Optimizing Industrial Gas Supply Logistics with a Zero-Intervention Monitoring System

Shridhar Deshpande[1], Sagar Wankhede[1,*]

[1] School of Mechatronics Engineering, Symbiosis Skills & Professional University Pune 412101

*Correspondence: svw8890@gmail.com

ABSTRACT

Industrial settings often suffer from operational lags owing to the unforeseen depletion of cylinders and manual tracking of their inventory. In this paper, an IoT-based zero-intervention monitoring technique that uses a strain-gauge load cell, HX711 amplifier, and ESP32 microcontroller to measure the cylinder weight is proposed. The measured weight values are then displayed on a P10 LED (Light Emitting Diode) display and sent via a TCP (Transmission Control Protocol) socket to a Python Flask server. The rate of gas consumption and remaining backup time are calculated by using a linear regression algorithm, and users are notified via SMTP (Simple Mail Transfer Protocol) alerts before depletion of cylinders. The effectiveness of the proposed scheme was verified through experimentations done with calibrated weights.

KEYWORDS: IoT, Industrial Gas Monitoring, ESP32, Load Cell, Flask Framework, TCP Socket, Predictive Analytics.

1. INTRODUCTION

The gas cylinders used in industries are usually checked manually or through pressure gauges, which cause delayed refill decisions and unanticipated stockouts. The latest IoT solutions have mainly concentrated on gas leakages and safety purposes, while the other aspect of continuous consumption monitoring and inventory prediction is less explored. Current GSM-based systems notify only the basic aspects and lack centralized monitoring and predictive capabilities. In addition, cloud-based approaches create delay and reliability issues for the industry. The new system uses ESP32 based edge nodes and Python Flask dashboard that use TCP sockets to monitor the weights continuously and estimate the backup and refill needs without using cloud services [1-3].

2. MATERIALS AND METHODS

The designed architecture comprises a sensing layer and analytics layer as shown in Figure 1. For cylinder weigh measurement, the strain-gauge load cell, along with the 24-bit ADC HX711 module, provides accurate measurements. The ESP32 microcontroller collects the sensor data and shows the measured weigh on a P10 LED matrix; at the same time, it sends telemetry to a TCP server once every five seconds. Python Flask framework serves as a backend that collects telemetry, saves historical data in the CSV file format, and shows the performance of the system in an interactive dashboard. For predicting gas usage speed and estimated remaining backup time, a linear regression model is used; meanwhile, SMTP-based notifications are sent if the predicted backup time becomes lower than the given threshold [4-8]

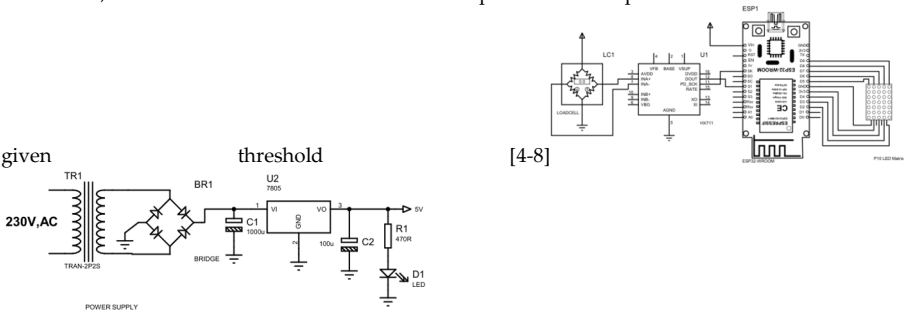


Figure 1. Circuit schematic detailing the connections between the ESP32 microcontroller, the HX711 amplifier, the load cell, the P10 display & the power supply circuit

© The Author(s) 2026
D. Nikam et al. (eds.), *Proceedings of the Global Symposium on Adaptive Manufacturing (GSAM 2026)*, Advances in Engineering Research 308,
https://doi.org/10.2991/978-94-6239-813-9_19

3. RESULTS & DISCUSSION

The developed system prototype (Figure 2) was calibrated using standard masses to create a linear correlation between ADC readings and actual mass while minimizing sensor drift. Simulations of depletion were done by progressively taking out calibrated masses to test the sensing, communication, and predictions pipeline. The ESP32 chip succeeded in sending weight information every five seconds, while the Flask server managed to display live telemetry and calculated the remaining backup hours. The interface allows viewing current level of gas, weight of cylinder, predictions about backup hours, depletion rate, and communication status. Data history (Figure 2) is being saved automatically for usage analysis purposes, while dynamic graphs allow visualizing the depletion trend. Successful testing was done for both manual refill requests and SMTP notifications, proving its reliability when predicted backup time reached the critical value. Thus, the architecture enables predictive inventory management without continuous manual control [9-10].



Figure 2. The experimental hardware prototype

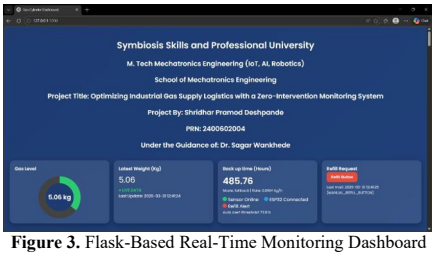


Figure 3. Flask-Based Real-Time Monitoring Dashboard

4. CONCLUSION

A zero-intervention IoT-based monitoring system for industrial gas cylinders has been designed and experimentally verified. The combination of a load cell, HX711 amplification circuit, ESP32 microcontroller, TCP sockets, and Python Flask-based dashboard allows for precise real-time monitoring and analytics. Experiments showed precise weight measurement, uninterrupted communication, proper estimation of backup time, and sending automatic emails. The proposed system minimizes manual involvement, facilitates inventory planning, and can be scaled to manage an industrial gas supply chain.

REFERENCES

1. A. O. Adedokun. Smart IoT-Based Monitoring System for CNG Cylinder Safety. 2025.
2. S. Kumar et al. IoT-Based Smart Weighing Device for LPG Monitoring. 2020.
3. K. B. Udaya Sree; E. Purushotham. Monitoring LPG Cylinder Weight and Detecting Leakages System. 2023.
4. Avia Semiconductor. HX711 Load Cell Amplifier Datasheet. 2019.
5. Espressif Systems. ESP32 Technical Reference Manual. 2023.
6. Flask Project Documentation. Pallets Foundation. 2023.
7. NIST. IoT Cybersecurity Framework. 2021.
8. M. Abishek et al. Estimation and Prediction of LPG Gas Using Data Analytics. 2022.
9. D. Balogun et al. LPG Smart Guard: An IoT-Based Solution for Real-Time Gas Cylinder Monitoring. 2024.
10. J. Klensin. Simple Mail Transfer Protocol (RFC 5321) IETF. 2008.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

