EXPERIMENT 13 - Arduino Project

Aim: To Implement air quality monitoring and smart air purifier using Arduino microcontroller.

THEORY:

AIR QUALITY MONITORING SYSTEM WITH AUTOMATED PURIFIER CONTROL

Component	Quantity	Purpose	Key Details
Arduino Uno	1	Main microcontroller	Controls sensors, LCD, LED, and buzzer
16×2 LCD Display	1	Shows AQI, status, and runtime	Uses 4-bit mode (RS, EN, D4-D7)
10kΩ Potentiometer	1	Simulates air quality sensor	Outputs 0–5V (mapped to AQI 0–500)
LED (Any Color)	1	Visual indicator for air purifier	Connected via 220Ω resistor to Pin 13
Piezo Buzzer	1	Sound alerts for purifier ON/OFF	Connected to Pin 8 (300ms beeps)
220Ω Resistor	1	Limits current for LED	Prevents LED burnout
Jumper Wires	-	Connects components	---

1. **System Overview**
   This microprocessor-based air quality monitoring system continuously measures airborne pollutants using an electrochemical sensor and automatically controls an air purifier based on predefined air quality thresholds. The Arduino microcontroller processes analog sensor data, classifies air quality into health risk categories, and provides both visual and auditory feedback to users. The system implements a closed-loop control mechanism where environmental measurements directly determine the purifier's operational state.

2. Sensor Principle and Signal Conditioning

The analog air quality sensor operates on electrochemical principles, generating a voltage output proportional to the concentration of target pollutants (typically volatile organic compounds and particulate matter). As pollutant concentration increases, the sensor's output voltage rises linearly. The microcontroller's 10-bit ADC converter samples this analog voltage (0-5V range mapped to 0-1023 digital values), which is then scaled to a standardized Air Quality Index (AQI) range of 0-500 through linear interpolation. This digital signal processing ensures compatibility with international air quality standards while accounting for the sensor's specific response characteristics.

3. Air Quality Classification Algorithm

The system implements a multi-tier classification scheme based on established environmental protection guidelines. The microcontroller compares the computed AQI against five critical thresholds: Good (<50), Moderate (50-100), Unhealthy for Sensitive Groups (100-150), Unhealthy (150-200), and Hazardous (>200). Each classification triggers corresponding system responses - the "Unhealthy" threshold (150 AQI) activates the purifier, while better air quality conditions deactivate it. This hysteresis-based control prevents rapid cycling of the purification system during borderline conditions.

4. Visual Feedback System

A 16x2 character LCD displays real-time air quality parameters through multiple visual elements. The primary display shows numerical AQI values alongside color-coded quality descriptors. A custom-designed bar-graph character provides instantaneous visual indication of pollution levels through height-proportional bars. The system also implements a runtime counter that displays the purifier's cumulative operating duration in seconds, allowing users to assess system utilization patterns. Display refresh occurs at 500ms intervals to balance information currency with processor load.

5. Auditory Feedback Mechanism

The system employs frequency-modulated sound alerts to signal state transitions. Purifier activation generates a 1000Hz tone (300ms duration), while deactivation produces an 800Hz tone (same duration). These distinct auditory cues provide immediate system status awareness without requiring visual monitoring. During initialization, a two-tone ascending sequence (1500Hz→2000Hz) confirms proper system startup. All audio feedback follows the principle of minimal intrusion, with brief durations that don't mask environmental sounds.

6. Breathing LED Control Algorithm

The purifier's status indicator implements a dynamic pulse-width modulation (PWM) pattern that simulates natural respiration. Using a sinusoidal brightness modulation function:

$\text{Brightness} = 128 + 127 \times \sin(2\pi t/2000)$

where t represents milliseconds in the 2000ms cycle. This produces a smooth, periodic intensity variation that is both visually distinctive and power-efficient. The effect continues throughout purifier operation, providing constant visual confirmation of system activity while consuming less power than a static full-brightness LED.

7. Control System Architecture

The firmware implements a state-machine architecture with three primary operational states:

Monitoring State: Continuously samples sensor input and updates displays
Purification State: Activates both the purifier and visual/auditory indicators
Idle State: Maintains display updates during good air quality conditions
State transitions occur automatically based on sensor readings, with all control decisions executing in the main program loop without blocking delays. The system achieves real-time responsiveness through millis()-based non-blocking timing for all periodic functions.

8. Power Management Considerations

The design optimizes energy efficiency through several techniques: selective peripheral activation (purifier only engages when needed), display backlight management, and processor sleep modes between sensor readings. The breathing LED effect itself reduces power consumption compared to constant illumination, while the buzzer's short activation periods minimize energy use for auditory feedback.

9. Applications and Significance

This system finds practical application in residential air quality management, workplace environmental monitoring, and educational demonstrations of embedded control systems. Its automated response mechanism provides continuous environmental regulation without user intervention, while the comprehensive feedback systems maintain user awareness. The project demonstrates effective integration of multiple microcontroller peripherals (ADC, PWM, digital I/O) into a cohesive monitoring and control solution.

Code Flow Overview

1. **Initialization (setup)**
   - Configures LCD, pins (sensor, button, buzzer, LED)
   - Shows startup message with sound
   - Prepares custom LCD character for air quality indicator
2. **Main Loop (repeats continuously)**
   - Read sensor → Classify air quality → Check button → Control purifier → Update display
   - Runs every 20ms (non-blocking delay)

Key Functions & Logic

1. Sensor Reading & Classification

- Takes analog reading from air quality sensor (0-1023)
- Maps value to AQI scale (0-500)
- Categorizes into 6 levels:
  - Good (<50)
  - Moderate (50-100)
  - Unhealthy-SG (100-150)
  - Unhealthy (150-200)
  - Very Unhealthy (200-300)
  - Hazardous (300+)

2. Purifier Control

- **Auto ON** when AQI ≥ 150 (default threshold)
- **Auto OFF** when AQI < 150
- Tracks runtime (seconds) while active

- Plays distinct sounds when turning ON/OFF

3. Visual Feedback

- **LCD Display Shows**:
    - Real-time AQI number
    - Air quality status text
    - Purifier runtime counter
    - Custom bar-graph indicator (shows pollution level)
- **LED Feedback**:
    - Breathing effect (brightness pulsates) when purifier is ON
    - Fully OFF when purifier is inactive

4. Auditory Feedback

- **Activation Sound**: High-pitch beep (1000Hz)
- **Deactivation Sound**: Low-pitch beep (800Hz)
- **Startup Sound**: Two-tone chime