College of Computing and Information Sciences	جامعة التقنية والعلوم التطبيقية University of Technology and Applied Sciences صحار Suhar

IoT based Smart Gas Booking System

A Project report submitted in partial fulfilment of the requirements for the award of the degree of Diploma of Network Computing and Security

Submitted by

Ali Sulaiman Al-Balushi (2021293061)

Loay Musabah Al-Sinani (2020293203)

Abdulaziz Salim Al-Kalbani (2021293039)

Osama Ali Al-kindi (2020293204)

Supervisor: **Dr Thirumurugan Shanmugam**

Project code: 24SP-DP3

MAY 2024

جامعة التقنية والعلوم التطبيقية
University of Technology and Applied Sciences
مسقط
Muscat

STUDENT DECLARATION

(Semester 4, Academic Year 2023/2024)

To: HOD of IT Department Thru: Lecturer/Supervisor, Ali Al-Balushi, 2021293061, Loay Al-Sinani, 2020293203, Abdulaziz Al-Kalbani, 2021293039, Osama Al-kindi, 2020293204 of Diploma Second year who belongs to Section 10 of the CSNW2203-DIPLOMA COURSE PROJECT offered by the IT department, hereby declare that my submission of IoT based Smart Gas Booking System as requirement(s) for the said course is a result of my own original work except for source materials explicitly acknowledged by proper citations. I also understand that plagiarism is an offense that can lead to disciplinary action depending on the seriousness of the case.

Signature:

Student 1 ……

Student 2…….

Student 3……

Student 4……

Date: 19/05/2024

Acknowledgment:

First thing, we are appreciating all that support that we got from all university staff, specialists, and administrators. you believe in us, make us go through this project to apply what we are learn in the past years, not only what we learn in college but, what we study to complete this project. To make you proud of us.

We want to thank the man how he devoted himself to make our project grow in best way. To be something that we are proud of it. Thank to Dr. Thirumurugan Shanmugam, the man that organized the us and use our passion to make this project see the light. As his students we are so grateful for his lead and under of him we built something from nothing. Thank you, our leader and guider, for all thing that you did to us.

we want to thank the member of the group Ali Al-Balushi, Loay Al-Sinani, Usama Al-Kindi and Abdulaziz Al-Kalbani. We are so proud of our work in this project, we build thing from nothing. We are happy that we collaborate with each other, the knowledges that we share make us improve in our career. We are happy of this journey.

In the last, we want to thank all people who they be with us. From the start of the project until it finishes. Your advice and ides are help us more than you think, all this is because of you. We build this.

Abstract:

With the advancement of technology, in years our project takes advantage of this progress by implementing the Internet of Things (IoT) to enhance gas management. In the past gas levels were manually measured, our system automates this task. It utilizes a load cell (HX711) to gauge gas levels and transmits the data to the owner's device through an Arduino Mega connected to a node MCU for Wi Fi connectivity. Furthermore, the system is equipped with a gas leak sensor (MQ 2) for leak detection. The setup transmits real time information to apps via IoT offering timely data to enhance gas management efficiency and safety measures.

1. Introduction

1.1 overview

In Oman, the management of gas resources. Ensuring safety poses a challenge due, to the limitations of the current system. These systems often lack real time monitoring capabilities and automated responses to dangers putting users at risk of gas leaks and fires. To address this pressing issue our top priority is to introduce a gas detection system. At its core is an automated gas level detection system that incorporates a load cell (HX711) to measure gas levels. This system offers real-time monitoring. Sends notifications through an app. Not does this comprehensive system enhance safety measures. It also optimizes gas resource management by offering users an intuitive mobile application interface. By integrating sensors for coverage and effective hazard response our project has the potential to revolutionize safety standards by being proactive and user friendly with automatic reservation being a key feature.

The main outcomes of the project include developing the system conducting in depth sensor analyses for danger detection and creating an interface for applications. Stringent testing protocols are in place to ensure reliability across real world scenarios ensuring alignment, between cutting edge advancements and safety standards.

1.2 Problem statement / background of project

Challenges arise when there is a demand for LPG gas cylinders leading to usage. This situation can disrupt work. Put homeowners in positions. The issue stems from the uncertainty of how much gass left in the cylinder, which forces restaurant owners to halt operations for refills and leaves homeowners waiting for gas deliveries for days. To address this concern a solution has been developed; a system that monitors gas levels in cylinders and alerts users before they run out. Additionally the system incorporates safety features to safeguard restaurant personnel and homeowners in case of gas leaks or fires at home or, in restaurants.

1.3 Project scope and limitation

The scope of our project includes the design, development, and implementation of an advanced gas detection system equipped with automated gas cylinder booking, real-time monitoring, and safety features. To be more precise the system will.

- **Detect Gas Leaks:** Employ sensors to precisely identify gas leaks and set off safety features including user alerts, air flow, and alarms.

- **Monitor Gas Levels:** Utilize load cell sensors (HX711) to continuously monitor the cylinder's gas levels and give users access to real-time data.

- **Automate Gas booking:** The booking process for a new gas cylinder will be started automatically when the gas level drops below a predetermined threshold.

- **Mobile application integration:** Offer a user-friendly mobile application that enables remote system control, alerting, and gas level monitoring.

- **Safety mechanisms:** To guarantee complete safety, incorporate safety measures like fire detection, automated window and door opening systems, and alarms.

Limitation of the project:

- **Sensor accuracy:** The caliber and calibration of the employed sensors determine how accurately gas is detected and levels are measured. There may occasionally be misleading readings or sensor issues.

- **Connectivity issues:** For the mobile application to provide real-time monitoring and notifications, dependable internet connectivity is necessary. Problems with connectivity could cause the system to stop working.

- **User adoption:** The new system must be embraced by users, who must engage with it daily. Users' lack of technical expertise or resistance to change could restrict the system's usefulness.

- **Maintenance Requirements:** In order to guarantee accuracy and dependability over an extended period of time, sensors and system components require routine maintenance and calibration.

1.4 Project objectives

The goals of this project's research are as follows; 1. Implement a Gas Detection System that includes cutting edge technology, like the load cell sensor (HX711) gas leak sensors, fire sensors and connectivity to a mobile app. 2 Showcase how effective the Gas Detection System's at monitoring gas levels in cylinders detecting gas leaks and quickly identifying fire incidents. 3 Study the impact of the Gas Detection System on enhancing safety measures and improving user experience in managing gas resources. 4 Send out automated alerts and notifications via the app when gas levels, leaks or fire incidents are detected to ensure response and action. 5 Assess the practicality of incorporating the Gas Detection System as a tool for monitoring gas and boosting safety across settings such as residential areas, businesses, and industrial sites.

1.5 Project description

In Oman, the management and safety of gas resources creates important issues based on the restrictions on existing systems. These systems often lack the potential of actual monitoring and the potential dangers, exposing users to the risk of gas leaks and fire. In response to this urgent need, our main objective is to introduce an innovative gas detection system with automatic gas cylinder backup as a fundamental feature. With the integration of a load cell (HX711), our system accurately measures gas levels, providing seamless booking automation and real-time notifications via a mobile app. This comprehensive system not only improves safety measures, but also optimizes gas resource management by providing users with an intuitive and user-centric mobile application interface. Thanks to the integration of multiple sensors, it provides comprehensive coverage and effective response to potential dangers. Our project has significant

potential to revolutionize security standards by making them proactive and user-friendly, with automatic booking being a key aspect.

