Final Project

IOT23, “Software Developer, Embedded Systems and IoT”, Nackademin

IoT-based Animal Tracking Solution Using Wi-Fi Positioning

Lucia Moonlit

Executive Summary

This project focuses on developing a real-time animal tracking system using Wi-Fi-based positioning. The device collects Wi-Fi metadata and sends it via the MQTT protocol to a cloud platform (Adafruit IO), where the data is stored. From the Adafruit IO feed, the data is retrieved into a JSON file, converted into geolocation (latitude and longitude), and visualized on a map. This system enables tracking of an animal's position using Wi-Fi positioning, which functions reliably in crowded areas, indoors, and is unaffected by weather conditions. The device has been successfully tested for accurate position tracking, and the data is displayed on a map for easy monitoring, providing a robust solution for animal location tracking.

Introduction

With the rise of pet tracking technologies, ensuring the safety and location of pets in real-time is becoming increasingly important. Many current tracking systems rely heavily on GPS, which can be unreliable in certain environments, such as indoors or during bad weather. This project was driven by the need to create a more reliable and accessible tracking solution.

Throughout the development process, I tested several tracking technologies, but encountered significant challenges:

- **SIM800 module (2G)**: This module failed to provide reliable connectivity, likely due to the limitations of 2G networks and poor signal reception in certain areas.
- **NEO GPS module**: Despite attempts to use this GPS module, I was unable to obtain any signal, especially in areas with poor satellite visibility.
- **Heltec module for LoRa**: I was unable to connect to any nearby LoRa hotspots, limiting the effectiveness of this module.
- **GPS module for outdoors**: While this module worked in clear weather, its performance was unreliable during poor weather conditions (e.g., cloudy or snowy days).

Given these challenges, I shifted to a **Wi-Fi-based positioning** solution that would work indoors, be less dependent on weather, and offer more consistent performance in urban environments.

Scope and Limitations:

- This project focuses solely on Wi-Fi-based positioning for tracking, excluding other technologies such as cellular, Bluetooth, and GPS.
- The system is designed primarily for tracking pets and animals, with an emphasis on portability and ease of use.

Implementation and Solution

Hardware:

- **LilyGO7000G**: A microcontroller with integrated LTE capabilities, used for data transmission.
- **LTE antenna**: Enables the device to connect to the LTE network for sending data.
- **LiPo battery** (18650) with external charging: Provides power to the device, allowing it to be portable.
- **SIM card** (Comviq, prepaid data): Provides cellular connectivity for the device.
- **USB-C cable**: Used for device programming and charging.

Software:

- **Platform IO**: A development environment used to write and upload code to the LilyGO7000G device. It uses the Arduino framework for easy integration with various sensors and modules.
- **Adafruit IO**: Cloud platform for storing and visualizing data received from the device, enabling easy monitoring of the pet's location.
- **MQTT**: A lightweight messaging protocol used for communication between the device and the cloud. This protocol ensures reliable and efficient data transfer.
- **Node.js**: Used for server-side scripting to retrieve, process, and store data from Adafruit IO. Node.js enables real-time updates and facilitates interaction with the front-end.
- **Leaflet**: A JavaScript library used to visualize the device's location on an interactive map. Leaflet is used to render geographic data from JSON files, making it easy to track the pet's location in real-time.
- **Google Maps Geolocation API (API key)** for converting Wi-Fi metadata into geographical coordinates
- **MongoDB:** A cloud-based NoSQL database platform used to store data from the device in a structured yet flexible format

