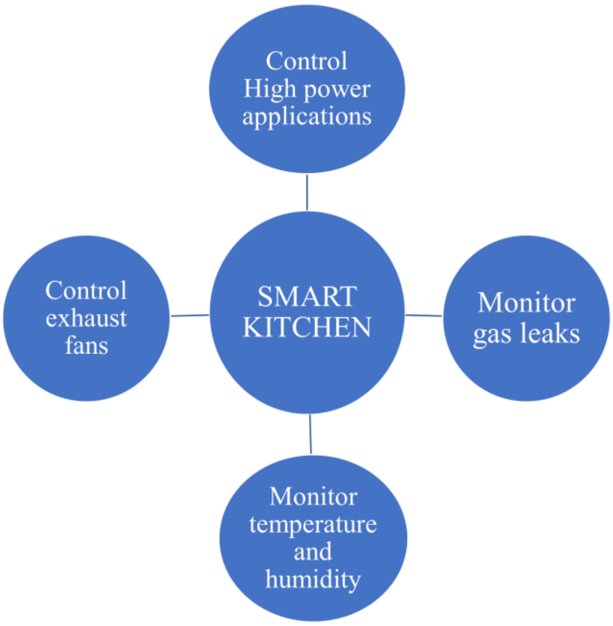
Assignment 2	**IOT based Smart Kitchen Automation and Monitoring System**
20/04/2023	

Aim

To make a prototype of IOT based Smart Kitchen Automation and Monitoring System using NodeMCU and Blynk application.


Control High power applications
Control exhaust fans
SMART KITCHEN
Monitor gas leaks
Monitor temperature and humidity

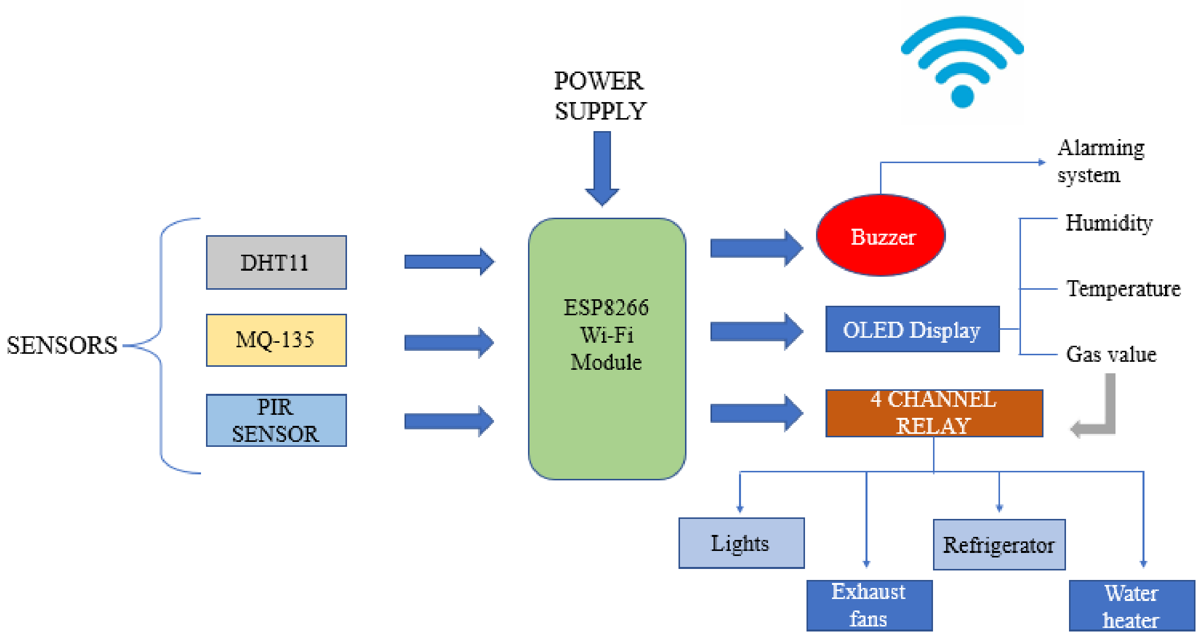
Components Required

S No	Hardware	Software
1.	ESP8266 Wi-Fi Module	Arduino IDE
2.	DHT11 Sensor	Wokwi
3.	MQ-135 Air Quality Sensor	Blynk
4.	0.96" I2C OLED Display	
5.	Buzzer 5V	
6.	4 Channel Relay Module	
7.	PIR Sensor	
8.	Exhaust Fan	

