

IoT Based Smart Trolley At Airport

Shwetha Madiwala¹, Niharika Honnalli², Priyanka Biradar³, Kavya Shetgar⁴,
Bhagyashree Shinde⁵

¹Professor, Department of Computer Science and Engineering, Faculty of Engineering and Technology
(exclusively for women), Sharnbasva University, Kalaburagi, India.

^{2, 3, 4, 5}Student, Department of Computer Science and Engineering, Faculty of Engineering and Technology
(exclusively for women), Sharnbasva University, Kalaburagi, India.

ABSTRACT

Airport baggage handling is a critical operational component of modern aviation infrastructure. Traditional baggage transport systems are susceptible to inefficiencies, human error, and high operational costs. This paper presents the design and implementation of an Arduino-based autonomous smart airport baggage carrier trolley that leverages Radio Frequency Identification (RFID), Internet of Things (IoT), and multi-sensor integration to automate baggage transportation within airport environments. The proposed system employs an Arduino UNO microcontroller as the central processing unit, interfaced with an RFID RC522 module for passenger identification and baggage authentication, an HC-SR04 ultrasonic sensor for real-time obstacle detection and avoidance, an L293D motor driver for precise DC motor control, an ESP32 CAM module for QR code scanning, and an ESP8266 (NodeMCU) Wi-Fi module for IoT-enabled remote monitoring via the Blynk platform. A scaled-down prototype was fabricated and tested, demonstrating autonomous navigation, RFID-based access control, obstacle avoidance, and wireless status reporting. Results confirm the system's capability to significantly reduce manual baggage handling effort, minimize mishandling incidents, improve airport ground logistics efficiency, and support sustainable electric-powered operations. The architecture is designed for scalability, making it suitable for deployment in large international airports. This work contributes to the advancement of smart airport automation and provides a cost-effective, open-source foundation for future IoT-integrated baggage management systems.

Keywords: Arduino UNO, RFID, IoT, Autonomous Navigation, Ultrasonic Sensor, Airport Baggage Handling, ESP8266, Smart Trolley

I. INTRODUCTION

The global aviation industry is experiencing unprecedented growth, with the International Civil Aviation Organization (ICAO) projecting annual passenger traffic to reach 8.2 billion by 2037. This surge places enormous strain on airport ground operations, particularly baggage handling — one of the most labor-intensive and error-prone aspects of airline logistics. According to SITA's Baggage IT Insights report (2022), the global rate of mishandled baggage stood at 4.35 bags per 1,000 passengers in 2021, resulting in billions of dollars in annual financial losses for airlines and airports through passenger compensation, bag recovery logistics, and reputational damage.

Traditional baggage handling systems rely heavily on manual labor, semi-automated conveyor belts, and mechanized tugs. While these solutions have improved over the decades, they remain vulnerable to human error, operational bottlenecks, and scalability limitations. The integration of microcontroller-based automation, wireless communication, and intelligent sensing offers a transformative pathway toward fully autonomous baggage transportation.

This paper presents a smart airport baggage carrier trolley built on the Arduino UNO platform, integrating RFID-based identification, ultrasonic obstacle detection, Wi-Fi-enabled IoT monitoring, ESP32 CAM-based QR scanning, and DC motor-driven autonomous mobility. The system transports passenger baggage from check-in counters to designated aircraft terminals autonomously, with real-time communication to a central monitoring dashboard via the Blynk IoT platform.

The proposed system addresses four critical pain points in current airport baggage operations: (1) reduction of manual labor and associated human error; (2) real-time baggage tracking and location transparency; (3) improved safety through autonomous obstacle avoidance; and (4) cost-efficiency through electrically powered operations. The prototype demonstrates the technical feasibility and provides a scalable architecture for real airport deployment.

II. BACKGROUND AND RELATED WORK

A substantial body of research has explored the evolution of airport baggage handling technologies. Smith et al. [1] laid foundational groundwork by examining early conveyor-based baggage systems, highlighting limitations of manual sorting and the need for automation. Johnson and Lee [2] demonstrated the feasibility of robotic navigation for airport applications, proposing systems with LiDAR and ultrasonic sensors for obstacle avoidance, confirming significant reduction in human error through autonomous robotics.

