Kammavari Sangham (R) - 1952

K.S.INSTITUTE OF TECHNOLOGY

(Affiliated to VTU, Belgaum. Approved by AICTE, New Delhi)

No.14, Raghuvanahalli, Kanakapura Main Road, Bengaluru - 560109

Phone : 080-28435722 / 24, Fax : 080-28435723, Email : principal.ksit@gmail.com, Website : www.ksit.ac.in

Gas Leakage Detector

Department of Computer Science & Design

Company Name: Cranes Varsity

Domain: IOT And Detection Of Gas Leakage

Presented By:

Kusum Pakira	-1KS21CG026
Shivani Adhikari	-1KS21CG042
Arun A	-1KS22CG400
Chandrasekar A	-1KS22CG401
Purushotham V Mitti	-1KS22CG404

Under the Guidance of:

Dr. Deepa S R, Head of Department, Computer Science & Design, KSIT

Dr. Surekha Byakod, Assistant Professor, Computer Science & Design Department, KSIT

AGENDA

1. Company Profile
2. Internship Objective
3. Internship Tasks
4. Skills Utilized
5. Project
6. Snapshots
7. Internship Impact
8. Conclusion

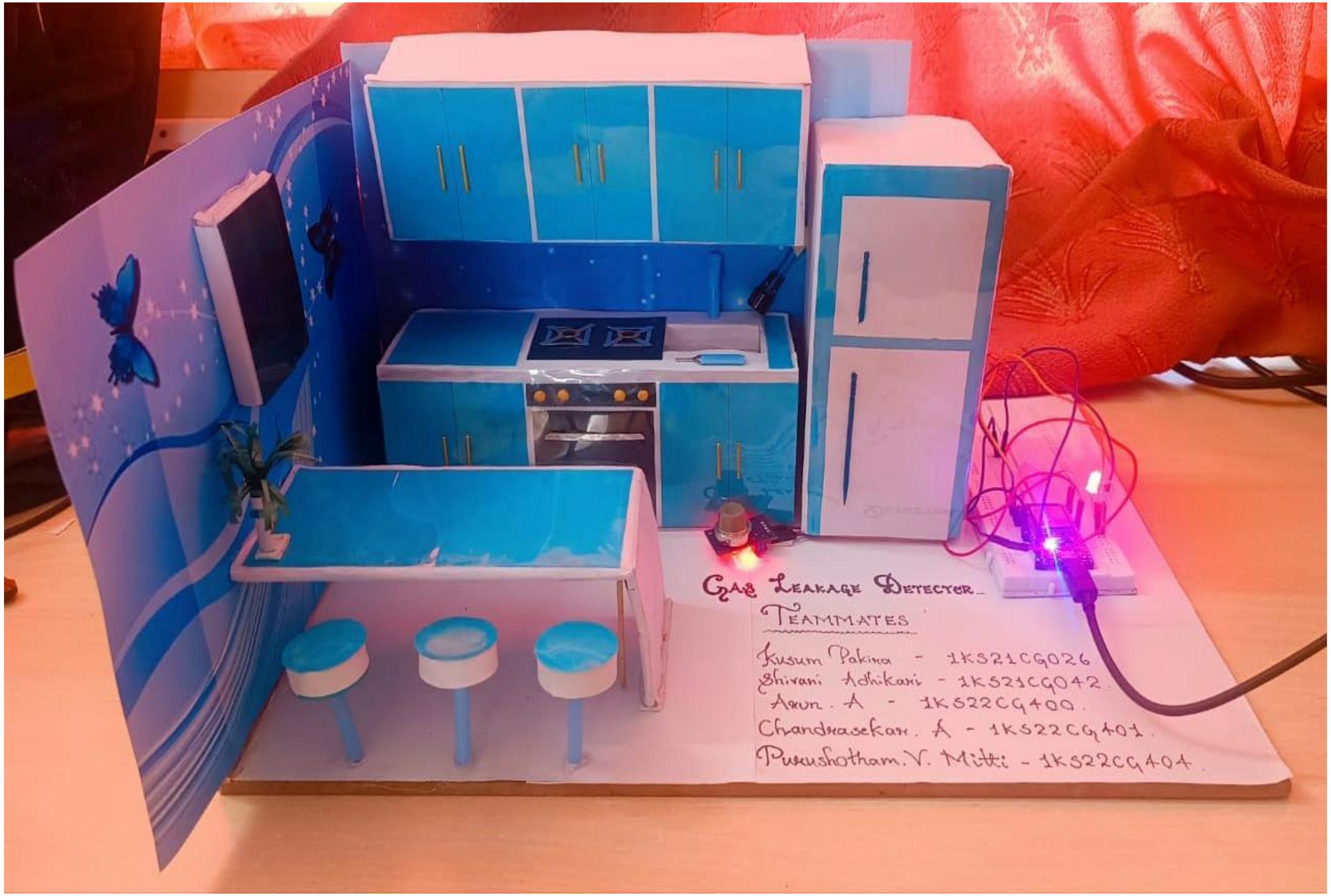
Company Profile

Cranes Varsity, a division of Cranes Software International Ltd, has empowered professionals with a perfect blend of technology and education **since 1998**. As an EdTech platform, we offer high-impact, hands-on training to graduates, universities, working professionals and corporate and defense sectors, bridging the gap between technology, academia and industry.

CRANES Cranes Varsity 'Where Technology Meets Excellence'

- The 4-week IoT Internship Program by Cranes Varsity, designed to equip engineers with practical knowledge and skills in the exciting realm of Internet of Things (IoT). The Internship Program undergo training on Python, Sensors, and IOT concept.
- This Internship Program offers a focused and intensive learning experience, covering essential aspects such as IoT architecture, sensor integration, data analytics, cloud computing, and security. Participants will engage in hands-on exercises, real-world projects, and interactive sessions, gaining valuable insights into IoT technologies.
- Under the mentorship of experienced professionals, we worked on cutting-edge IoT projects, enhancing our skills in system design, prototyping, and implementation. This hands-on experience fostered our critical thinking, problem-solving abilities, and teamwork skills.

INTERNSHIP OBJECTIVE


Gas Leakage Detector
Teammates
Kusum Pakina - 1KS21CG026
Shivani Adhikari - 1KS21CG042
Arun . A - 1KS22CG400
Chandrasekar . A - 1KS22CG401
Purushotham.V. Mitti - 1KS22CG404

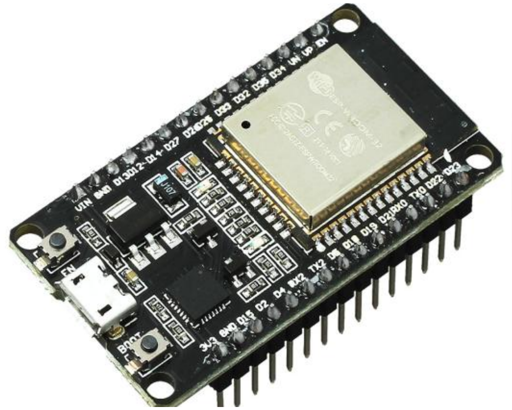
- Early Detection
- Safety Assurance
- Alarm System
- Continuous Monitoring
- Accuracy and Sensitivity
- Automatic Shutoff
- Data Logging
- Remote Notification
- Compliance with Standards
- User-Friendly Interface