Literature Survey

Home automation refers to the incorporation of technology into the home environment to enhance convenience, comfort, security, and energy efficiency for its inhabitants. In recent times, the advent of IoT has resulted in an increase in research and implementation of home automation. To embed intelligence in the home, several wireless technologies, including Bluetooth, Wi-Fi, RFID, and cellular networks, have been utilized to support remote data transfer, sensing, and control. Many research and projects have been made regarding this smart kitchen, However, most of them only suggest a concept of working of a smart kitchen. Some researches that succeeded in making a system that can work but still use ATmega32 as a microcontroller with a limited accessibility. Here we are going to use NodeMCU ESP8266 WIFI board and connect our smartphones with our kitchen.

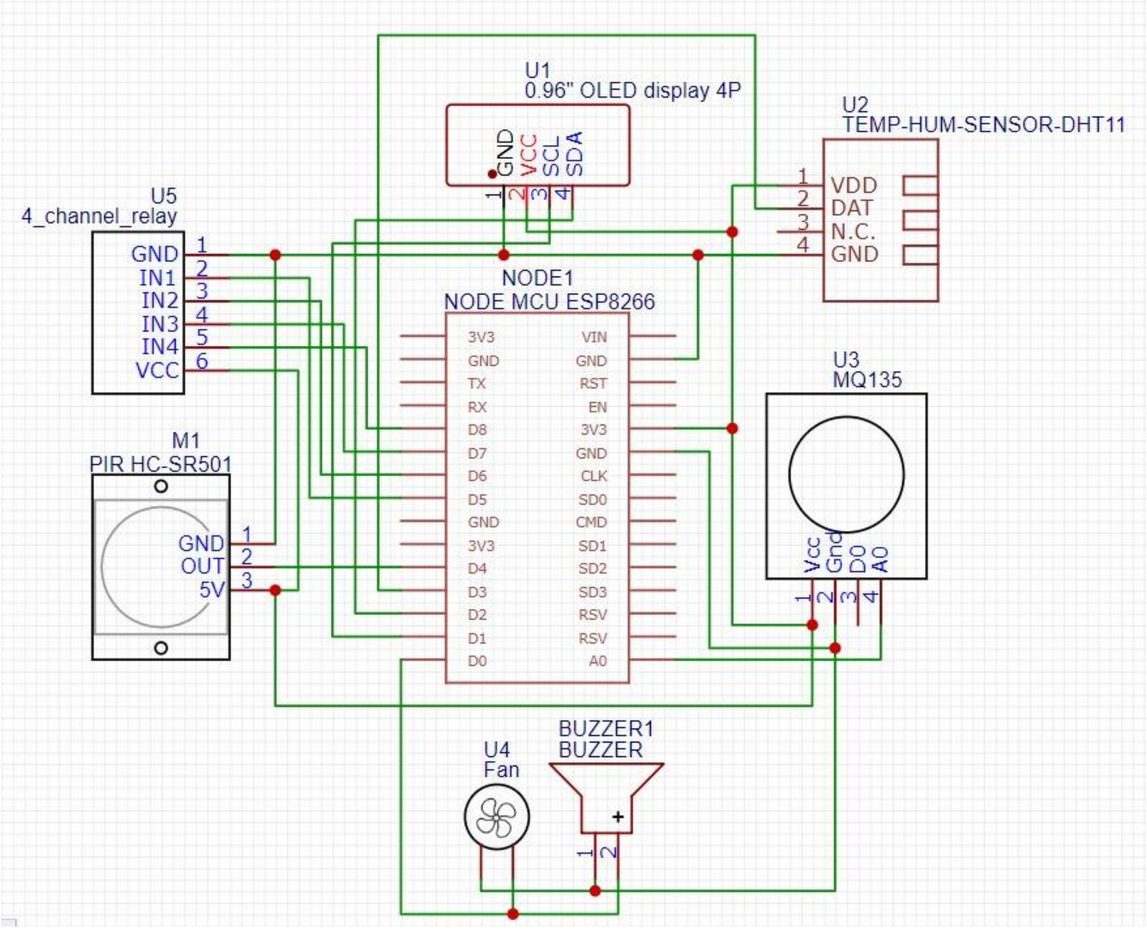
Methodology



Based on background and review from research that has been described, we need a system that can monitor and send various information about the condition of the kitchen continuously, even though we are in a place far from the kitchen. A system which can detect changes in temperature and fire caused by the use of gas stoves in the kitchen. In this system, the DHT11 sensor is mounted to detect temperature changes, an MQ-135 gas sensor to detect Liquefied Petroleum Gas

