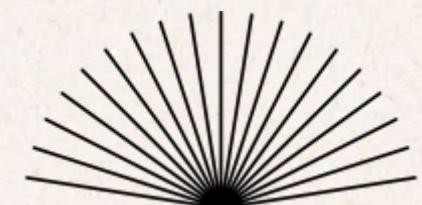


EMT PROJECT

ESP32-BASED RFID PARKING MANAGEMENT AND TRACKING SYSTEM

TEAM MEMBERS:

ARPIT RAJ	2301EC07
AYUSH KUMAR	2301EC08
ABHITESH SHUKLA	2301EE52
ANKIT KUMAR	2301EC05
DHIRENDRA PRATAP SINGH	2301EE49



CONTENT

03	Idea and Timeline
04	Principle
05	EMT Perspective
06	Assembly
07	Circuit Diagram
08	Prototype and Output
09	Observation
10	Future Aspects

Idea:

- 01 To design a smart RFID-based parking management system using an ESP32 and two RFID readers for real-time vehicle tracking at entry and exit points.**
- 02 To log every RFID scan event (Card ID, Date, Time) automatically to Google Sheets using the ESP32's built-in Wi-Fi, eliminating manual registers.**
- 03 To develop a cost-effective, scalable IoT solution that enhances security, efficiency, and automation in parking management systems.**



Timeline:

System Design and Planning:

- Define system architecture: ESP32 as the central controller, RFID readers for entry and exit, and Google Sheets as the cloud database.

Hardware Integration:

- Connect two RFID readers to the ESP32 through SPI.
- Configure GPIO pins to handle both readers simultaneously.
- Add LED indicators/buzzer for scan confirmation and gate control signals

Software Development:

- Develop ESP32 firmware using Arduino IDE.
- Implement Wi-Fi connectivity for the ESP32 to communicate with Google Sheets via Google Apps Script API.
- Code logic for reading card IDs, capturing timestamps, and sending structured data (Card ID, Date, Time).

Cloud Integration (Google Sheets):

- Create a Google Apps Script to receive HTTP POST requests from the ESP32.
- Format and store incoming data in a structured manner.
- Enable real-time monitoring and historical data analysis.

Principle



The project is based on the principle of RFID communication and IoT-based data logging. When an RFID tag is brought near the reader, the RFID module generates a time-varying magnetic field that induces a current in the tag's coil (as described by Faraday's Law of Induction). This allows wireless transmission of the tag's unique ID to the ESP32 microcontroller, which processes and sends the data over Wi-Fi to a cloud platform (Google Sheets).

The system combines electromagnetic coupling for short-range identification with wireless Internet communication for real-time data recording. This integration demonstrates practical application of Electromagnetic Theory and IoT, ensuring reliable, contactless, and efficient tracking of vehicles entering and exiting the parking premises.

Electromagnetics perspective:

05

1. RFID Working Principle:

- The RFID reader operates at 13.56 MHz in the High-Frequency (HF) band. It generates an alternating magnetic field that links with the RFID tag through inductive coupling. According to Faraday's Law of Electromagnetic Induction, this changing magnetic flux induces a voltage in the tag coil, powering it wirelessly and allowing data exchange.

2. Mutual Inductance and Coupling:

- The interaction between the reader coil and tag coil depends on the coupling coefficient (k) and alignment. Strong coupling ensures efficient energy transfer and stable tag detection – a direct application of magnetic field coupling studied in EMT.

