AeroLens

Jana Godieh 0223457 (Team Leader)

Leen Almasarweh 0222209

Dana Abu-Alruz 0224982

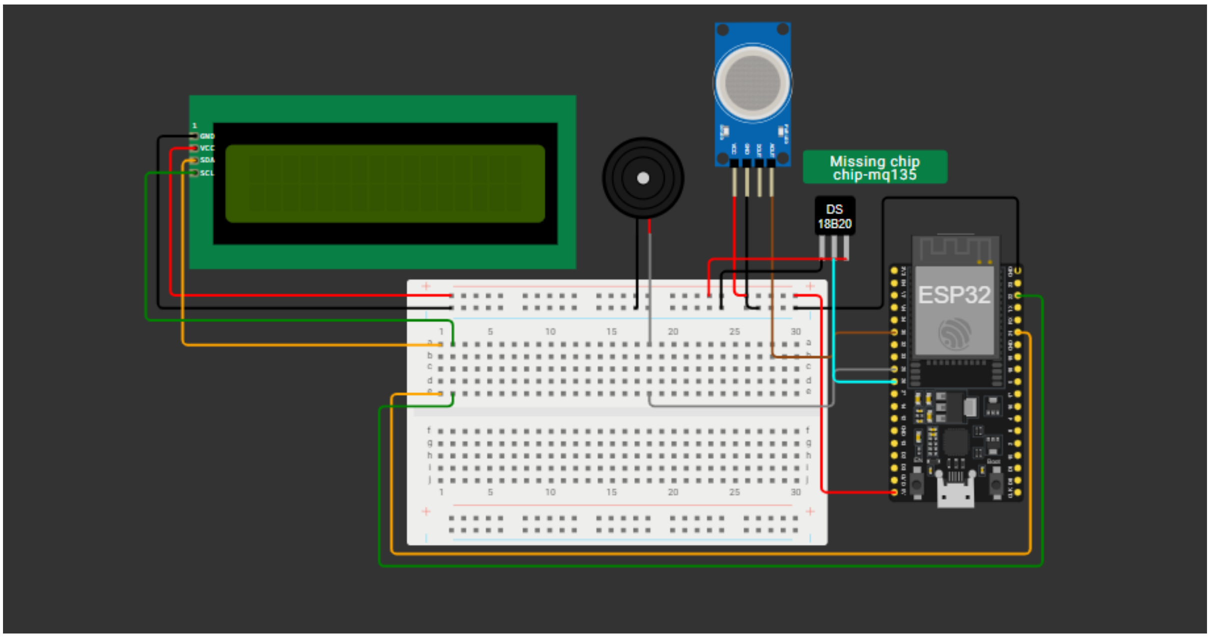
1 Summary of Deliverable1

1.1 Wokwi network

For the simulation phase of this project, Wokwi was employed to replicate the network environment.

