ESP32 Base Server for Drone Borne Application with LoRa

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

in

Electronics and Communication Engineering

by

Sudharson Aswin N K (21BEC0025)

Srivatsa S A (21BEC0312)

Tushar Sharma (21BEC0490)

Under the guidance of

Dr. ROHIT MATHUR

November, 2024

DECLARATION

I hereby declare that the thesis entitled "ESP32 base server for Drone Borne application with LoRa" submitted by me, for the completion of the course "BECE497J – Project 1" to the school of electronics engineering, vellore institute of technology, vellore is bonafide work carried out by me under the supervision of Dr.Rohit Mathur.

I further declare that the work reported in this thesis has not been submitted previously to this institute or anywhere.

Place: Vellore
Date: 12.11.2024

Signature of the Candidate

CERTIFICATE

This is to certify that the thesis entitled "ESP32 base server for Drone Borne application with LoRa" submitted by Tushar Sharma (21BEC0490), Srivatsa S A (21BEC0312) and Sudharson Aswin N K (21BEC0025) ,School of Electronics Engineering, VIT, for the completion of the course "BECE497J – Project 1", is a bonafide work carried out by him / her under my supervision during the period, 15. 07. 2024 to 12.11.2024, as perthe VIT code of academic and research ethics.

I further declare that the work reported in this thesis has not been submitted previously to this institute or anywhere.

Place : Vellore

Date : 12.11.2024

Signature of the Guide

Internal Examiner

ACKNOWLEDGEMENTS

Student Name

Executive Summary

This project presents a file server for drones based on ESP32 module to facilitate file transfer within remote areas with minimal network connectivity. The file management system allows uploads and downloads over WiFi (softAP) and LoRa modulation, even where there is no support of the internet. The 433 MHz LoRa offers a long range and low power advantage where there are otherwise hard to reach areas such as forests and disaster struck areas. A microSD card module is included to store more files because it is impossible to store large files due to the internal memory of the ESP32.

This is a secured file serving system and an interface to the users is presented in a simple HTML page. Such solution fits well with the principles of Sustainable Development Goals aiming on building resilience infrastructure, management of emergencies and environmental protection, attending to visual communication in areas with limited connectivity. The provided system is an efficient tool to cater for emergency response and environmental monitoring services as well as communication in remote places, providing solutions to a critical problem of sharing information without networks available

TABLE OF CONTENTS

List of Figures

(In the chapters, figure caption should come below the figure and table caption should come above the table. Figure and table captions should be of font size 10.)

List of Tables

List of Abbreviations

Table 1 Abbrevations

SPI	Serial Peripheral Interface
SPIFFS	Serial Peripheral Interface Flash File System
SoftAP	Software Access Point
LoRa	Long Range
SDG	Sustainable Development Goals
HTML	Hypertext Markup Language

1. INTRODUCTION

This project is focused on creating an ESP32-based file server for drones, facilitating file sharing in remote and network-limited areas using WiFi and LoRa. Users can upload, download, and manage files via a direct WiFi connection (softAP) without internet, while LoRa extends access to isolated locations with weak signals, such as forests or disaster zones. Ad-hoc networking ensures uninterrupted data exchange between drones, base stations, and users during network outages, while expanded storage with a microSD card allows for larger files. The system addresses gaps in existing solutions, offering resilient, infrastructure-free communication in challenging environments

1.1.LITERATURE REVIEW

LoRa technology, which has a long-range and low-power characteristic, has become the transformational solution in Internet of Things (IoT) applications, especially in remote areas where traditional communication methods are lacking. Its integration with Unmanned Aerial Vehicles (UAVs) has demonstrated substantial advantages in disaster management and environmental monitoring, with extended communication range and improved signal quality. Despite payload capacity and battery limitations, LoRa's potential in real-time data transmission remains promising.

One notable application of LoRa in agriculture is through systems that monitor soil moisture and air temperatures as well as other environmental parameters over distant areas. An example is the case where remote agricultural monitoring using LoRa is utilized alongside DHT-11 and FC-28 sensors in which data is transmitted to the microcontroller for visualization on IoT platforms, reducing the problem of distance in rural farming.

The technology has also been used in smart home and industrial applications, beating traditional methods such as Wi-Fi and Bluetooth with better range, lower power consumption, and even higher precision when controlling appliances. In addition, it has been applied to micro-grid systems where its combination with AES encryption allows for secure communication and overcomes cyber threats in the energy and remote applications.

Energy efficiency studies reveal that LoRa definitely performs better than Wi-Fi in a few contexts, such as agriculture, when the settings for transmitting data are optimized. In addition, smart irrigation systems that rely on LoRa are transforming the current methods of water management through automatic processes based on real-time soil data as against older traditional technologies.

In parallel, the ESP32 microcontroller is opted because of its low price, versatility, and power efficiency. This makes it widely used in numerous IoT applications, and finally, it improves UAV communication, environmental monitoring, solar energy systems, and disaster response solutions. In all, it is appropriate because it offers real-time data processing and long-range communication. The ESP32 now integrates well with LoRa capabilities, particularly with remote operations, to become a low-power and reliable solution that cuts across different fields.