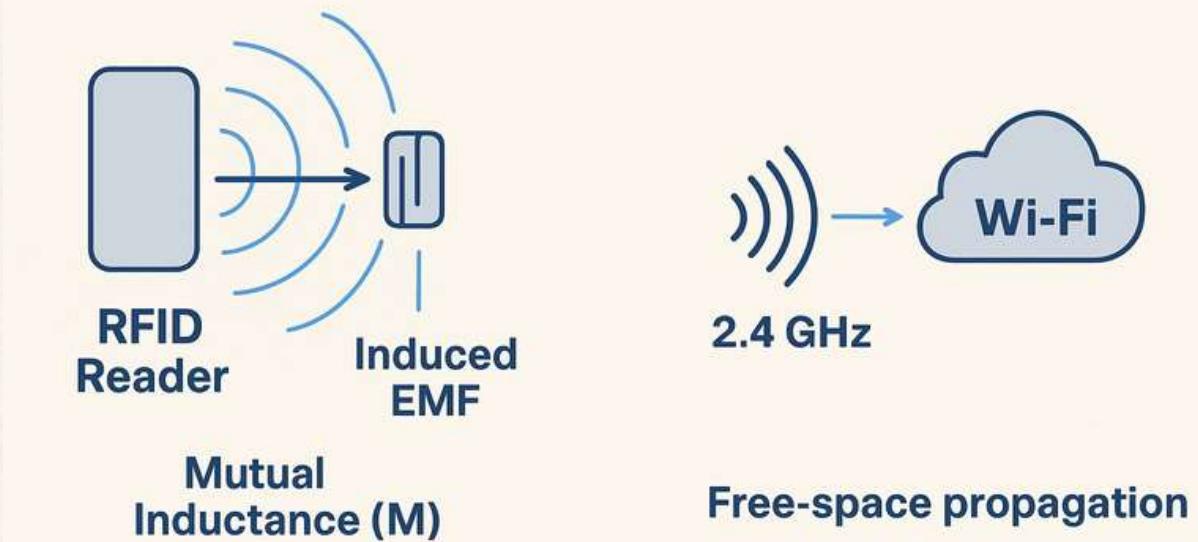
3. Wi-Fi Communication (2.4 GHz):

- The ESP32 module transmits data over 2.4 GHz electromagnetic waves in the ISM band. The transmitted power reduces with distance as described by the Friis transmission equation, which models free-space propagation loss – a key EMT concept in wireless systems.

4. Signal Interference and Field Behavior:

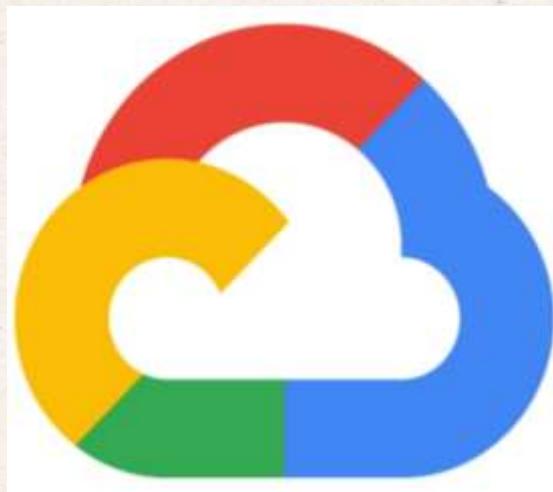
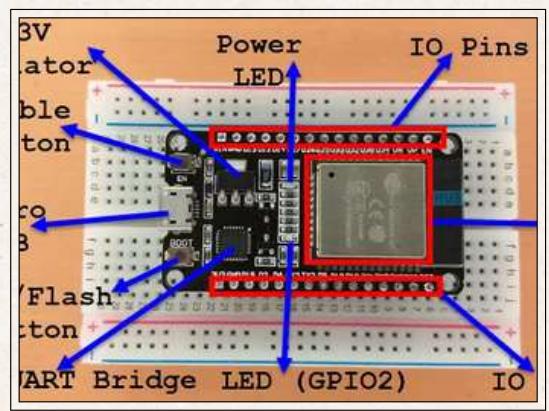
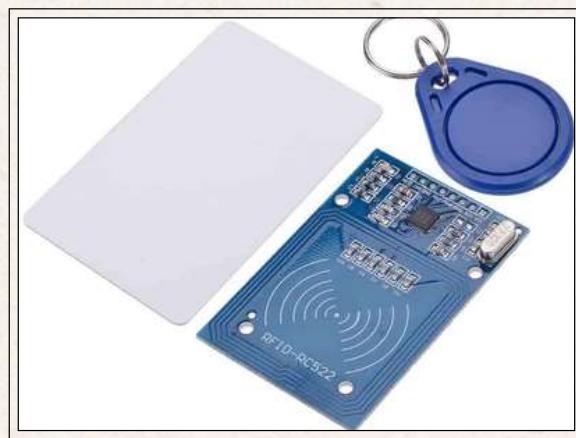
- Proper spacing between the two RFID readers prevents electromagnetic interference (EMI) and ensures clear signal reception. This demonstrates the importance of field interaction and attenuation in practical electronic systems.