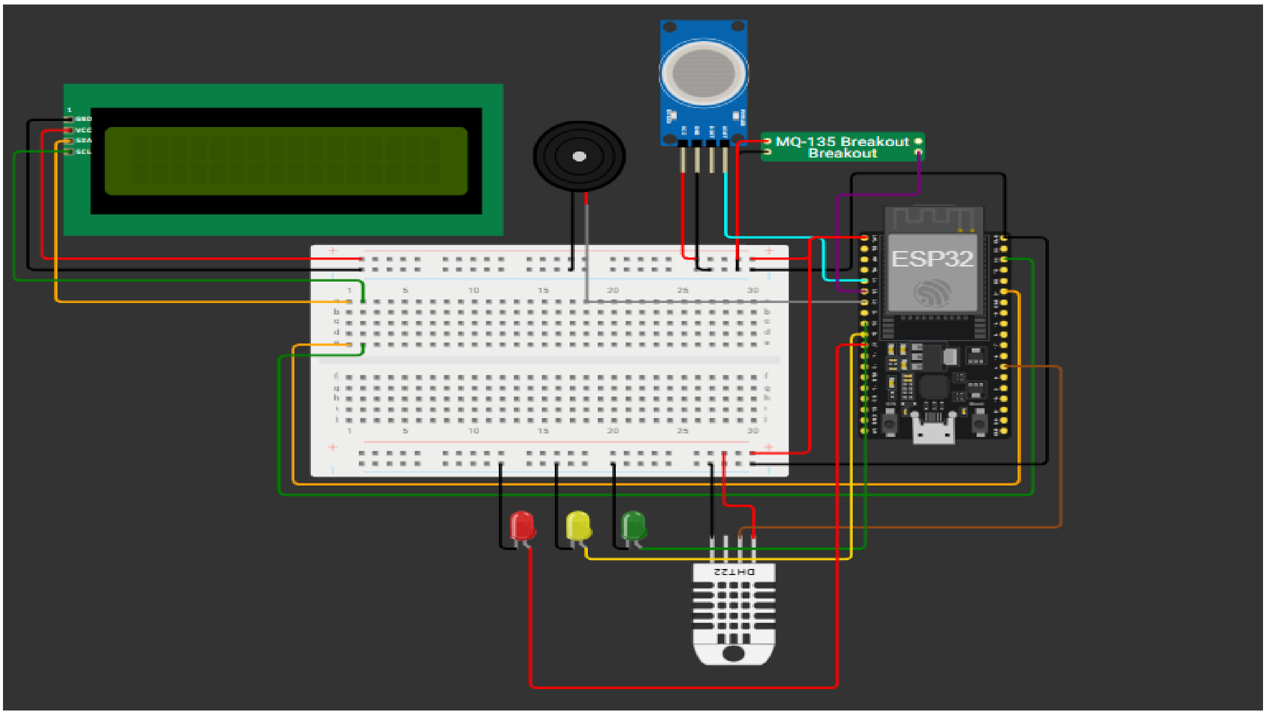
Our first network configuration includes an ESP32, buzzer, a temperature sensor, MQ2, and MQ135. Because certain parts were unavailable in the Wokwi library, we used these sensors as temporary placeholders to validate the network's logic.

The first network link: https://wokwi.com/projects/305569599398609473



To enhance the system’s capabilities, we integrated a flame sensor, a DHT humidity sensor, and a tri-color LED indicator (Red, Yellow, and Green) to provide real-time status updates.

The Final network link: https://wokwi.com/projects/452516632139075581



1.2 How Sensor Collect Data

Our system collects data by interfacing various sensors with the ESP32 microcontroller. The gas and flame sensors monitor the environment for hazards by sending signal voltages to the controller. At the same time, the humidity sensor provides digital data packets regarding ambient conditions. The ESP32 acts as the central hub, continuously polling these sensors, analyzing the data, and determining if the environmental parameters fall within safe limits

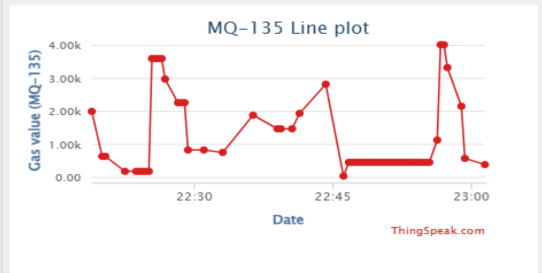
1.3 Thingspeak visualizations

1. Real-Time Monitoring (Line Plots): The first three charts are Time-Series Line Plots. Their purpose is to track how specific environmental variables change over a period of time

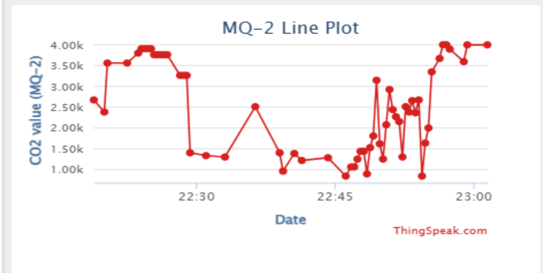
2. Relationship Analysis (Scatter Plots): These charts are used to find correlations between two different variables. They help you determine if one factor influences another.