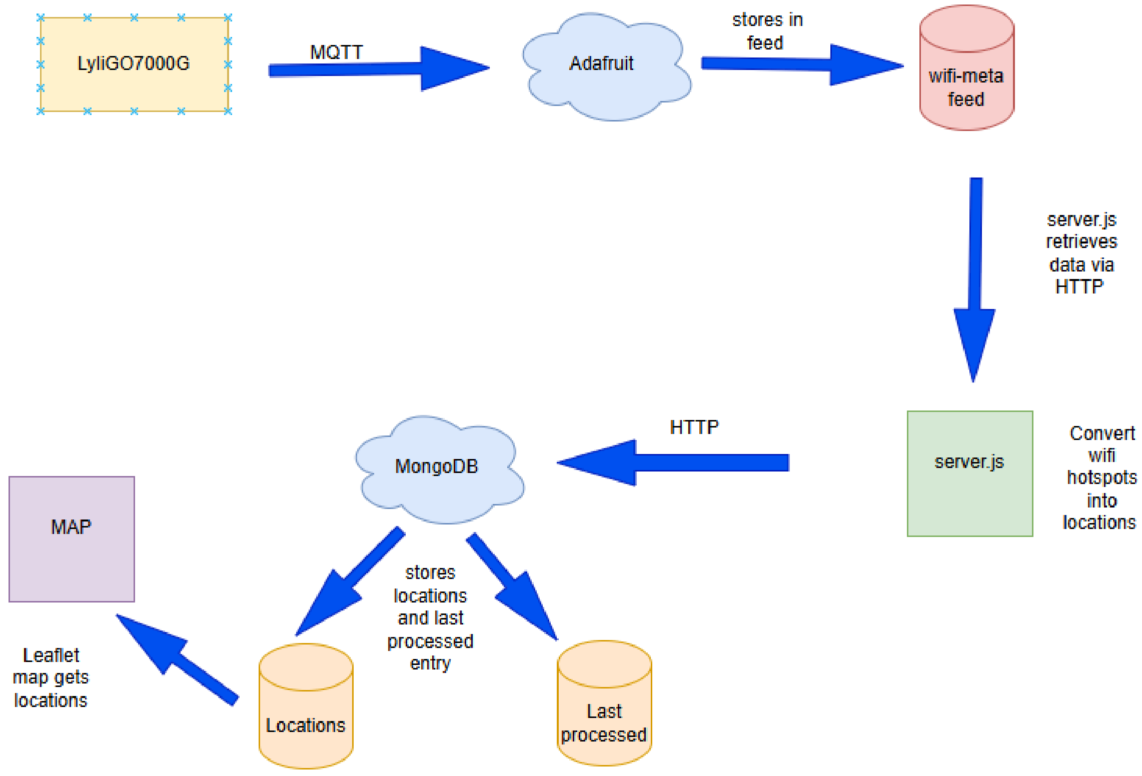
Process:

1. **Device Setup**: The device is connected to the LTE network using an APN and a SIM card for internet access.
2. **MQTT:** Device collects Wi-Fi-access points (which is Mac address and Signal Strength (RSSI) of hotspots) and send them into Adafruit IO feed via MQTT protocol.

3. **Retrieval and conversion of metadata:** Data from Adafruit feed are retrieved into **server.js**, convert them into location data (latitude, longitude) with help of **Google API key**, and store them in **MongoDB** cluster.

4. **Visualisation of location:** Data stored in MongoDB cluster is used for visualisation on the map using Leaflet library.

How are all parts connected?



Starting HTTP server:

To retrieve and process data, it is needed to run the **server.js** script. The server will constantly check for updates and process new data.

1. **Install node.js**
2. **Start the HTTP server**
    - Navigate to the project directory in your terminal and run command

      node server.js

    - This will start your server and begin retrieving data from Adafruit IO, processing it, and storing it in ***MongoDB cluster*** for visualization.

