## **Código de Arduino**

#include <LiquidCrystal.h>

#include <Servo.h>

#include <SoftwareSerial.h>

// ---------- Configuración de Pines ---------- //

// LCD en modo paralelo: RS, E, D4, D5, D6, D7

LiquidCrystal lcd(2, 3, 4, 5, 6, 7); // Usando los pines especificados en la tabla

// Sensor MQ6

const int gasAnalogPin = A0; // Lectura analógica

const int gasDigitalPin = 11; // Lectura digital (D0)

// Actuadores

const int ledPin = 13; // LED de alarma

const int servoPin = 9; // Servomotor

// ---------- Configuración de Bluetooth ---------- //

#define BT\_RX 8 // RX del Arduino ← TX del Bluetooth

#define BT\_TX 10 // TX del Arduino → RX del Bluetooth (con divisor de voltaje)

SoftwareSerial bluetooth(BT\_RX, BT\_TX);

// ---------- Objetos y variables ---------- //

Servo miServo;

const int umbralGas = 300; // Umbral de gas (ajustable)

String receivedData = ""; // Para almacenar los datos recibidos desde Bluetooth

void setup() {

// Comunicación serial para monitoreo

Serial.begin(9600);

bluetooth.begin(9600); // Comunicación con el módulo Bluetooth

// Configurar pines

pinMode(gasDigitalPin, INPUT);

pinMode(ledPin, OUTPUT);

// Inicializar servo

miServo.attach(servoPin);

miServo.write(0); // Ventana cerrada

// Inicializar LCD

lcd.begin(16, 2);

lcd.setCursor(0, 0);

lcd.print("Inicializando...");

delay(2000);

}

void loop() {

// Leer valores del sensor MQ6

int valorAnalogico = analogRead(gasAnalogPin);

int valorDigital = digitalRead(gasDigitalPin);

// Mostrar valor en LCD

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Gas: ");

lcd.print(valorAnalogico);

if (valorAnalogico > umbralGas || valorDigital == LOW) {

// Gas detectado: activar alarma y abrir ventana

digitalWrite(ledPin, HIGH);

miServo.write(90); // Abrir ventana

lcd.setCursor(0, 1);

lcd.print("¡Alerta de gas!");

} else {

// No hay gas: apagar alarma y cerrar ventana

digitalWrite(ledPin, LOW);

miServo.write(0); // Cerrar ventana

lcd.setCursor(0, 1);

lcd.print("Sin gas.");

}

// Mostrar en monitor serial

Serial.print("Valor MQ6 (A0): ");

Serial.print(valorAnalogico);

Serial.print(" | D0: ");

Serial.println(valorDigital);

// ---------- Comandos Bluetooth ---------- //

if (bluetooth.available()) {

char c = bluetooth.read();

if (c == '\n') {

processBluetoothCommand(receivedData); // Procesar comando cuando se recibe un salto de línea

receivedData = ""; // Limpiar la variable para nuevos datos

} else {

receivedData += c; // Acumular caracteres recibidos

}

}

delay(1000); // Espera entre lecturas

}

// Función para procesar los comandos recibidos por Bluetooth

void processBluetoothCommand(String command) {

Serial.print("Comando recibido: ");

Serial.println(command);

// Comando para abrir la ventana

if (command == "abrir") {

miServo.write(90); // Mover el servo a 90° (ventana abierta)

bluetooth.println("Ventana abierta.");

}

// Comando para cerrar la ventana

if (command == "cerrar") {

miServo.write(0); // Mover el servo a 0° (ventana cerrada)

bluetooth.println("Ventana cerrada.");

}

// Puedes agregar más comandos si es necesario

}