1.6 Significance of project

- The project combines technology with sensors to detect gas leaks accurately and promptly enhancing safety measures through early warnings and hazard prevention.Features mobile app compatibility, enhancing safety standards.

- With mobile app compatibility users can remotely monitor gas levels. Receive alerts ensuring response, to potential leaks and bolstering safety protocols even when off site.

- Efficient gas monitoring and management provided by the system contribute to reducing wastage and minimizing impact promoting eco gas usage practices.

- This initiative sets the stage for research in environmental monitoring fields showcasing the potential of IoT applications in enhancing safety standards and sustainability efforts fostering innovation, in these domains.

1.7 Project Methodology

Planning phase.

- **Problem identification:**
  - Analyze current gas monitoring and detection systems in extreme detail.
  - Determine the requirements and difficulties pertaining to automatic booking, leak detection, and gas safety.

- **Requirement gathering:**
  - Talk to potential users to learn about their needs and expectations.
  - Describe the system's functional and non-functional requirements.

- **Feasibility study:**
  - Examine the project's viability considering its technical specifications, available funding, and resource availability.
  - Determine possible risks and create plans for mitigating them.

Design phase:

- **System architecture design:**
  - Create the entire system architecture, considering the MCU, SBC, actuators, and sensors.
  - Create block diagrams and schematics to show the parts of the system and how they work together.

- **Component selection:**
  - Choose the proper actuators (such alarm, servos, etc.), hardware and sensors (such as gas sensors, flame sensor, and load cell sensor HX711).
  - Select the SBC or MCU that will serve as the system's core components.

Development phase:

- **Hardware development:**
  - Assemble the hardware components based on the design specifications.
  - Integrate sensors, actuators and MCUs.

- **Software development:**
  - Create firmware for the MCU to manage actuator control and sensor data processing.
  - Create a mobile application for user interactions, notifications, and real-time monitoring.

- **Integration:**
  - Make sure that the hardware and software components integrate seamlessly.
  - Create dependable lines of communication for the mobile app, MCU, actuators, and sensors.

Test phase:

- **Unit testing:**
  - Check that each component—sensors, actuators, and MCU—is operating as intended by testing it separately.

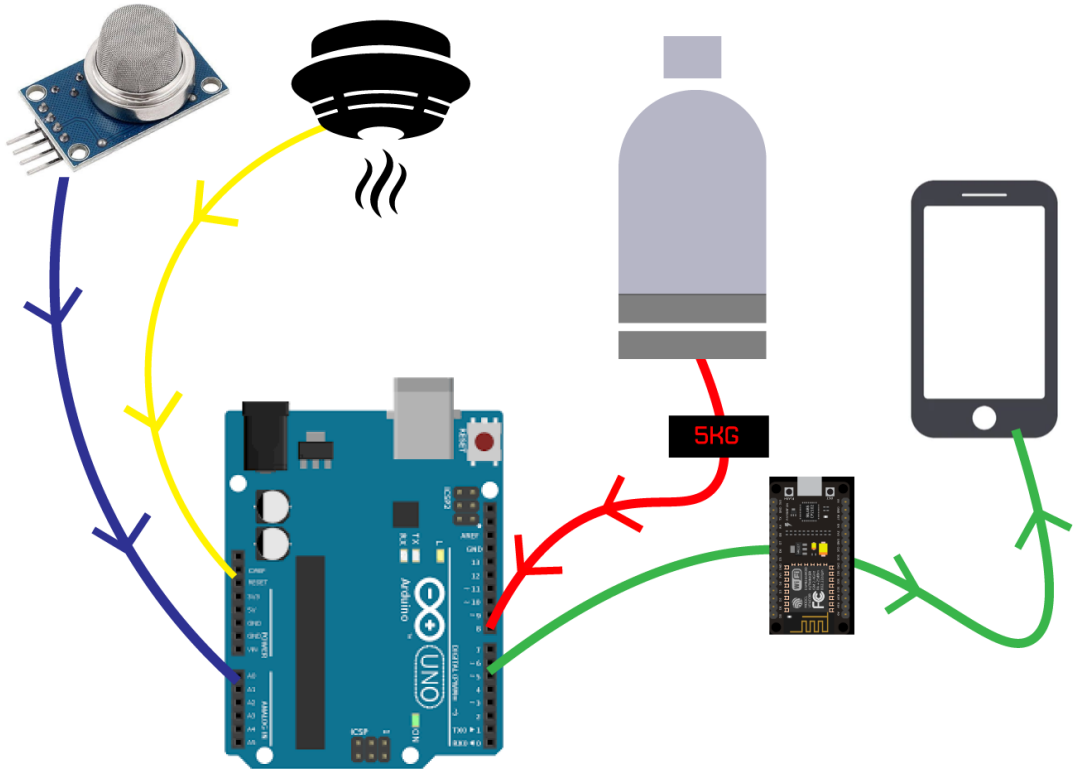
- **System design:**
  - Conduct comprehensive system testing to ensure all components work together as expected.
  - Simulate various scenarios (e.g., gas leaks, low gas levels) to test system responses.

Deployment phase:

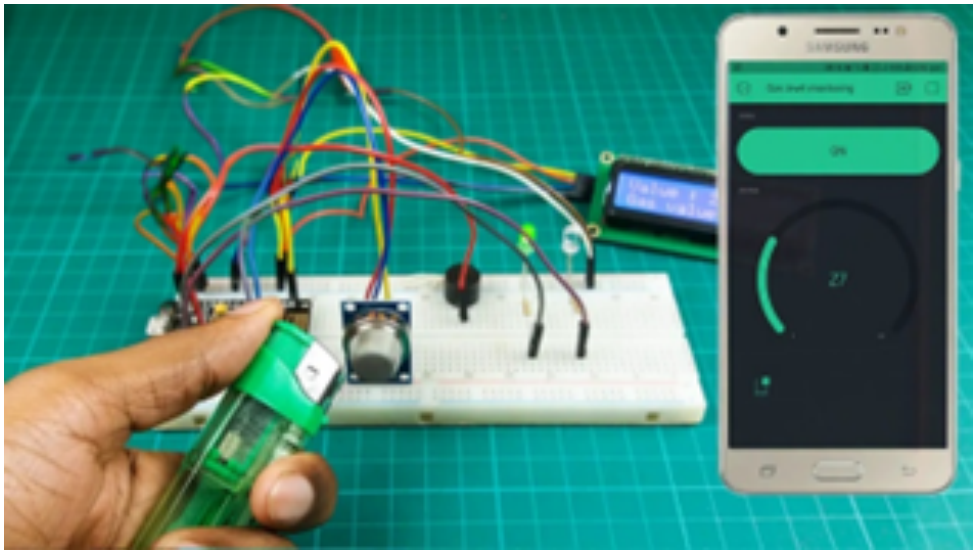
- **Monitoring and maintenance:**
  - Monitor system performance and address any issues that arise.

