IOT & Hardware Documentation

for the Design and Implementation of a Secure, Decentralized Early Warning System for Disasters and Crises

Prepared by

Reem Nazeh – IoT & Hardware Track Lead
Computer Science Department – Future Academy
Academic Year: 2024/2025

About This File

This documentation focuses specifically on the **IoT and Hardware Implementation** part of the graduation project. The complete system consists of multiple integrated tracks (Sensor Network , AI, Backend, Frontend, Mobile,Web , Smart Contract), and this document explains the circuit design, sensor configuration, and embedded software development done using ESP32 and Arduino IDE.

Executive Summary:

The goal of EWS (Early Warning Systems) is to detect problems early and enable rapid, automated responses to minimize risks.

The system relies on environmental sensors to collect real-time data. This data is processed locally to trigger an initial response, then passed to an AI model that validates the alert's accuracy before forwarding it to the software applications.

---

1. Project Overview

A real-time monitoring system using ESP32 to detect:

- Toxic gases (MQ2)
- Carbon Monoxide (MQ7)
- Temperature/Humidity (DHT11)
- Flame (IR sensor)

Data is streamed to Blynk Web/Mobile Dashboards and a custom API for alerts.

---

2. Hardware Components

Component	Specification	Purpose
ESP32	Wi-Fi	Main controller
MQ2 Gas Sensor	Analog output (LPG, smoke)	Detect flammable gases
MQ7 Gas Sensor	Analog output (CO)	Detect carbon monoxide
DHT11	Digital output	Measure temp/humidity
Flame Sensor	Digital output (HIGH on flame)	Fire detection
LED	5mm Red	Visual alarm

Wiring Diagram
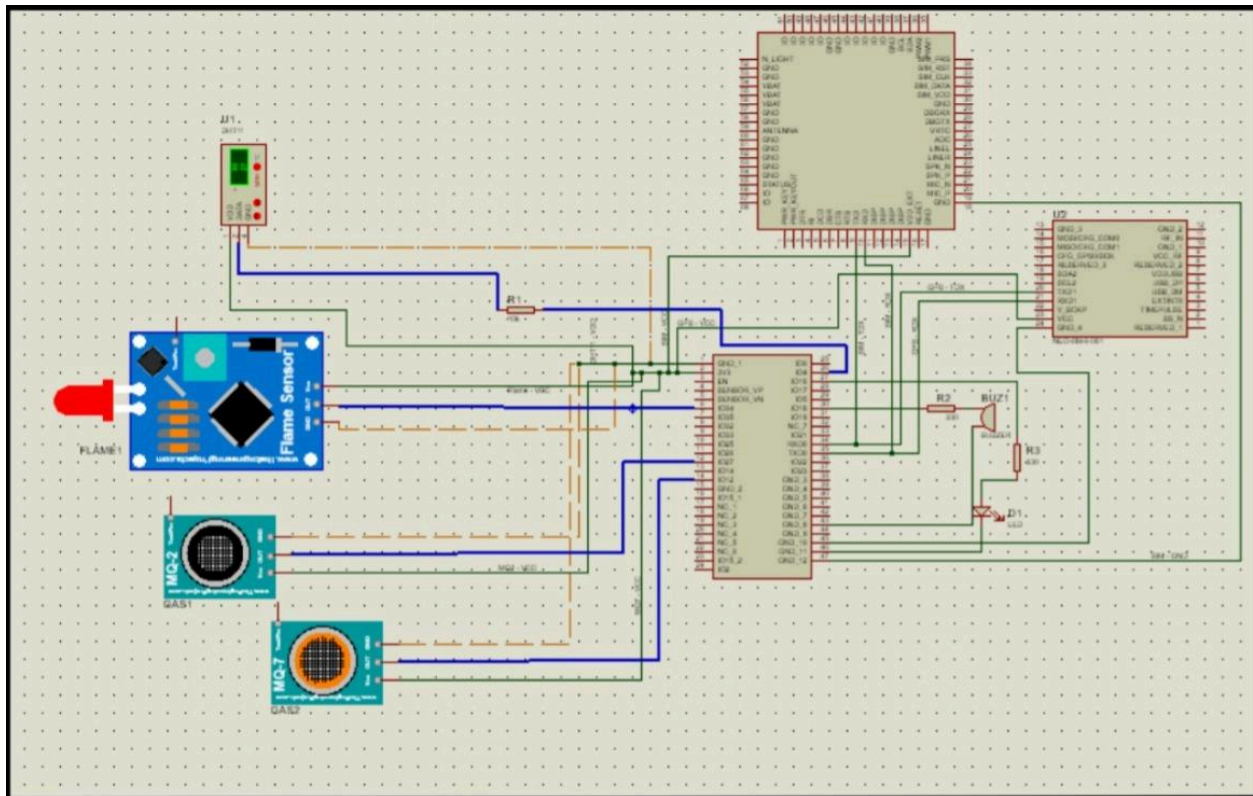
ESP32 Pins:

