Smart Water/Fuel Monitoring System

Revanshu Pusadkar
Electronics & Computer Science
Ramdeobaba University
Nagpur, India
revanshu444@gmail.com

Mohommad Moinuddin
Electronics & Computer Science
Ramdeobaba University
Nagpur, India
24moinuddinm@rbunagpur.in

Zaid Sheikh
Electronics & Computer Science
Ramdeobaba University
Nagpur, India
24sheikhz@rbunagpur.in

Abstract

This project presents an IoT-based Smart Water/Fuel Monitoring System using ESP32 and the Blynk platform. The system is designed to measure fuel or water levels using an ultrasonic sensor (HC-SR04), track usage percentage, calculate volume, and trigger alerts when fluid levels drop below a critical threshold. It also estimates remaining capacity and shows historical data trends on the mobile app. The TP4056 module is used for battery management, enabling wireless monitoring. The system aims to provide real-time tracking for water tanks or fuel storage, enhancing efficiency and avoiding sudden outages.

Keywords—ESP32, IoT, Ultrasonic Sensor, Fuel Monitoring, Blynk, TP4056

I. INTRODUCTION

Recent developments in IoT-based monitoring have led to smart and remote fuel/water level detection solutions. Various systems have been proposed using Arduino or ESP32 microcontrollers combined with ultrasonic sensors. Traditional analog systems lack the ability to notify users remotely or show usage trends. This system uses the Blynk platform to visualize real-time sensor data, send push notifications, and remotely monitor system health. The inclusion of a TP4056 charging module ensures mobile and autonomous operation.

Proposed Solution: Our system combines ESP32, HC-SR04, and Blynk with TP4056 for power, creating a compact and cost-effective smart tank monitor.

II. SYSTEM ARCHITECTURE

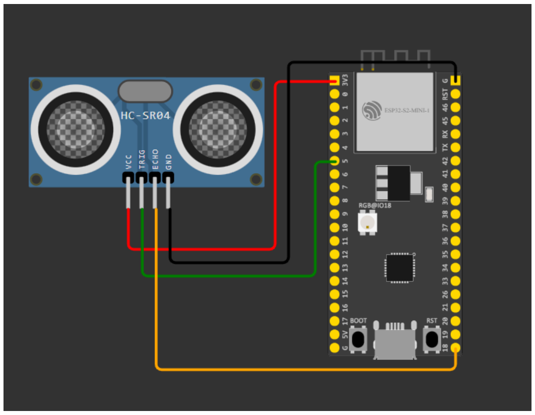
The architecture of this system is based on a **microcontroller-based IoT model** using the **ESP32**, which communicates with an ultrasonic sensor and a power supply module (TP4056), and interfaces with the **Blynk IoT platform** for remote monitoring.

Main Components:

- **ESP32**: The core controller responsible for data processing, sensor reading, and wireless communication.
- **HC-SR04 Ultrasonic Sensor**: Measures the distance between the sensor and the surface of the liquid (fuel/water).
- **TP4056 Module**: Manages battery charging and power delivery to the ESP32.
- **3.7V Li-ion Battery**: Acts as the power source for the system.
- **Blynk IoT**: Used to visualize real-time sensor data on mobile through widgets like gauges, labels, and event-based alerts.

Data Flow:

1. **Ultrasonic Sensor** collects distance data.
2. **ESP32** calculates water/fuel level percentage and usage.
3. **Data is sent over Wi-Fi** using Blynk to a mobile dashboard.
4. Users receive **real-time updates** and **alerts** (e.g., low fuel).
5. All powered by a battery system managed by the **TP4056 module**.


HC-SR04
VCC
TRIG
ECHO
GND
ESP32-S2-MINI-1
3V3 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 5V G
G RST 46 45 RX TX 42 41 40 39 38 37 36 35 34 33 26 21 20 19 18
RGB@IO18
BOOT
RST

III. METHODOLOGY

Hardware Assembly:

- The **HC-SR04** sensor is connected to **GPIO5 (TRIG)** and **GPIO18 (ECHO)** of the ESP32.
- The **TP4056 module** connects to the battery and powers the **VIN and GND** of the ESP32.
- No resistive voltage divider was used in this case, but the battery voltage remained within safe limits.

Distance Measurement Logic:

- The HC-SR04 sends out an ultrasonic pulse and calculates the time it takes to bounce back.