2. Review of related literature and questions

Sample:



Iot real prototype:



2.1 Literature Review

2.1.1 Research 1

Using an integrated sensor, Tamizharasan.V, Sandeep.R, Ravichandran.T, Saravanavel.K, and Sowndariya.M developed an Internet of Things-based gas level detection system for booking management. The system integrates sensor data with Node MCU and ESP8266 for Wi-Fi connectivity. It is described in "IoT Based Level Detection of Gas for Booking Management Using Integrated Sensor" by V. Mariselvam and M. Siva Dharshini from the Department of Electronics and Communication Engineering at M. Kumarasamy College of Engineering, Karur, India. Because this configuration immediately notifies the distributor when the gas level is low, it guarantees timely refills and avoids problems before or after booking.

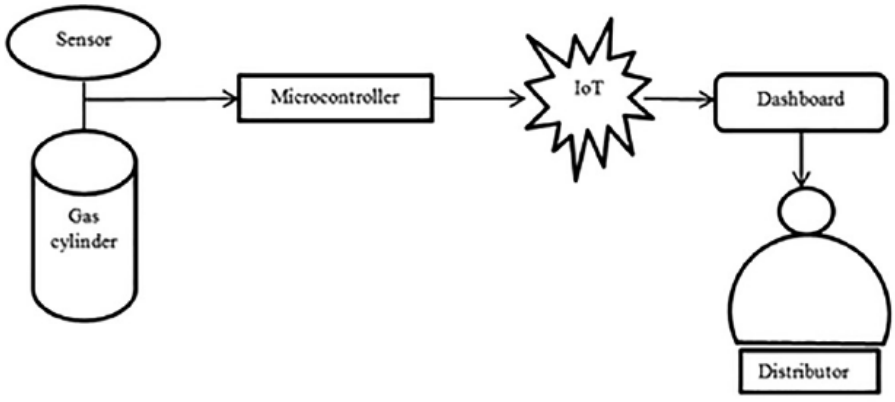
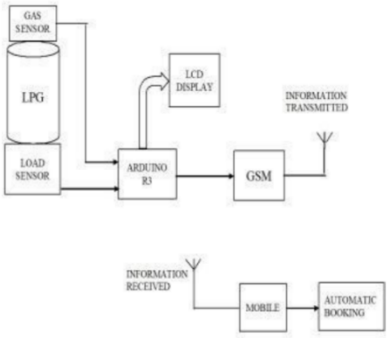


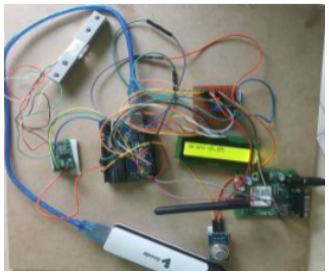
Fig. 2. Block diagram of proposed system.

2.1.2 Research 2

A scientific paper called “Gas level detection and automatic booking using IoT” for Tamizharasan V, Ravichandran.T, Sowndariya.M, Sandeep.R, and Saravanavel.K that talk about measuring gas level using load sensor (SEN-10246) and the output of this sensor is

sends to Arduino R3 by use of GSM module. The information of the gas is sends to user by SMS and the automatic booking is done by dialing the registered gas booking number, also the paper include a security measurement like adding a gas leakage detecting system by using of gas sensor (MQ-6 sensor), and the value is displayed on a LCD screen, and finally the user is alerted if the gas level become less than 20%.





2.2 Questionnaires

We conduct a questionnaire with restaurant owners, houses owners, and gas cylinders providers to get more ideas about the gas cylinder and the traditional ways to measure gas level. Questions and answers were as follow:

How does the customer know the gas is almost over?

Most answers were as follow: The customer knows if the gas is over by many ways, first if there is no gas flow over the stove, second, if the gas cylinder becomes light weight.

How are they getting gas from suppliers?

Most answers were as follows: Customers can get the gas cylinder in many ways, first they call gas dealer and order a gas. Second, they go to the factory and refill the gas, and lastly, they use applications like Awan gas for ordering gas.

Whether they have any security for gas leakage

for a security purpose from gas leakage there is a system called Gas Leak Detection a gas detector is an important tool for safety. It checks for dangerous gases in an area. It works with control systems to stop processes if there's a leak. If it finds a problem, it makes a loud noise so people can leave quickly.

Any Technical detail off gas cylinder and cooker?

In Oman, we have two most used gas cylinder. The first one is the middle one (22Kg) which is used mostly in small places or houses, which uses by local people to cook food. This gas cylinder is made from steel. It has 50L capacity and the total weight with the gas is probably 51 kg, it has %50.1 percent of gas on it. The second one is the big one (44kg) which is mostly used in economic sectors like restaurants. It is also made from steel, and it is taller than the middle one. It has 100L capacity and the total weight with the gas is probably 94 kg, it has %53.1 percent of gas.

1. 48L

2. 108L

2- Question	Daily	Weekly	Monthly	Rarely	Never
How often do you monitor the gas levels in your home / restaurant?	13%	35%	9%	23%	20%
	Strong Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
How concerned are you about the potential risks associated with gas usage, such as leaks, fire hazards, or environmental impact?	8%	17%	32%	23%	20%
How confident are you in the accuracy of your current gas level monitoring and leak detection methods?	9%	19%	29%	33%	10%
How important do you consider real-time monitoring of gas levels in ensuring safety and efficiency?	20%	7%	13%	37%	23%
Would you be interested in a gas detection system that provides automatic alerts for gas leaks and real-time monitoring of gas levels in your home or workplace?	21%	11%	8%	36%	24%

Table Output of survey

3. Requirement

3.1 Hardware requirements

For hardware we go with:

- MCU/SBC:
  - Arduino Mega: for connecting all sensors and actuators together.
  - NodeMCU: for connecting Arduino Mega to internet and send data to blynk app.

- Sensors:
  - Load cell sensor-HX711: measure gas level.
  - 2 Flame sensors: for detecting flame happened in gas room or kitchen.
  - 2 Gas leak sensor-MQ2: for detecting any leak happen inside gas room or kitchen.

- Actuators and others:
  - LED: for gas level monitoring and incident alerts.
  - Buzzer: for making an alarm if any leak of fire happened.
  - Servo motor. For opening and closing doors and windows

3.2 Software requirements

For software requirements we use:

- Arduino IDE: for writing the codes.
- Blynk app: for connecting our system to internet and to mobile devices.

3.3 Functional requirements

- The system should be able to detect gas leaks and fire incident accurately and make action on that.

- The system should give accurate gas level percentages and give and alerts if the gas level is low.
- User should be able to access the system through mobile application.

3.4 Non-functional requirements

- Reliability: The system should be highly reliable, with minimal downtime and accurate detection of gas leaks and level.
- Performance: It should have fast response times for detecting leaks and processing booking requests.
- Scalability: The system should be scalable to accommodate a growing number of users and sensors.
- Compliance: The system should comply with relevant safety standards and regulations for gas detection and management.

