NAVODAYA INSTITUTE OF TECHNOLOGY, RAICHUR

DEPARMENT OF COMPUTER SCIENCE & ENGINEERING

IOT Lab

Program - 08

08 Develop a program to detect the gas leakage in the surrounding environment

🔧 Components

- Arduino Uno/Nano
- MQ gas sensor module (MQ-2 is a good default)
- Buzzer **or** LED + 220 Ω **or** Relay module (to drive fan/alarm)
- Breadboard + jumper wires
- (Optional) 0.1 µF cap between A0 and GND for extra noise filtering

⚠️ MQ sensors have a heater and draw ~150–200 mA at 5 V. Power the sensor from **5 V** (not 3.3 V) and avoid weak USB hubs.

🔌 Wiring

- **Sensor module**: VCC → 5V, GND → GND, **AO → A0** (analog), DO (optional) → D4
- **Alarm**: D7 → buzzer/LED(or relay IN) → GND (LED needs a 220 Ω series resistor)
- (Relay module: VCC → 5V, GND → GND; switch your external load via COM/NO)

🧪 Steps to Do the Experiment

1. **Assemble the circuit** as above.
2. **Warm-up** the MQ sensor: power it for **5–10 minutes** (for quick demos). For best stability, initial "burn-in" can be **24–48 h** when new.
3. **Upload** the Arduino sketch below and open **Serial Monitor @ 115200**.
4. **Observe baseline** analog readings in clean air for ~1–2 minutes; note the average (e.g., 180–260).
5. **Calibrate threshold**: briefly waft a tiny bit of test gas (e.g., an *unlit* lighter near the intake) and note the spike (e.g., 420–700). Set THRESHOLD midway between baseline and spike.

6. **Test alarm**: approach with the same source → buzzer/relay should activate; remove source → alarm clears after hold time.
7. (Optional) Use the **DO** pin and its onboard potentiometer as a hardware threshold; the code supports both analog and digital modes.

💻 Arduino Program (analog smoothing + hysteresis + alarm hold)

```
/* Gas Leak Detector (MQ-2/MQ-5/MQ-135) with Arduino
   - Sensor AO -> A0 (analog), DO -> D4 (optional)
   - Alarm (buzzer/LED/relay) -> D7
   - Serial at 115200 for live logs
*/

const uint8_t PIN_GAS_AO = A0;
const uint8_t PIN_GAS_DO = 4;    // optional digital comparator from module
const uint8_t PIN_ALARM  = 7;

const uint16_t SAMPLES = 20;     // moving average window
const uint16_t SAMPLE_DELAY_MS = 20;

int THRESHOLD = 380;             // set after calibration
int HYST = 20;                   // hysteresis to avoid chatter (+/-)
unsigned long ALARM_HOLD_MS = 8000;

unsigned long alarmUntil = 0;
bool alarmState = false;

uint16_t readSmoothedAnalog() {
  uint32_t sum = 0;
  for (uint16_t i = 0; i < SAMPLES; i++) {
    sum += analogRead(PIN_GAS_AO);
    delay(SAMPLE_DELAY_MS);
  }
  return sum / SAMPLES;
}

void setAlarm(bool on) {
  alarmState = on;
  digitalWrite(PIN_ALARM, on ? HIGH : LOW);
}

void setup() {
  pinMode(PIN_ALARM, OUTPUT);
  pinMode(PIN_GAS_DO, INPUT); // safe even if unused
  setAlarm(false);
  Serial.begin(115200);
  delay(500);

  Serial.println(F("Gas Leak Detector - MQ Sensor"));
  Serial.println(F("Warm up the sensor for a few minutes before trusting readings..."));
}

void loop() {
  // Read analog with smoothing
  uint16_t a = readSmoothedAnalog();
```

```
  // Optional digital comparator from module (active LOW on many boards)
  int doVal = digitalRead(PIN_GAS_DO);

  // Hysteresis logic around THRESHOLD
  bool tripAnalog = false;
  if (!alarmState && a >= (THRESHOLD + HYST)) tripAnalog = true;
  if (alarmState && a >= (THRESHOLD - HYST)) tripAnalog = true;

  // Combine analog decision with optional digital comparator
  bool trip = tripAnalog || (doVal == LOW); // adjust if your board's DO is active HIGH

  unsigned long now = millis();
  if (trip) {
    alarmUntil = now + ALARM_HOLD_MS; // latch for a bit
  }
  setAlarm(now < alarmUntil);

  // Logs
  Serial.print(F("Analog: ")); Serial.print(a);
  Serial.print(F("  DO: ")); Serial.print(doVal);
  Serial.print(F("  Threshold: ")); Serial.print(THRESHOLD);
  Serial.print(F("  Alarm: ")); Serial.println(alarmState ? F("ON") : F("off"));
}
```

🛠️ Calibration Tips

- After warm-up in clean air, note baseline (e.g., **220**).
- Expose briefly to target gas and note peak (e.g., **600**).
- Set THRESHOLD ≈ **(baseline + peak) / 2** (here, ~410).
- Adjust HYST (10–40) to reduce chattering. Increase SAMPLES for steadier readings (slower but smoother).