(LPG) leak and passive infrared (PIR) sensors to detect human activity in the kitchen. In this system, there is also a relay to control the fan that serves to control the temperature and blow out of gas in the event of a gas leak or smoke from the kitchen in case of fire. This system can be controlled and monitored via the internet directly from laptops or smartphones anytime, even from a place far from the kitchen.

Work Proposed



- DHT11 Humidity Temperature Sensor, MQ-135 Gas Sensor, Passive IR sensor is used to monitor the Indoor Air Quality Parameters.
- 5V buzzer can work as an alarming system.
- Using a 4-channel relay, an automatic exhaust Fan is connected which activates automatically when the gas level exceeds the threshold value. The remaining 3 relays can be connected to kitchen appliances like Mixer, Refrigerator, Oven, Water Heater, Induction stove, etc.
- A 0.96″ I2C OLED is used to display room temperature, humidity, and air

quality.

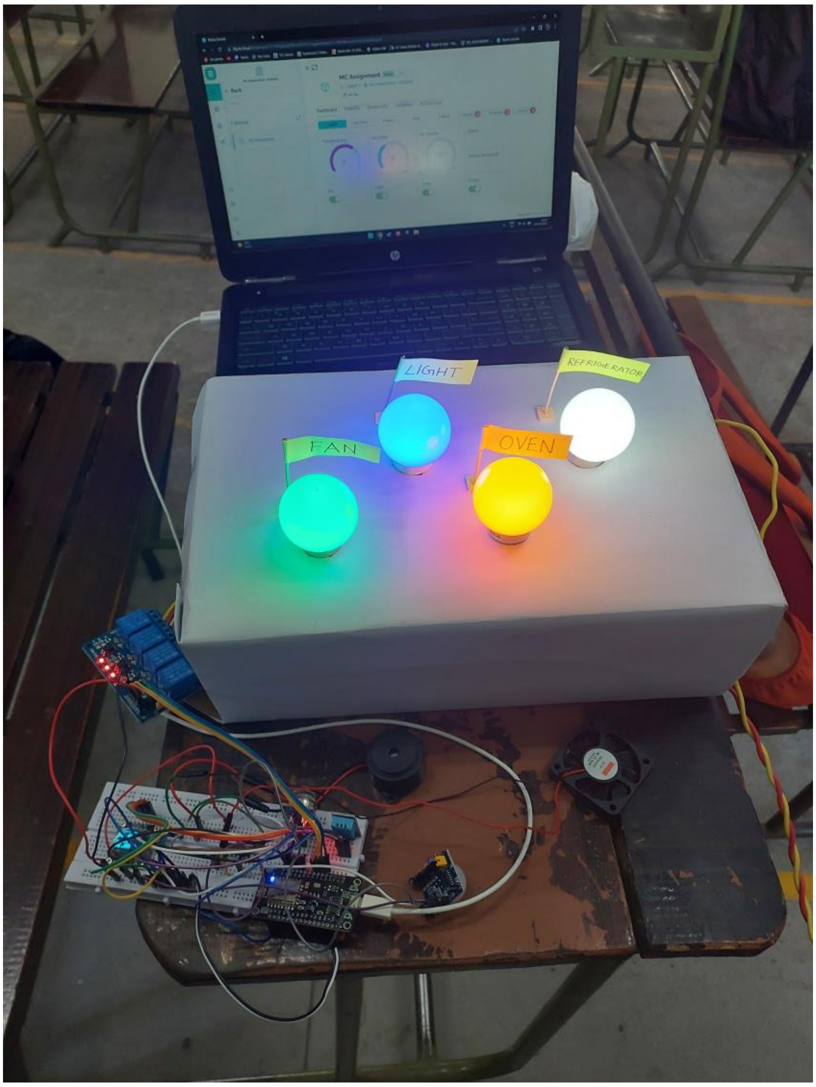
- We will use a NodeMCU ESP8266 Board. The ESP8266 chip connects to the WiFi Network and establishes a connection with Blynk Application.