1.2. RESEARCH GAP

The research gap addressed by this project lies in the lack of integrated solutions for file sharing and communication in remote, infrastructure-limited environments, particularly for drone-based applications. Existing systems rely heavily on stable network infrastructure, which is often unavailable in dense forests, disaster zones, or isolated areas. Additionally, there is limited research on combining LoRa long-range communication with drone-based file servers to facilitate real-time data exchange without relying on conventional internet, thus offering a solution for reliable communication in network-free regions.

1.3 PROBLEM STATEMENT

In remote and network-constrained environments, such as dense forests, disaster zones, or isolated regions, reliable communication and data exchange remain a significant challenge. Existing communication systems depend heavily on stable network infrastructure, which is often unavailable in these areas. This project aims to develop a robust ESP32-based file server integrated with WiFi and LoRa technologies to enable file storage, upload, download, and management in environments with limited or no network connectivity. The solution seeks to bridge the gap in remote communication by providing infrastructure-independent, long-range communication for drones and users in challenging environments.

1.3.1 RELEVANCE OF PROBLEM STATEMENT W.R.T SDG

This project aligns with several Sustainable Development Goals (SDGs) by providing an innovative, infrastructure-independent communication system for remote and network-limited regions. It supports SDG 9 (Industry, Innovation, and Infrastructure) by enabling reliable communication in challenging environments. The system enhances SDG 11 (Sustainable Cities and Communities) by improving disaster resilience, facilitating data exchange during emergencies to enhance community safety. It contributes to SDG 13 (Climate Action) through environmental monitoring and real-time data collection in isolated areas, while also supporting SDG 15 (Life on Land) by aiding conservation efforts and habitat protection. Additionally, the project supports SDG 3 (Good Health and Well-being) by providing reliable communication for medical and aid teams, improving coordination in disaster relief and emergency response.

2. PROJECT OBJECTIVE

The aim of this endeavor is to create and deploy an ESP32 hardware evidence of the base which utilizes not only WiFi but also the LoRa technologies for the purposes of efficient data communication and file sharing in remote, low network areas. There will be a system which will let users comfortably store, upload, download and delete files without having to use the traditional internet system; hence ensuring accessibility in file management in areas that have poor or no networks at all.

Users of the file server will be connected to the network through a Wi-Fi network implemented within the device by operating in softAP mode. The system is also designed to have a LoRa module working at 433 MHz that will facilitate low-power equipment communications at extended ranges for transmission of data from drones and base stations to remote users even when cellular networks are very poor or not available. This makes it feasible to communicate in real time with others in tough places such as forests, mountains, or even in situations of calamity.

As the storage limits of the ESP32 microcontroller pose a challenge to the reclamation of storage, another microSD card module would be employed to enable the use of storage expansion, thus the system could operate with more sizeable files. This additional storage will be in addition to the internal SPIFFS storage of the ESP32 and will assist in the offset of data management and larger transfers.

A simple HTML interface will be designed as part of the system enabling users to do file operations such as secure upload, download, manage, and erase files and keep track of the data in transit. This aspect of the design will help ensure that the system can be used with ease even by users in remote areas with low levels of technical sophistication.

Ultimately, the goal of this project is to provide a resilient, infrastructure-independent communication and file-sharing solution, ensuring reliable data exchange in areas where traditional communication systems are unavailable, making it a vital tool for various applications, including remote communication, disaster response, and environmental monitoring.

3. PROPOSED WORK

3.1 Design Approach/System Model/Algorithm

Design Approach:

- **System Overview**:
  - In this project, two ESP32 microcontrollers are utilized, designated as **ESP32 A** and **ESP32 B**. ESP32 A is configured to serve as both a file server and a LoRa communication module, while ESP32 B functions as a remote LoRa-based controller to issue commands and request files. This setup allows ESP32 B to communicate with ESP32 A to retrieve files stored on an SD card, and access them via a web server hosted on ESP32 A.
  - The main objective is to design a **reliable, long-range communication system** that supports file transfer and remote access in areas with limited connectivity.
- **Roles of Each Module**:
  - **ESP32 A**: Equipped with an SD card module and a LoRa transceiver, ESP32 A acts as a data storage server, holding files for download or deletion. It also hosts a web server, accessible via WiFi, for user interaction and file management.
  - **ESP32 B**: This module has a LoRa transceiver for issuing commands to ESP32 A. It requests specific files or file lists and controls file operations on ESP32 A without needing direct physical access.
- **Communication Flow**:
  - ESP32 B initiates communication by sending commands to ESP32 A via LoRa. Commands include “LIST” to get a directory of files and “FILE:<filename>” to

request a specific file.

- Upon receiving a command, ESP32 A retrieves the requested file from the SD card, divides it into smaller data packets, and transmits them back to ESP32 B over LoRa.
- The ESP32 A's web server updates accordingly, showing available files, enabling download links, and providing real-time file management for the user.