This time is converted into distance using the formula:

nginx
CopyEdit

```
distance (cm) = (duration / 2) / 29.1
```

-

Data Processing:

- Fuel/Water level = Max Level – Distance
- Water/Fuel Used % = ((Start – Current Level) / Max) × 100
- Remaining Capacity = (Current Level / Max) × Tank Volume

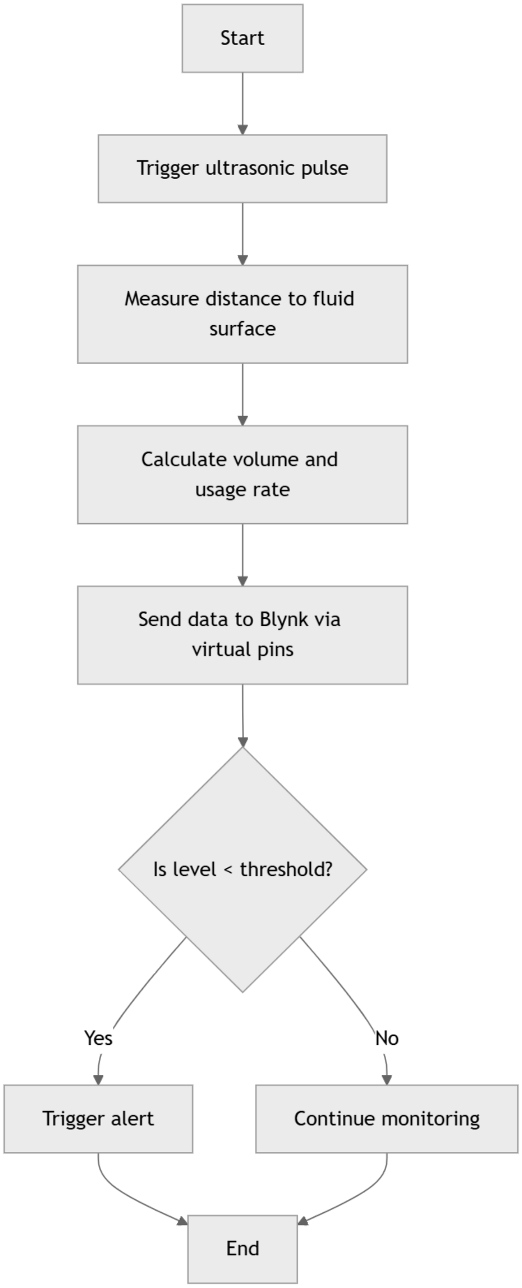
Blynk Dashboard Setup:

- Virtual Pins are configured for each data parameter.
- Widgets like Gauge, Value Display, and Events are used.
- An alert event is triggered if level goes below a predefined threshold.

Simulation & Testing:

- Initial simulation done on **Wokwi**.
- Final implementation done using real hardware components.

The algorithm follows these steps:


Start
Trigger ultrasonic pulse
Measure distance to fluid surface
Calculate volume and usage rate
Send data to Blynk via virtual pins
Is level < threshold?
Yes
No
Trigger alert
Continue monitoring
End

IV. RESULT & DISCUSSIONS

The system successfully monitored the water level using the **ultrasonic sensor** and displayed it on the **Blynk IoT app**.

Readings were stable and updated every 5 seconds, except for fuel usage percentage, which updated every 5 minutes for better accuracy.

Low fuel alert was implemented using the Blynk **event feature**.

The **mobile UI** clearly showed:

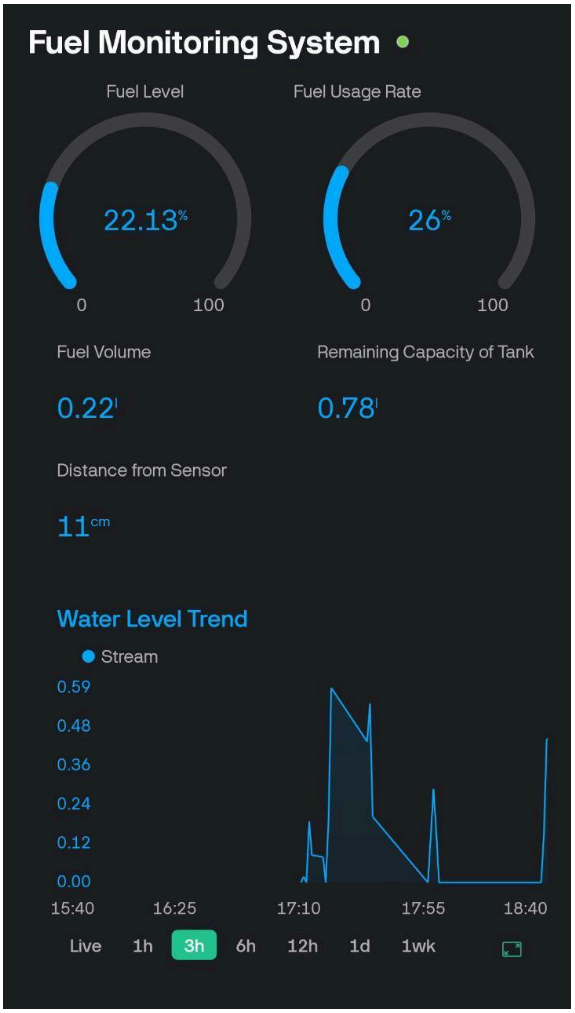
- Real-time fuel level percentage
- Fuel usage rate
- Volume of fuel used
- Remaining tank capacity
- Water/fuel trend (rise or fall)