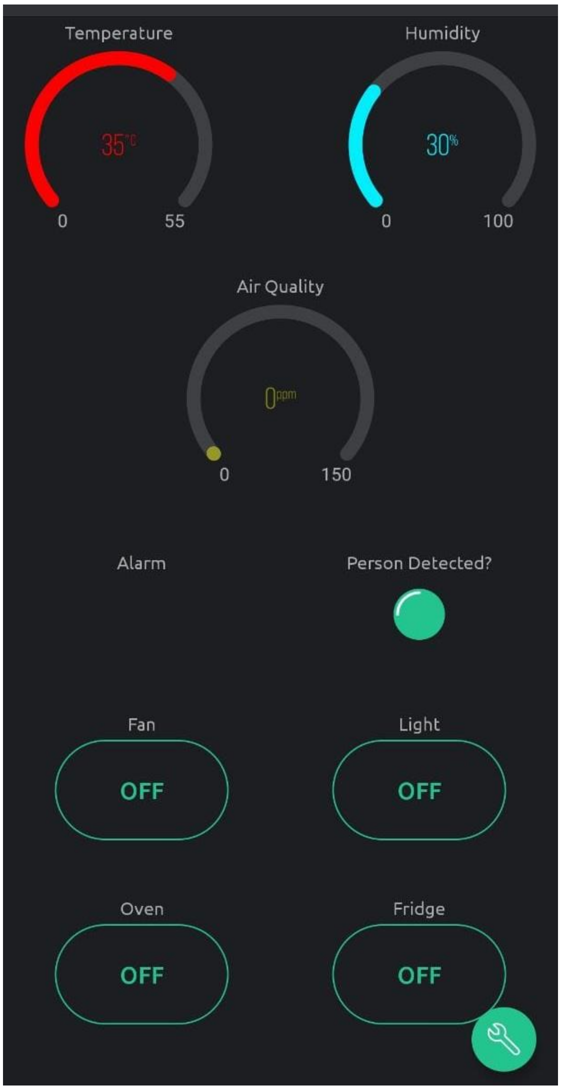
Program

```
// Connections
// OLED D1 - SCL, D2 - SDA
// DHT D3
// MQ135 A0
// PIR D4
// EXHAUST FAN, BUZZER D0
// RELAYS FAN D5, LIGHT D6, OVEN D7, FRIDGE D8

#include <DHT.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_Sensor.h>
#include <MQ135.h>
#include <BlynkSimpleEsp8266.h>
#include <ESP8266WiFi.h>

// Blynk
#define BLYNK_AUTH_TOKEN "p3Fr3mOaKXGoozm_eaKgujPN4uFyAgq4"

// WIFI
#define SSID "Logesh S"
#define PW "87654321"

// Blynk Virtual Pins
// V0 Temperature, V1 Humidity, V2 AQ, V3 Fan, V4 Light
// V5 Oven, V6 Fridge, V7 Alarm, V8 Person Detected

// Relay
#define FAN D5
#define LIGHT D6
#define OVEN D7
#define FRIDGE D8
```

```
// Time delay in ms
const int DELAY = 2000;

// OLED
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

// DHT
#define DHT_PIN D3
#define DHT_TYPE DHT11
DHT dht(DHT_PIN, DHT_TYPE);

// MQ135
#define MQ_PIN A0
MQ135 gasSensor = MQ135(MQ_PIN);
int ALARM_STATE;
const int AQ_LIMIT =  5;

// Exhaust Fan and Buzzer
#define ALARM D0

// PIR
#define PIR_PIN D4
int PIR_VALUE = 0;

// First time setup
void setup()
{
  Serial.begin(9600);

  Blynk.begin(BLYNK_AUTH_TOKEN, SSID, PW);

  // Configure IO pins
  pinMode(PIR_PIN, INPUT);
  pinMode(ALARM, OUTPUT);
  pinMode(FAN, OUTPUT);
  pinMode(LIGHT, OUTPUT);
  pinMode(OVEN, OUTPUT);
  pinMode(FRIDGE, OUTPUT);

  digitalWrite(ALARM, LOW);
  digitalWrite(FAN, HIGH);
  digitalWrite(LIGHT, HIGH);
  digitalWrite(OVEN, HIGH);
  digitalWrite(FRIDGE, HIGH);

  dht.begin();
  // Loop infinitely until connection is established with OLED
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C))
  {
    Serial.println(F("SSD1306 allocation failed"));
    for(;;);
  }

  delay(DELAY);

  display.clearDisplay();
  display.setTextColor(WHITE);
}

// Blynk State Syncing
BLYNK_CONNECTED()
{
  Blynk.syncVirtual(V3);  // will cause BLYNK_WRITE to be executed
  Blynk.syncVirtual(V4);
  Blynk.syncVirtual(V5);
  Blynk.syncVirtual(V6);
}

// Repeatedly call
void loop()
{
  Blynk.run();

  dht_run();
  delay(DELAY);
  display.clearDisplay();

  mq_run();
  delay(DELAY);

  pir_run();

  display.clearDisplay();
}

void dht_run()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  if (isnan(h) || isnan(t))
  {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }

  Serial.print("Temperature in C: ");
  Serial.println((dht.readTemperature()));
  Serial.print("Humidity in %: ");
  Serial.println((dht.readHumidity()));

  // Blynk display
  Blynk.virtualWrite(V0, t);
  Blynk.virtualWrite(V1, h);

  // OLED Display temperature
  display.setTextSize(1);
  display.setCursor(0, 0);
  display.print("Temperature: ");
  display.setTextSize(2);
  display.setCursor(0, 10);