- MQ2 → GPIO34 (Analog)
- MQ7 → GPIO35 (Analog)
- DHT11 → GPIO22
- Flame Sensor → GPIO13 (Digital)
- LED → GPIO23 (with 220Ω resistor)

---

3. Software Requirements

- **Arduino IDE** (v2.3.2+)
- **Libraries**:
  - `BlynkSimpleEsp32`
  - `DHT sensor library`
  - `MQUnifiedsensor`
  - `Arduinojson`
  - `WiFi`
- **Blynk IoT App** (Android/iOS)
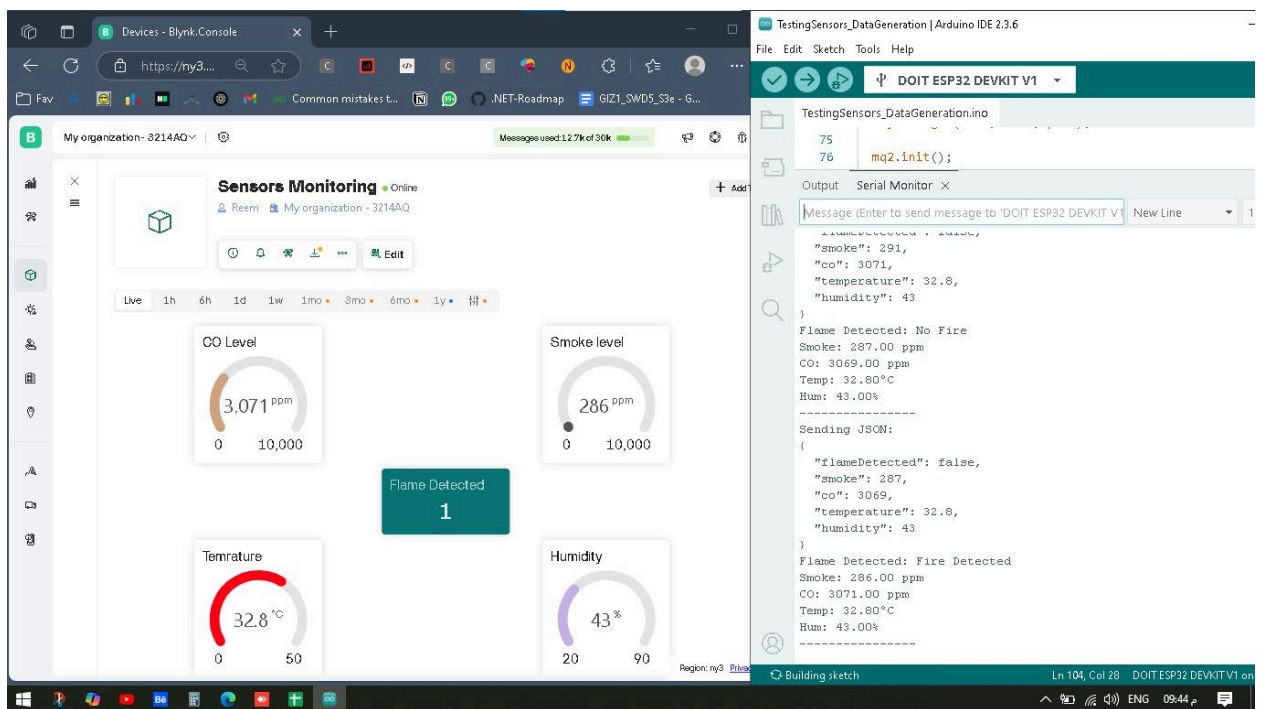- **Blynk.Console WebDashboard**