3. Comparative Trends (Dual-Axis Plots):The final two charts use a Dual-Axis format, which is excellent for comparing the behavior of two different datasets on a single timeline for example overlaying two gas sensors on the same graph, we can easily see if a specific event triggered both sensors at the exact same moment.

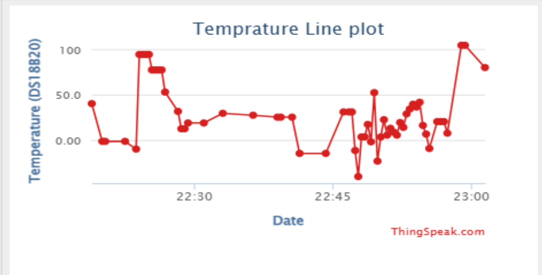
Field 1 Chart



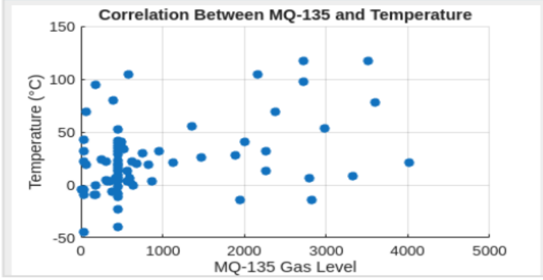
Field 2 Chart



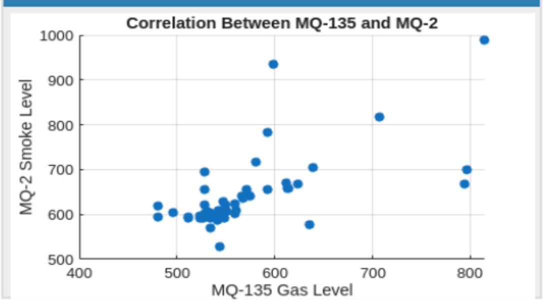
Field 3 Chart



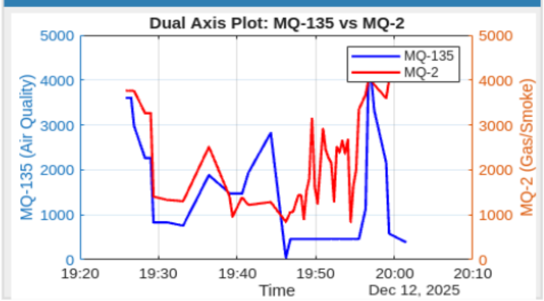
scatter plot of MQ-2 and Temprature



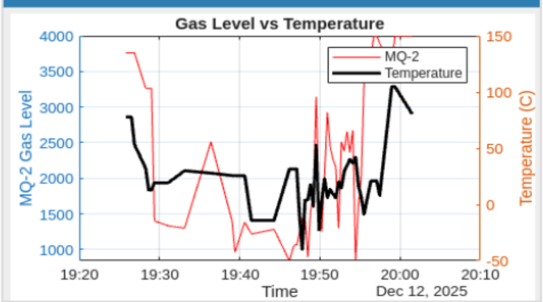
scatter plot of MQ-135 and MQ-2



Dual-axis plot of MQ-2 and MQ-135



Dual-axis plot of MQ-2 and Temperature



2 Summary of Deliverable2

2.1 description of the dataset

For the second phase of our project we choose Gas Sensors Measurements Dataset (Kumar) from github,The data includes readings from various gas sensors (MQ2, MQ3, MQ5, MQ6, MQ7, MQ8, MQ135) along with a serial number and the type of gas detected. The link of the dataset:
https://github.com/TakMashhido/Gas-Sensors-Measurements-Dataset/tree/main

2.2 description of the visualizations

1. Time-Series Line Plots: These are used to track real-time fluctuations in environmental variables such as temperature and specific gas concentrations (MQ-2, MQ-3, MQ-6, and MQ-135) over a defined period. Their primary purpose is to identify sudden "spikes" that may indicate a gas leak or fire.