Brown [3] conducted an economic analysis of mishandled baggage using IATA data, quantifying multi-billion-dollar annual losses and emphasizing urgent need for automation. Taylor [4] examined wireless communication protocols for autonomous systems, evaluating Wi-Fi, Zigbee, and Bluetooth for large-scale deployment, recommending redundant communication architectures for reliability. Martinez and Kim [5] proposed energy-efficient designs for autonomous robots, advocating low-power microcontrollers and renewable energy integration.

Clark et al. [6] demonstrated IoT integration's impact in airport logistics, showing real-time tracking reduces misplacement rates. Patel and Green [7] reviewed Arduino-based robotics, establishing its suitability as a cost-effective open-source platform for scalable automation. Lee et al. [8] showed superiority of AI-driven navigation in dynamic environments. Zhang [9] emphasized scalability and interoperability as critical goals for airport automation, highlighting open-source platforms as key enablers.

Despite this rich literature, few studies have integrated RFID authentication, real-time IoT monitoring, QR scanning, and autonomous navigation into a single unified low-cost prototype. This paper bridges that gap by presenting a comprehensive hardware-software integrated system built on commodity components and open-source frameworks.

III. SYSTEM ARCHITECTURE AND METHODOLOGY

A. System Overview

The proposed system consists of a motorized trolley chassis controlled by an Arduino UNO R3 microcontroller. The trolley is equipped with: an RFID RC522 module for user authentication; an HC-SR04 ultrasonic sensor for obstacle detection; an L293D H-Bridge motor driver for DC motor control; an ESP32 CAM module for QR code baggage identification; an ESP8266 NodeMCU for Wi-Fi connectivity and IoT integration via Blynk; a 16x2 LCD for real-time status display; a buzzer for audible alerts; and a 12V 1.2 Ah sealed maintenance-free rechargeable battery for portable power.

The system operates in two primary modes: (1) Autonomous path-following mode, where the trolley navigates predefined routes using sensor feedback; and (2) RFID-triggered interactive mode, where the trolley awaits RFID card authentication before initiating movement. The ESP8266 module continuously transmits telemetry to the Blynk cloud platform for remote operator monitoring.

B. Hardware Components and Specifications

Arduino UNO R3 (ATmega328): Central processing unit with 14 digital I/O pins, 6 analog inputs, 16 MHz clock, 32 KB flash, 2 KB SRAM. Interfaces all peripheral modules via digital, SPI, and I2C protocols.

RFID RC522 Module: Operates at 13.56 MHz (HF). Reads passive RFID tags via SPI protocol. On valid tag detection, grants access and logs baggage data. Read range approximately 5-10 cm.

HC-SR04 Ultrasonic Sensor: Generates 8-cycle ultrasonic burst at 40 kHz. Detects obstacles within 2-400 cm range with ± 3 mm accuracy. Distance calculated using: $\text{Distance (cm)} = (\text{Echo_pulse_width_us} \times 0.034) / 2$. Triggers motor halt when obstacle is detected within 30 cm safety threshold.

L293D Motor Driver IC: Dual H-Bridge supporting two DC motors simultaneously with bidirectional PWM speed control. Accepts logic-level signals from Arduino and drives 12V wheel motors.

ESP32 CAM AI-Thinker: Dual-core 32-bit LX6 CPU at 160 MHz. Integrated Wi-Fi and Bluetooth. OV2640 camera for QR code scanning and baggage label recognition.

ESP8266 NodeMCU: Low-cost Wi-Fi SoC with full TCP/IP stack. Programmed via Arduino IDE. Transmits live trolley status, RFID events, and sensor data to Blynk cloud dashboard. Supports 10 GPIO pins, 3.3V operation.

16x2 LCD (HD44780): Provides real-time textual feedback on trolley status, RFID read results, and obstacle alerts. Interfaced in 4-bit mode.

12V 1.2 Ah SMF Battery: Sealed gel-electrolyte battery providing independent, cable-free power for the complete trolley system during operation.