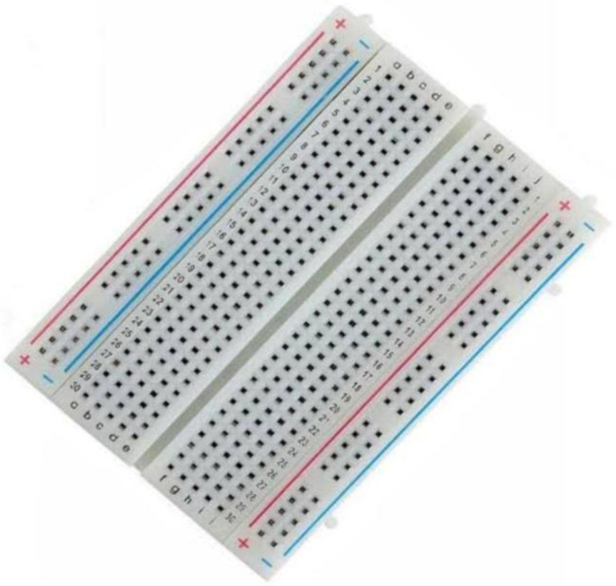
INTERNSHIP TASKS

- A gas leakage detector performs several critical tasks. Its primary function is to detect the presence of potentially hazardous gases such as natural gas, propane, and carbon monoxide in the air.
- Upon detecting these gases, it immediately alerts occupants through audible alarms, visual indicators, or connected smart devices, enabling prompt action to mitigate risks.
- The detector continuously monitors the environment for gas leaks, ensuring constant protection. In some advanced systems, it can also automatically shut off the gas supply to prevent escalation of the leak into a more dangerous situation.
- This proactive approach helps in preventing accidents, protecting health, and preserving property.

Project : Gas Leakage Detection

- Gas leakage detection addresses a significant problem faced by humans: the risk of undetected gas leaks.
- Without proper detection, leaks of gases such as natural gas, propane, and carbon monoxide can lead to severe consequences, including explosions, fires, and toxic poisoning.
- These gases are often colorless and odorless, making them difficult to detect without specialized equipment.
- The lack of early detection can result in delayed responses, putting lives at risk and causing extensive property damage.
- Therefore, the absence of effective gas leakage detection systems presents a serious safety hazard in both residential and commercial settings.