2. Correlation Scatter Plots: These charts plot two variables against each other (e.g., MQ-135 Gas Level vs. Temperature) to determine if a relationship exists. This is essential for understanding if temperature changes are influencing sensor accuracy or if gas levels rise independently.

3. Dual-Axis and Overlay Plots: By plotting two different datasets (like Gas Level and Temperature) on a single timeline with two Y-axes, the system allows for direct comparison of trends. This helps distinguish between different types of emergencies, such as high heat without gas (overheating) versus high heat with gas (fire).

4. Sensor Correlation Heatmap: This visualization provides a statistical overview of how closely the different MQ-series sensors mirror each other. A high correlation (close to 1.00) indicates that multiple sensors are detecting the same environmental change, increasing the reliability of the system's alerts.

5. Bar Plot Status Analysis: This chart categorizes data points into "Normal" or "Warning/Alert" states. It justifies the system's performance by showing the frequency of dangerous conditions relative to safe operating levels.

2.3 description of the action

To ensure real-time safety and rapid response, the system is integrated with an automated email notification service via the ThingSpeak Alerts API. When any connected sensor—such as the MQ6 or MQ135—exceeds a pre-defined safety threshold, the system triggers a state change from Normal (Green) to either Warning (Yellow) or High Alert (Red). For instance, if the MQ6 sensor detects gas levels reaching a critical raw value, a 'High Alert' email is immediately dispatched to the user. These notifications include a detailed snapshot of all current environmental data, including temperature, humidity, and flame sensor status, enabling the user to take immediate corrective action even when away from the physical setup


MathWorks Logo

Alert: ⚠️ WARNING (YELLOW): Environmental Monitor

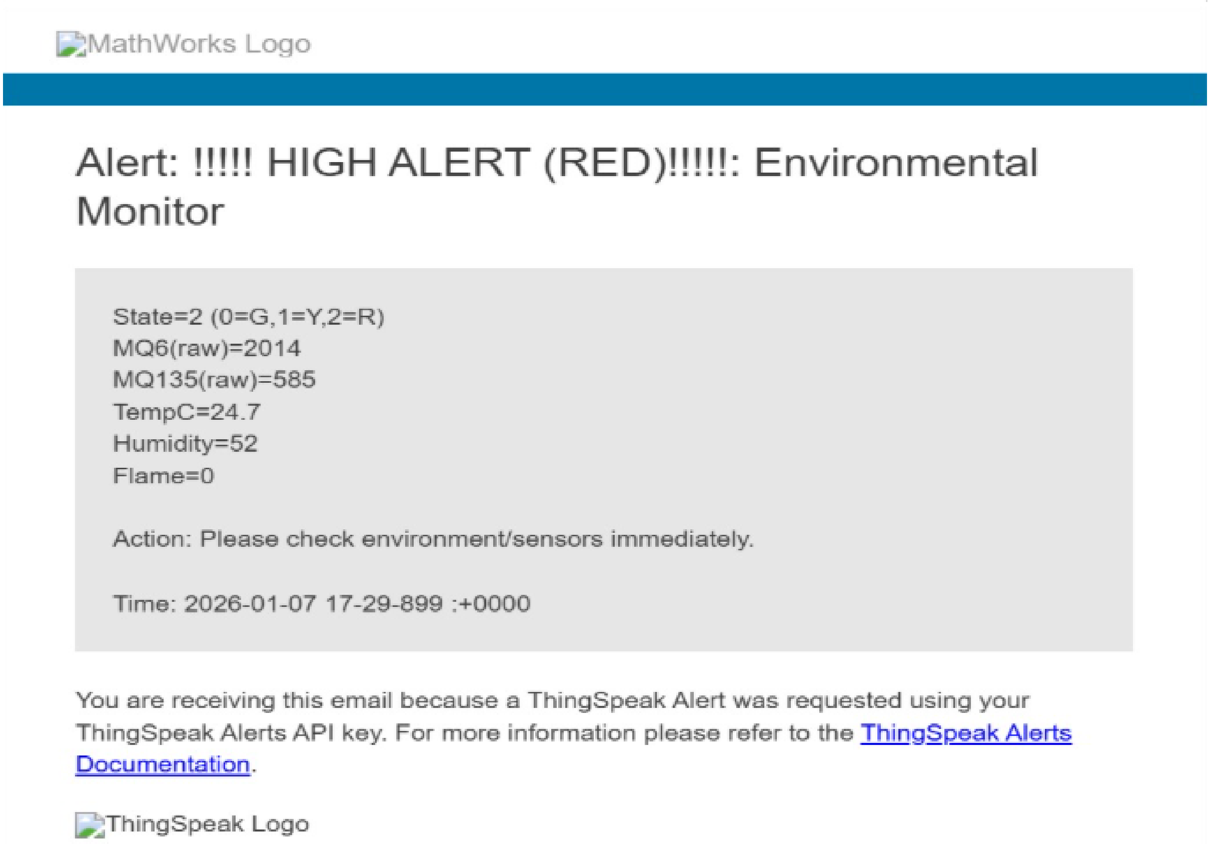
State=1 (0=G,1=Y,2=R)
MQ6(raw)=707
MQ135(raw)=817
TempC=24
Humidity=50
Flame=0

