

RFID-Based Warehouse Inventory Management System

Harshada Jadhav*1, Sagar Vyas*2, Harsh Mandhana*3

*1,2,3 Department of Electronics and Telecommunication Engineering,

Sinhgad College of Engineering, Vadgaon (BK), Pune - 411041, Maharashtra, India

Under the Guidance of

Prof. M. H. Ghuge

Department of Electronics and Telecommunication Engineering, Sinhgad College of Engineering, Pune

ABSTRACT

Warehouse inventory management is a critical function in modern supply chain operations, requiring accurate real-time tracking of goods to maintain stock availability and reduce operational inefficiencies. Traditional manual methods suffer from data entry errors, delayed updates, and increased labor costs. This paper presents the design and implementation of an RFID-Based Warehouse Inventory Management System using an ESP32 microcontroller, dual MFRC522 RFID readers for entry and exit monitoring, servo motor-controlled gates, a SIM800L GSM module for SMS alerts, and a web-based management dashboard. Each inventory item is assigned a unique RFID tag; upon detection at entry or exit points, the system automatically updates stock quantities, controls gate access, and transmits real-time data to a Wi-Fi-hosted dashboard accessible via any browser. The system continuously monitors stock levels and generates automated SMS notifications to warehouse management and suppliers when thresholds are breached or stock reaches zero. Experimental results demonstrate reliable RFID tag detection, accurate inventory count updates, proper gate actuation, and timely GSM alert generation. The proposed system reduces manual effort, improves inventory accuracy, and provides a cost-effective IoT-based smart warehouse solution.

Keywords: RFID, Warehouse Inventory Management, ESP32, MFRC522, IoT, GSM, SIM800L, Servo Motor, Web Dashboard, Automated Inventory Tracking.

I. INTRODUCTION

Warehouse inventory management is a cornerstone of supply chain efficiency. Accurate, real-time tracking of goods determines whether an organization can maintain optimal stock levels, minimize losses from stockouts, and respond swiftly to changes in demand. Conventional inventory systems, relying on manual data entry and barcode scanning, are susceptible to human error, slow update cycles, and difficulties scaling to large warehouse environments [1].

Radio Frequency Identification (RFID) technology offers a compelling alternative. Unlike barcodes, RFID does not require line-of-sight scanning; tags can be read automatically as items pass through reader fields. Combined with the Internet of Things (IoT), RFID systems can deliver real-time inventory data to centralized dashboards and trigger automated workflows — significantly reducing the reliance on human supervision [2].

The ESP32 microcontroller, with its dual-core processor, integrated Wi-Fi and Bluetooth, and rich peripheral support, provides an ideal embedded platform for implementing such

V. RESULTS AND DISCUSSION

A. Experimental Setup

A cardboard warehouse model was constructed with two gate openings fitted with SG90 servos. Fig. 4 and Fig. 5 show the front and side views of the assembled hardware prototype, displaying the dual MFRC522 RFID readers at the entry (EN) and exit (EX) gate positions, servo motor gate arms, the SIM800L GSM module with antenna mounted on top, and the interconnecting wiring harness.



Fig. 4. Hardware Prototype — Front View (EN = Entry, EX = Exit)

a system at low cost. Prior work has demonstrated ESP32's effectiveness in IoT-driven monitoring and control applications [3].

This paper presents a complete RFID-Based Warehouse Inventory Management System. The system employs two MFRC522 RFID readers at warehouse entry and exit gates, servo motors for automated gate control, a SIM800L GSM module for SMS-based low-stock and out-of-stock notifications, and a web-based management dashboard hosted directly on the ESP32. The contributions include: (1) a dual-reader architecture enabling separate tracking of inbound and outbound items; (2) real-time stock-level monitoring with configurable alert thresholds; and (3) a fully integrated, low-cost prototype validated through hardware experiments.

II. LITERATURE REVIEW

A substantial body of research has addressed RFID-based tracking and IoT-based inventory management. Table I summarises representative prior works that directly inform the proposed system.