System Model:

Architecture Diagram:

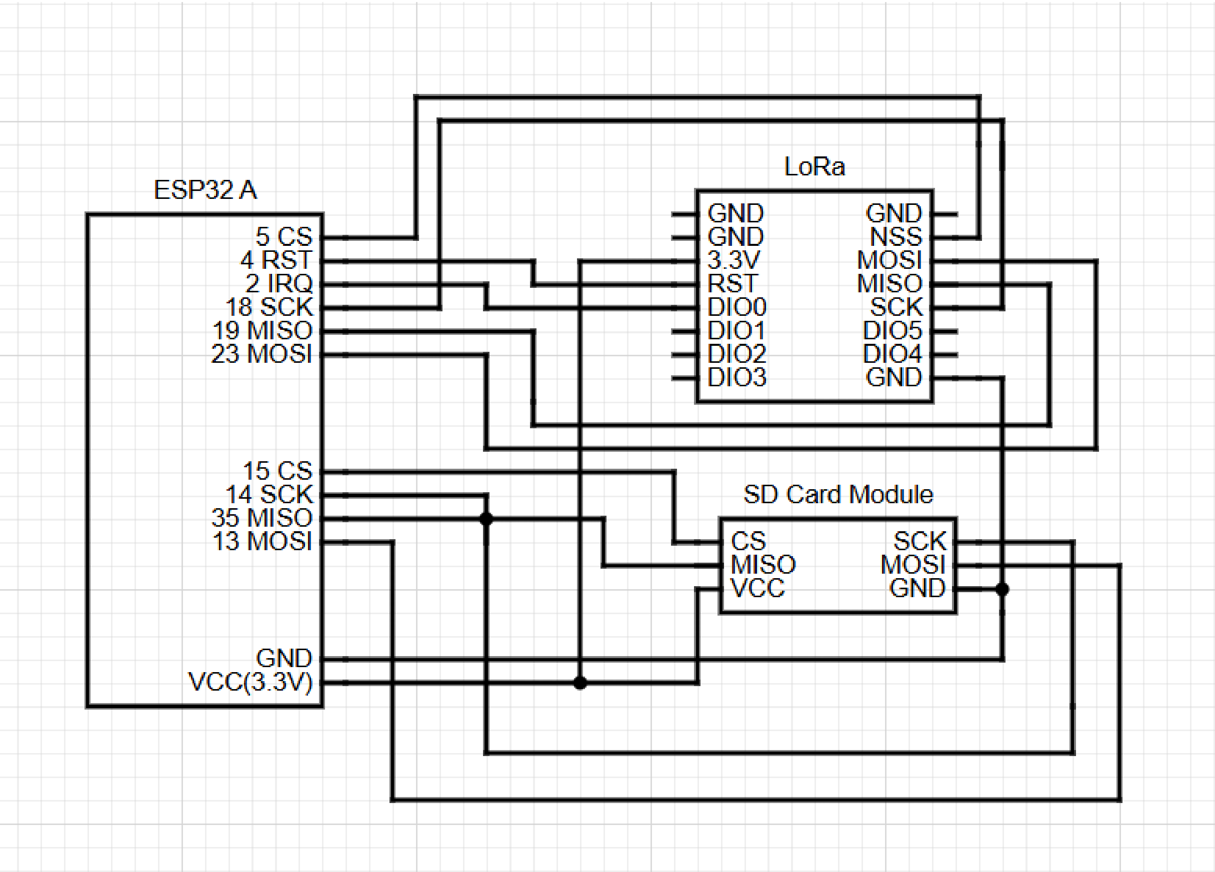

Figure 1ESP32 A CIRCUIT

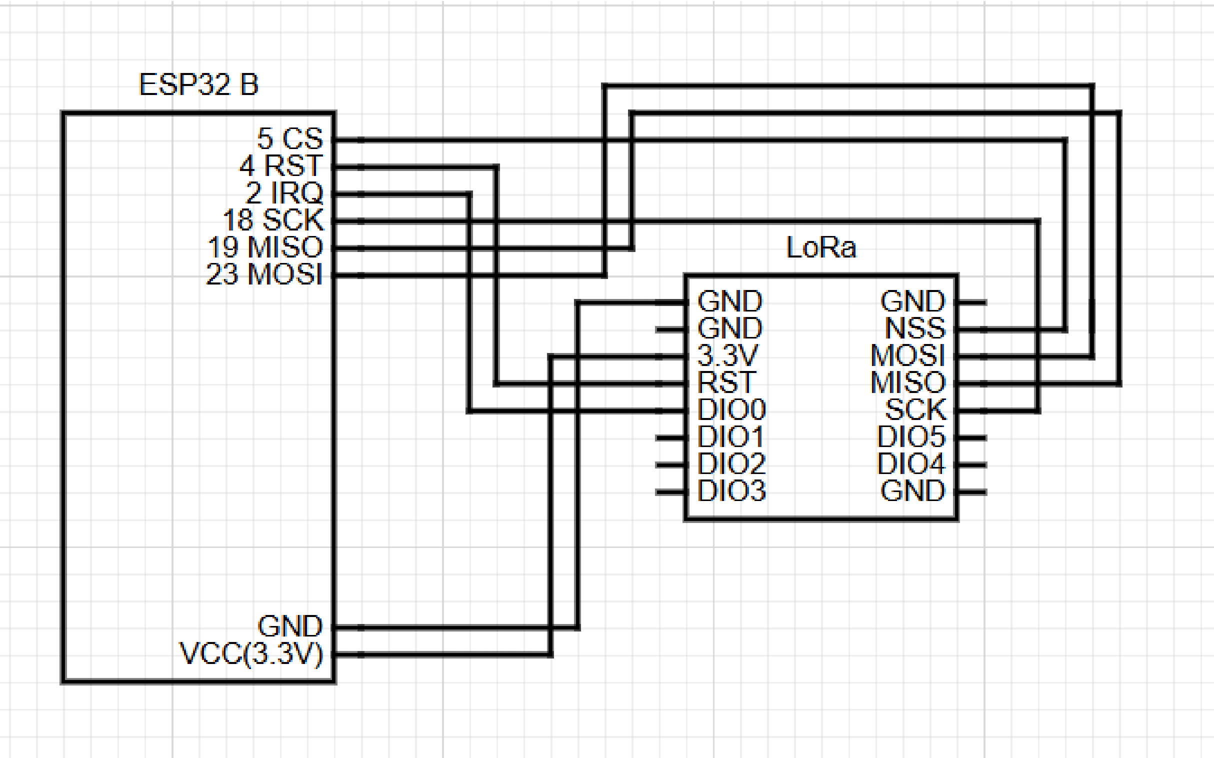

Figure 2ESP32 B CIRCUIT

Flowchart:

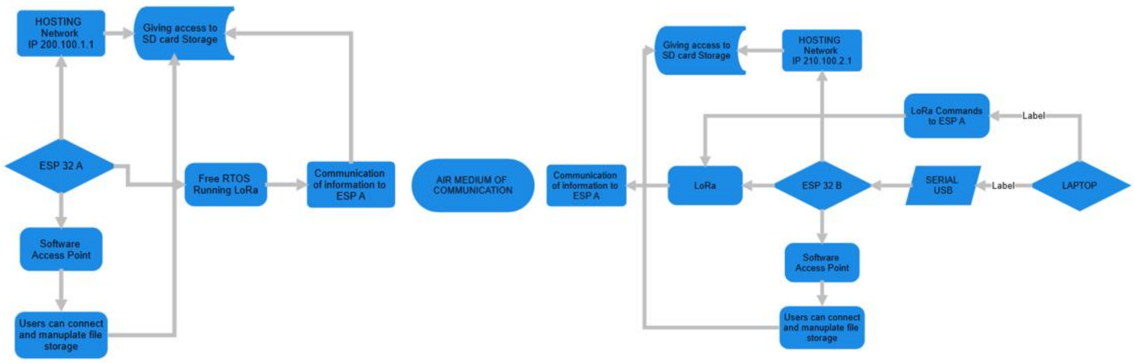

Figure 3system flow diagram

3.2 Technical Description

ESP A and ESP B Setup:

- **ESP32 A Configuration**:
  - **SD Card Module**: ESP32 A reads/writes data to an SD card module for persistent storage. Connections to the SD card are via SPI (Serial Peripheral Interface), enabling high-speed data transfer.
  - **Web Server Setup**: ESP32 A hosts a web server accessible over WiFi, allowing users to view, download, or delete files on the SD card. HTML and JavaScript

are embedded in the ESP32 A code to enable an interactive web interface.

    - **LoRa Configuration**: LoRa module on ESP32 A is set to operate on a pre-determined frequency, with specific parameters for power, bandwidth, and spread factor to optimize for long-range, low-power communication.

- **ESP32 B Configuration**:
    - **LoRa Module**: Configured similarly to ESP32 A for compatibility, ESP32 B's LoRa module only transmits and receives commands. It lacks an SD card or web server, relying on ESP32 A for file access.

LoRa Communication Protocol:

- **Frequency and Bandwidth**:
    - Frequency is set to 915 MHz (or region-specific frequency), optimized for low interference and longer range. Bandwidth and spreading factor are tuned to balance range and data rate.
- **Packet Structure**:
    - LoRa packets are structured with headers identifying the command type and payload size. The payload contains the command or data being sent.
    - Commands like "LIST" or "FILE:<filename>" are embedded in headers, while data packets have file contents split by CHUNK_SIZE.
- **Data Integrity**:
    - Each packet includes a **CRC** checksum for error detection. If the packet fails CRC verification, ESP32 B requests a resend.

Data Handling:

- **SD Card File Management**:
    - ESP32 A reads/writes to the SD card, creating, deleting, or retrieving files upon command.
    - File directory information is cached for faster access, especially when sending a file list to ESP32 B.
- **Web Server Integration**:
    - ESP32 A's web server has a simple HTML interface listing all files on the SD card. Users can click on a file name to download or delete it.
    - **JavaScript and AJAX**: Enables file management without reloading the page, showing the latest files in real time.
- **Data Chunking**:

- **File Transfer Optimization**: Large files are split into data chunks, typically of 512 bytes, due to LoRa's limited payload capacity.
- **Reassembly**: ESP32 B reassembles these chunks to form the complete file, verifying each chunk's integrity before assembling them.

4. HARDWARE/SOFTWARE TOOLS USED

4.1 Hardware Components

- **ESP32 Microcontroller**: The ESP32 is a powerful, dual-core microcontroller with built-in WiFi and Bluetooth, used for hosting the web server on ESP32 A and enabling remote data commands via LoRa on ESP32 B.
- **LoRa Module (SX1278)**: This module provides long-range, low-power communication, allowing ESP32 A and ESP32 B to send commands and transfer data over the ISM frequency band without WiFi.
- **SD Card Module**: Connected to ESP32 A, this module stores files accessible via the web server, allowing retrieval and transmission of requested files to ESP32 B over LoRa.
- **Power Supply**: A stable 5V USB or battery supply powers the ESP32 boards, ensuring reliable operation for both the web server on ESP32 A and LoRa communication.