4. Project plan

4.1 Action plan

Step 1: Training and Self-learning

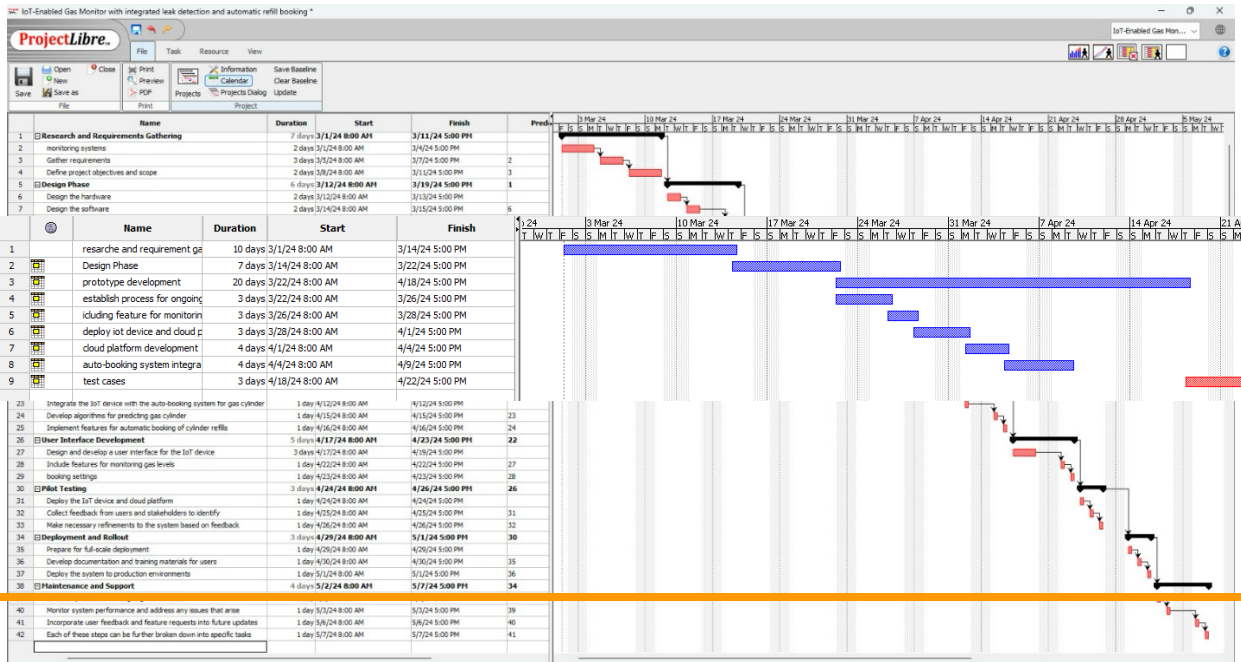
- Understanding how internet of things work through self-learning, online courses, and in-site workshop.
- Try to understand the tools and codes that will be implemented in our project.
- Determine which cloud service is better for our project by conducting research on that.

Step 2: Literature Review

- Compile and review the literature on Internet of things and gas level detection system.
- Study the relevant resources to identify the best techniques and tools for the project.

Step 3: Prototype Development

- Develop a prototype of project that will be able to identify gas level and detect any leak or fire incident.



Step 5: Data Processing and Analysis

- Process the data output from sensors through Arduino mega and nodeMCU in order to analyze data and push the correct notification to user mobile phone through cloud services.
- Complete and submit the project.

5. Analysis and design

5.1 Problem statement

Existing systems face challenges in managing gas resources and ensuring safety due to their limitations. The absence of automated responses and real time monitoring for threats leaves users vulnerable to dangers like gas leaks and fire incidents. Moreover, outdated methods of measuring gas levels are cumbersome, inefficient and susceptible to error making it difficult to ensure gas replenishment and effective leak detection.

Real Time Monitoring Shortcomings.
The lack of real time monitoring of gas levels and leaks in current gas detection systems results in delayed responses during situations.

Manual Gas Level Measurement.
Although manual gas level measurement is a practice it is laborious, inaccurate and prone, to errors.

5.2 Solution

An effective solution proposed for addressing these issues involves developing a enabled system for gas detection and monitoring. This system offers features like automated notifications, real time monitoring and user-friendly interfaces. To ensure efficient gas management, cutting edge sensors, an MCU and a mobile app will be utilized.

Real time monitoring.
Continuous monitoring of gas levels using sensors like fire, gas leak detectors and the load cell sensor HX711 to promptly detect any leaks.

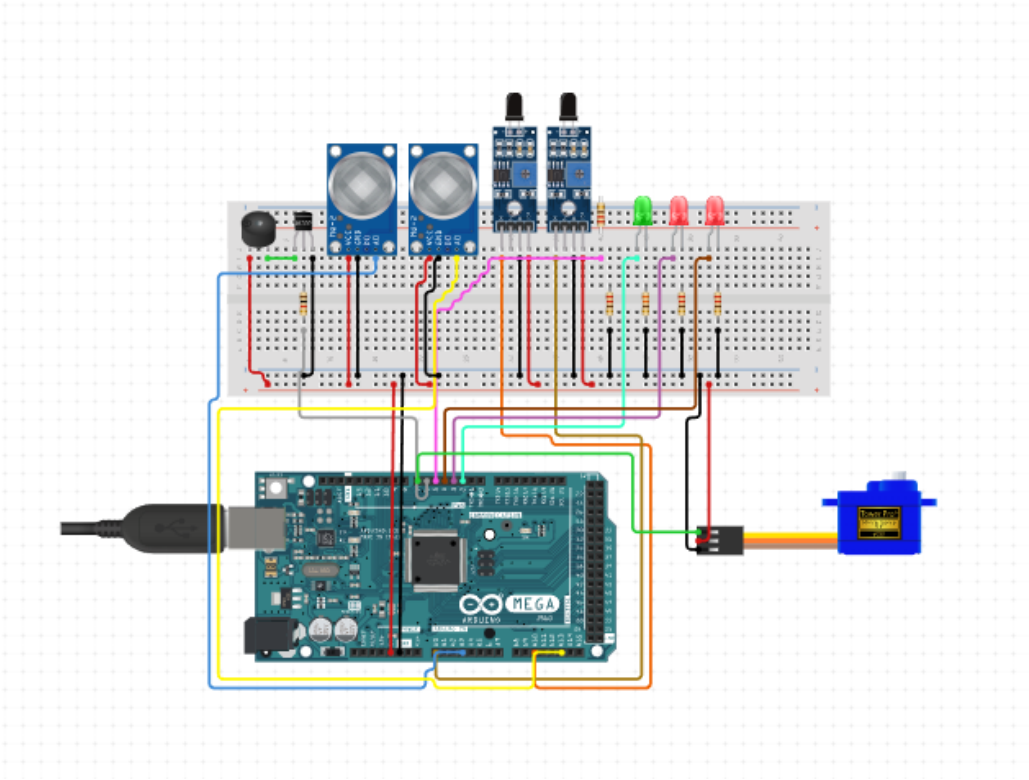
Automated gas level measurement.
measurement of gas levels through the use of the load cell sensor HX711 eliminating the need for measurements.