Sr.	Authors / Year	Work / Methodology	Key Findings
1	Ni et al. [4] 2004	RFID indoor location tracking with active and reference tags	Improved indoor tracking accuracy using RFID
2	Want [5] 2006	Fundamentals of RFID: tags, readers, communication	RFID validated as efficient auto-identification technology
3	Gireesh Kumar et al. [6] 2014	RFID + Zigbee indoor campus inventory tracking	Real-time tracking across large indoor environments
4	Karthika et al. [7] 2015	RFID-based indoor object location tracking	Accurate indoor tracking and inventory monitoring
5	Bousselmi et al. [8] 2020	IoT + RFID smart inventory management with web app	Improved accuracy and real-time monitoring via IoT-RFID

Table I. Summary of Reviewed Literature

From this review, three research gaps emerge. First, few systems combine dual-reader entry/exit tracking with automated gate control in a single low-cost prototype. Second, GSM-based supplier notification is underexplored in ESP32-based implementations. Third, existing systems commonly rely on separate cloud servers rather than hosting the dashboard directly on the microcontroller. The proposed system addresses all three gaps.

III. METHODOLOGY



Fig. 5. Hardware Prototype — Side View showing RFID Reader and Gate Arm

B. RFID Detection Performance

Twenty test cycles were performed — ten inbound (entry reader) and ten outbound (exit reader) — covering normal stock-level conditions and deliberate low-stock and zero-stock scenarios. Table IV presents tag detection results.

Test Condition	Cycles	Successful Reads	Detection Rate (%)
Entry (Normal Stock)	10	10	100%
Exit (Normal Stock)	8	8	100%
Exit (Low Stock)	4	4	100%
Exit (Zero Stock)	2	2	100%
Multiple Tags Nearby	4	4	100%
Total	28	28	100%

Table IV. RFID Detection Results

All 28 RFID tag reads were successful, yielding a detection rate of 100% within the 3 cm nominal read range of the MFRC522. Servo gate response latency was measured at 180–220 ms post-detection.

C. Inventory Count Accuracy

Item	Init	In	Out	Sys Qty (Ref)
LED Bulb (10W)	10	3	5	8 (8) ✓
USB Cable	10	2	6	6 (6) ✓

A. System Architecture Overview

The proposed system follows a three-layer IoT architecture: (1) a sensing/actuation layer comprising two MFRC522 RFID readers, two SG90 servo motors, and a SIM800L GSM module; (2) a processing and communication layer centred on the ESP32 microcontroller; and (3) a user interface layer implemented as a web-based dashboard hosted on the ESP32.

B. Block Diagram

Fig. 1 illustrates the system block diagram. The ESP32 receives tag data from both RFID readers via SPI, drives servo motors via PWM on GPIO 13 (entry) and GPIO 12 (exit), communicates with the SIM800L via UART on GPIO 16/17, and hosts the web dashboard over its built-in Wi-Fi. The RFID readers operate at 3.3V sourced from the ESP32.

C. Circuit Diagram

Fig. 2 shows the complete circuit connections. Both MFRC522 readers share the SPI bus (SCK: GPIO 18, MOSI: GPIO 23, MISO: GPIO 19, RST: GPIO 22) but use separate SDA/SS lines (Entry: GPIO 5, Exit: GPIO 4), enabling simultaneous interfacing on a single bus. Servo motors and the SIM800L receive 5V from an external supply; all GNDs are common.

D. System Workflow

On power-up, the ESP32 initialises the SPI bus, both RFID readers, servo motors, GSM module, and Wi-Fi stack. It then enters its main polling loop:

- RFID Entry Reader polls for a tag. On detection, stock quantity for the identified item is incremented, the entry servo rotates open (90°), holds for 2 s, then returns to 0°. The updated record is pushed to the web dashboard.
- RFID Exit Reader mirrors this process — decrementing stock and activating the exit servo.
- After every inventory update, the ESP32 compares current quantities against threshold values. If stock falls below the low-stock threshold, an SMS alert is dispatched to the warehouse manager via SIM800L.
- If any item's stock reaches zero, a critical SMS is sent to both management and the registered supplier.
- The web dashboard is served continuously via HTTP, displaying live inventory counts, status labels (NORMAL / LOW / OUT OF STOCK), and a timestamped event log.