4.2 Software Components

- **Arduino IDE**: The primary development platform for ESP32 programming, compiling, and debugging, allowing easy integration of web server, SD card, and LoRa functionalities.
- **ESPAsyncWebServer Library**: This library powers the web server on ESP32 A, supporting responsive, asynchronous HTTP requests for file viewing, uploading, and downloading.
- **SPIFFS (SPI Flash File System)**: Used on ESP32 A to store web server files (HTML/JavaScript), enabling client-side access to web server interfaces without extra storage.
- **LoRa Library (Arduino LoRa Library)**: Provides APIs for managing LoRa communications between ESP32 A and ESP32 B, enabling reliable long-distance data exchange.
- **SD.h Library**: Supports SD card operations on ESP32 A, enabling file storage and retrieval in response to LoRa-based requests from ESP32 B.

4.3 Development and Testing Tools

- **Serial Monitor and Debugging Tools**: Used to log and troubleshoot system behavior on both ESP32 units during development, particularly for LoRa communication and file handling.
- **Network Analyzer**: Monitors and evaluates WiFi and LoRa network performance, identifying issues like packet loss to ensure optimized communication.

5. RESULT ANALYSIS

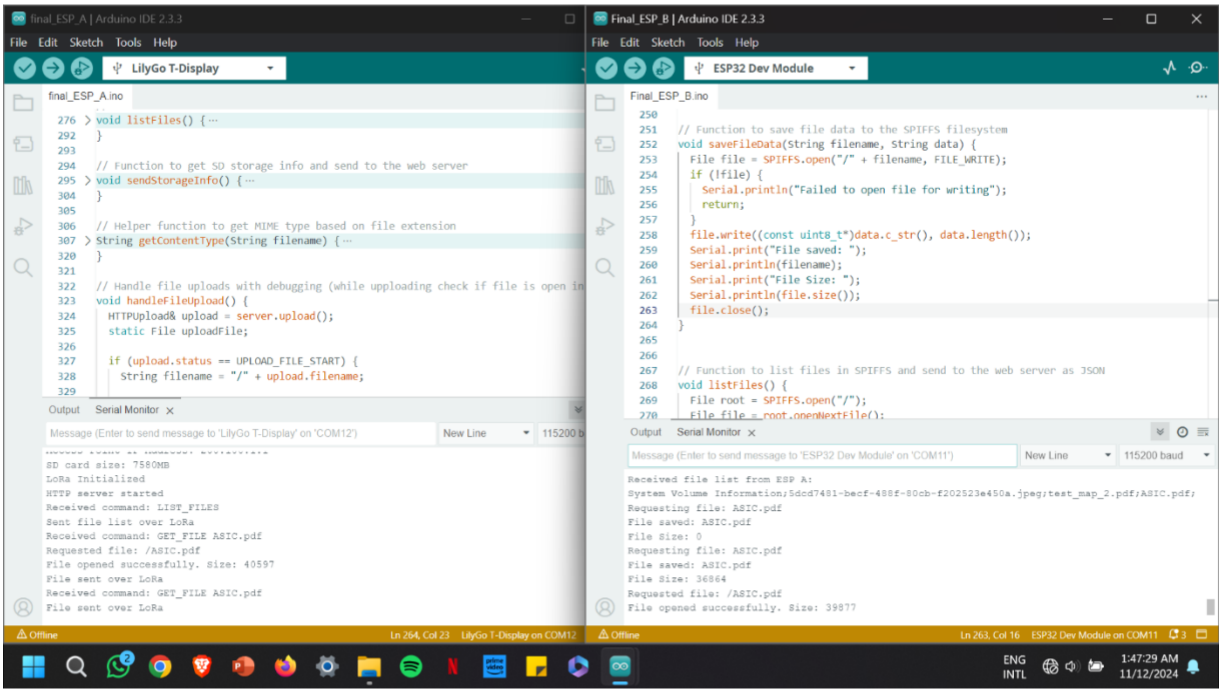

Figure 4serial moniter of each esp nodes

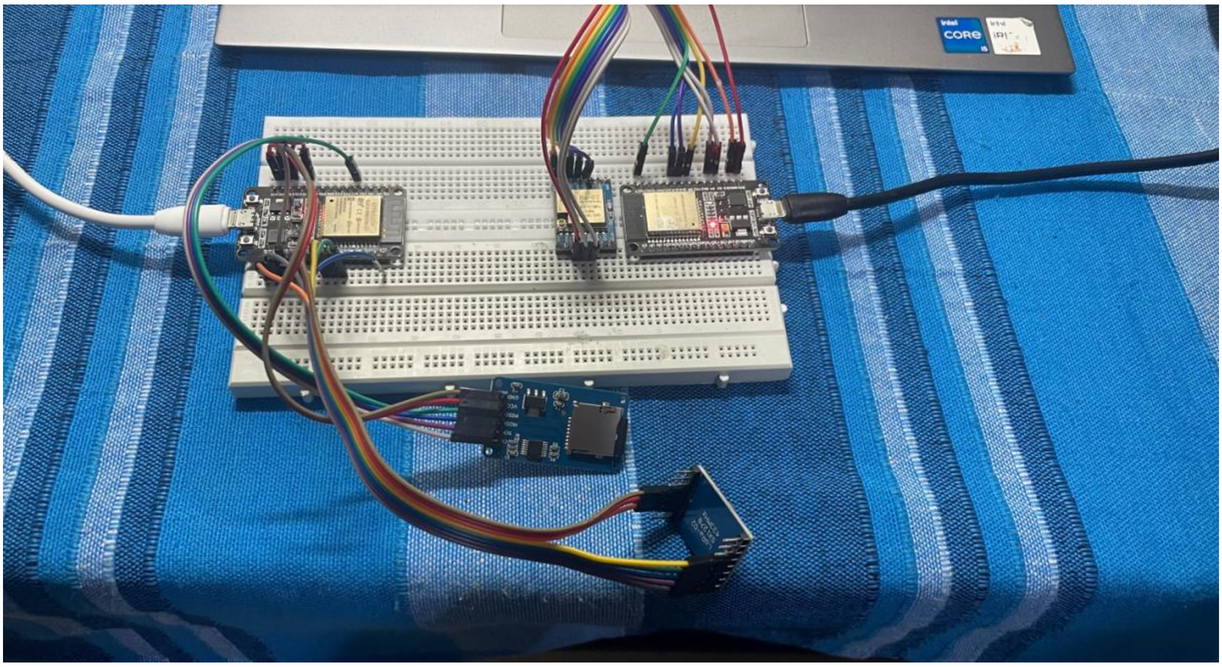

Figure 5Setup and connecton of the prototype

