

## README – Rider Telemetry System (ESP32 + IMU + Local Web Dashboard)

### Overview

This project is a wearable rider telemetry system using an ESP32 and MPU6050 IMU. It measures rider motion, braking behaviour, and movement state in real time, and displays the data on a local web dashboard. All data can be downloaded as a CSV file for analysis. It runs fully offline with no cloud required. GPS support is included safely, but the system continues to work even if GPS doesn't lock.

### Hardware Needed

- ESP32 development board
- MPU6050 IMU module
- (Optional) NEO-6M GPS module
- USB cable or 5V power bank
- Jumper wires

Basic wiring:

MPU6050 → SDA to GPIO 21, SCL to GPIO 22, VCC to 3.3V, GND to GND

GPS (optional) → TX to GPIO16, RX to GPIO17, VCC to 5V, GND to GND

### Software Requirements

- Arduino IDE installed
- ESP32 board package installed in Arduino IDE
- Library required: TinyGPSPlus (only if using GPS)

### Setup Instructions

1. Open the code in Arduino IDE.
2. Install the required library if using GPS.
3. **(VERY IMP)** Update the Wi-Fi hotspot name and password in the code.
4. Upload the code to the ESP32.
5. Turn on your phone hotspot and keep your phone connected.
6. Open the Serial Monitor to see the IP address printed by the ESP32.
7. Enter that IP address in a browser to open the dashboard.

Example browser link format:

[http://<IP shown in Serial Monitor>/ \(most probably – 10.117.126.199 \)](http://<IP shown in Serial Monitor>/)

### How It Works

- Connect ESP32 to your phone hotspot.
- Access the live dashboard in any browser.
- Pitch, roll, filtered acceleration, linear acceleration, state detection, and braking status update live.

- You can view a results summary page and download the recorded data as a CSV file.

If GPS is connected and gets a satellite lock outdoors, speed and location will also appear. If not, it safely shows “Searching for GPS...” and logs zeros instead, without affecting IMU functions.

### Customization for Other Users

- Change Wi-Fi hotspot name and password at the top of the code.
- Edit the logging interval (default 500 ms) to store data more or less frequently.
- Increase or reduce sample storage size if needed.
- GPS can be disabled by removing the GPS-related lines; the code still works fully with IMU only.

### Important Notes

- GPS only works outdoors and may take 30–90 seconds for a satellite fix.
- The system works completely without GPS, so it remains stable in all conditions.
- CSV download contains all recorded sensor values and timestamps for analysis.