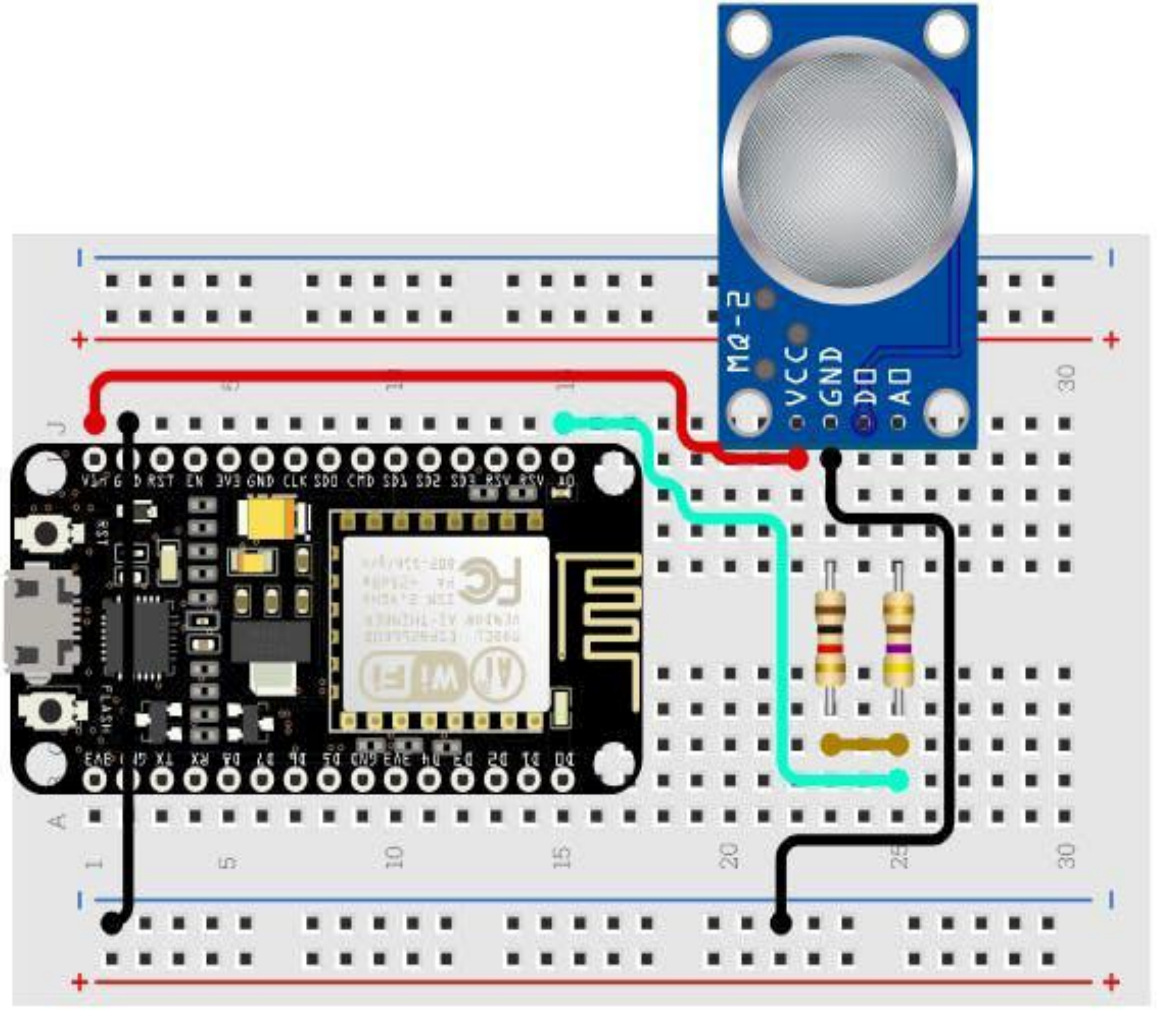


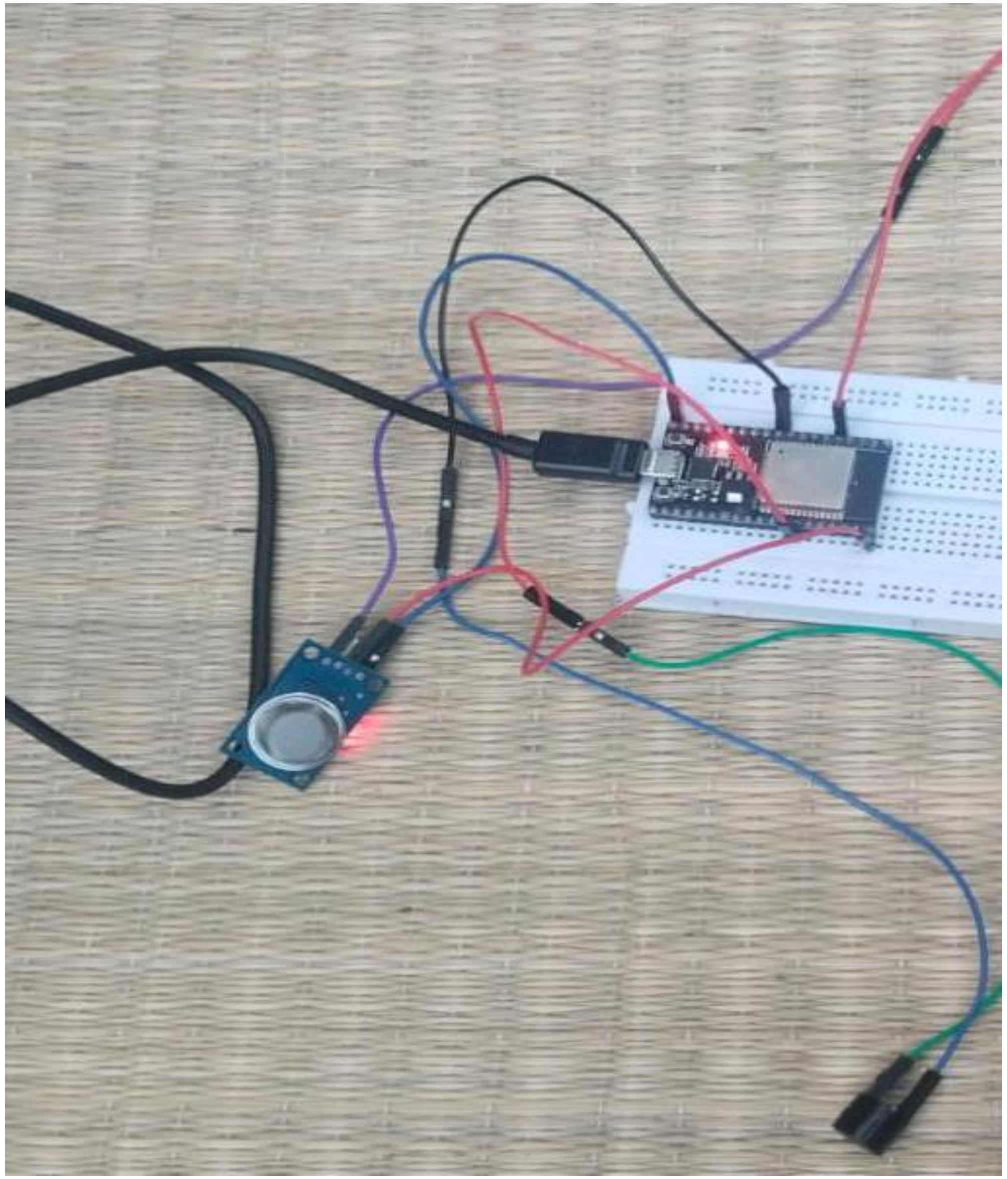
HARDWARE REQUIREMENTS

- **ESP32 Development Board**
- **Gas Sensor**
- **Buzzer**
- **LED Bulb**
- **Power Supply**
- **Connecting Wires**
- **Breadboard**
- **PCB (Printed Circuit Board)**
- **USB Cable**

SOFTWARE REQUIREMENTS

- **Thonny**
- **ESP32 Board Support**
- **ESP32 Libraries**
- **Cloud Service Account**
- **Wi-Fi Configuration**
- **Notification Configuration**
- **Safety Considerations in Code**
- **Continuous Monitoring Loop**

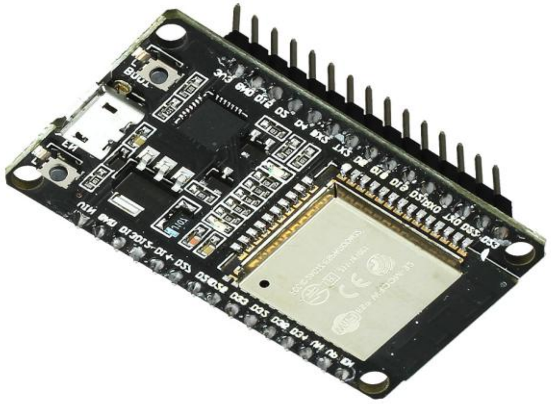




CONNECTION REQUIRED

- **Esp 32 3 ground pins 2 VCC pin 5v 3v**
- **Esp 32 Ground pin connected to Gas sensor Ground pin**
- **Gas sensor vss pin connected to ESP 32 vcc (5v)**
- **Gas sensor A0 pin connected to ESP32 Gpio pin (X)**
- **Gas sensor D0 pin empty**
- **Buzzer have 2 connection pins long and short pin**
- **Long pin Gpio pin,short pin Ground pin**
- **Long pin is connected to Gpio pin ESp32**
- **Short pin is connected to ground pin**
- **Led bulb pins one is positive , another one negetive**

Working:



1. **Programming the ESP32:**

Install the necessary libraries for the ESP32 and the gas sensor.

Write a program to initialize the sensor, read its analog output, and process the data.

Set a threshold value for the gas concentration. If the sensor's reading exceeds this value, it indicates a gas leak.

2. **Wi-Fi Notifications:**

Configure the ESP32 to connect to a Wi-Fi network.

Set up a web server or use MQTT to send alerts to a mobile app or a monitoring system.

Example: Sending an email or SMS via an online service like IFTTT.

(contd.)

3. Testing and Calibration:

Test the detector in a controlled environment with known gas concentrations.

Calibrate the sensor by adjusting the threshold value for accurate detection.

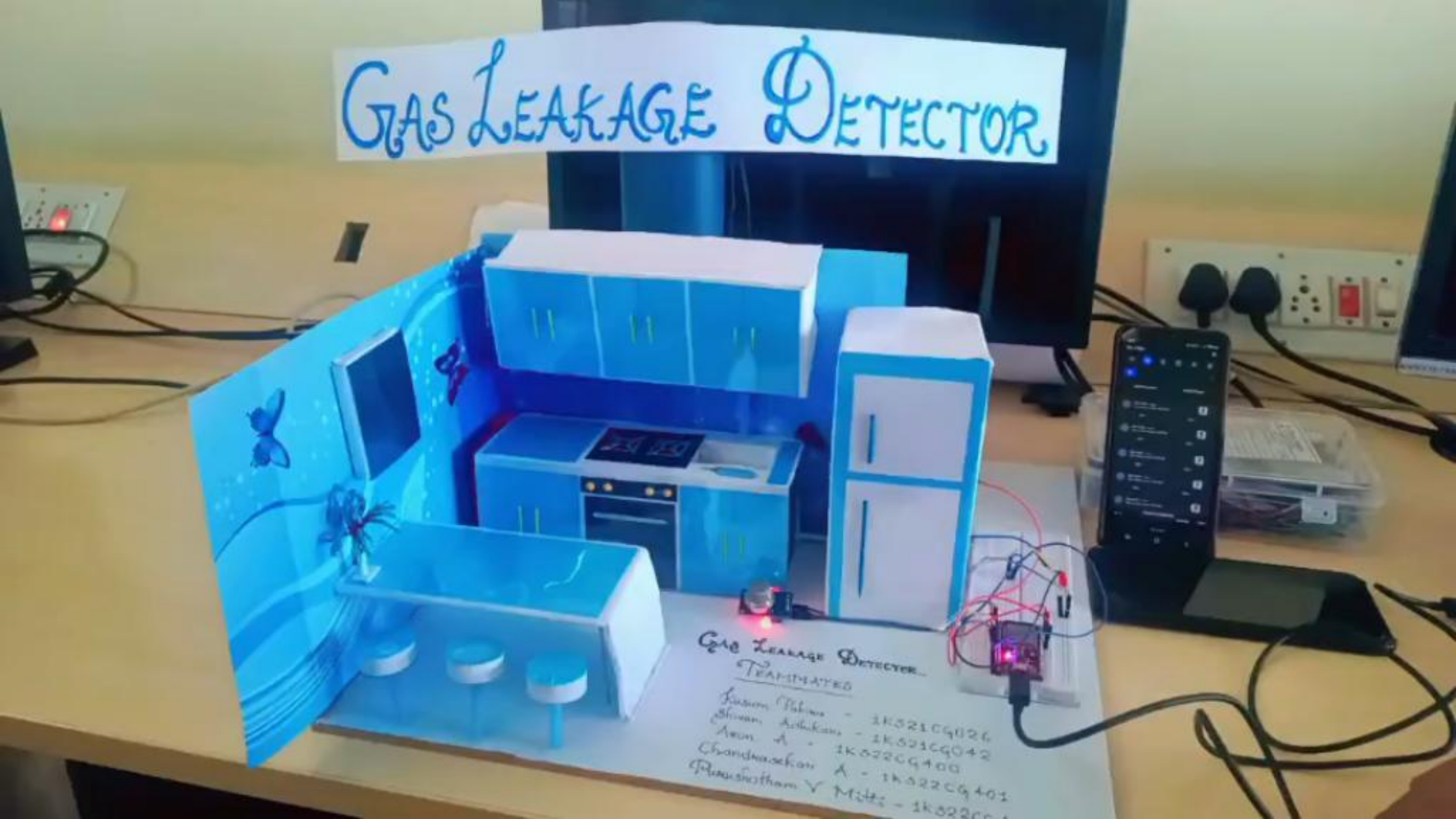
4. Deployment:

Install the detector in the desired location.