Action: Please check environment/sensors immediately.

Time: 2026-01-07 18-53-044 :+0000

You are receiving this email because a ThingSpeak Alert was requested using your ThingSpeak Alerts API key. For more information please refer to the ThingSpeak Alerts Documentation.

ThingSpeak Logo


MathWorks Logo

Alert: !!!!! HIGH ALERT (RED)!!!!!: Environmental Monitor

State=2 (0=G,1=Y,2=R)
MQ6(raw)=2014
MQ135(raw)=585
TempC=24.7
Humidity=52
Flame=0

Action: Please check environment/sensors immediately.

Time: 2026-01-07 17-29-899 :+0000

You are receiving this email because a ThingSpeak Alert was requested using your ThingSpeak Alerts API key. For more information please refer to the ThingSpeak Alerts Documentation.

ThingSpeak Logo

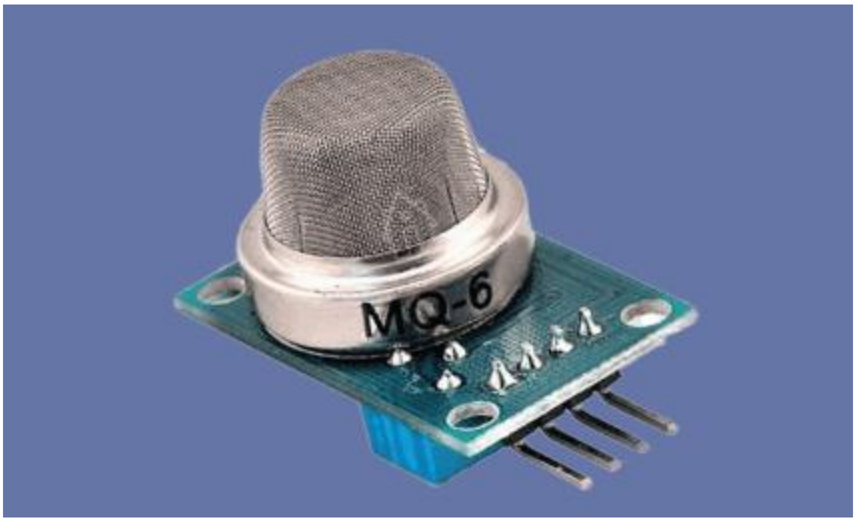
2.4 How thingspeak channel support our project

The ThingSpeak IoT platform serves as the primary backbone for this air pollution monitoring system, providing a centralized infrastructure for data aggregation, real-time analysis, and remote accessibility. By utilizing Channel 3209217, the system simultaneously logs data from multiple sources, including various gas sensors and temperature readings, through a robust multi-field data intake architecture that allows for parallel monitoring environmental variables. This centralized hub features an integrated visualization engine that automatically converts raw numeric values into time-series line charts, providing a critical tool for identifying immediate anomalies such as gas leaks or sudden temperature spikes.

