

## 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.

### Architecture Diagram:



### Flowchart:

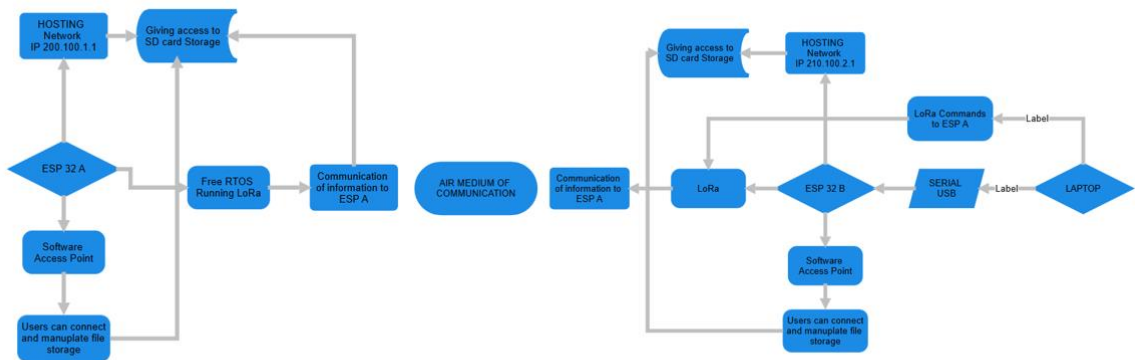


Figure 3system flow diagram

## 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.

