EE5305 – Sensors and Transducers

Smart Kitchen Monitoring System

Project Report

Department of Electrical and Information Engineering

Faculty of Engineering

University of Ruhuna

Contents

Project Title

Smart Kitchen Monitoring System.

Objective

The objective of this project is to design and implement a smart kitchen monitoring system using the NodeMCU ESP32 microcontroller. The system will monitor key parameters in the kitchen such as temperature, humidity and gas levels and enable monitoring over essential appliances, enhancing kitchen safety and convenience.

Project Description

This ESP32-based Smart Kitchen System integrates sensors with two online dashboards for real-time monitoring. It provides key functions such as gas leak detection, automatic triggering of kitchen appliances according to threshold values of temperature, humidity and gas PPM values. It also provides remote monitoring using the ThingsBoard Demo Version and it can also provide real time system data using the ESP-Dash Web Interface without requiring an internet connection in which the users can monitor kitchen conditions when they are inside the house. The ThingsBoard can be used to monitor the system data by connecting to the internet and the users can track the system status and be on alert at anytime from anywhere. The system alerts users through an onboard alarm when at least one of the threshold parameter values is exceeded such as temperature, humidity or the gas level in the kitchen and it also activates a relay module to switch off the connected electrical appliances such as light bulbs, refrigerators, ovens, rice cookers, heaters etc. This project demonstrates a practical and effective way to manage kitchen safety and energy efficiency through IoT technology.

Project Scope

<u>Temperature and Humidity Monitoring</u>

Uses the DHT11 sensor to display kitchen temperature and humidity on an OLED screen and web dashboards.