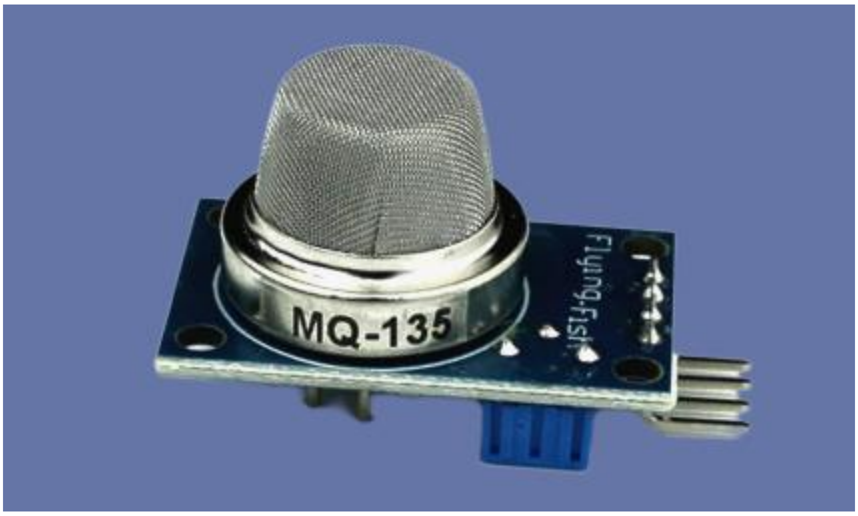
3 Summary of Deliverable3

For the final phase of our project we implemented the following hardware :