  display.print(t);
  display.print(" ");
  display.setTextSize(1);
  display.cp437(true);
  display.write(167);
  display.setTextSize(2);
  display.print("C");

  // OLED Display humidity
  display.setTextSize(1);
  display.setCursor(0, 35);
  display.print("Humidity: ");
  display.setTextSize(2);
  display.setCursor(0, 45);
  display.print(h);
  display.print(" %");

  display.display();
}

void mq_run()
{
  float AQI = gasSensor.getPPM();
  Serial.print("Air Quality in PPM: ");
  Serial.println(AQI);

  // Blynk display air quality
  Blynk.virtualWrite(V2, AQI);

  // OLED display air quality
  display.setTextSize(1);
  display.setCursor(0, 0);
  display.print("Air Quality: ");
  display.setTextSize(2);
  display.setCursor(0, 15);
  display.print(AQI);
  display.print(" PPM");

  if(AQI > AQ_LIMIT)
  {
    Serial.println("Smoke Detected!!!");
    display.setTextSize(1);
    display.setCursor(0, 45);
    display.print("Smoke Warning!");
    digitalWrite(ALARM, HIGH);
    Blynk.virtualWrite(V7, 1);
  }
  else
  {
    digitalWrite(ALARM, LOW);
    Blynk.virtualWrite(V7, 0);
  }

  display.display();
}

void pir_run()
{
  PIR_VALUE = digitalRead(PIR_PIN);

  if (PIR_VALUE == 1)
  {
    Serial.println("Motion Detected");
    Blynk.virtualWrite(V8, 1);
  }
  else if (PIR_VALUE == 0)
  {
    Serial.println("No Motion");
    Blynk.virtualWrite(V8, 0);
  }
}

// Fan
BLYNK_WRITE(V3) // Executes when the value of virtual pin changes
{
  if(param.asInt() == 1)
  {
    // execute this code if the switch widget is now ON
    digitalWrite(FAN, LOW);  // Set digital pin 2 HIGH
  }
  else
  {
    // execute this code if the switch widget is now OFF
    digitalWrite(FAN, HIGH);
  }
}

// Light
BLYNK_WRITE(V4) // Executes when the value of virtual pin changes
{
  if(param.asInt() == 1)
  {
    // execute this code if the switch widget is now ON
    digitalWrite(LIGHT, LOW);  // Set digital pin 2 HIGH
  }
  else
  {
    // execute this code if the switch widget is now OFF
    digitalWrite(LIGHT, HIGH);
  }
}

// Oven
BLYNK_WRITE(V5) // Executes when the value of virtual pin changes
{
  if(param.asInt() == 1)
  {
    // execute this code if the switch widget is now ON
    digitalWrite(OVEN, LOW);  // Set digital pin 2 HIGH
  }
  else
  {
    // execute this code if the switch widget is now OFF
    digitalWrite(OVEN, HIGH);
  }
}

// Fridge
BLYNK_WRITE(V6) // Executes when the value of virtual pin changes
{
  if(param.asInt() == 1)
  {
    // execute this code if the switch widget is now ON
    digitalWrite(FRIDGE, LOW);  // Set digital pin 2 HIGH
  }
  else
  {
    // execute this code if the switch widget is now OFF
    digitalWrite(FRIDGE, HIGH);
  }
}
```

