.begin(GPS_BAUD, SERIAL_8N1, GPS_RX, GPS_TX);
  Serial.println("GPS and Bluetooth Serials started");
}
void motor1(int direction) {
  if (direction == 1) {
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    bluetoothSerial.println("Driller ON, Blockage Detected");
  } else {
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
    //bluetoothSerial.println("Driller OFF");
  }
}

void motor2(int direction) {
  if (direction == 1) {
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
    Serial.println("BO Motor Forward");
    bluetoothSerial.println("BO Motor Forward");
```

```
  } else if (direction == -1) {
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);
    Serial.println("BO Motor Backward");
    bluetoothSerial.println("BO Motor Backward");
  } else {
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);
    Serial.println("BO Motor Stopped");
    bluetoothSerial.println("BO Motor Stopped");
  }
}

long getDistance() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  long duration = pulseIn(ECHO_PIN, HIGH);
  return duration * 0.034 / 2;
}

bool checkGas() { return digitalRead(MQ2_PIN); }
bool checkLight() { return digitalRead(LIGHT_PIN); }
bool checkMoisture() { return !digitalRead(MOISTURE_PIN); }

void sendGPSData() {
  unsigned long start = millis();
  while (millis() - start < 1000) {
    while (gpsSerial.available() > 0) {
      gps.encode(gpsSerial.read());
    }
    if (gps.location.isUpdated()) {
      float latitude = gps.location.lat();
      float longitude = gps.location.lng();

      bluetoothSerial.println("GPS Data:");
      Serial.println("GPS Data:");
      bluetoothSerial.print("LAT: ");
      bluetoothSerial.println(latitude, 6);
      bluetoothSerial.print("LONG: ");
      bluetoothSerial.println(longitude, 6);
      bluetoothSerial.print("SPEED (km/h): ");
      bluetoothSerial.println(gps.speed.kmph());
      bluetoothSerial.print("ALT (m): ");
      bluetoothSerial.println(gps.altitude.meters());
      bluetoothSerial.print("HDOP: ");
```

```
      bluetoothSerial.println(gps.hdop.value() / 100.0);
      bluetoothSerial.print("Satellites: ");
      bluetoothSerial.println(gps.satellites.value());
      bluetoothSerial.print("Time in UTC: ");
      bluetoothSerial.println(String(gps.date.year()) + "/" +
                              String(gps.date.month()) + "/" +
                              String(gps.date.day()) + "," +
                              String(gps.time.hour()) + ":" +
                              String(gps.time.minute()) + ":" +
                              String(gps.time.second()));

      bluetoothSerial.println("\nGoogle Map Link:");
      Serial.println("\nGoogle Map Link:");
      bluetoothSerial.print("https://www.google.com/maps?q=");
      Serial.println("https://www.google.com/maps?q=");

      Serial.println("GPS Data sent via Bluetooth");
      return;
    }
  }
  bluetoothSerial.println("No valid GPS data available");
}

void loop() {

  if (bluetoothSerial.available()) {
    char command = bluetoothSerial.read();
    switch(command) {
      case 'F':
        motor2(1);
        break;
      case 'B':
        motor2(-1);
        break;
      case 'P':
        motor2(0);
        break;
      case '$':
        sendGPSData();
        break;
      default:
        //bluetoothSerial.println("Invalid command");
        break;
    }
  }

  long distance = getDistance();
  Serial.print("Distance: ");
```

```
  Serial.print(distance);
  Serial.println(" cm");
  if (distance < 30 && distance > 0) {
    motor1(1);
    Serial.println("DC Motor ON - Object detected");
  } else {
    motor1(0);
    Serial.println("DC Motor OFF");
  }

  if (checkGas()) {
    Serial.println("GAS is Detected");
    bluetoothSerial.println("GAS is Detected");
  }
  if (checkLight()) {
    Serial.println("Crack Detected");
    bluetoothSerial.println("Crack Detected");
  }
  if (checkMoisture()) {
    Serial.println("Leakage Alert");
    bluetoothSerial.println("Leakage Alert");
  }

  delay(100);
}
```

TGL