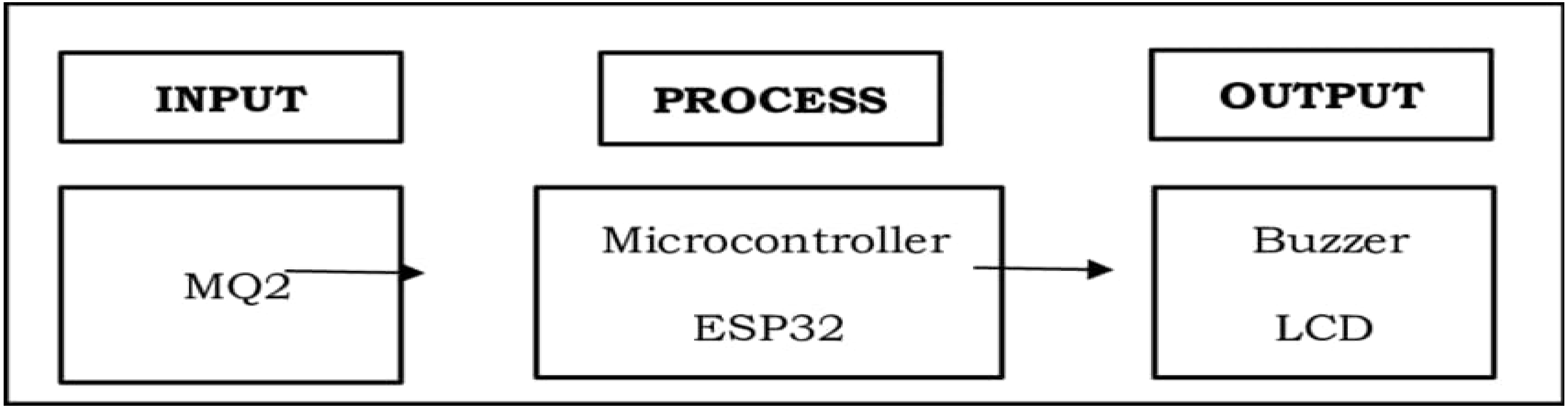
Ensure it is positioned where gas leaks are likely to occur and can be detected early.

Regularly maintain and test the system to ensure ongoing reliability.

This setup provides a cost-effective and efficient gas leakage detection system using the ESP32, with the added benefit of connectivity for remote monitoring and alerts.


GAS LEAKAGE DETECTOR
GAS LEAKAGE DETECTER..
TEAMMATES
Kusum Pabina - 1KS21CG026
Shivam Adhikari - 1KS21CG042
Arun A - 1KS22CG400
Chandrasekar A - 1KS22CG401
Purushotham V Mitti - 1KS22CG