Figure 1: DHT11 Temperature and Humidity Sensor

Figure 2: 0.96" OLED Display

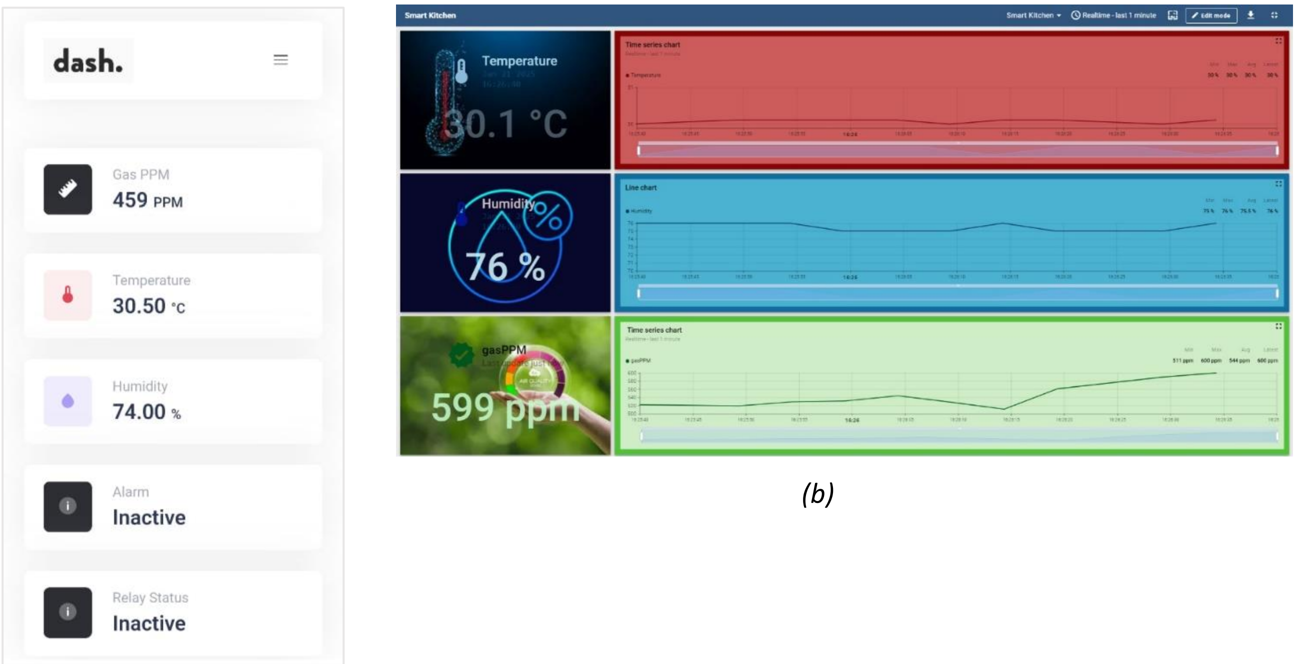
Gas Detection and Alarm

Employs the MQ-135 gas sensor to monitor air quality. If gas levels exceed the threshold, the system triggers the relay.

Figure 3: MQ-135 Gas Sensor

Web Dashboards for Monitoring and Control

A mobile-accessible web dashboard allows users to view kitchen status and monitor appliances in real time.



(a) *(b)*

Figure 4: (a) ESP-Dash Web interface (b) ThingsBoard Demo Version Web Interface

Methodology

Micro controllers, Sensors and Electronic Modules Required

NodeMCU ESP32	Main controller and Wi-Fi interface
DHT11 Temperature and Humidity Sensor	Measures kitchen temperature and humidity levels
MQ-135 Gas Sensor	Detects gas levels and provides air quality information
0.96" I2C OLED Display	Displays temperature, humidity, and gas levels locally
Relay module	Triggering electrical appliances to turn off for safety
5V Buzzer	Sounds an alarm in case of high temperature or gas levels
Logic level converter	Convert 3.3v levels to 5V signals to operate Relay