Automated leak and fire detection.
Combining gas leak sensors and flame detectors to swiftly identify leaks and fires while triggering safety protocols, like alerts, opening/closing windows and doors and notification systems.

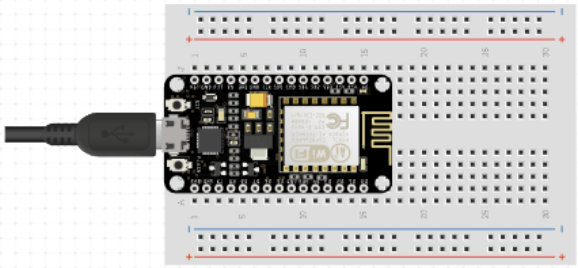
5.3 Prototype design.

Circuit Design:

All sensors and actuators are connected to Arduino Mege, in addition, the nodeMCU is also connected to Arduino Mega through port (2,3):

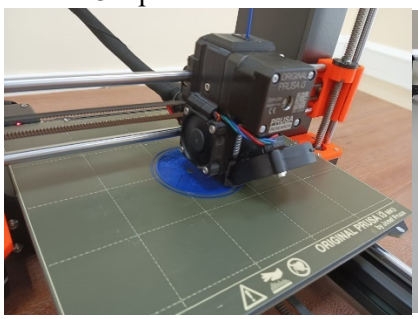
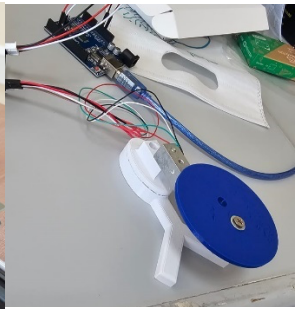
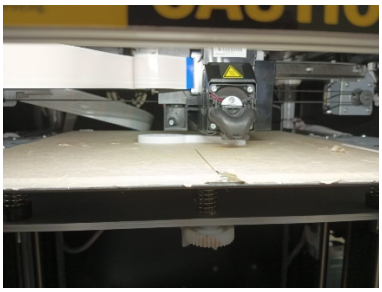


NodeMCU will get values from Arduino mega and send them to blynk app using wifi:



Mechanical design:

We build module for kitchen to hold the measuring and alarming system and also create a cylinder room to hold the HX711-load cell sensor with flame and gas sensor, and in order to make the measurement of load cell sensor write we contact with engendering department to make a 3D printed bass for that.

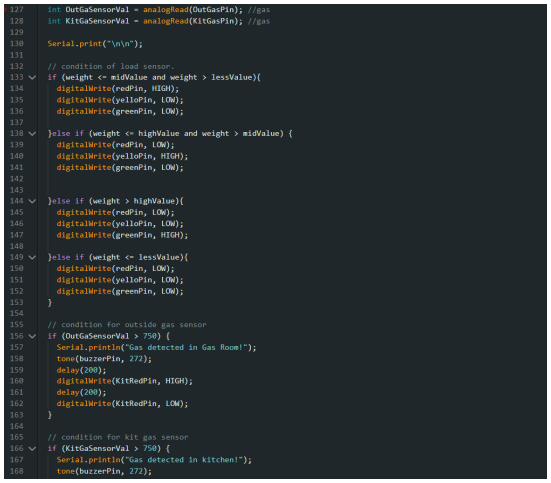
Code overview:

Code for setting up Arduino mega and all other sensor, actuators, and nodeMCU.

```
//HX711 constructor (dout pin, sck pin)
HX711_ADC LoadCell(HX711_dout, HX711_sck); // load

const int calVal_eepromAdress = 0; // eeprom address for calibration value (4 bytes) // load
long t; // load


void setup() {
  // Debug console
  Serial.begin(9600);
  nodemcu.begin(9600);//connect arduino to MCU//

  //buzzer
  pinMode(buzzerPin,OUTPUT);

  //lights
  pinMode(yelloPin, OUTPUT);
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(KitRedPin, OUTPUT);

  // load
  delay(10); // load
  Serial.println(); // load
  Serial.println("Starting..."); // load
  float calibrationValue; // calibration value // load
  calibrationValue = 696.0; // Uncomment and set this value if you want to set it in the sketch // load
  LoadCell.begin(); // load
  long stabilizingtime = 2000; // Tare precision can be improved by adding a few seconds of stabilizing time // load
  boolean tare = true; //set this to false if you don't want tare to be performed in the next step // load
  byte loadcell_rdy = 0; // load
  while (!loadcell_rdy) {
      loadcell_rdy = LoadCell.startMultiple(stabilizingtime, tare);
  }
  if (LoadCell.getTareTimeoutFlag()) {
      Serial.println("Timeout, check MCU>HX711 wiring and pin designations");
  }
  LoadCell.setCalFactor(calibrationValue); // Set calibration value (float)
  Serial.println("Startup is complete");

  // flame
  pinMode(OutFlamePin,INPUT); // flame
```

```

#include <Servo.h>
#include <HX711_ADC.h>
#include <SoftwareSerial.h> //connect arduino to MCU//
#include <ArduinoJson.h> //connect arduino to MCU//

//Initialise Arduino to NodeMCU (5=Rx & 6=Tx)
SoftwareSerial nodemcu(2, 3); //connect arduino to MCU//

//servo Open/Close
const int openServo = 90;
const int closeServo = 0;

//load values
const int lessValue = 2;
const int midValue = 26;
const int highValue = 40;

//HX711 pins
const int HX711_dout = 4; // mcu > HX711 dout pin // load
const int HX711_sck = 5; // mcu > HX711 sck pin // load

//light pins
const int yelloPin = 8;
const int redPin = 9;
const int greenPin = 10;

//kitlight
const int KitRedPin = 11;

// buzzer pin
const int buzzerPin = 7;

const int OutGasPin = A0;  // Analog pin connected to MQ-2 output // Outgas

const int KitGasPin = A2;  // Analog pin connected to MQ-2 output // Kitgas

const int baselineValue = 0; // Adjust this after initial calibration //gas
const int GASthreshold = 600;   // Adjust this threshold based on desired sensitivity //gas

const int OutFlamePin = A1; // Outflame
const int KitFlamePin = A3; // Kitflame
```

```
pinMode(OutFlamePin,INPUT); // flame
pinMode(KitFlamePin,INPUT); // flame

}

void loop() {
noTone(buzzerPin);
digitalWrite(KitRedPin, LOW);

static boolean newDataReady = 0; // load
const int serialPrintInterval = 0; // Increase value to slow down serial print activity // load
// Check for new data/start next conversion: // load
if (LoadCell.update()) newDataReady = true; //load
// Get smoothed value from data set //load
float weight;
if (newDataReady) { //load
    if (millis() > t + serialPrintInterval) { // load
        weight = LoadCell.getData(); // load
        Serial.print("Load cell output value: "); // load
        Serial.println(weight); // load
        newDataReady = 0; // load
        t = millis(); // load
    }
}
// Receive command from serial terminal, send 't' to initiate tare operation:
if (Serial.available() > 0) { // load
    char inByte = Serial.read(); // load
    if (inByte == 't') { // load
        LoadCell.tareNoDelay(); // load
    }
}
// Check if last tare operation is complete
if (LoadCell.getTareStatus() == true) { // load
    Serial.println("Tare complete"); // load
}

// flame
int threshold=200; // flame
int OutFlameSensorVal = analogRead(OutFlamePin); // flame
int KitFlameSensorVal = analogRead(KitFlamePin); // flame

// gas
int OutGaSensorVal = analogRead(OutGasPin); //gas
```