Block Diagram



INPUT

PROCESS

OUTPUT

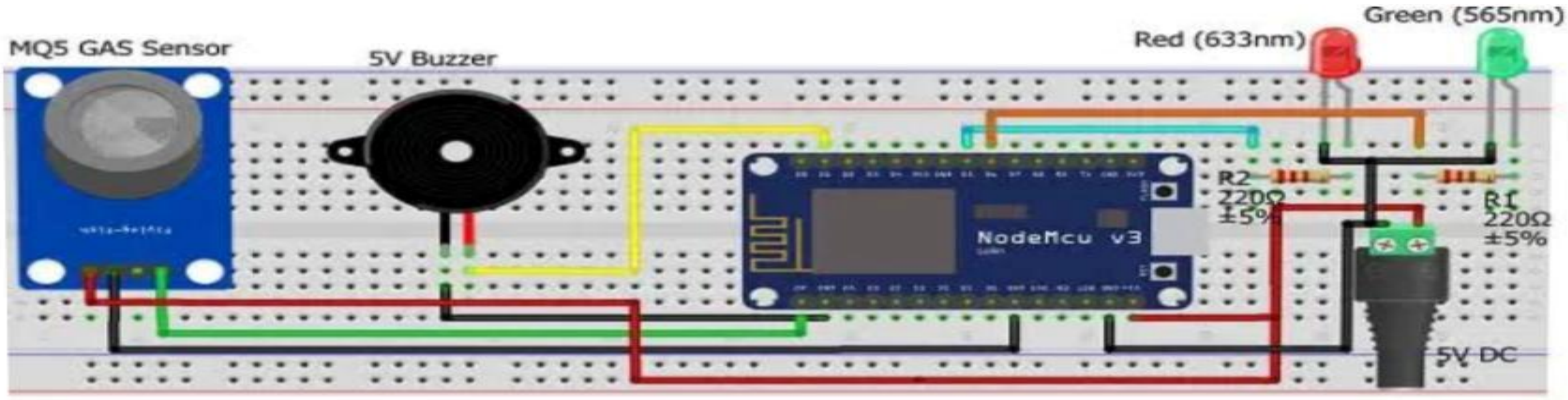
MQ2

Microcontroller

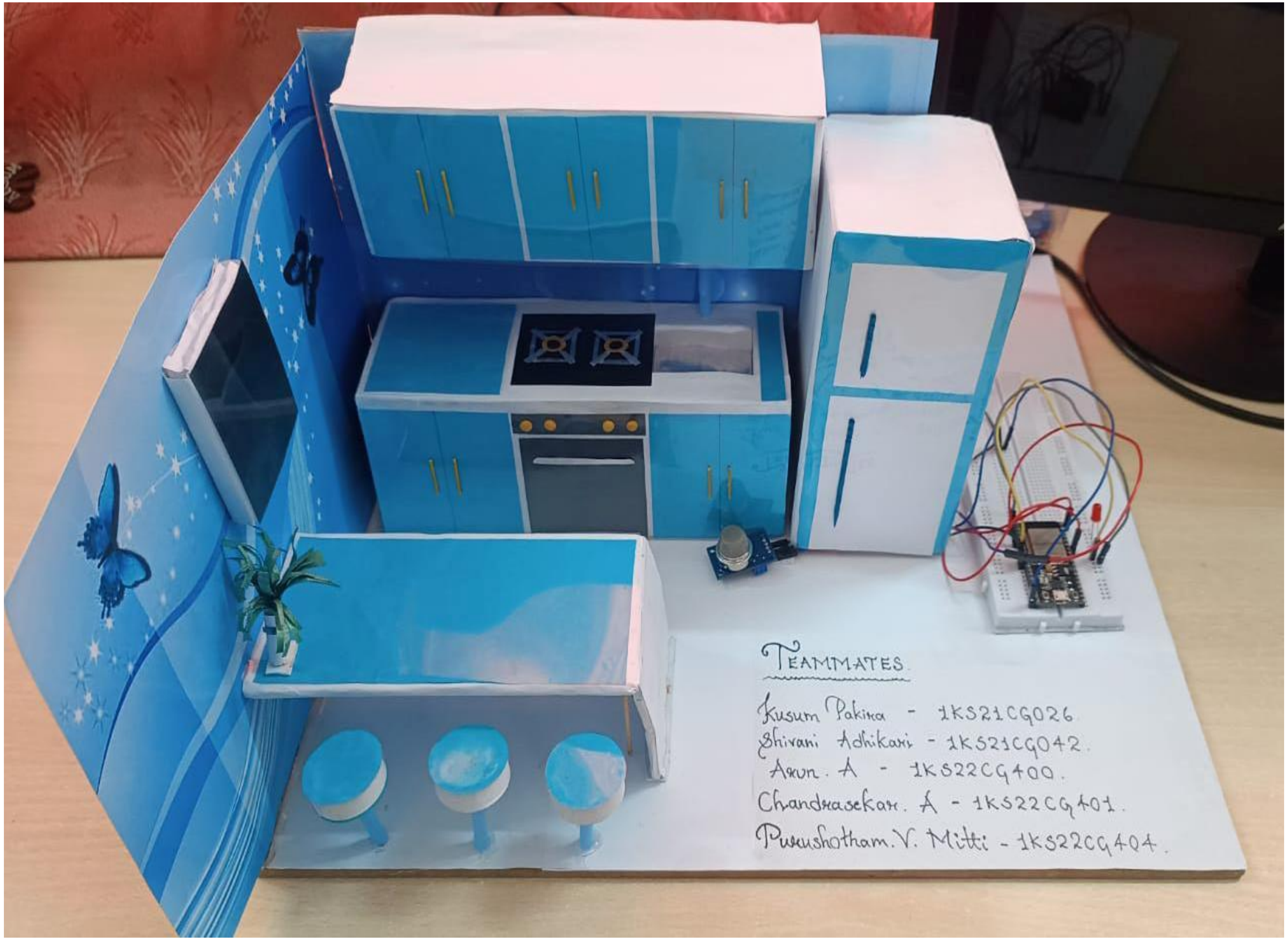
ESP32

Buzzer

LCD

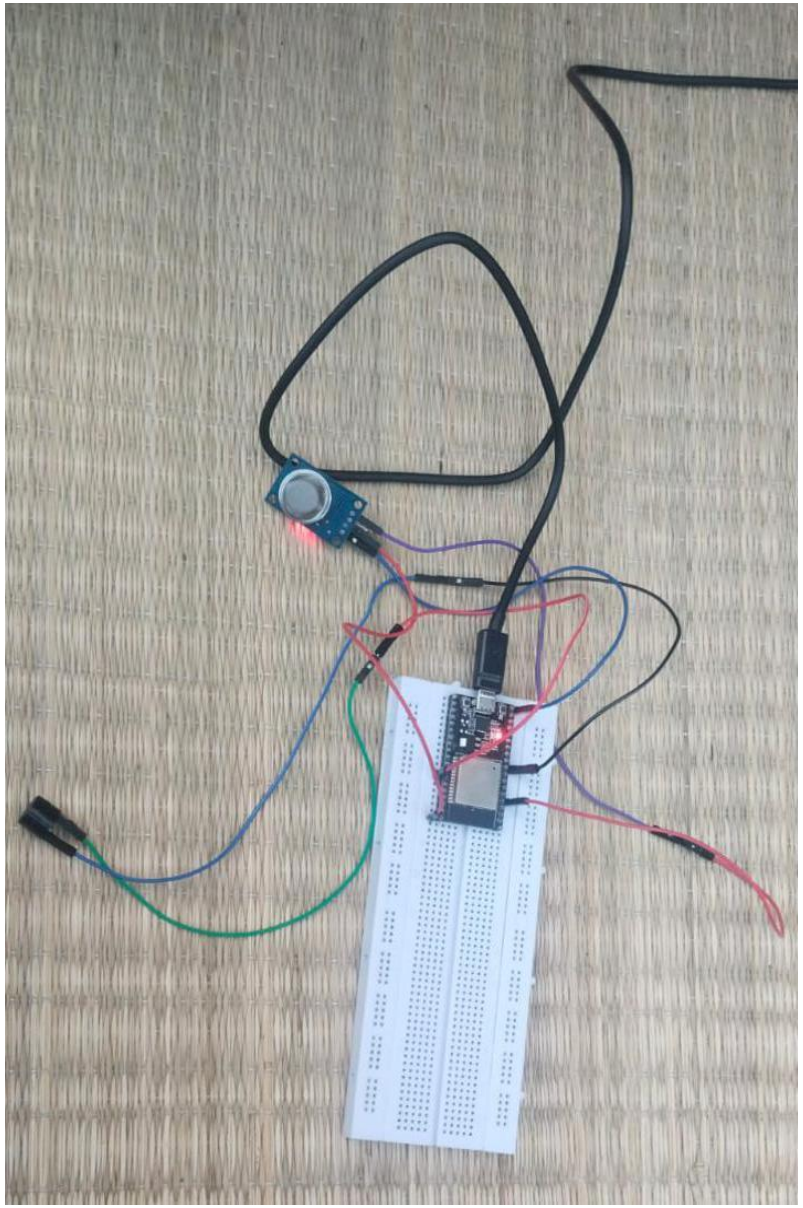


Circuit Setup & Snapshot:

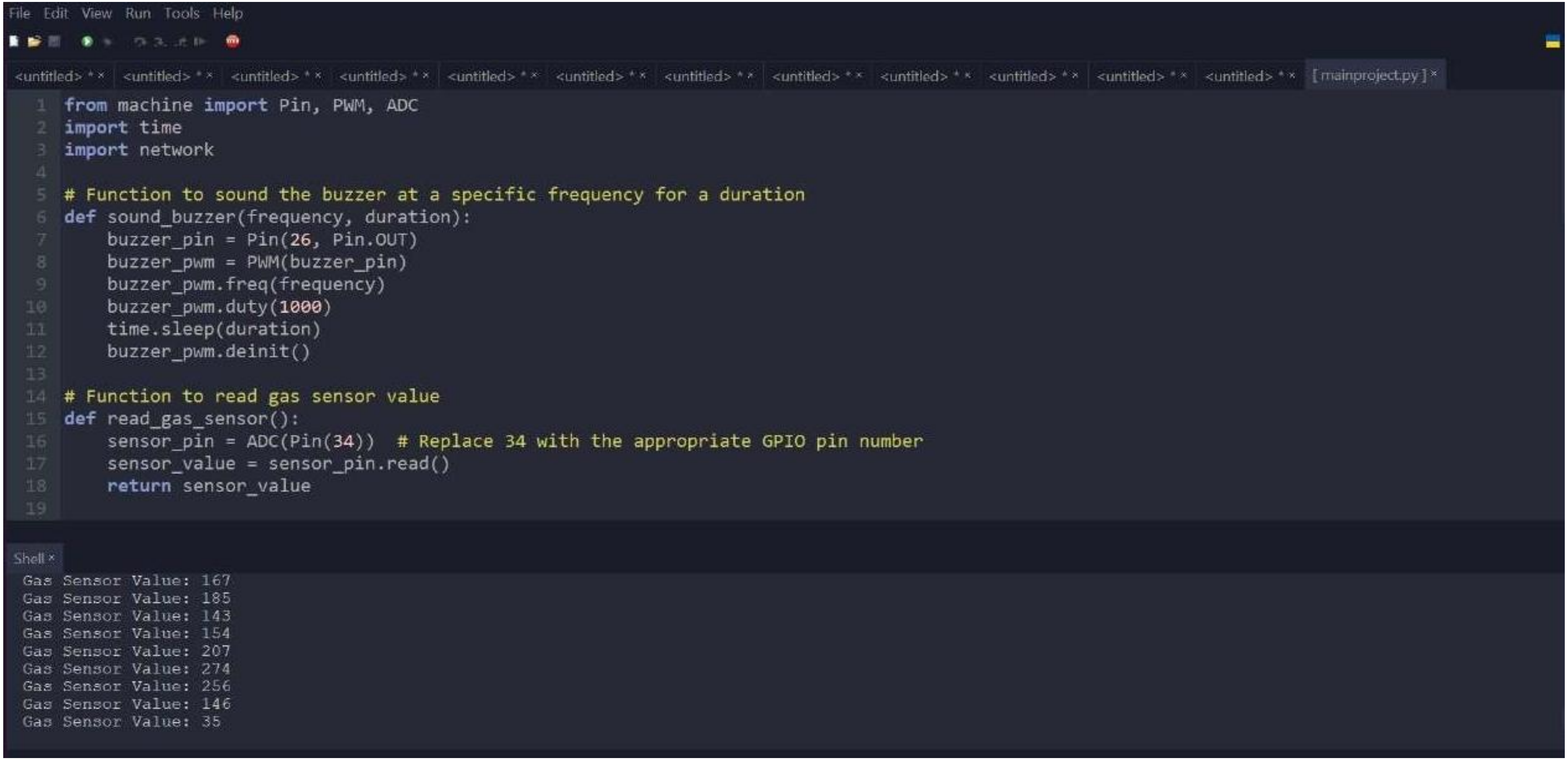


TEAMMATES

Kusum Pakira - 1KS21CG026.
Shivani Adhikari - 1KS21CG042.
Arun. A - 1KS22CG400.
Chandrasekar. A - 1KS22CG401.
Purushotham. V. Mitti - 1KS22CG404.