### 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.
- 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.

## 5. RESULT ANALYSIS

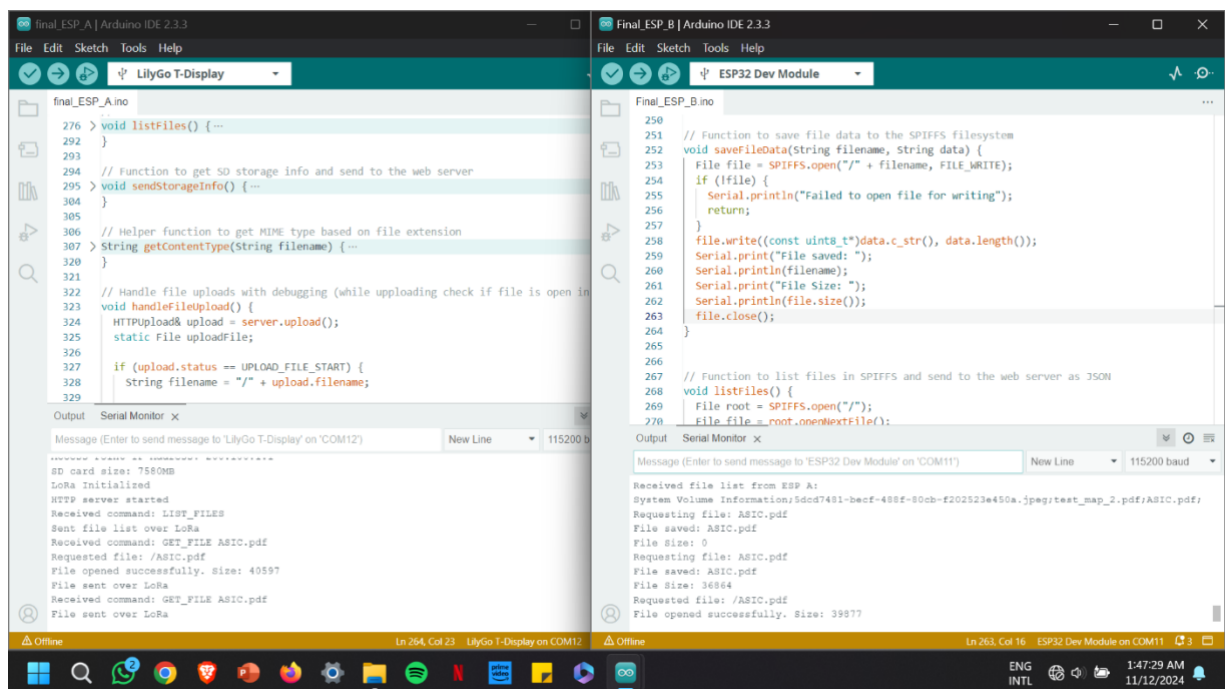


Figure 4 serial monitor of each esp nodes

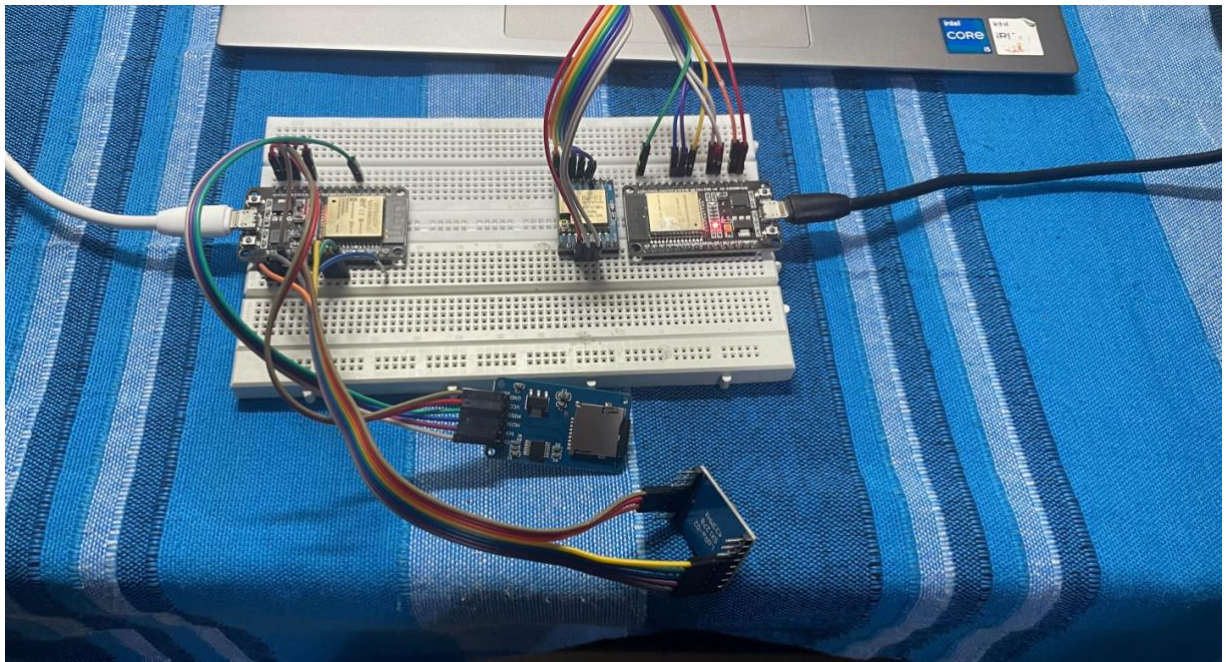