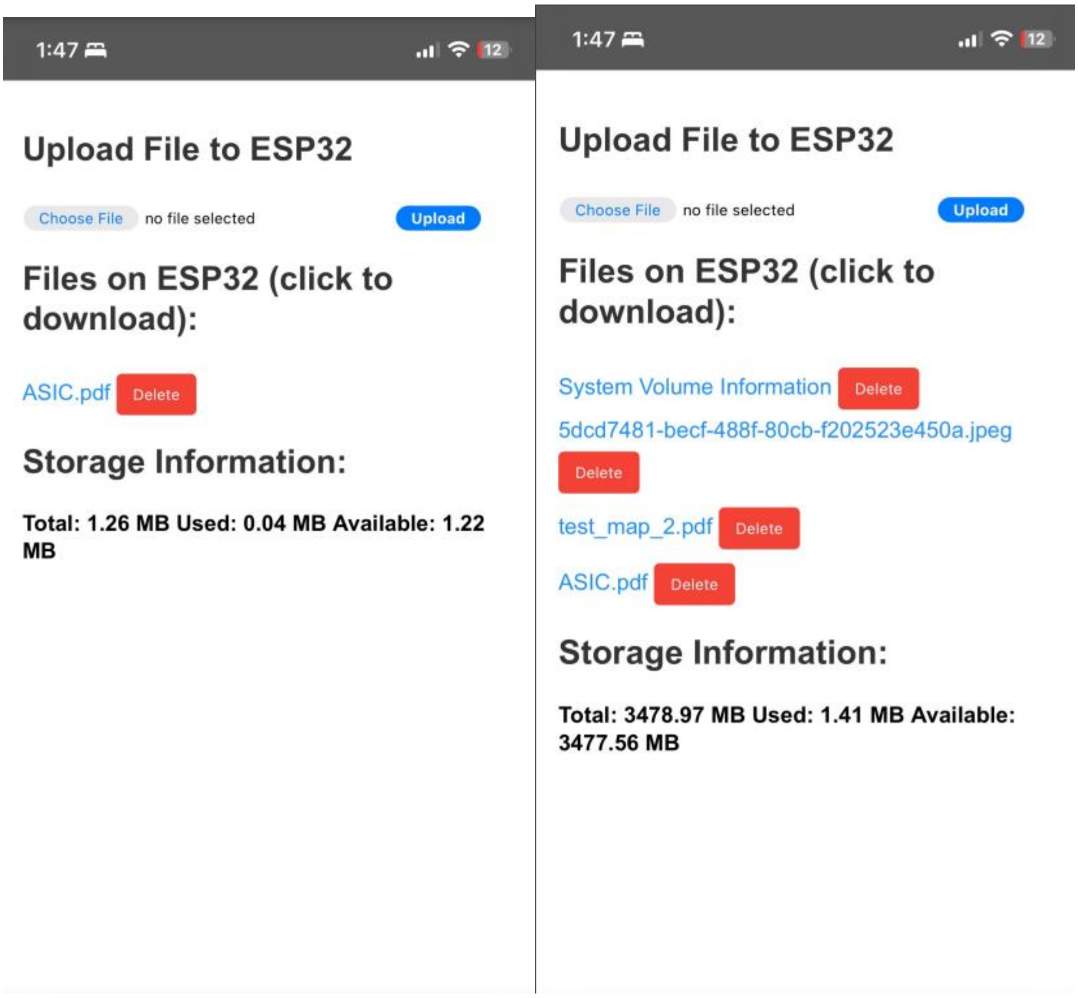

Figure 6webserver from espB

Figure 7webserv from esp A

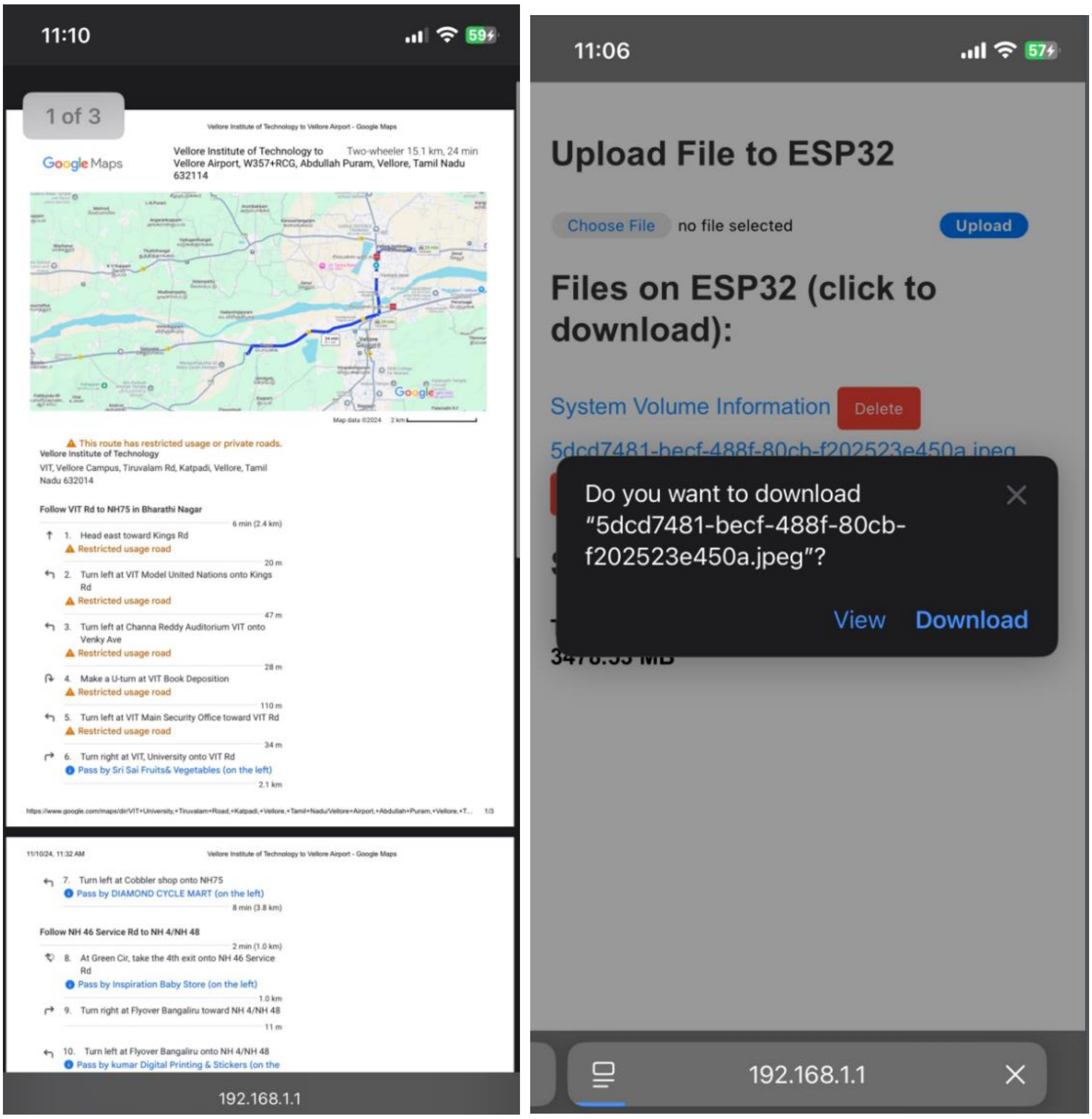

Figure 8Sample pictures of functon of webserver

6. CONCLUSION AND FUTURE WORK

6.1 Summary

To conclude, this project successfully created a working prototype of an ESP32-based file server, integrating both WiFi and LoRa communication technologies. This system allows for file storage, uploading, downloading, and deleting, all without relying on traditional internet infrastructure. This makes it ideal for use in remote areas, such as during disaster response or environmental monitoring, where network connectivity is limited or nonexistent.

A standout feature of the project is the use of LoRa technology, which provides long-range, low-power communication in areas where other communication methods fail. Whether it's dense forests or remote mountain regions, LoRa enables reliable data exchange between drones, base stations, and isolated users, even when network signals are weak or unavailable. This makes it a crucial tool for real-time communication in challenging environments.

The ESP32 microcontroller effectively manages both the file server functions and LoRa communication, ensuring smooth operation. The addition of a microSD card module also extends the system's storage capacity, allowing it to handle larger files and making it more practical for field use.

Overall, this project shows how combining ESP32, LoRa, and an intuitive HTML interface can create a practical, infrastructure-independent solution for remote data exchange. It successfully addresses the challenge of working in areas without reliable network access and offers a valuable communication tool for emergency situations, environmental monitoring, and remote operations. The project's successful implementation demonstrates the potential for this technology to fill a critical gap in communication capabilities in network-limited environments.

6.2 Limits and Constraints

The ESP32-based file server with WiFi and LoRa communication offers a reliable solution for remote data exchange but has several limitations. LoRa's range can be affected by terrain and interference, reducing effectiveness in some environments. Its low data transmission speed (0.3 to 27 kbps) makes it unsuitable for high-bandwidth tasks like video streaming or large file transfers.

While the system's storage is extended with a microSD card, it remains limited for larger files or long-term storage. Power consumption is another concern, as a stable power source is required, which may not always be available in remote areas. The reliance on WiFi in SoftAP mode and LoRa excludes advanced communication protocols like cellular or satellite, which may be more reliable in certain locations. LoRa's performance can also be affected by weather and interference.

