Operational Document

Cement Mixer Monitoring System

1. Introduction

This document outlines the complete operational workflow for deploying, managing, and maintaining the Cement Mixer Monitoring System developed during the internship. It is intended to be a standalone reference for anyone continuing this work, ensuring that setup, configuration, and troubleshooting are all clearly documented.

2. System Overview

The Cement Mixer Monitoring System is designed to track the real-time operational status of cement mixers through two custom-built IoT circuits:

- Circuit 1: Measures angular velocity and linear acceleration using the MPU6050 sensor.

- Circuit 2: Measures current drawn by the mixer motor using the SCT013 current sensor.

Data Flow:
Sensor → ESP8266 → Wi-Fi → Flask Server → CSV Log File

Both ESP8266-based circuits transmit sensor data over Wi-Fi to a local Flask server running on a laptop, which logs the data into a CSV file for further analysis.

3. Hardware Setup

Components:

- ESP8266 (NodeMCU) – 2 units

- MPU6050 Accelerometer + Gyroscope – 1 unit

- SCT013 Current Sensor – 1 unit

- External LEDs – 2 units

- Resistors, breadboard, jumper wires

- 9V batteries with snap connectors

Placement:

- MPU6050 Sensor: Attached using double-sided tape to any rotating surface of the cement mixer. For future deployments, this can be PCB-mounted and bolted for stability.

- SCT013 Sensor: Placed on the ground with a single wire (live or neutral) running through its core. Do not insert both live and neutral.

LED Indicators:

- External LED: Glows when the circuit is powered.

- Internal ESP8266 LED: Blinks when data is transmitted.

Note: Currently, no external casing is used. Weatherproof enclosures are recommended for future deployments.

4. Power Supply

Each ESP8266 board is powered by a 9V battery using a battery snap. Power setup is independent of the cement mixer (which runs on 230V AC). For long-term deployment, LiPo batteries or other rechargeable options are recommended. Always check battery level before data collection; low power can lead to sensor failure.

5. Wi-Fi Configuration

The ESP8266 must connect to a local Wi-Fi network to transmit data. SSID and Password must be updated in the Arduino code before uploading. Both circuits send data to the IP address of the laptop running the Flask server. If the server is moved to another system, update the IP in both circuit codes and re-upload.

6. Flask Server Setup

Requirements:

- OS: Windows

- Python: Version 3.11+

- Libraries: Flask, datetime, csv

Steps:

1. Clone the GitHub repo or copy the flask_server.py file from the /Codes folder.

2. Open terminal or command prompt in that directory.

3. Install Flask: pip install flask

4. Run the server: python flask_server.py

5. The server runs at: http://<your_ip>:5000/

Note: The server must be running before activating the circuits.

Server Notes: Manually run the server each time data collection is started.

7. Data Transmission Flow

Circuit 1 (MPU6050) sends:

- x_gyro
- y_gyro
- z_gyro
- accel_euclidean_sum

Circuit 2 (SCT013) sends:

- current

CSV Logging: The data is logged to a file named in flask_server.py. It appends to the same file. Change the filename in the code to log data to a new file. Do not open the CSV in Excel during data collection as this will block write access and result in data loss.

8. Testing Procedures

Before Deployment:

- Upload the test sketches from the /Tests folder.

- Ensure Flask server receives the expected POST data.

- Use LED behavior to confirm functioning.

Debugging blocks are present in each circuit's code to help verify sensor values and connectivity during testing.

9. Troubleshooting

Common Issues:

- Garbage MPU6050 values: Fix unstable wiring or solder properly.

- Server not receiving data: Check IP/network match.

- No data in CSV: Avoid opening CSV in Excel while collecting data.

- ESP disconnects: Replace low battery.

- LED off: Reupload code or check wiring.

10. Shutdown & Restart Instructions

Starting:
1. Start Flask server.
2. Power on the circuits.

Stopping:
1. Power off the circuits.
2. Stop Flask server manually.

11. Maintenance Guidelines

Daily Before Use:

- Check Wi-Fi credentials and IP configuration.

- Use test script or serial monitor to verify startup.

- Observe LED activity for confirmation.

Battery Maintenance: Replace batteries as needed.

Data Management: A new data point is logged every 5 seconds. Manually archive the CSV file after each session to avoid data loss.