Figure 5 Setup and connection of the prototype

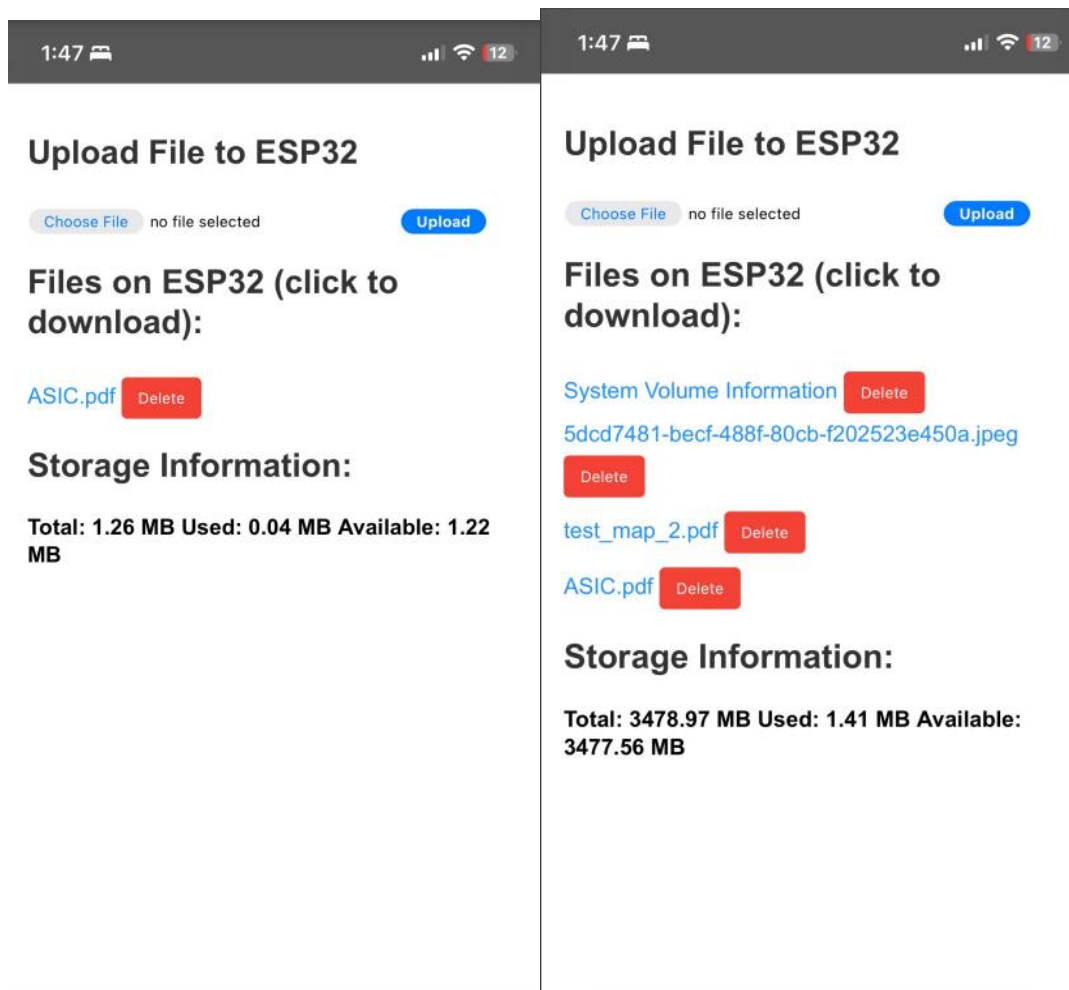


Figure 6 webserver from espB

Figure 7webserv from esp A

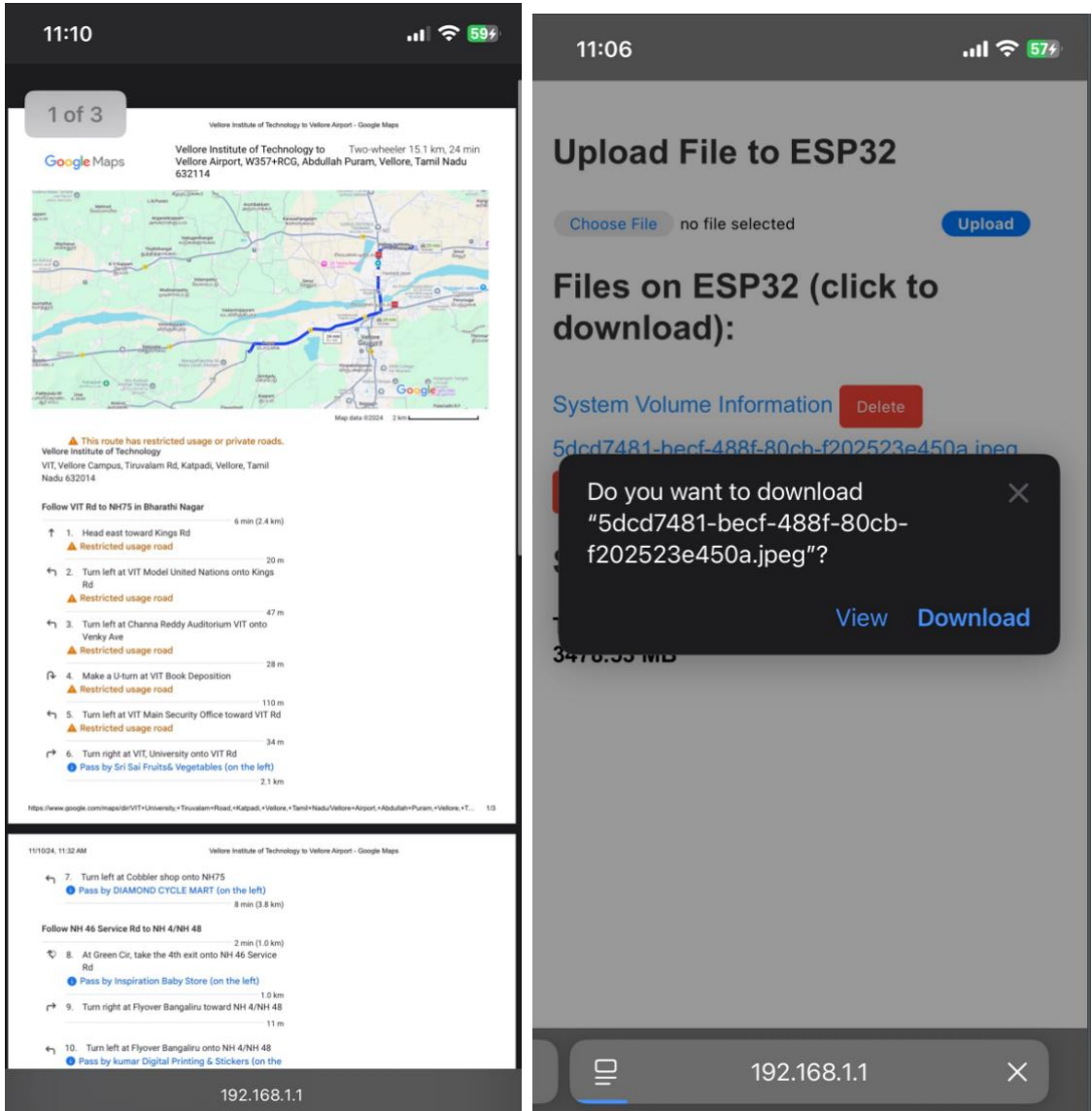


Figure 8Sample pictures of function of webserver