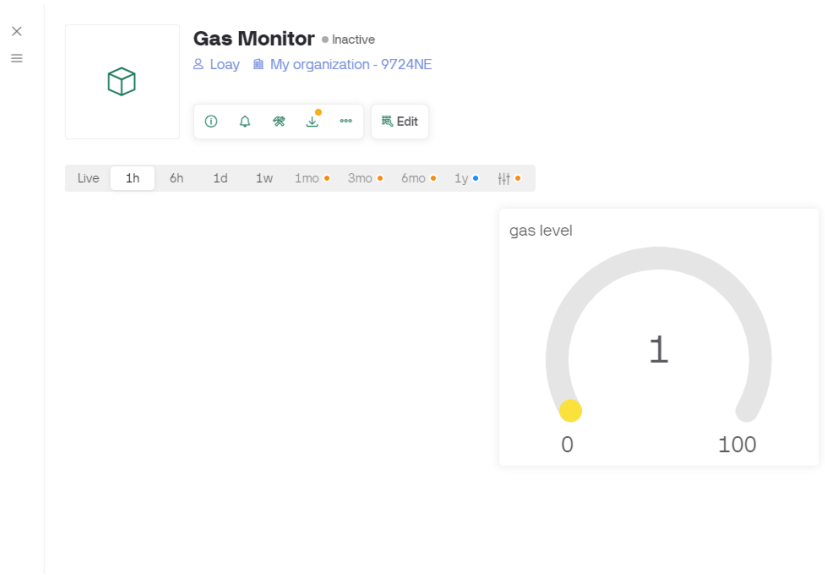
```
int OutGaSensorVal = analogRead(OutGasPin); //gas
int KitGaSensorVal = analogRead(KitGasPin); //gas

Serial.print("\n\n");

// condition of load sensor.
if (weight <= midValue and weight > lessValue){
  digitalWrite(redPin, HIGH);
  digitalWrite(yelloPin, LOW);
  digitalWrite(greenPin, LOW);

}else if (weight <= highValue and weight > midValue) {
  digitalWrite(redPin, LOW);
  digitalWrite(yelloPin, HIGH);
  digitalWrite(greenPin, LOW);


}else if (weight > highValue){
  digitalWrite(redPin, LOW);
  digitalWrite(yelloPin, LOW);
  digitalWrite(greenPin, HIGH);

}else if (weight <= lessValue){
  digitalWrite(redPin, LOW);
  digitalWrite(yelloPin, LOW);
  digitalWrite(greenPin, LOW);
}

// condition for outside gas sensor
if (OutGaSensorVal > 750) {
  Serial.println("Gas detected in Gas Room!");
  tone(buzzerPin, 272);
  delay(200);
  digitalWrite(KitRedPin, HIGH);
  delay(200);
  digitalWrite(KitRedPin, LOW);
}

// condition for kit gas sensor
if (KitGaSensorVal > 750) {
  Serial.println("Gas detected in kitchen!");
  tone(buzzerPin, 272);
```

```
    delay(200);
    digitalWrite(KitRedPin, HIGH);
    delay(200);
    digitalWrite(KitRedPin, LOW);
  }

  Serial.println("kitchen flame: ");
  Serial.println(KitFlameSensorVal);

  Serial.println("Gas Room flame: ");
  Serial.println(OutFlameSensorVal);

  // condition for gas sensor
  if (KitFlameSensorVal <= 100) {
    Serial.println("Flame detected in kitchen!");
    tone(buzzerPin, 272);
    digitalWrite(KitRedPin, LOW);
    delay(200);
    digitalWrite(KitRedPin, HIGH);
    delay(200);
  }

  if (OutFlameSensorVal <= 100) {
  Serial.println("Flame detected in Gas Room!");
  //digitalWrite(buzzerPin, HIGH);
  tone(buzzerPin, 272);
  digitalWrite(KitRedPin, LOW);
  delay(200);
  digitalWrite(KitRedPin, HIGH);
  delay(200);
  }
  nodemcu.write(weight);
// gas
  delay(1000); // Read sensor value every 500 milliseconds
}
```

Code for setting up nodeMCU by receiving load cell sensor value and send it to blynk app through wifi:

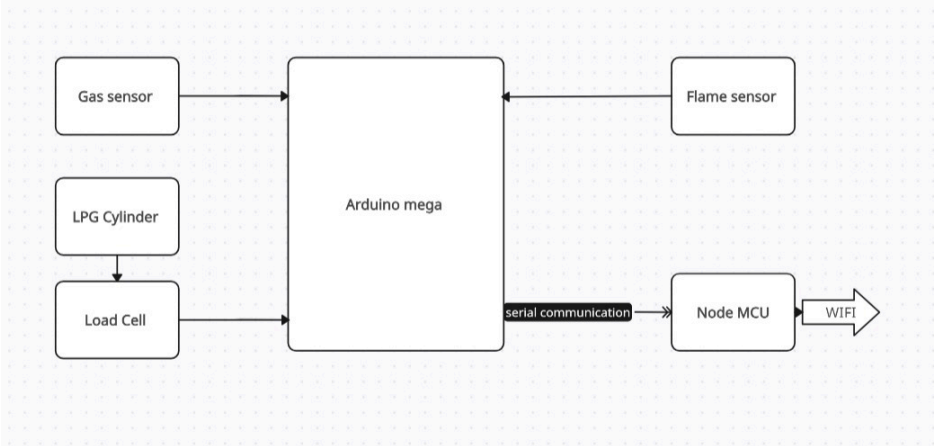
```ino
#define BLYNK_TEMPLATE_ID "TMPL6P190oxJJ"
#define BLYNK_TEMPLATE_NAME "gas monitor"
#define BLYNK_AUTH_TOKEN "GQ7ozdS3TI4xfjVB7ilodCxWBgNbL4Za"
#define BLYNK_PRINT Serial


#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <SoftwareSerial.h>
#include <ArduinoJson.h>

char ssid[] = "bxl";
char pass[] = "a12345678";

BlynkTimer timer;

SoftwareSerial nodemcu(D2, D3);
void setup() {
  Serial.begin(9600);
  nodemcu.begin(9600);
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  //while (!Serial) continue;
}

void loop() {
  if ( nodemcu.available() > 0){
    int load = nodemcu.read();
    Serial.println(load);
    Blynk.virtualWrite(V0,load);
  }

  Blynk.run();
  timer.run(); // Initiates BlynkTimer
  delay(1000);


}
```

Blynk app interface:

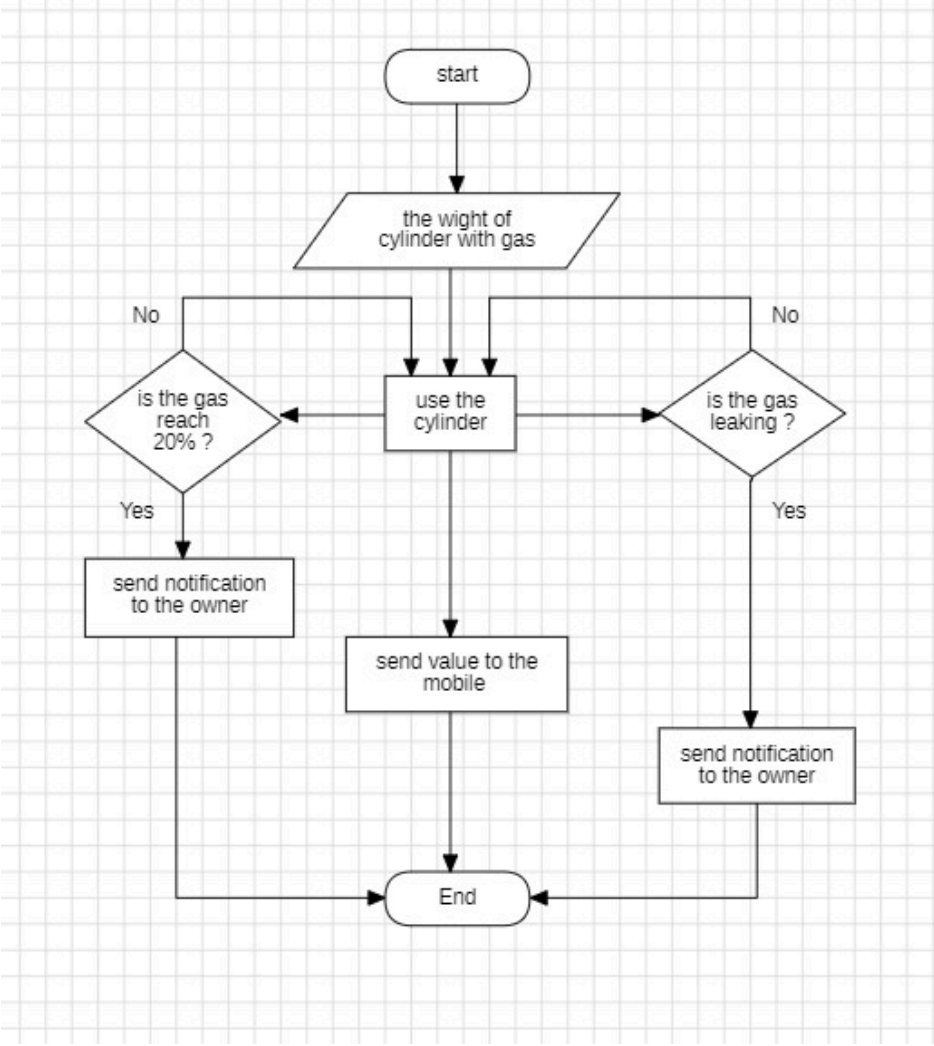


5.4 F/B D

Block diagram:


Gas sensor
Flame sensor
Arduino mega
LPG Cylinder
Load Cell
serial communication
Node MCU
WIFI

Flowchart:


start
the wight of cylinder with gas
No
No
is the gas reach 20% ?
use the cylinder
is the gas leaking ?
Yes
Yes
send notification to the owner
send value to the mobile
send notification to the owner
End

5.5 Sensor/actuator used and linked via MCU.

- The flame sensor, in the kitchen and gas room is connected to the MEGA board using three ports. One for power (5V) one for GND and the last one for analog port 3 for the kitchen sensor and A1 for the gas room sensor.

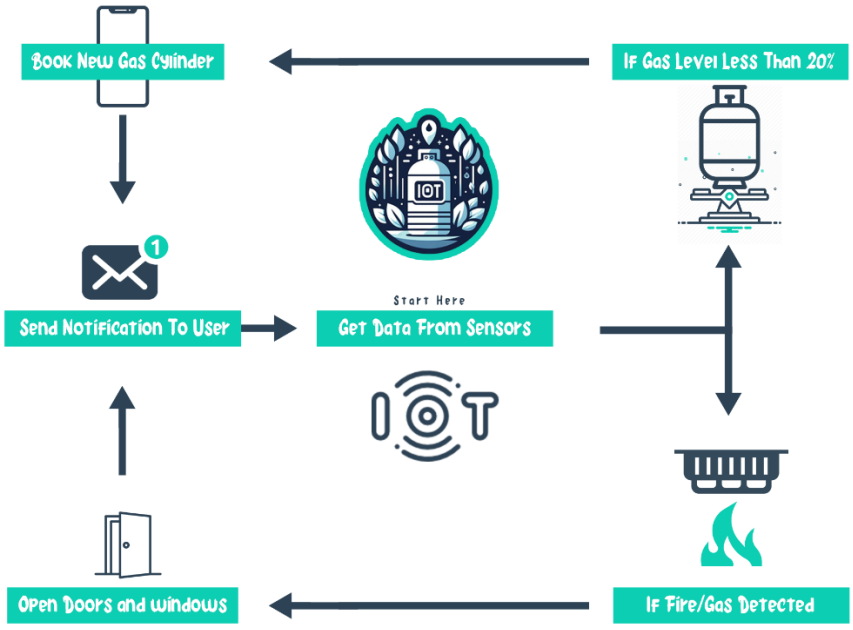
- As for the gas sensor (MQ2) in both the kitchen and gas room they are linked to the MEGA board through four ports. One for power (5V) another for GND, one connected to TX port and the last one going to analog port 2 for the kitchen sensor and A0 for the gas room.

- The load cell sensors (HX711) are hooked up to the MEGA board through four ports. Two each for GND and 5V with two (4, 5) used to send data to the MEGA.

- The buzzer is attached to the MEGA using two ports. One controls it through port 7 while another connects it to ground (GND).

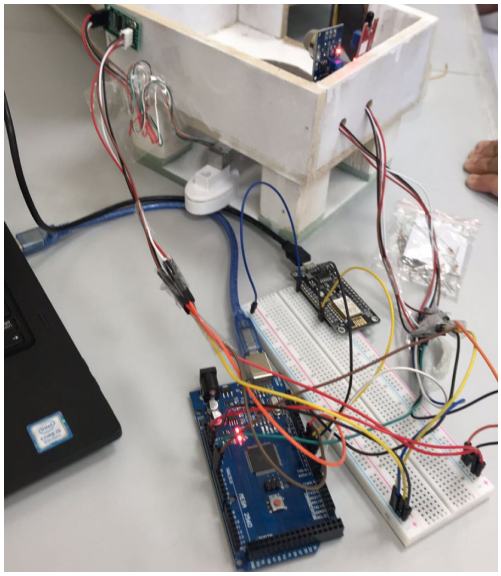
- Additionally, four lights are connected to MEGA via two ports. One linked with ground (GND) with a 330 Ohm resistance while the other connects them to ports. The light on D10 signifies light on D8 monitors gas percentage light, on D9.
- A red signal light has been installed at location D11 in case of any events.