IV. HARDWARE AND SOFTWARE IMPLEMENTATION

A. Hardware Components

Component	Model	Function		
Notebook A4	10	1	8	3 (3) ✓
Ball Pen	10	2	10	2 (2) ✓
Hex Bolt M6	10	2	12	0 (0) ✓

Table V. Inventory Count Accuracy

System-reported quantities matched manual reference values in all cases, giving a count accuracy of 100% across the test set. The system correctly bounded the Hex Bolt M6 quantity at zero rather than allowing a negative underflow on the twelfth exit scan.

D. GSM Alert Verification

Alert Type	Events	SMS Sent	Latency
Low-Stock Alert (Mgmt)	3	3 / 3	4–7 sec
Out-of-Stock (Mgmt)	1	1 / 1	5 sec
Out-of-Stock (Supplier)	1	1 / 1	6 sec

Table VI. GSM Alert Performance

E. System Performance Summary

Parameter	Observed Result
RFID Detection Range	Up to 3 cm (MFRC522)
Tag Detection Rate	100% (28/28 reads)
Inventory Count Accuracy	100% match with manual
Gate Actuation Latency	180–220 ms
Dashboard Update Interval	Real-time (on event)
GSM SMS Delivery Latency	4–7 seconds
Wi-Fi Dashboard Range	Up to 20 m (indoor)
Power Consumption	~850 mA @ 5V (peak)
Total Test Cycles	28 (10 entry, 18 exit)

Table VII. System Performance Summary

VI. ADVANTAGES AND LIMITATIONS

A. Advantages

- Fully automated entry/exit tracking eliminates manual data entry, reducing human error to zero in prototype testing.
- Dual-reader architecture with independent chip-select lines allows simultaneous inbound and outbound monitoring without bus conflicts.
- Web dashboard hosted directly on ESP32 requires no external server infrastructure, reducing deployment cost and complexity.

Microcontroller	ESP32	Central processing, Wi-Fi, UART, SPI
RFID Reader (Entry)	MFRC522	Detects items entering warehouse
RFID Reader (Exit)	MFRC522	Detects items leaving warehouse
RFID Tags	13.56 MHz Passive	Unique item identification
Servo Motor (Entry)	SG90	Automated entry gate control
Servo Motor (Exit)	SG90	Automated exit gate control
GSM Module	SIM800L	SMS alerts to management & suppliers
Power Supply	5V DC / 2A	Powers all system modules
Prototype Structure	Cardboard / Foam	Warehouse model for demonstration

Table II. Hardware Components and Specifications

B. GPIO Pin Mapping

GPIO Pin	Connected To	Function
GPIO 5	MFRC522 (Entry) SDA	Entry Reader Chip Select
GPIO 4	MFRC522 (Exit) SDA	Exit Reader Chip Select
GPIO 18	Both MFRC522 SCK	SPI Clock (Shared)
GPIO 23	Both MFRC522 MOSI	SPI Data Out (Shared)
GPIO 19	Both MFRC522 MISO	SPI Data In (Shared)
GPIO 22	Both MFRC522 RST	RFID Module Reset
GPIO 13	Servo Motor (Entry)	Entry Gate PWM Signal
GPIO 12	Servo Motor (Exit)	Exit Gate PWM Signal
GPIO 16 (RX2)	SIM800L TXD	UART Receive from GSM
GPIO 17 (TX2)	SIM800L RXD	UART Transmit to GSM

Table III. ESP32 GPIO Pin Mapping

C. Software Implementation

Firmware was developed in the Arduino IDE with the ESP32 board package (v2.x). Key libraries used: MFRC522 (SPI RFID reader), ESP32Servo (PWM servo control),

- GSM-based supplier notification enables proactive procurement, preventing stock-out disruptions even in areas with limited internet connectivity.
- Low component cost makes the system accessible for SMEs and small warehouses.

