#include <AFMotor.h>

// Motores individuales

AF\_DCMotor motor1(1); // M1

AF\_DCMotor motor2(2); // M2

AF\_DCMotor motor3(3); // M3

AF\_DCMotor bomba(4);  // M4

// Pines del sensor ultrasónico

#define TRIG\_PIN A4

#define ECHO\_PIN A5

// Sensor de nivel de agua

#define SENSOR\_AGUA A0

#define UMBRAL\_AGUA 410  // 40%

bool sistemaActivo = true;

bool retrocesoRealizado = false;

void setup() {

  // Configurar motores

  motor1.setSpeed(255);

  motor2.setSpeed(255);

  motor3.setSpeed(255);

  bomba.setSpeed(255);

  pinMode(TRIG\_PIN, OUTPUT);

  pinMode(ECHO\_PIN, INPUT);

  Serial.begin(9600);

}

void loop() {

  int nivelAgua = analogRead(SENSOR\_AGUA);

  Serial.print("Nivel de agua: ");

  Serial.println(nivelAgua);

  if (nivelAgua < UMBRAL\_AGUA && !retrocesoRealizado) {

    // Agua por debajo del 40% → apagar todo

    apagarTodo();

    Serial.println("Nivel bajo de agua. Sistema apagado.");

    // Retroceder 8 segundos

    motor1.run(BACKWARD);

    motor2.run(FORWARD);

    motor3.run(BACKWARD);

    Serial.println("Retrocediendo por 8 segundos...");

    delay(8000);

    // Apagar motores nuevamente

    motor1.run(RELEASE);

    motor2.run(RELEASE);

    motor3.run(RELEASE);

    retrocesoRealizado = true;

    while (true) {

      Serial.println("Esperando nivel adecuado de agua...");

      delay(2000);

    }

  }

  if (sistemaActivo) {

    long distancia = medirDistancia();

    if (distancia <= 13 && distancia > 0) {

      // Detener motores

      motor1.run(RELEASE);

      motor2.run(RELEASE);

      motor3.run(RELEASE);

      Serial.println("Obstáculo detectado. Activando bomba.");

      // Activar bomba 10 segundos

      bomba.run(BACKWARD);

      delay(4000);

      bomba.run(RELEASE);

      delay(1000); // Pausa antes de reanudar movimiento

    } else {

      // Avanzar

      motor1.run(FORWARD);

      motor2.run(BACKWARD);

      motor3.run(FORWARD);

    }

  }

  delay(100); // Pausa

}

// Función para medir distancia

long medirDistancia() {

  digitalWrite(TRIG\_PIN, LOW);

  delayMicroseconds(2);

  digitalWrite(TRIG\_PIN, HIGH);

  delayMicroseconds(10);

  digitalWrite(TRIG\_PIN, LOW);

  long duracion = pulseIn(ECHO\_PIN, HIGH, 30000);

  if (duracion == 0) return -1;

  long distancia = duracion \* 0.034 / 2;

  Serial.print("Distancia: ");

  Serial.print(distancia);

  Serial.println(" cm");

  return distancia;

}

// Apaga todos los motores

void apagarTodo() {

  motor1.run(RELEASE);

  motor2.run(RELEASE);

  motor3.run(RELEASE);

  bomba.run(RELEASE);

  sistemaActivo = false;

}