EMT Perspective



COMPONENTS:

06



Hardware Components:

- ESP32 Development Board: Serves as the main controller and Wi-Fi interface for cloud communication.
- RFID Readers (2 × RC522 Modules): Used at entry and exit points to scan RFID tags.
- RFID Tags/Cards: Unique identifiers assigned to each vehicle.
- LED Indicators & Buzzer: Provide real-time feedback on successful scans.
- Breadboard, Jumper Wires & Power Supply Module: Used for circuit prototyping and power distribution.
- (Optional) Servo Motor/Relay Module: Can be used for automated gate control.

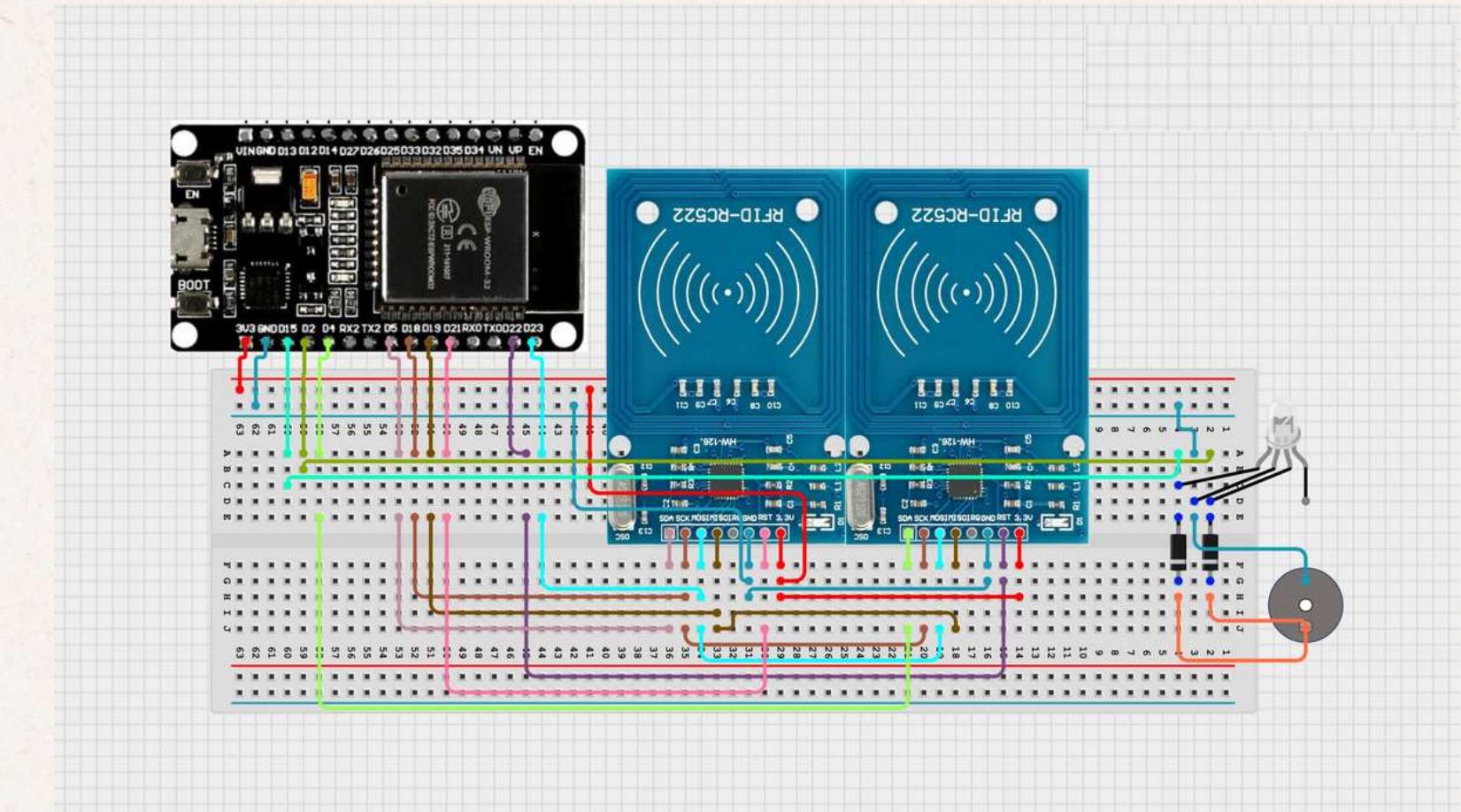
Software Components:

- Arduino IDE: For coding and uploading firmware to ESP32.
- Google Sheets with Google Apps Script: For real-time cloud data logging and visualization.
- Wi-Fi Network: Enables communication between ESP32 and the cloud server.

ASSEMBLY

07

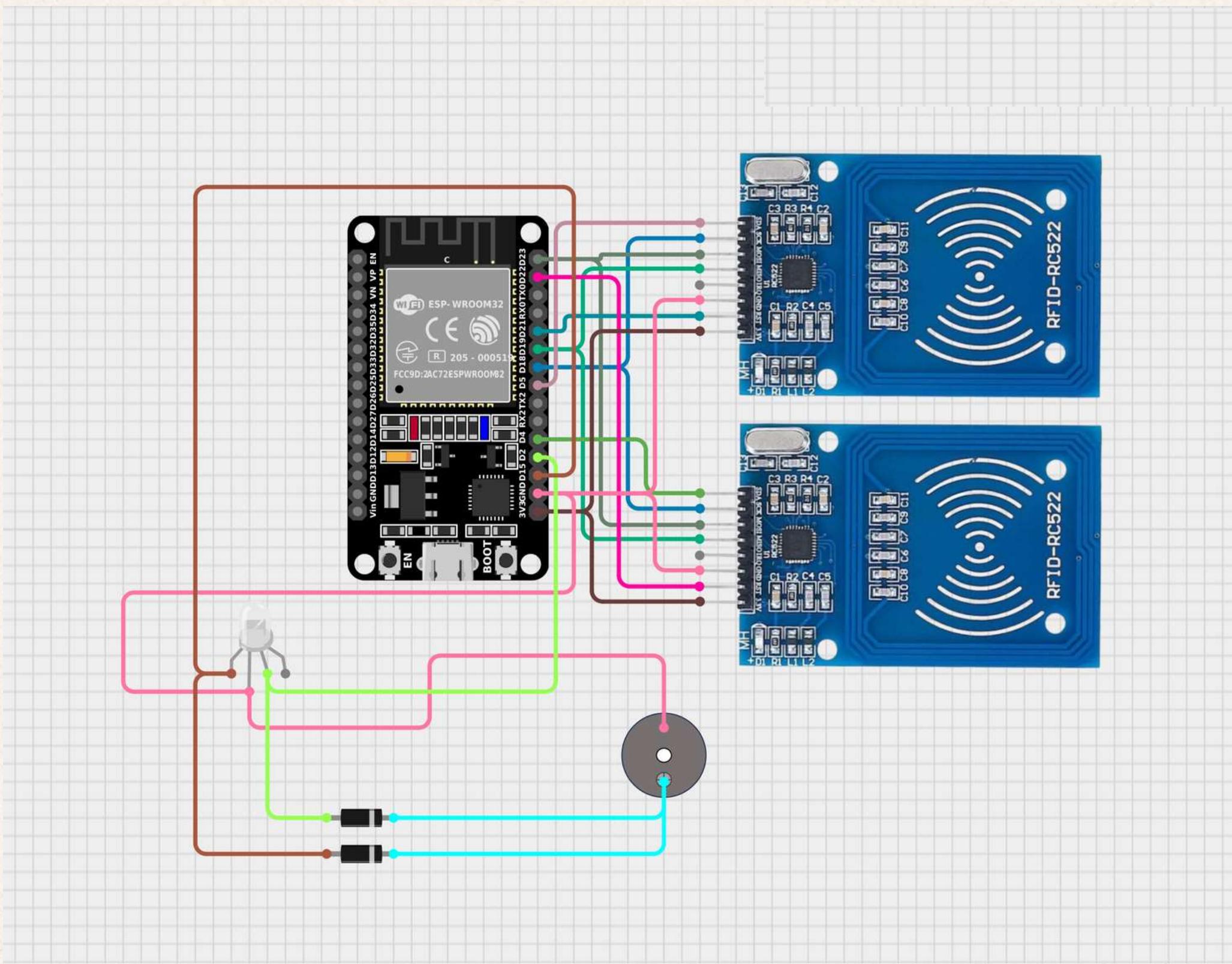
- The ESP32 microcontroller acts as the main controller and is programmed using the Arduino IDE.
 - Two RFID readers (RC522 modules) are connected to the ESP32 via the SPI interface – one at the entry gate and the other at the exit gate.
 - LED indicators and a buzzer are connected to GPIO pins to give feedback during card scanning.
 - A servo motor or relay module is optionally added to demonstrate automatic gate control.
 - All components are powered through a 5V regulated power supply and assembled on a breadboard using jumper wires.
 - The ESP32 reads the RFID tag ID, timestamps the event, and sends it to Google Sheets through Wi-Fi connectivity.