---

4. Troubleshooting

Issue	Solution
No Blynk connection	Check Auth Token/Wi-Fi signal

MQ7 always reads 0	Verify analog pin & 5V power

5. Blynk IoT

- ● **WebDashBoard**

- **Hardware Testing**

```
24 ->     "smoke": 582,
24 ->     "co": 4095,
24 ->     "temperature": 29.8,
24 ->     "humidity": 43
24 -> }
046 -> API Response: {"receivedData":{"id":0,"temperature":29.8,"humidity":43,"smokeLe
100 -> Flame Detected: Fire Detected
100 -> Smoke: 555.00 ppm
100 -> CO: 4095.00 ppm
100 -> Temp: 29.80°C
100 -> Hum: 43.00%
100 -> ----------------
```
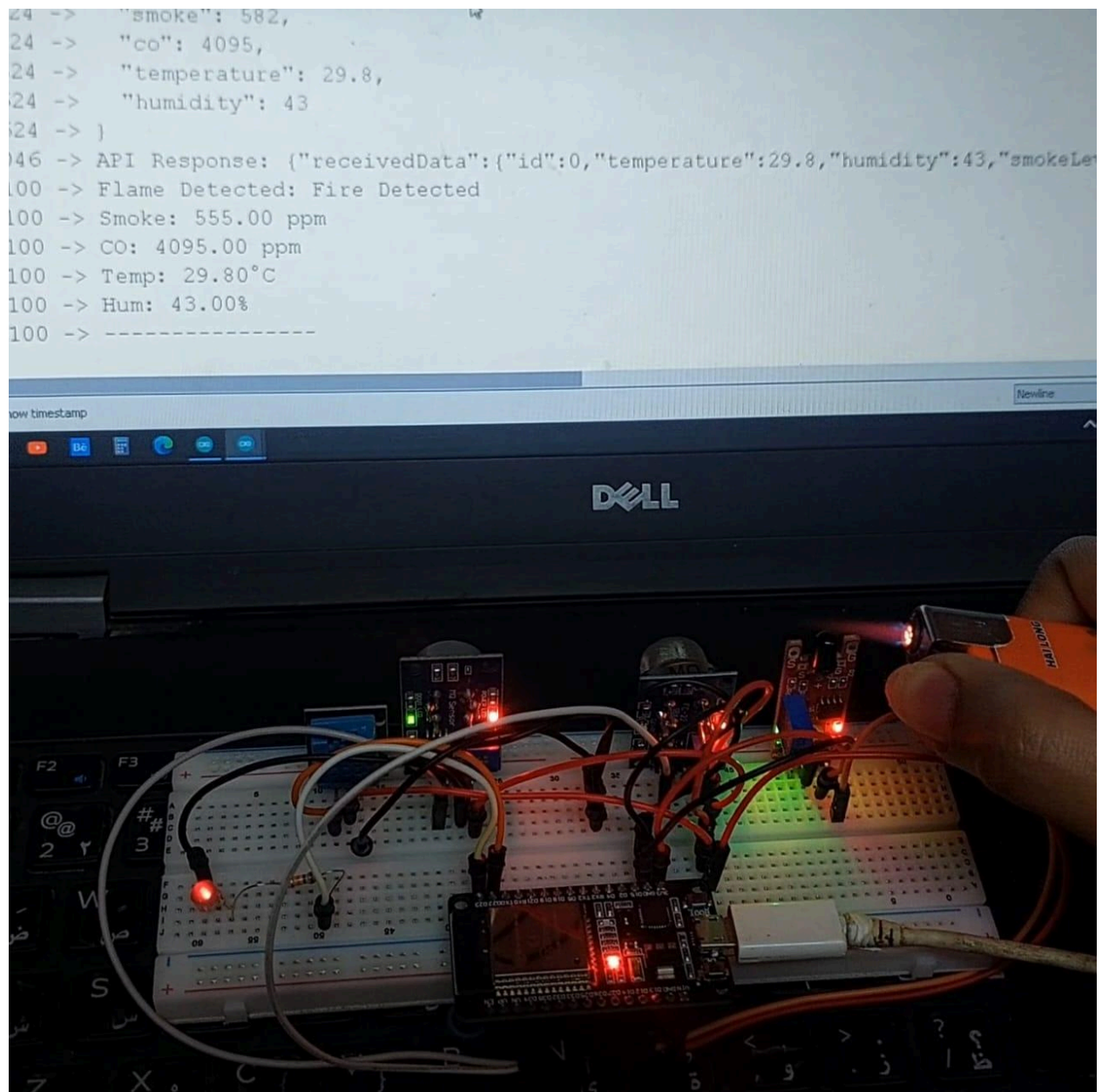
- **MobileApp**
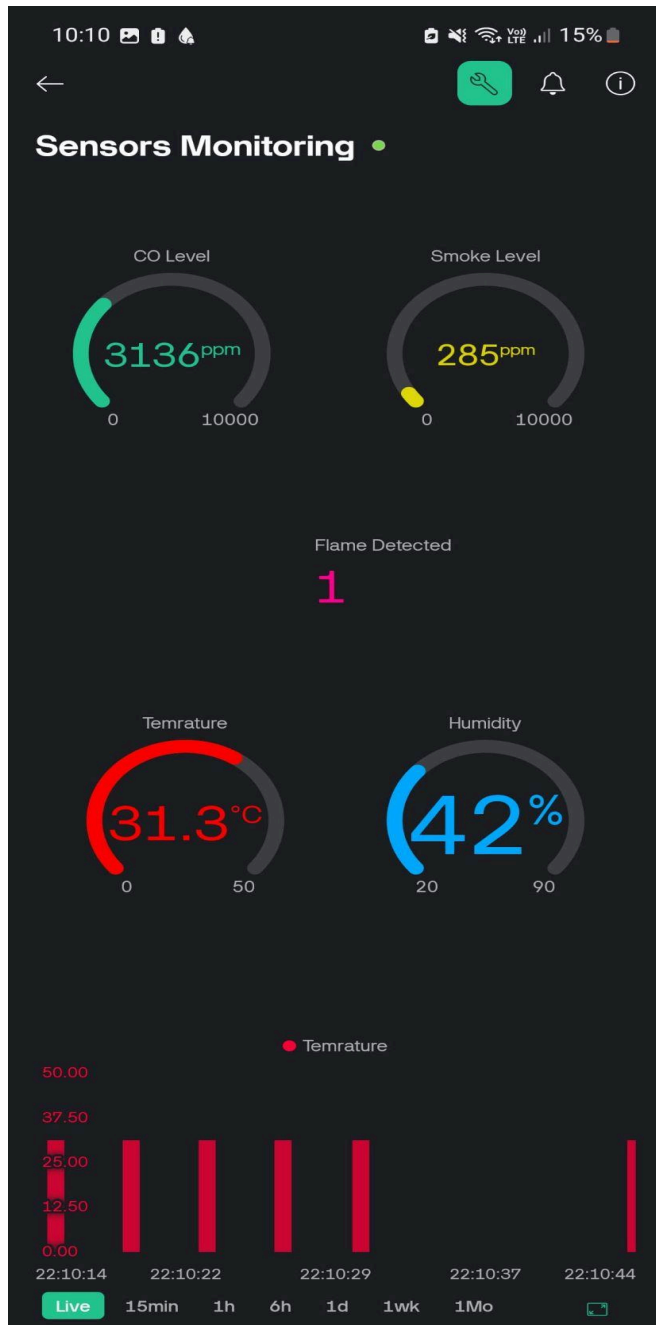
6. References

- **MQ2 Datasheet [https://www.pololu.com/file/0J309/MQ2.pdf]**
- **MQ7 DataSheet [https://cdn.sparkfun.com/assets/b/b/b/3/4/MQ-7.pdf]**
- **DHT11 DataSheet [https://www.alldatasheet.com/datasheet-pdf/pdf/1132088/ETC2/DHT11.html]**
- **Blynk Documentation[https://docs.blynk.io/]**
- **GitHub Repository[https://github.com/your-repo]**

🏆Project Achievements

This system is part of a graduation project that was awarded **First Place** in the qualifying round of **the Summit Forum for Scientific and Engineering** Innovations at the level of **the Ministry of Higher Education Institutes.**

📄Full Project Documentation.

Full Repo Link

📩 Contact

Reem Nazeh

reem1nazeh@gmail.com | GitHub: Reemnazeeh | LinkedIn: Reem Nazeh

📌 Final Note

This IoT subsystem is a crucial component of a larger, integrated solution addressing disaster and crisis management.
It combines embedded systems, real-time data processing, and wireless communication to provide actionable alerts and support rapid decision-making.

For more technical details, full system design, and sub systems integrations , please refer to the full project documentation or contact the team via the links above