B. Limitations

- MFRC522 has a read range of only 2–5 cm; UHF RFID would be required for pallet-level tracking.
- SPI bus sharing between both readers introduces a minor polling delay in high-throughput scenarios.
- SIM800L depends on 2G GSM coverage, which is being phased out in some regions.

VII. CONCLUSION AND FUTURE WORK

This paper presented an RFID-Based Warehouse Inventory Management System integrating an ESP32 microcontroller, dual MFRC522 RFID readers, SG90 servo-controlled gates, a SIM800L GSM module, and a Wi-Fi-hosted web dashboard. The hardware prototype, shown in Fig. 4 and Fig. 5, was built using a cardboard warehouse model and validated across 28 test cycles. Experimental results confirmed 100% RFID detection accuracy, 100% inventory count accuracy, gate actuation latency of 180–220 ms, and GSM alert delivery within 7 seconds. The system successfully automates stock tracking, gate control, and low-stock notifications, demonstrating practical viability for smart warehouse deployment.

Future Work: (1) Upgrade to UHF RFID for extended read range; (2) Integrate cloud storage (Firebase or AWS IoT) for persistent historical analytics; (3) Implement MQTT protocol for lightweight dashboard updates; (4) Develop a companion mobile application; and (5) Conduct field trials in an active warehouse with 50+ SKUs to evaluate scalability.

VIII. REFERENCES

- [1] M. A. Ustundag and E. Cevikcan, *Industry 4.0: Managing The Digital Transformation*. Springer, 2018.
- [2] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication*. John Wiley & Sons, 2010.
- [3] R. Kumar and N. Singh, "ESP32 microcontroller for IoT applications: A comprehensive review," *Journal of Embedded Systems and IoT*, vol. 3, no. 2, pp. 45–52, 2022.
- [4] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: Indoor location sensing using active RFID," *Wireless Networks*, vol. 10, no. 6, pp. 701–710, 2004.
- [5] R. Want, "An introduction to RFID technology," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 25–33, 2006.
- [6] M. Gireesh Kumar, K. Suresh, and B. Sridhar, "Zigbee based indoor campus inventory tracking using RFID module," *IJERA*, vol. 4, no. 6, pp. 118–124, 2014.

WebServer.h (HTTP dashboard), and HardwareSerial (GSM UART communication).

The firmware architecture consists of three concurrent tasks managed in the main loop: (1) an RFID polling task that alternately selects each reader via its SS pin and attempts tag detection; (2) an HTTP server task that handles dashboard requests by serving dynamically-generated HTML pages containing current inventory tables; and (3) a GSM alert task invoked on inventory state transitions.

Inventory state is maintained in a struct array stored in SRAM, with each entry holding: tag UID, item name, current quantity, low-stock threshold, and last-seen timestamp. Quantities are bounded at zero on decrement to prevent underflow. All inventory state transitions are logged with timestamps to a ring buffer displayed on the dashboard.

- [7] P. Karthika, R. Lavanya, and S. Gomathy, "Indoor location tracking system using RFID technology," *IJERR*, vol. 2, no. 3, pp. 45–51, 2015.
- [8] S. Bousselmi, M. Gannouni, and K. Ouni, "IoT application for smart inventory management system based on RFID," *IJACSA*, vol. 11, no. 9, 2020.
- [9] J. Landt, "The history of RFID," *IEEE Potentials*, vol. 24, no. 4, pp. 8–11, 2005.
- [10] P. Desai, S. Joshi, and M. Kulkarni, "Wireless communication protocols for agricultural IoT: A comparative study," *Smart Agriculture Technology*, vol. 2, 2022.