كليـــــة العلـــوم والتقنيـــات – مراكش
FACULTÉ DES SCIENCES
ET TECHNIQUES – MARRAKECH

Setup Document for UGV Real-Time Monitoring Project

---

1. Introduction

This project focuses on the development of a real-time monitoring system for an Autonomous Unmanned Ground Vehicle (UGV). The UGV is equipped with solar panels for energy harvesting and various sensors to monitor environmental conditions, track its performance, and optimize energy use.

The key components of the project are:

1. **UGV Monitoring**: Real-time tracking of the UGV's performance, including speed, battery levels, and GPS position.
2. **Solar Panel Monitoring**: Monitoring solar energy production, panel orientation, and historical data.
3. **Environmental Data Monitoring**: Collecting and visualizing environmental metrics like temperature, humidity, and light levels.

The project integrates a **Streamlit dashboard** for data visualization, supported by a **FastAPI backend** to manage real-time data transfers between the UGV and the dashboard. The UGV sends data from its sensors to the FastAPI server, which processes the data for visualization and alerting in the dashboard.

---

2. Prerequisites

Ensure the following are installed on your system:

- **Python 3.8+**
- **Streamlit** (pip install streamlit)
- **FastAPI** (pip install fastapi)
- **Uvicorn** (pip install uvicorn)
- **Additional Libraries**:
  - pvlib for solar panel monitoring (pip install pvlib)
  - pandas, numpy for data processing (pip install pandas numpy)

- requests for HTTP requests from the UGV to the FastAPI server (pip install requests)

---

3. Project Directory Structure

Create a project directory with the following structure:

- UGV_Monitoring_Project

dashboard_main.py : Main dashboard script (integrated solar panel monitoring)
api_server.py : FastAPI server script

ugv_monitoring.py : UGV monitoring module

environmental_data_monitoring.py: Environmental data monitoring

reports_alerts.py : Reports and alerts module

ugv_data_sender.py : UGV data sender script (executed on the UGV)

---

4. Setting Up the UGV Data Sender

The UGV is equipped with a script that sends real-time data, such as speed, battery levels, GPS position, and IMU sensor readings (pitch, yaw, roll), to the FastAPI server. This script runs continuously on the UGV to simulate and transmit data at regular intervals.

Ensure the UGV's system has the necessary environment set up with **Python 3.x** and the requests library for HTTP communication.

---

5. Setting Up the FastAPI Server

The FastAPI server handles receiving data from the UGV, storing it temporarily, and making it available for the dashboard to fetch and display in real-time. This is a key component for enabling the UGV data to be transmitted and visualized efficiently on the dashboard.

To run the FastAPI server:

1. Navigate to the project directory.
2. Run the following command:

```
uvicorn api_server:app --reload
```

---

6. Setting Up the Dashboard

The **Streamlit** dashboard is the user interface where all the UGV data is visualized. It consists of multiple sections:

- **UGV Monitoring**: Displays speed, battery status, and GPS data in real-time.
- **Solar Panel Monitoring**: Shows energy production, panel orientation, and history.
- **Environmental Data Monitoring**: Visualizes temperature, humidity, and light levels.
- **Reports and Alerts**: Manages notifications for critical conditions (e.g., low battery) and generates reports.

To start the dashboard:

1. Navigate to the project directory.
2. Run the following command:

```
streamlit run dashboard_main.py
```

---

7. Running the Project

1. **Start the FastAPI server:**
   - Navigate to the project directory.
   - Run the following command to start the FastAPI server:

   ```
   uvicorn api_server:app --reload
   ```

2. **Start the UGV data sender:**
   - On the UGV system, execute the ugv_data_sender.py script:

   ```
   python ugv_data_sender.py
   ```

3. **Run the Streamlit dashboard**:
   - In the project directory, run the following command to start the dashboard:

   ```
   streamlit run dashboard_main.py
   ```

4. **Access the dashboard:**
   - Open a web browser and navigate to http://localhost:8501 (or use the server IP if remote).