Battery-powered design proved efficient and flexible.

The system can be scaled for **fuel tank monitoring** in bikes/cars with minimal changes.

V. CONCLUSIONS

This project demonstrates a **low-cost, IoT-based fuel/water monitoring system** using the ESP32 and ultrasonic sensor. By integrating it with **Blynk**, real-time monitoring becomes user-friendly and accessible via smartphones. The system is accurate for applications like **fuel tanks in vehicles** or **overhead water tanks**, and the modular structure allows future additions like temperature sensing or leak detection. With improvements, it could be deployed in real-world smart vehicle or home automation systems.



Fuel Monitoring System

Fuel Level: 22.13%

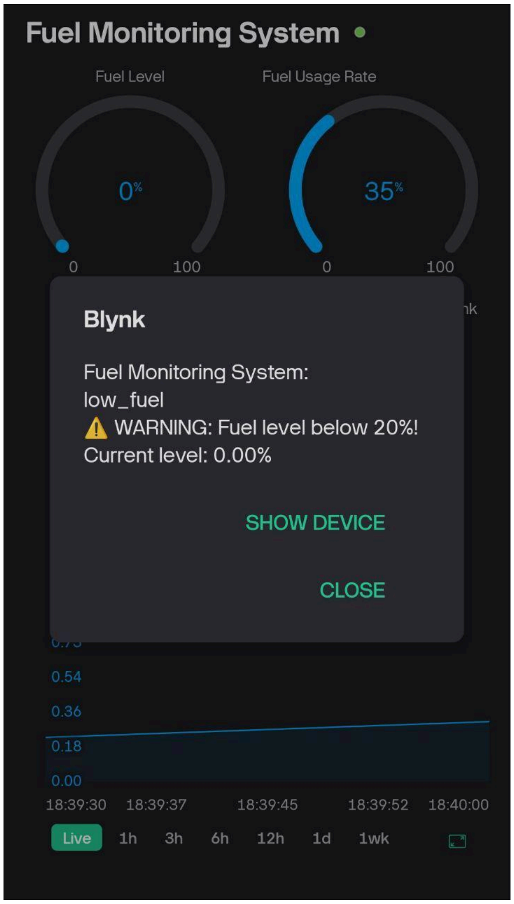
Fuel Usage Rate: 26%

Fuel Volume: 0.22l

Remaining Capacity of Tank: 0.78l

Distance from Sensor: 11cm

Water Level Trend



Blynk

Fuel Monitoring System: low_fuel
⚠️ WARNING: Fuel level below 20%!
Current level: 0.00%

REFERENCES

1. Wokwi Simulator – Simulate ESP32 and sensor circuits online 👉 https://wokwi.com

2. Blynk IoT Platform – Build mobile apps and dashboards for IoT control 👉 https://blynk.io

3. TP4056 Charging Module Datasheet (PDF)👉 https://www.electrokit.com/uploads/productfile/41091/TP4056.pdf

4. HC-SR04 Ultrasonic Sensor Datasheet (PDF)👉 https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf

5. ESP32 Microcontroller Datasheet by Espressif (PDF)👉 https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

6. Battery Voltage Monitoring using Voltage Divider (Tutorial)👉 https://circuitdigest.com/microcontroller-projects/battery-voltage-monitor-using-arduino

7. Blynk Events and Automations – Configure alerts & triggers👉 https://docs.blynk.io/en/getting-started/using-events

8. Blynk Virtual Pins Guide – Sending and receiving data👉 https://docs.blynk.io/en/getting-started/virtual-pins

9. How Ultrasonic Sensors Work (HC-SR04) 👉 https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/

10. ESP32 ADC Input Reading (Analog Pins)👉 https://randomnerdtutorials.com/esp32-adc-analog-read-arduino-ide/

11. Lithium-Ion Battery Charging Basic👉 https://www.adafruit.com/product/1904 (Adafruit Guide)

Github Repository: Link