Design Implementation of the Circuit in Easy-EDA Software

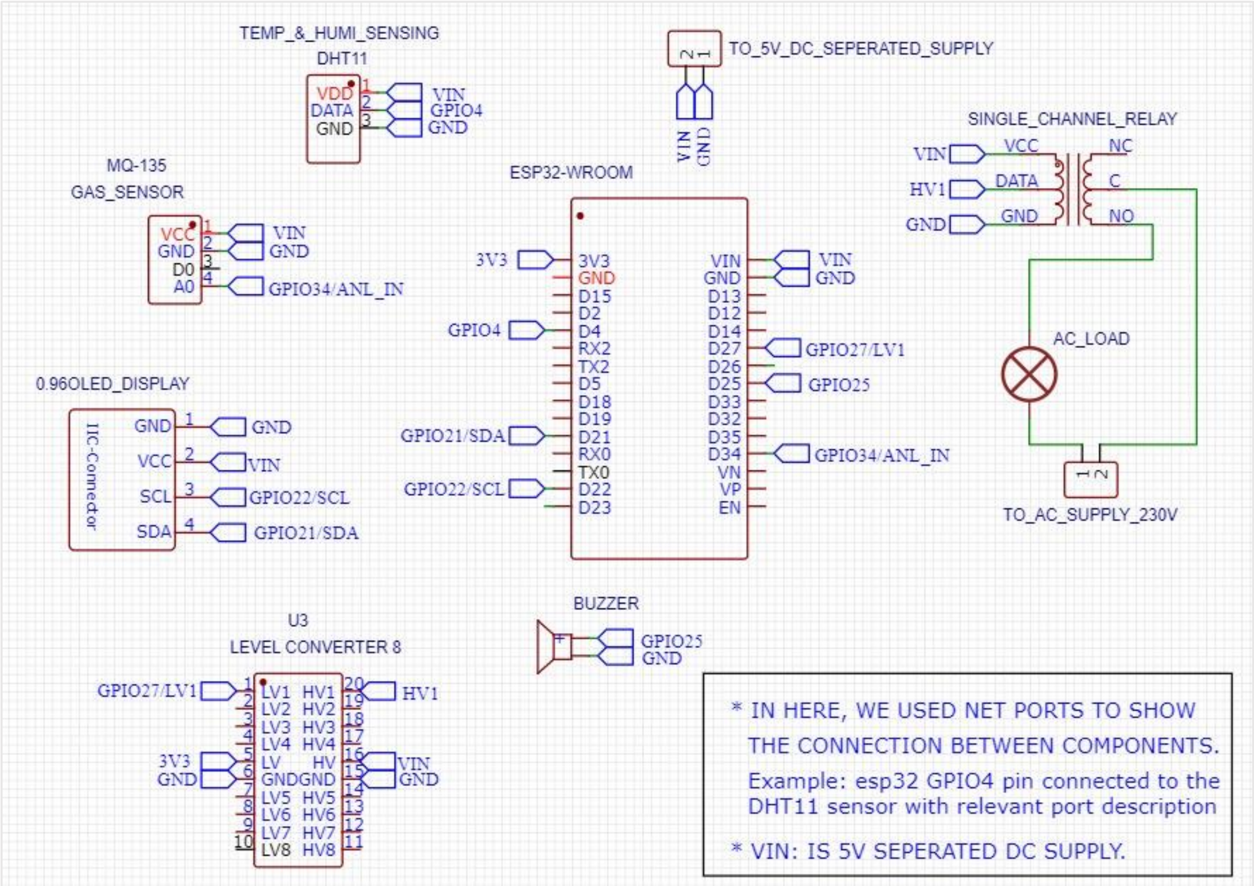


Figure 5: Designed Circuit Using the Easy-EDA Platform

Arduino Code

Below is the GitHub repository URL of the Arduino code implemented in the microcontroller.

https://github.com/Deshan-Lokuge01/IoT-based-Kitchen-Monitoring-System

Libraries Used in the Code:

1. WiFi.h: Provides functions to manage Wi-Fi connectivity in the ESP32. This library is used to connect to a Wi-Fi network (for ThingsBoard Interface functionality) and also create an Access Point (for ESP-Dash interface functionality) for local communication.
2. PubSubClient.h: This is a lightweight MQTT library for ESP32. This library is used to publish telemetry data (temperature, humidity and gas levels) to the ThingsBoard IoT platform.
3. Wire.h: This is used to handle I2C communication to enable the ESP32 to interact with devices like SSD1306 OLED display.
4. Adafruit_GFX.h: This is a graphics library for rendering shapes, text and images. This library is used as a dependency for the OLED display to format and render data visually.
5. Adafruit_SSD1306.h: This library is used for controlling SSD1306 OLED display. This enables the display to show real time temperature, humidity and gas level on the screen.
6. Adafruit_Sensor.h: This library is used for the integration between the sensors. This is used as dependency for the DHT sensor library to manage temperature and humidity readings.
7. DHT.h: This library works as an interface with DHT11 temperature and humidity sensor. It reads and processes temperature and humidity data from the DHT11 sensor.
8. MQ135.h: This library is used to handle the MQ-135 gas sensor. It converts analog gas sensor data into usable values like PPM (parts Per Million).
9. AsyncTCP.h: This library provides asynchronous TCP support for the ESP-Dash library in order to handle efficient communication.
10. ESPAsyncWebServer.h: This library is used to create an asynchronous web server on the ESP32. This is used to host the ESP-Dash interface for real time local monitoring.
11. ESPDash.h: This is a lightweight library for designing web-based dashboards on the ESP32. This is used to create the dashboard showing temperature, humidity, gas levels and device statuses like alarm and the relay.

Working Principle

The system uses the Wi-Fi capabilities of ESP32 board to create a local web server accessible through any Wi-Fi-enabled device. When the system detects a gas leak, it activates an alarm. Temperature, humidity and gas data are displayed on an OLED screen and the web dashboard. Additionally, users can turn control from the web interface, allowing for easy, remote monitoring of kitchen equipment.

Conclusion

This project is expected to deliver critical household kitchen safety protocols such as real-time monitoring of kitchen environment metrics (temperature, humidity, gas levels) through an OLED display and web dashboards, immediate alert and control in case of gas leakage, remote monitoring of kitchen conditions via a user-friendly web interface.