Results





Temperature 35°C (0–55)

Humidity 30% (0–100)

Air Quality 0ppm (0–150)

Alarm

Person Detected?

Fan OFF

Light OFF

Oven OFF

Fridge OFF

The proposed work of developing a smart kitchen system has been successfully implemented, and a mobile application using Blynk has been created to control various kitchen appliances and monitor important kitchen parameters. This system is designed to improve the convenience, safety, and energy efficiency of the kitchen environment, and to provide real-time information on key parameters such as temperature, humidity, and gas levels. The system composed of various sensors, including a DHT11 humidity and temperature sensor, an MQ-135 gas sensor, and a passive infrared sensor, which continuously monitor indoor air quality parameters and detect any potential safety hazards such as gas leaks or smoke from the kitchen. An automatic exhaust fan controlled by a 4-channel relay is also integrated into the system to maintain proper ventilation and to quickly expel any harmful gas or smoke. Moreover, the system can be easily controlled and monitored through the mobile application created using Blynk. The application displays real-time information on temperature, humidity, and gas levels, and allows users to remotely control various kitchen appliances such as mixers, refrigerators, ovens, water heaters, and induction stoves. Users can also receive alerts through the application if any safety hazards are detected, and can take immediate action to prevent any potential harm.

Resources

- Design of Smart Kitchen Management System Using IOTs - https://www.irjmets.com/uploadedfiles/paper/issue_7_july_2022/28317/final/fin_irjmets1658136599.pdf
- IoT based system for detection of gas leakage and house fire in smart kitchen environments - https://ieeexplore.ieee.org/abstract/document/8971021
- Smart Kitchen: Real Time Monitoring of Kitchen through IoT - https://ieeexplore.ieee.org/abstract/document/9853161
- Smart Kitchen System using IOT - https://www.ijeast.com/papers/378-383,Tesma411,IJEAST.pdf
- IOT Based Smart Kitchen Automation And Monitoring System - https://www.irjmets.com/uploadedfiles/paper/issue_6_june_2022/27173/final/fin_irjmets1656515544.pdf
- Internet of things based system for Smart Kitchen - http://www.ttcenter.ir/ArticleFiles/ENARTICLE/3838.pdf
- Automatic and Monitoring Smart Kitchen Based on Internet Of Things - https://iopscience.iop.org/article/10.1088/1757-899X/384/1/012007/pdf
- IOT Based Smart Kitchen Pantry - https://iijsr.com/data/uploads/99958.pdf