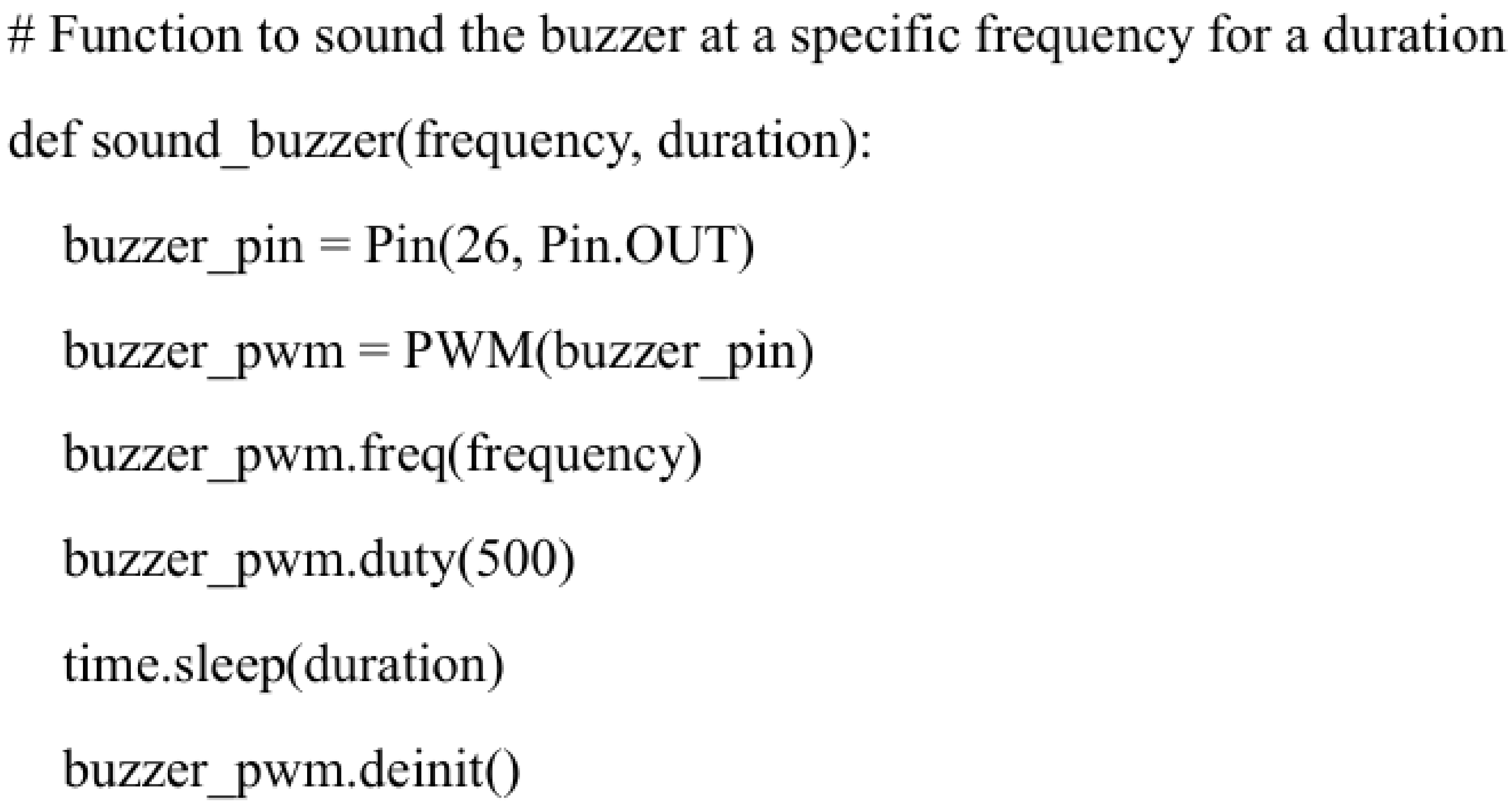
2.THONNY IDE OUTPUT



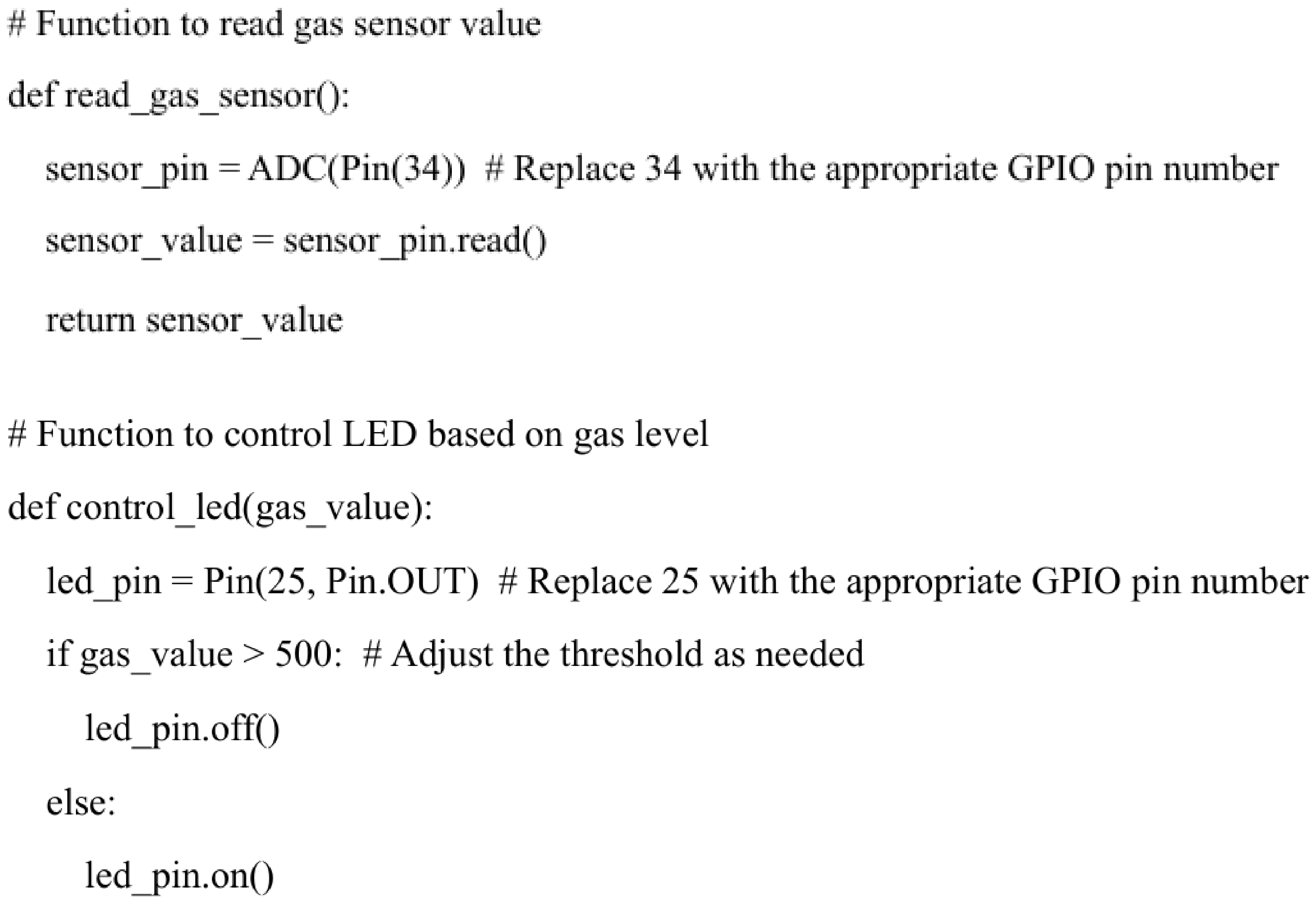
```python
from machine import Pin, PWM, ADC
import time
import network

# Function to sound the buzzer at a specific frequency for a duration
def sound_buzzer(frequency, duration):
    buzzer_pin = Pin(26, Pin.OUT)
    buzzer_pwm = PWM(buzzer_pin)
    buzzer_pwm.freq(frequency)
    buzzer_pwm.duty(1000)
    time.sleep(duration)
    buzzer_pwm.deinit()

# Function to read gas sensor value
def read_gas_sensor():
    sensor_pin = ADC(Pin(34))  # Replace 34 with the appropriate GPIO pin number
    sensor_value = sensor_pin.read()
    return sensor_value

```

```
Gas Sensor Value: 167
Gas Sensor Value: 185
Gas Sensor Value: 143
Gas Sensor Value: 154
Gas Sensor Value: 207
Gas Sensor Value: 274
Gas Sensor Value: 256
Gas Sensor Value: 146
Gas Sensor Value: 35
```

CODE SNIPPET



```
# Function to sound the buzzer at a specific frequency for a duration
def sound_buzzer(frequency, duration):
    buzzer_pin = Pin(26, Pin.OUT)
    buzzer_pwm = PWM(buzzer_pin)
    buzzer_pwm.freq(frequency)
    buzzer_pwm.duty(500)
    time.sleep(duration)
    buzzer_pwm.deinit()
```



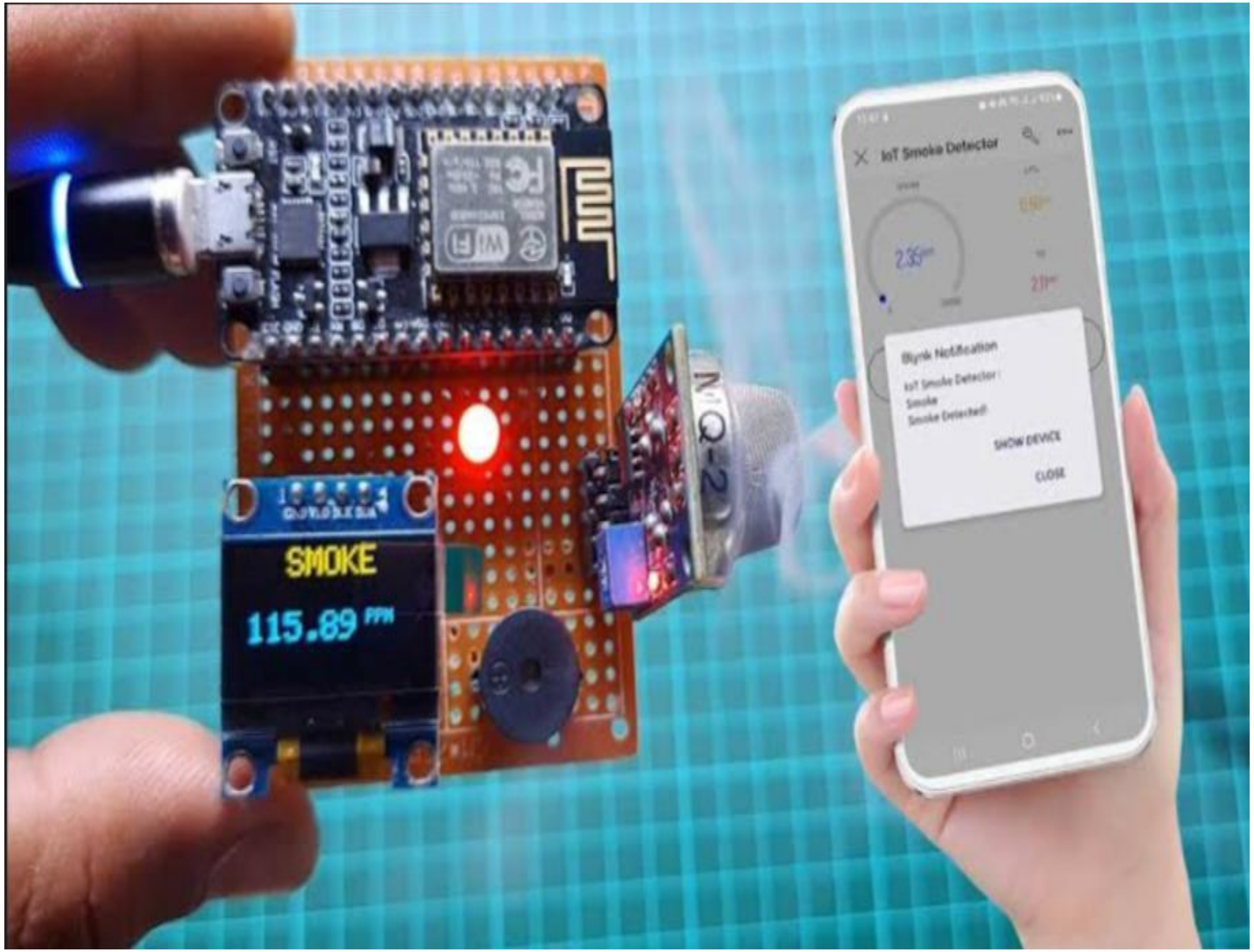
```
# Function to read gas sensor value
def read_gas_sensor():
    sensor_pin = ADC(Pin(34))  # Replace 34 with the appropriate GPIO pin number
    sensor_value = sensor_pin.read()
    return sensor_value

# Function to control LED based on gas level
def control_led(gas_value):
    led_pin = Pin(25, Pin.OUT)  # Replace 25 with the appropriate GPIO pin number
    if gas_value > 500:  # Adjust the threshold as needed
        led_pin.off()
    else:
        led_pin.on()
```

```
# Function to connect to Wi-Fi
def connect_wifi(ssid, password):
    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    if not wlan.isconnected():
        print('Connecting to Wi-Fi...')
        wlan.connect(ssid, password)
        while not wlan.isconnected():
            pass
    print('Wi-Fi connected:', wlan.ifconfig())

connect_wifi('OPPO F11', 'shoo3533')
```

```
# Function to send a Pushbullet notification
access_token = "o.ugfzvjf5x1HNecG2JUqzOOasawYh22eq"

def send_notification(title, body):
    url = "https://api.pushbullet.com/v2/pushes"
    headers = {
        "Access-Token": access_token,
        "Content-Type": "application/json"
    }
    data = {
        "type": "note",
        "title": title,
        "body": body
    }
    try:
        response = requests.post(url, json=data, headers=headers)
        print("Notification sent:", response.text)
    except Exception as e:
        print("Failed to send notification:", e)
```

```
# Function to play "La La La" music on the buzzer

# Main loop
try:
    while True:
        gas_value = read_gas_sensor()
        print("Gas Sensor Value:", gas_value)

        # Sound the buzzer if gas value is above a certain threshold
        if gas_value > 500:
            sound_buzzer(3, 6)  # Sound the buzzer at 1000Hz for 1 second

        # Control LED based on gas value
        control_led(gas_value)

        # Send Pushbullet notification with gas value
        send_notification("Gas Alert", "Gas sensor value: {:.2f}".format(gas_value))

                    time.sleep(1)

            except KeyboardInterrupt:
                pass
```