C. System Architecture and Block Diagram

The system architecture follows a hub-and-spoke model. Input modules (RFID RC522, HC-SR04 ultrasonic sensor, ESP32 CAM, user input switch, Bluetooth module) feed data to the Arduino UNO. The Arduino processes inputs and drives output modules: L293D motor driver controlling the robot chassis, 16x2 LCD display, and buzzer. The ESP8266 bidirectionally communicates with the Blynk cloud server for IoT monitoring. The entire system is powered by the 12V SMF battery through appropriate voltage regulation circuits.

D. RFID Authentication Workflow

Each passenger's baggage is labeled with a passive RFID tag encoded with a unique UID. When the RC522 reader detects a card within its electromagnetic field, it reads the UID via SPI and compares against an authorized list in Arduino flash memory. On successful match, the trolley initiates the transport sequence and the event is logged and transmitted via Wi-Fi. Unauthorized access triggers the buzzer alarm and an LCD alert. This ensures only authenticated baggage accesses the trolley compartment.

E. Obstacle Avoidance Algorithm

The ultrasonic sensor continuously measures the forward path distance. If measured distance falls below the 30 cm safety threshold, the Arduino immediately: (1) halts DC motors via L293D; (2) activates the buzzer for a 500 ms alert; and (3) displays an obstacle warning on the LCD. The trolley remains stationary until the obstacle is cleared, after which autonomous operation resumes automatically. This fail-safe prevents collisions with passengers, equipment, and other vehicles.

F. IoT Integration via Blynk

The ESP8266 NodeMCU connects to the Blynk cloud server via standard TCP/IP over Wi-Fi. The Blynk mobile application provides a widget-based dashboard displaying real-time trolley status, RFID authentication logs, ultrasonic distance readings, and motor state. Bidirectional communication enables operators to remotely halt the trolley via the app in emergency scenarios. Average data transmission latency measured at under 1.5 seconds during testing.

G. Development Process

The system was developed in eight systematic steps: (1) Component selection based on performance and cost; (2) Arduino IDE setup with required libraries (MFRC522.h, NewPing.h, LiquidCrystal.h, BlynkSimpleEsp8266.h); (3) Hardware assembly on a four-wheeled robotic chassis; (4) Connection testing via digital multimeter and serial monitor; (5) Library installation and dependency resolution; (6) Firmware development covering RFID handling, motor control logic, obstacle avoidance, and IoT transmission; (7) System integration testing under simulated airport conditions; and (8) Final assembly, calibration, and enclosure mounting.

IV. RESULTS AND DISCUSSION

A scaled-down prototype was fabricated and tested under laboratory conditions simulating airport terminal scenarios. The prototype successfully demonstrated all key functional requirements across structured test cases.

Autonomous Navigation: The trolley reliably followed predefined paths using motor control algorithms. Navigation achieved satisfactory path adherence over simulated short terminal corridors.

Obstacle Detection and Avoidance: The HC-SR04 detected static and dynamic obstacles with measured accuracy of ± 2 cm. The system consistently halted within the 30 cm safety boundary, eliminating all test-case collision incidents.

RFID Authentication: The RC522 module identified registered baggage tags within approximately 8 cm read range. Unauthorized tags were rejected with 100% accuracy. Authentication latency consistently below 200 milliseconds.

IoT Monitoring: The ESP8266 established stable Wi-Fi connectivity and transmitted real-time telemetry to the Blynk dashboard with average latency under 1.5 seconds. Remote monitoring of all system parameters was confirmed via mobile application.

Battery Performance: The prototype operated continuously from battery power for a sustained test period exceeding 45 minutes per charge cycle, sufficient for short-range terminal transport tasks.

Identified limitations include: sensor dependency (single ultrasonic sensor covers only forward direction); short RFID read range; battery runtime for extended operations; and Wi-Fi dependency for IoT functionality. These provide clear directions for future improvement through LiDAR integration, multi-sensor arrays, high-capacity battery packs, and 4G/5G cellular backup.

V. AIRPORT DEPLOYMENT CONSIDERATIONS

Translating the prototype into full airport deployment requires addressing additional engineering challenges. Large-terminal navigation demands GPS or IMU integration for precise positioning. Fleet coordination of multiple trolleys requires a centralized management system. Integration with existing Airport Operational Database (AODB) systems enables automated baggage assignment and departure gate linking.