Additionally, the system lacks advanced encryption or secure communication protocols, exposing data to unauthorized access. The HTML interface, though functional, may not be user-friendly for non-technical users, particularly in emergency situations.

These constraints should be considered when deploying the system, as they may impact performance and reliability in specific scenarios.

6.3 Improvement/Future Work

There are several key areas for improvement to make the system more reliable and user-friendly. First, adding encryption to LoRa communication will enhance security, ensuring data remains private, especially in remote areas. Expanding storage capacity by integrating a microSD card will allow the system to handle larger files, addressing the limitations of the ESP32's internal storage. The user interface can also be improved to be more intuitive, making file management easier for non-technical users, particularly in emergencies. Lastly, further testing of LoRa communication in challenging environments like forests will ensure reliable data transfer over long distances. These improvements will make the system more effective and practical for remote data exchange.

7. SOCIAL AND ENVIRONMENTAL IMPACT

This project has significant social and environmental impacts. It enhances communication in remote areas, improving emergency response and disaster relief efforts, especially in regions with limited infrastructure. By enabling data exchange in isolated environments, it can support medical, conservation, and climate monitoring efforts. Environmentally, the system can aid in monitoring ecosystems, promoting sustainable land management and conservation efforts. The ability to operate without extensive network infrastructure also reduces reliance on energy-intensive solutions, contributing to sustainability in challenging environments.

8. WORK PLAN

8.1 Timeline

Phase	Month	Description
Project Initiation	July-Aug	Defining project scope, objectives, and setting up initial environment.
Research and Literature Review	July-Aug	Review existing literature on LoRa, ESP32, SD card usage, and web servers.
Hardware Setup	Aug	Setting up and configuring ESP32, LoRa modules, and SD card. Testing hardware components.
Software Development	Aug-Sep	Coding for ESP32 web server, LoRa communication, SD card integration, and client-side HTML/JS.
Review - 1	Sep	
Software improvement	Sep - Nov	Testing and debugging the entire system—ESP32 A as server, ESP32 B as LoRa client.
Documentation	Nov	Writing and finalizing project documentation, including final report and presentation.
Project Completion	Nov	Final project review, adjustments, and presentation preparation.

8.2 Individual Contribution

The work for this project was equally distributed among three members, with each contributing to different aspects:

1. **Srivatsa S A**: Responsible for the overall **documentation**, **hardware setup**, and **code**

development, including ESP32 web server hosting, LoRa communication, and SD card integration.

2. **Sudharson Aswin N K**: Focused on **software development**, **LoRa communication protocols**, and conducted **testing** to ensure proper functionality of data transfer and the web server.
3. **Tushar Sharma**: Handled **system integration**, **UI/UX design** for the web interface, and led **final testing** to ensure smooth operation across all components.

9. COST ANALYSIS

Item	**Quantity**	**Unit Cost (INR)**	**Total Cost (INR)**
ESP32 Development Board	2	500	1000
LoRa Module (RA-02)	2	450	900
SD Card Module	1	200	200
MicroSD Card (16GB)	1	300	300
USB Connectors	2	150	300
Breadboard	1	250	250
Jumper Cables	1 Set	100	100
Software/License	-	0	0
Total Estimated Cost (INR)	-	-	**3050(Rs)**

References

- Design and implementation of a low cost web server using ESP32 for real-time photovoltaic system monitoring (https://ieeexplore.ieee.org/abstract/document/8286184)

- Design of Standalone Asynchronous ESP32 Web-Server for Temperature and Humidity Monitoring (https://ieeexplore.ieee.org/abstract/document/9441845)

- Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things (https://ieeexplore.ieee.org/abstract/document/8101926)

- An IoT Based Smart Parking System Using LoRa (https://ieeexplore.ieee.org/abstract/document/8644697)

- Low-cost and Secure Communication System for Remote Micro-grids using AES Cryptography on ESP32 with LoRa Module (https://ieeexplore.ieee.org/abstract/document/8598380)

- LoRa Network Based Multi-Wireless Sensor Nodes and LoRa Gateway for Agriculture Application (https://ieeexplore.ieee.org/abstract/document/9559804)

- WiFi and LoRa Energy Consumption Comparison in IoT ESP 32/ SX1278 Devices (https://hal.science/hal-04083109/)

- Smart Monitoring and Controlling of Appliances Using LoRa Based IoT System (https://www.mdpi.com/2411-9660/5/1/17)

- Design and Implementation of small monitoring wireless network system based on LoRa (https://ieeexplore.ieee.org/abstract/document/8997877)

- Development of air temperature and soil moisture monitoring systems with LoRA technology (https://iopscience.iop.org/article/10.1088/1742-6596/1825/1/012029/meta)

- Architecture for Drone Assisted Emergency Ad-hoc Network for Disaster Rescue

Operations (https://ieeexplore.ieee.org/abstract/document/9352814)

- A Systematic Review of Real-Time Deployments of UAV-Based LoRa Communication Network (https://ieeexplore.ieee.org/abstract/document/9530545)
- LoRa Based Smart Irrigation System (https://ieeexplore.ieee.org/abstract/document/8777583)
- WebServers on ESP32 (https://www.hackster.io/425297/webservers-on-esp32-edffef)
- (https://github.com/LeeLeahy2/SdCardServer)
- (https://github.com/espressif/arduino-esp32/tree/master/libraries/WebServer)