INTERNSHIP IMPACT



1.Results Summary

- Enhanced Safety
- Health Protection
- Environmental Protection
- Cost Savings
- Regulatory Compliance
- Peace of Mind
- Automation Integration
- Reduced Downtime
- Insurance Benefits
- Technological Advancements

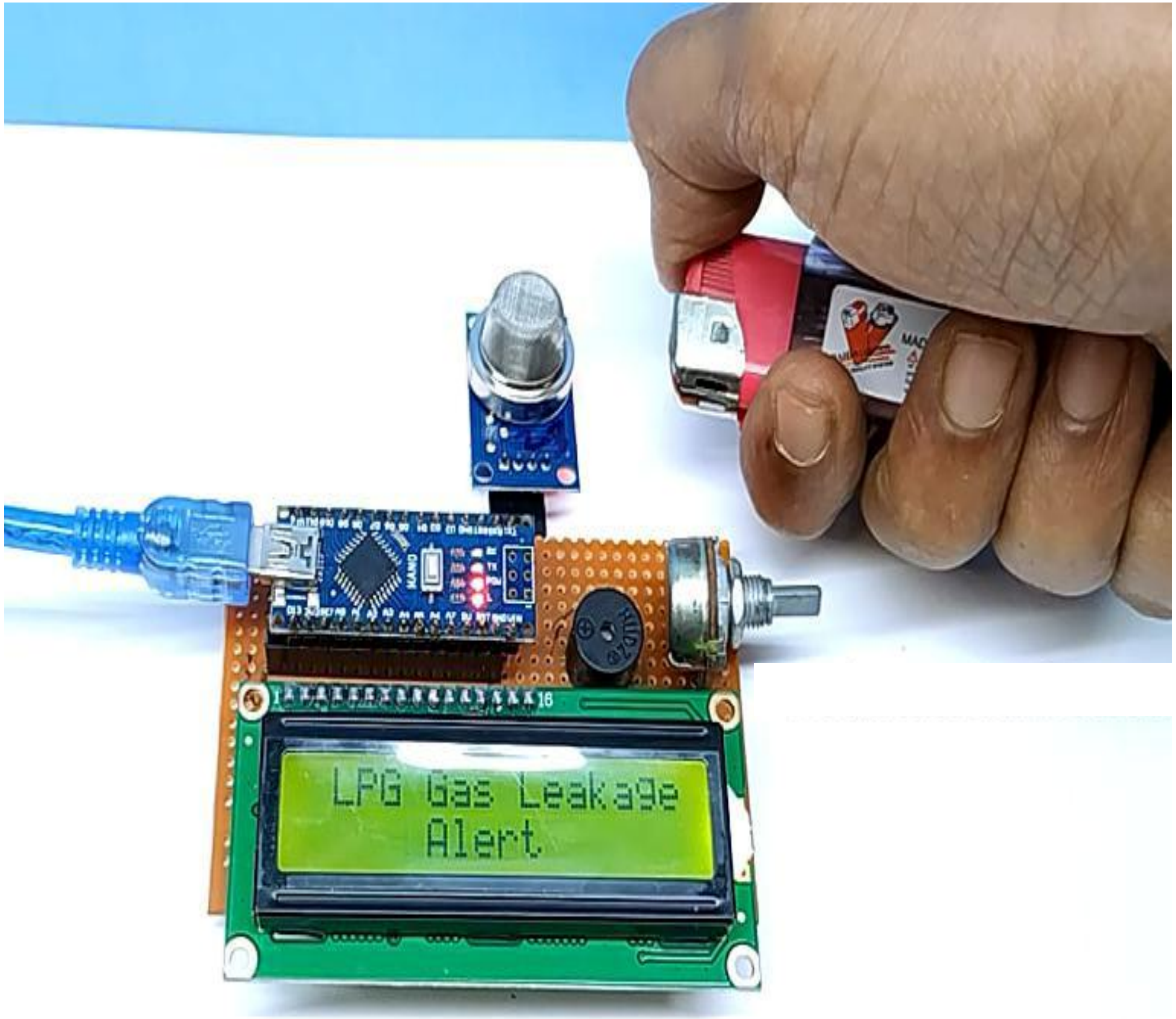
2.Challenges Faced

i. Gas leakage detectors face significant challenges in balancing sensitivity and selectivity. Detecting small leaks requires high sensitivity, but this can lead to false alarms if the detector cannot distinguish between hazardous gases and harmless substances. Environmental factors such as humidity, temperature variations, and dust further complicate this balance by potentially causing inaccurate readings or reduced sensitivity.

ii. Another major challenge is ensuring continuous and reliable operation, especially in remote or hazardous locations where power supply and energy efficiency are critical. High-quality gas detectors can be expensive, limiting their accessibility for smaller businesses and low-income households.

3.Future Scope

- Enhanced Sensitivity and Accuracy
- Wireless and IoT Integration
- AI and Machine Learning
- Energy Efficiency
- Integration with Building Management Systems
- Portable and Wearable Detectors
- Regulatory and Compliance Enhancements
- Environmental Impact Monitoring
- Cost Reduction and Accessibility
- Multi-Gas Detection Capabilities



CONCLUSION

- The Gas Detection Monitoring System using the ESP32 microcontroller represents a cutting-edge solution in the realm of safety technology.
- By seamlessly combining accurate gas detection, local alerts, and remote notifications, the system not only mitigates the immediate risks associated with gas leaks but also sets the stage for future advancements in safety protocols.
- As we strive for safer living and working environments, this system stands as a testament to the potential of IoT-driven solutions in ensuring the well-being of individuals and communities.
- The Gas Detection Monitoring System is not merely a technological innovation; it is a commitment to a safer, more secure future.

REFERENCES

- ESPRESSIF SYSTEMS. (N.D.). ESP32 RESOURCES.
- HTTPS://WWW.ESPRESSIF.COM/EN/PRODUCTS/SOCS/ESP32/RESOURCES
- PUSHBULLET. (N.D.). PUSHBULLET API DOCUMENTATION.
- HTTPS://DOCS.PUSHBULLET.COM/
- MQ GAS SENSORS DATASHEET. (EXAMPLE: MQ-2) – GAS SENSOR MANUFACTURER’S WEBSITE OR DISTRIBUTOR.
- YOUTUBE VIDEOS
- HTTPS://WWW.YOUTUBE.COM/WATCH?V=YZ1EPCLFYGM&T=162S

HTTPS://WWW.YOUTUBE.COM/WATCH?V=DQZCYXSUOBK&T=14S

THANK YOU!