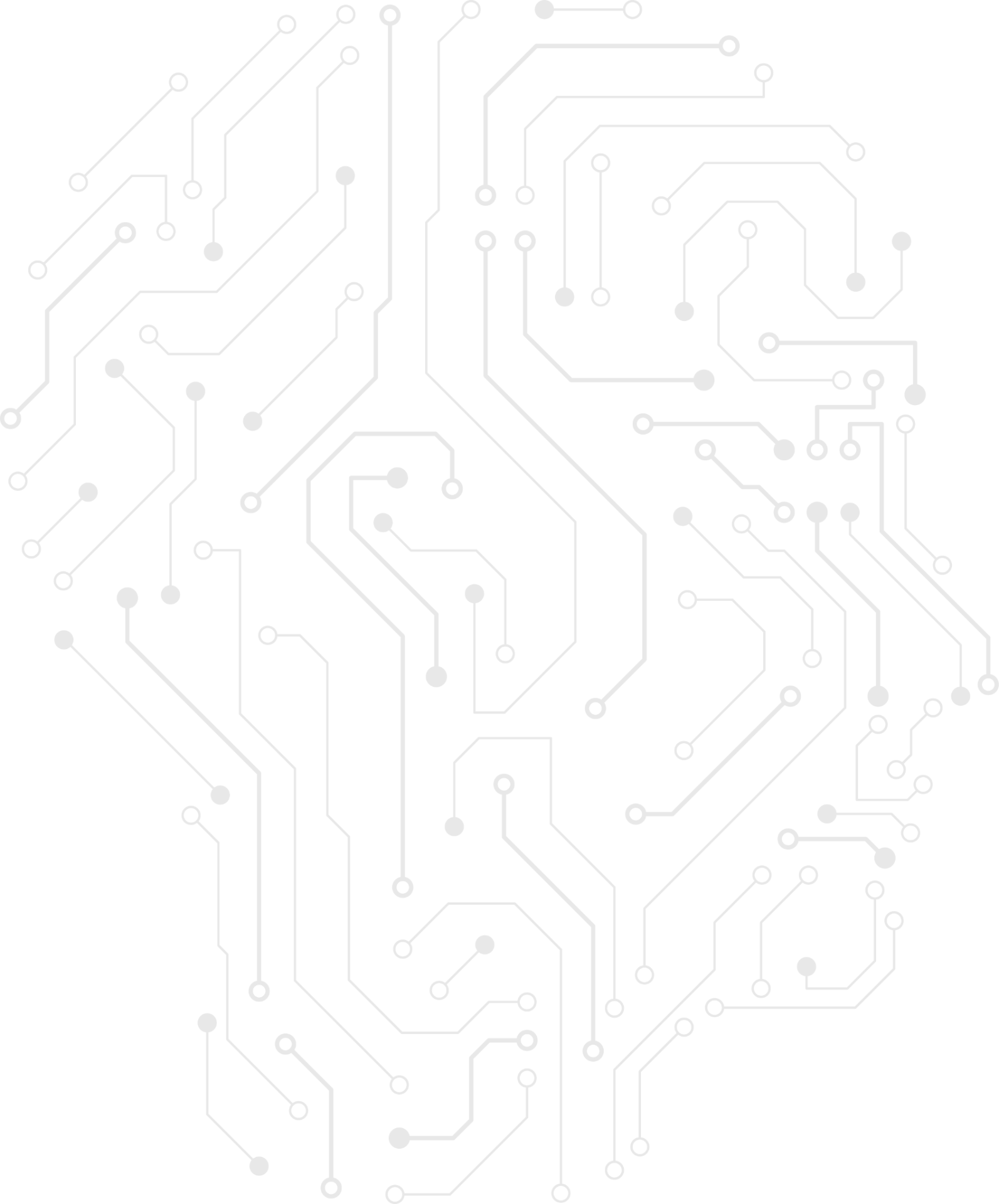
Open-Source Technologies Continuous Assessment-03

## Faculty In-Charge:

Prof. Amol Kamble

## Submitted by:

Atahrv Shelar  (24070123501)



Chavan Dhanraj (24070123501) Gulamnabi Khan(24070123501) Mayur Bhapkar (24070123501)

Kunal Ranjan (24070123510)



# Introduction:

The **Head–Tail Light Synchronization** project was developed as part of the Open-Source Technologies course to apply collaborative open-source workflows using **GitHub**.

The goal was to build an efficient, wireless, and **real-time** lighting synchronization system that ensures both the headlight and taillight of a smart helmet or vehicle operate in **perfect coordination**.