- Sensors :
  - MQ6

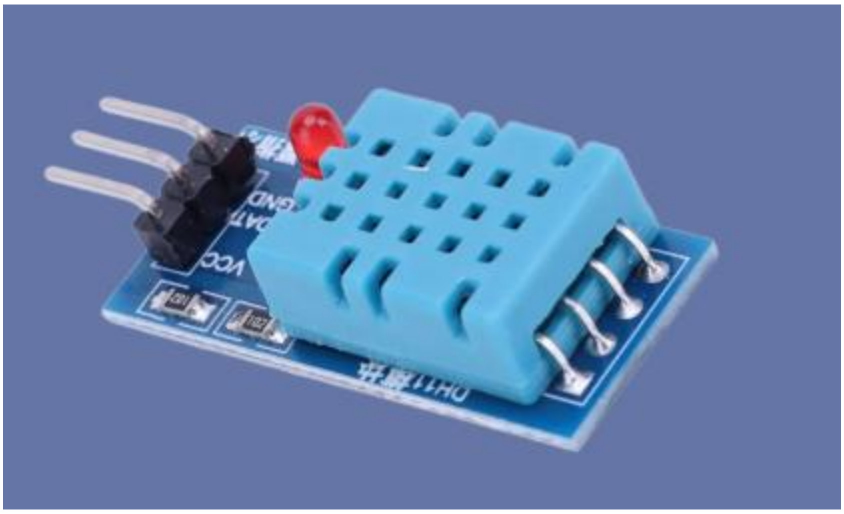


  - MQ135



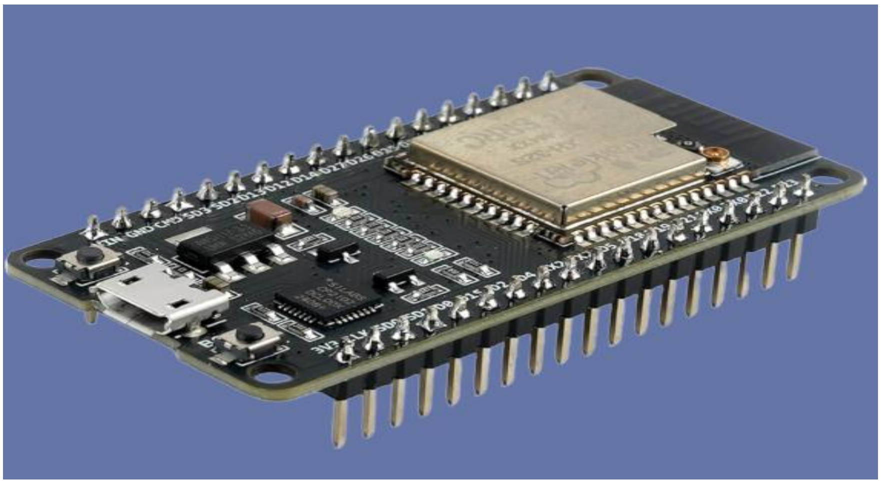
➢ Flame sensor

➢ Humidity sensor



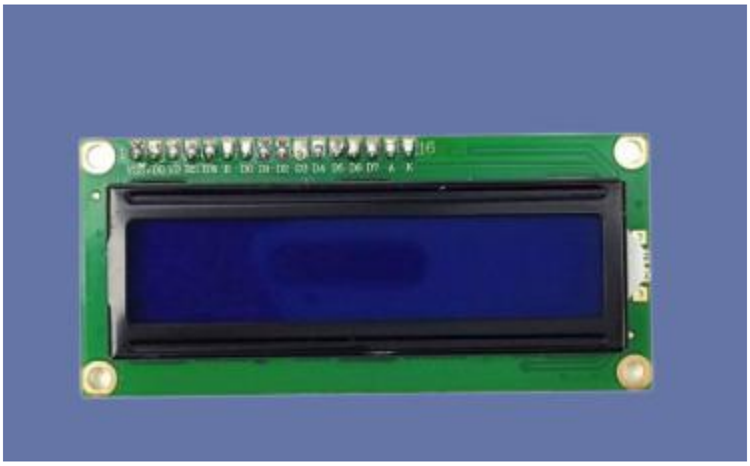
❖ Microcontroller

➢ ESP32

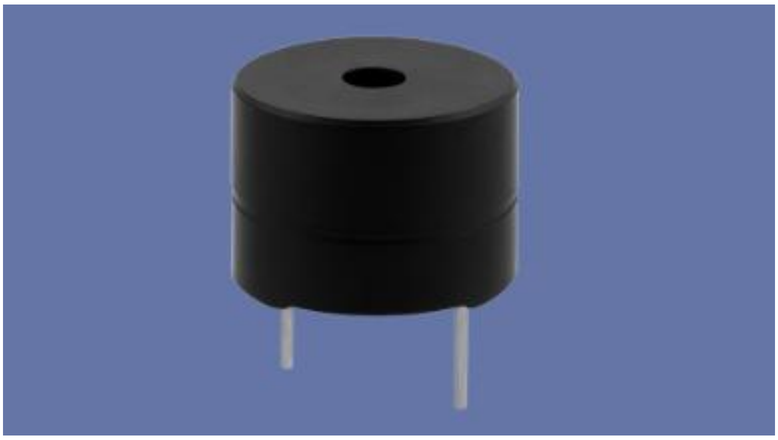


- Actuators
  - Traffic Light LED Module.

  - LCD



  - Buzzer



Works Cited

Wokwi - World's most advanced ESP32 Simulator, https://wokwi.com/. Accessed 8 January 2026.

IoT Analytics - ThingSpeak Internet of Things, https://thingspeak.mathworks.com/. Accessed 8 January 2026.

Kumar, Koustav. “Gas Sensor Measurement.” *github*, 2024, https://github.com/TakMashhido/Gas-Sensors-Measurements-Dataset/tree/main. Accessed 8 9 2024.