5.6 IoT Navigation Diagram

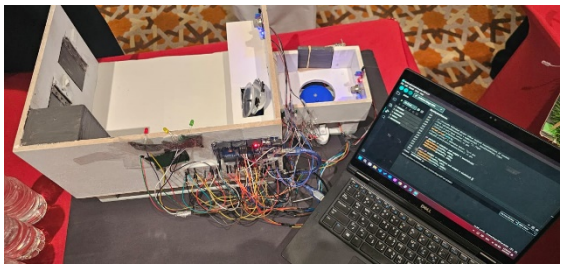


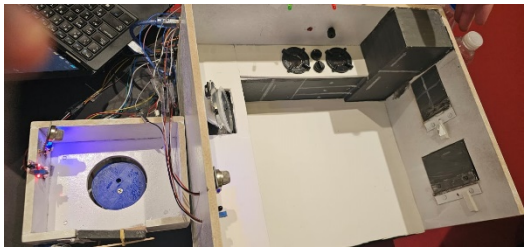
6. Implementation

6.1 Link sensors-actuators-MCU-SBC



6.2 Sample of project snapshots





7. Test and validate.

7.1 Final test cases

Test case id	Test case description	Expected result	Status
T1	Is Flame sensor work?	- flame sensor work -flame sensor detect fire	Pass Pass
T2	Is gas sensor work?	- gas sensor work - Gas sensor detect gas	Pass pass
T3	Is load sensor work?	- Load sensor work - Load sensor calculate the wight - Load sensor output is correct	Pass Pass Pass
T5	Merging the program codes together	- testing if program is working	Pass
T6	Connect nodeMCU to Arduino MEGA	- data will sends from ArduinoMEGA and nodeMCU - wifi is working	Pass Pass
T7	Blynk app connect to nodeMCU	- connecting nodeMCU to Blynk app - data from nodeMCU displayed on Blynk app	Pass Pass
T8	Connect servo motors	- the servo motors will open doors and windows	Fail
T9	Alert Notification	-after detect gas leakage blynk will show alert	Fail

8. Conclusion and future work

Conclusion:

The IoT-enabled gas monitoring, leak detection and automatic refill control project addresses critical issues in gas management and security, especially in regions like Oman. The system provides a complete solution for real-time monitoring and incident prevention by integrating the latest sensors such as MQ-2 for leak detection and HX711 strain gauge for gas level measurement. The system's ability to automatically schedule gas refills and provide information to users through a mobile app enhances convenience and safety for homes and businesses.

The main features of our project - accurate gas level monitoring, quick leak detection, automatic reservation of gas stations and user notifications via mobile application - were successfully demonstrated in a prototype. By combining these functions into a single system, gas control technology has advanced significantly and raised the standards of operational efficiency and safety.

future work:

field testing and validation:

- Conduct extensive real-world testing to validate the system's performance in various environments and conditions.
- Collect feedback from the users to reveal any possible areas of improvement and ensure that the system meets their requirements.

Mobile app enhancements:

- Improve the mobile app's user design and experience to make it more intuitive and user-friendly.
- Incorporate extra features like predictive maintenance warnings and historical data analysis.

Advanced analytics and AI:

- To increase the precision of leak detection and predictive maintenance, we're intend to integrate AI algorithms and advanced data analytics.

- Also to optimize gas consumption, examine usage trends using machine learning models.

9. Roles and contribution the student group

Project manager – Ali Al-Belushi:
Ali coordinated all team meetings, set team tasks, and ensured timely completion of works. He was responsible for maintaining project schedules.

Lead researcher – Loay Al-Sinani, Abdulaziz Al-Kalbani, Osama Al-Kindi:
Loay, Abdulaziz, and Osama conducted extensive research on existing gas detection system and conduct a survey for taking real user experience with traditional safety and gas leveling ways, also they work on determining the requirements.

Hardware developers - Loay Al-Sinani and Abdulaziz Al-Kalbani:
Loay and Abdulaziz were responsible for selecting and integrating the hardware requirement. They test and ensure that all sensors are connected write and they can work together.

Software and mobile app developer - Ali Al-Belushi and Loay Al-Sinani:
Ali work on making all sensors, actuators and MCUs works together by make the software and adding the condition for them, also he work on connecting the nodeMCU to Arduino IDE and send the date from Arduino nodeMCU to Blynk app. Loay make all the functunality of blynk app and prepare it for connecting with nodeMCU and displaying the sensor values, also he makes the user graphical interface for blynk app.

Tester – Ali Al-Balushi and Loay Al-Sinani:
Ali and Loay separate the work between each other and test all component of the system separately and then test the full system as full system.

Documentation specialist – Abdulaziz Al-Kalbani and Osama Al-Kindi:
Abdulaziz and Osama work on documenting every work the team was doing, until finally they give as this document that, has all work the team do, will do, and couldn’t do.

10. Appendices

The load cell sensor, known as HX711 is crucial, for measuring gas levels. It communicates through an interface. Works with an operating voltage between 2.7 V and 5.5 V offering an output data rate of, up to 80 Hz.. The HX711 sensor datasheet is available at https://www.digikey.com/htmldatasheets/production/1836471/0/0/1/hx711.html.

Our project combines an Arduino Mega microcontroller with an Arduino mega2560 MCU. It runs on 5 volts. Comes with 54 I/O pins 16 analog input pins and 256 kilobytes of flash memory. The Arduino Mega datasheet is available at https://docs.arduino.cc/resources/datasheets/A000067-datasheet.pdf

We also make use of a NodeMCU microcontroller that relies on the chip. Running on 3.3 volts it offers 11 I/O pins, 1 analog input pin. Has a flash memory capacity of 4 MB. The NodeMCU facilitates connectivity, for your system with its support, for Wi Fi 802.11 b/g/n.. The NodeMCU datasheet is available at https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet

11. References

1. V.Mariselvam and Dharshini, IoT based level detection of gas for booking management using integrated sensor (2021) https://www.sciencedirect.com/science/article/abs/pii/S2214785320344400

2. PUNEETH, SAI ROHITH, and PREMA, Gas Level Detection and Automatic Booking Using IoT 2022, https://www.irjet.net/archives/V9/i3/IRJET-V9I3158.pdf

3. V. Tamizharasan, T. Ravichandran, M. Sowndariya, R. Sandeep, and K. Saravanavel, Gas Level Detection and Automatic Booking Using IoT (2019) https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8728532

4. ilteo85, Arduino forum, Apr 2020, HX711 with load cell calibration https://forum.arduino.cc/t/hx711-with-load-cell-calibration/650485

5. ArduinoModules, KY-026 FLAME SENSOR MODULE, Dec 3, 2021, https://arduinomodules.info/ky-026-flame-sensor-module/

6. INOVATRIX, How to use MQ2 Gas Sensor with Arduino and Serial Monitor, Apr 2023, https://www.youtube.com/watch?v=8DQQDpaZYj8

7. Robotica DIY, Send Data From Arduino to NodeMCU and NodeMCU to Arduino Via Serial Communication, May 5, 2020, https://www.youtube.com/watch?v=EfzZOiNBQmI