The project focuses on embedded system development and demonstrates practical skills in **IoT design**, **ESP32 programming,** and wireless data communication using the **ESP-NOW protocol**, with an emphasis on issue tracking, enhancement management, and teamwork through GitHub.

# Project Overview:

**Project Title:** Head–Tail Light Synchronization using ESP32

To design and implement a wireless synchronization system that ensures the headlight and taillight of a smart helmet or vehicle operate in perfect coordination.  
The project focuses on achieving **real-time LED synchronization** using **ESP32 microcontrollers** communicating via the **ESP-NOW protocol**.  
This enhances rider safety and visibility by maintaining consistent lighting behaviour during various modes such as steady, blinking, and hazard operations.

**Key Goals:**

* Develop an embedded system using ESP32 microcontrollers for real-time wireless communication.
* Implement synchronized LED control using PWM and MOSFET drivers.
* Practice GitHub workflows: branching, issues, commits, and merges for open-source collaboration.
* Demonstrate teamwork and project management in an open-source hardware development environment.

**Features Implemented:**

|  |  |
| --- | --- |
| **Feature** | **Description** |
| Wireless Synchronization | Real-time communication between head and tail modules using the ESP-NOW protocol. |
| Multiple Light Modes | Supports Steady, Blinking, Brake, and Hazard light operations. |
| Phase Matching | Ensures both lights blink in perfect timing using synchronized PWM signals. |
| Low Power Consumption | Optimized for battery-powered operation with efficient MOSFET LED drivers. |
| Fail-safe Mode | Tail module retains the last known state in case of wireless signal loss. |
| Compact Circuit Design | Simple, portable design using ESP32, MOSFETs, and Li-ion power supply. |
| Open-Source Collaboration | Entire project managed collaboratively via GitHub with commits, issues, and version control. |

**Tech Stack:**

|  |  |
| --- | --- |
| **Component** | **Technology Used** |
| Microcontroller | ESP32 (Head & Tail modules) |
| Wireless Protocol | ESP-NOW (peer-to-peer) |
| IDE / Code Editor | Arduino IDE / PlatformIO |
| Programming Language | C++ / Arduino Framework |
| Debugging / Logs | Serial Monitor |
| Power Supply | Li-Ion / Li-Po Battery (3.7V–5V) |
| Version Control | Git & GitHub |

**Project Structure:**

HeadTailSync/

│

├── head.ino # Code for Head (Master) ESP32

├── tail.ino # Code for Tail (Slave) ESP32

├── espnow.h # Header file for shared ESP-NOW functions / structs

├── espnow.cpp # Implementation of shared functions (optional)

└── README.md # Project overview and setup

# GitHub Repository and Issue Management:

The project was maintained using open-source best practices:

* Feature branching for each enhancement.
* Issue tracking for bugs and enhancements.
* Collaborative documentation of problems and resolutions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Issue** | **Type** | **Status** | **Reported by** |
| 1 | Add IoT and GPS Extension Documentation | Enhancement | Closed | @Atharv185 |
| 2 | Add Library Dependencies and Setup Instructions | Enhancement | Closed | @gulamnabifiroz |
| 3 | Add Safety and Error Handling Features | Enhancement | Closed | @kunalranjanbtech2023-svg |
| 4 | Add Project Images and Demonstration Video | Enhancement | Open | @Mayur4404 |
| 5 | Add Circuit Diagram and Pin Configuration Details | Enhancement | Closed | @Atharv185 |
| 6 | Clarify Code Naming and File Roles | Enhancement | Open | @Atharv185 |



# Challenges Faced:

# Ensuring real-time synchronization between Head and Tail ESP32 modules over ESP-NOW.

# Handling lost or delayed packets to maintain accurate “Head-Tail” state.

# Managing MAC address registration and peer configuration reliably.

# Debugging and monitoring ESP-NOW communication using Serial Monitor.

# Learnings and Outcomes:

Through this project, the team learned:

* Practical implementation of **ESP-NOW protocol** for low-latency peer-to-peer communication between ESP32 modules.
* How to design and debug **real-time synchronization systems** in embedded hardware projects.
* Effective use of **Git & GitHub** for collaborative development, version control, and issue tracking.
* Importance of **modular code design**, clear documentation, and structured problem-solving.
* Techniques to **handle wireless packet loss, peer registration, and reliable data transmission**.

# Conclusion:

The Head-Tail Sync project successfully demonstrates real-time synchronization between two ESP32 modules using the **ESP-NOW protocol**. The project highlights effective teamwork, modular coding, and open-source collaboration through **GitHub version control and issue tracking**. It stands as an example of how **low-latency, peer-to-peer communication systems** can be implemented efficiently in embedded IoT projects, providing practical insights into wireless data transfer, synchronization challenges, and hardware-software integration.