Extended operations favor fast-charging lithium-ion packs or solar-hybrid charging. Structural reinforcement supports standard baggage weights (15-32 kg). Compliance with IATA baggage handling standards and electromagnetic compatibility (EMC) regulations for aircraft systems is mandatory. The electric-powered design aligns with the ACI initiative to reduce ground operation carbon footprints by up to 40%, contributing to ICAO's net-zero carbon 2050 commitment.

VI. CONCLUSION

This paper presented the design, implementation, and testing of an Arduino-based autonomous smart airport baggage carrier trolley integrating RFID authentication, IoT remote monitoring, ultrasonic obstacle avoidance, QR code scanning, and wireless communication. The prototype successfully validated all core functional requirements, demonstrating that a cost-effective open-source hardware platform can deliver reliable autonomous baggage transport.

The system offers significant advantages over traditional manual handling: reduced human error, continuous operation capability, real-time tracking, improved ground safety, and lower operational costs. The modular scalable architecture supports future enhancement with AI-based navigation, predictive maintenance, biometric authentication, and airport management database integration.

Future work will focus on GPS-based outdoor navigation, multi-trolley fleet coordination algorithms, extended battery life optimization, and formal compliance testing for real-world airport certification. This research provides a validated technical foundation for next-generation smart airport baggage management systems.

REFERENCES

- [1] Smith, J. (2005). Early automation in airport baggage handling: Conveyor systems and their impact. *Journal of Airport Operations*, 2(1), 15-29.
- [2] Johnson, R., & Lee, H. (2007). Robotic applications for airport navigation and obstacle avoidance. *IEEE Transactions on Robotics and Automation*, 23(4), 712-725.
- [3] Brown, A. (2009). Economic impact of mishandled baggage in aviation. *International Aviation Economics Review*, 5(2), 88-102.
- [4] Taylor, P. (2012). Wireless communication protocols for large-scale automated systems. *IEEE Communications Surveys & Tutorials*, 14(3), 1234-1251.
- [5] Martinez, C., & Kim, S. (2014). Energy-efficient designs for autonomous robots in industrial environments. *Robotics and Autonomous Systems*, 62(8), 1187-1195.
- [6] Clark, D., Williams, T., & Patel, R. (2016). IoT integration in airport logistics. *International Journal of Aviation Management*, 3(1), 44-58.
- [7] Patel, V., & Green, M. (2018). Arduino-based robotics: A review of advancements in cost-effective automation. *Journal of Open-Source Hardware*, 6(2), 78-91.
- [8] Lee, K., Park, J., & Nguyen, T. (2020). AI-driven navigation for autonomous systems. *Applied Intelligence*, 50(9), 2981-2997.
- [9] Zhang, W. (2022). Scalability and interoperability in airport automation. *Smart Airport Technology Journal*, 11(1), 22-37.
- [10] Kumar, S., Rajeev, K., & Singh, R. (2019). Automation of baggage handling system using IoT and RFID. *International Journal of Engineering Research & Technology*, 8(2), 125-131.

-
- [11] Patil, A., & Deshpande, N. (2021). Smart airport baggage transportation system using RFID and Arduino. *International Journal of Recent Trends in Engineering & Research*, 7(3), 98-105.
- [12] Rashid, M., & Khandaker, M. (2020). Autonomous baggage carrier using Arduino and ultrasonic sensors. *Proceedings of ICICCS 2020, IEEE*, 556-561.
- [13] SITA. (2022). Baggage IT Insights 2022. SITA Air Transport IT Insights Report.
- [14] IATA. (2021). World Air Transport Statistics. International Air Transport Association.
- [15] Airports Council International. (2021). Achieving net zero carbon emissions in airport ground operations. ACI World Policy Report.
- [16] Siegwart, R., Nourbakhsh, I. R., & Scaramuzza, D. (2011). *Introduction to Autonomous Mobile Robots* (2nd ed.). MIT Press.
- [17] Monk, S. (2017). *Programming Arduino: Getting Started with Sketches* (2nd ed.). McGraw-Hill Education.
- [18] Arduino.cc. (2023). Guide to motor control with Arduino for autonomous vehicles. Retrieved from <https://www.arduino.cc>