[Github Link](#)

Circuit Diagram:

08



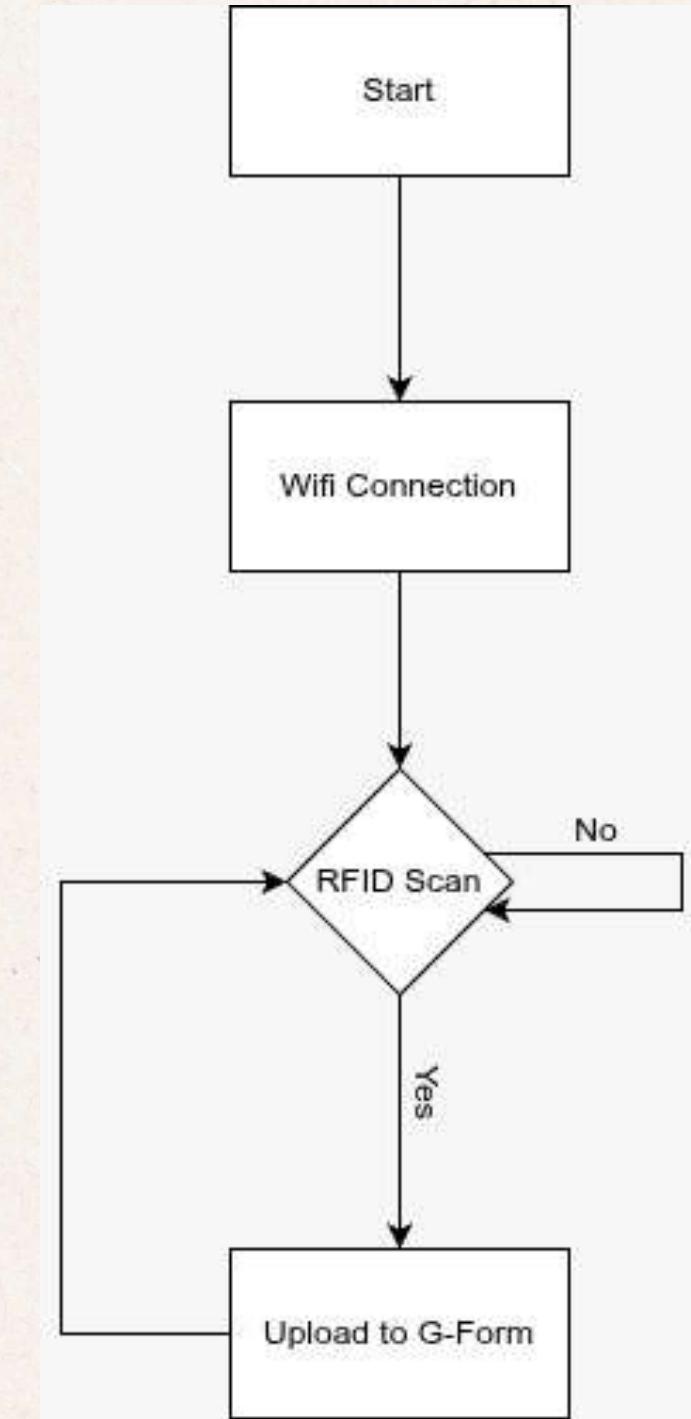
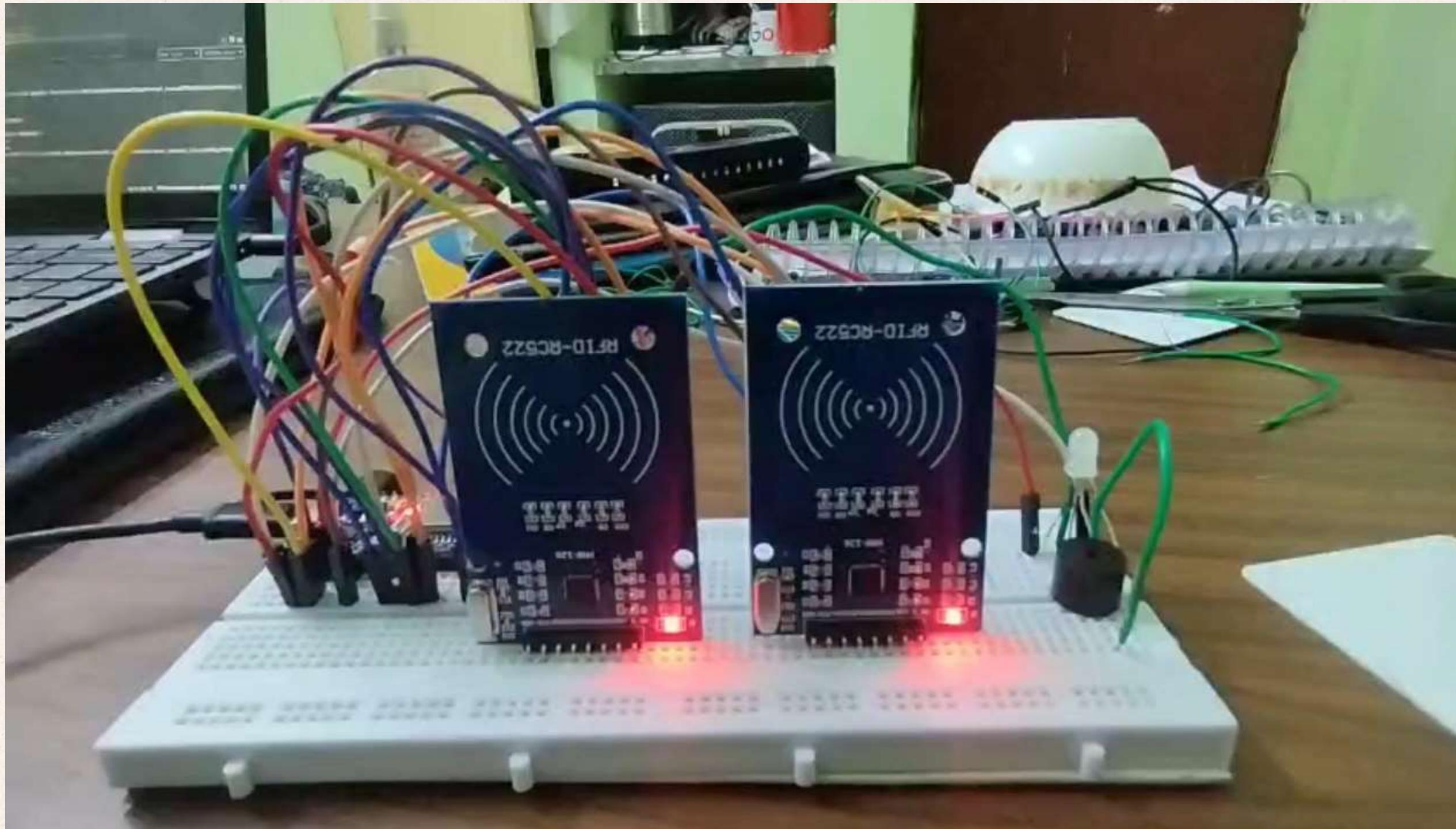
Connections Summary:

- Two RC522 RFID readers are connected to the ESP32 via the SPI interface (SCK, MISO, MOSI, SDA).
 - Both readers share 3.3V power and GND connections.
 - LED and buzzer are linked to GPIO pins for scan indication.

DEMO VIDEO:

[Video Link](#)

09



Observation:

10

- The project successfully tracks the ID using two RFID readers connected to a single ESP32 board.
- Each scan logs the card ID, time, and lane directly to Google Sheets, creating a real-time digital record.
- Communication between the ESP32 and RFID readers was handled through the SPI protocol, ensuring fast and reliable data transfer between devices.
- Each RFID tag's unique ID was read and converted into hexadecimal format, which simplified data handling and logging in Google Sheets.
- All scan details – including card ID (in hex), timestamp, and lane – were successfully recorded in real time on Google Sheets via Wi-Fi.
- The built-in Wi-Fi of ESP32 reduced hardware complexity and cost while maintaining smooth connectivity.
- LEDs and buzzers provided instant scan feedback, improving user experience and confirmation accuracy.
- Tests with multiple RFID tags showed consistent readings, accurate timestamps, and minimal lag.
- The system proved cost-effective (around ₹800 total) while remaining scalable for larger parking setups.
- Occasional delays during network drops highlighted a reliance on Wi-Fi stability for continuous data logging.

Future Aspect:

11

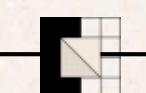
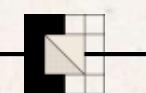
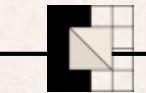
- The system design is highly scalable – additional RFID readers can be integrated with minimal hardware changes to manage multiple lanes or parking zones.
- Its low-cost and modular structure makes it practical for schools, offices, apartments, and public parking lots.
- By using the ESP32's Wi-Fi and cloud capabilities, administrators can monitor parking data from anywhere, making the solution practical for real-world deployment.
- The project can easily expand into a networked parking management system, connecting several ESP32 nodes to a common cloud dashboard.
- The use of open-source tools (Arduino IDE), Google Sheets, and Apps Script ensures affordability and ease of maintenance for future users.
- With minor software upgrades, the system can support real-time space availability tracking, enhancing convenience for drivers.
- The design's flexibility and cost-effectiveness make it suitable for both small private setups and larger smart city applications.
- Overall, the project demonstrates a practical, scalable, and future-ready IoT solution for efficient vehicle tracking and parking management.

CONTRIBUTIONS

12

Hardware Integration

Arpit Raj
Dhirendra Pratap Singh
Ankit Kumar



System Designing

Arpit Raj
Ayush Kumar
Abhitesh Shukla

Hardware Integration

Arpit Raj
Dhirendra Pratap Singh
Ankit Kumar

Cloud Integration

Abhitesh Shukla
Ankit Kumar

Software Development

Arpit Raj
Dhirendra Pratap Singh
Ayush Kumar

Documentation & Final Report

Ankit Kumar
Abhitesh Shukla
Arpit Raj
Dhirendra Pratap Singh
Ayush Kumar

Thank You