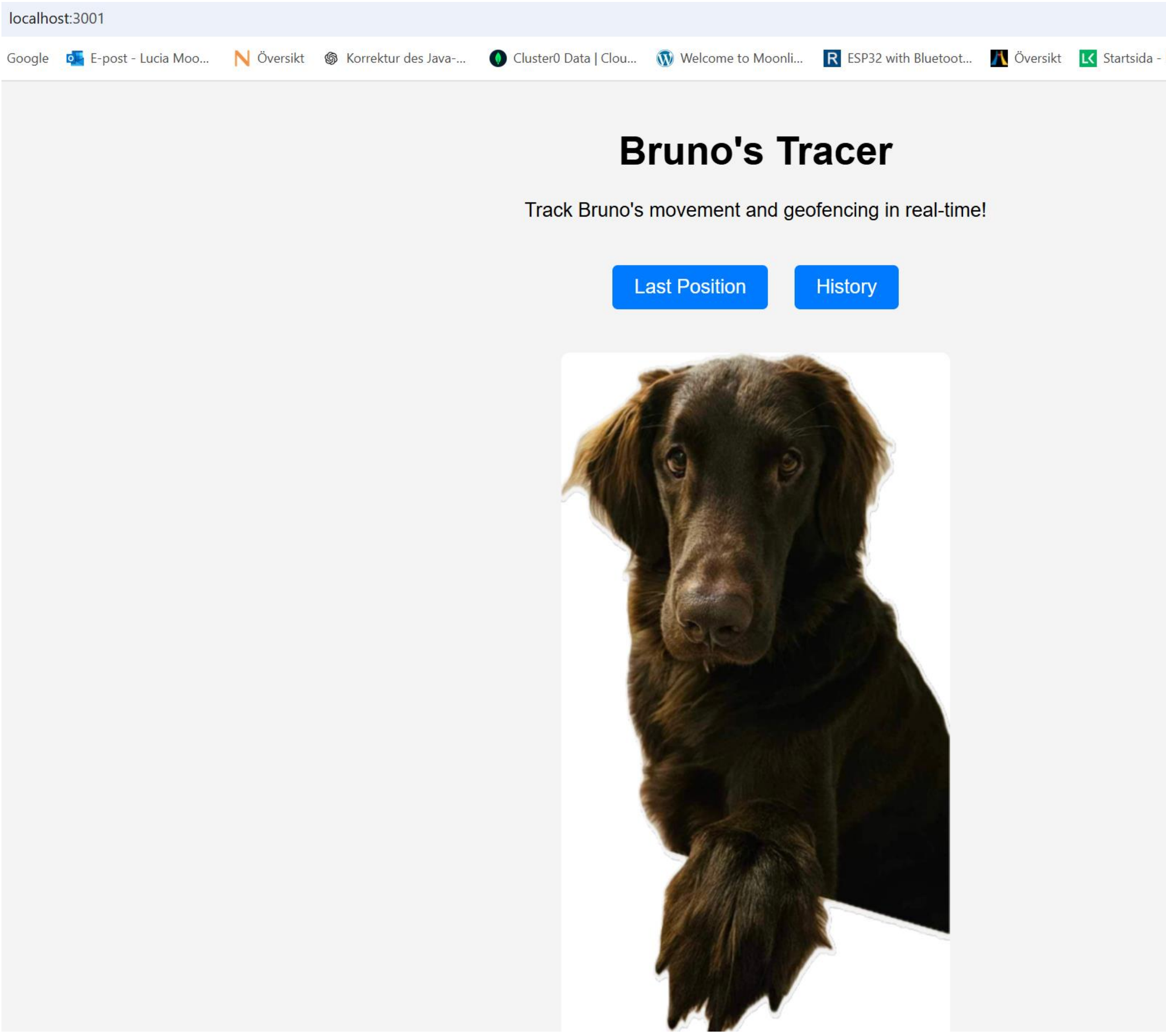
3. **Keeping the Server Running with pm2:**
    - To ensure that your server runs continuously in the background (even after you close the terminal or restart the machine), you can use pm2 (Process Manager 2).
    - Install pm2 globally: **npm install pm2 -g**
    - Start your server: **pm2 start server.js**
    - Save pm2 processes: **pm2 save**

- To ensure pm2 restarts your server automatically after a reboot: **pm2 startup** This will configure your system to automatically restart the server on reboot.

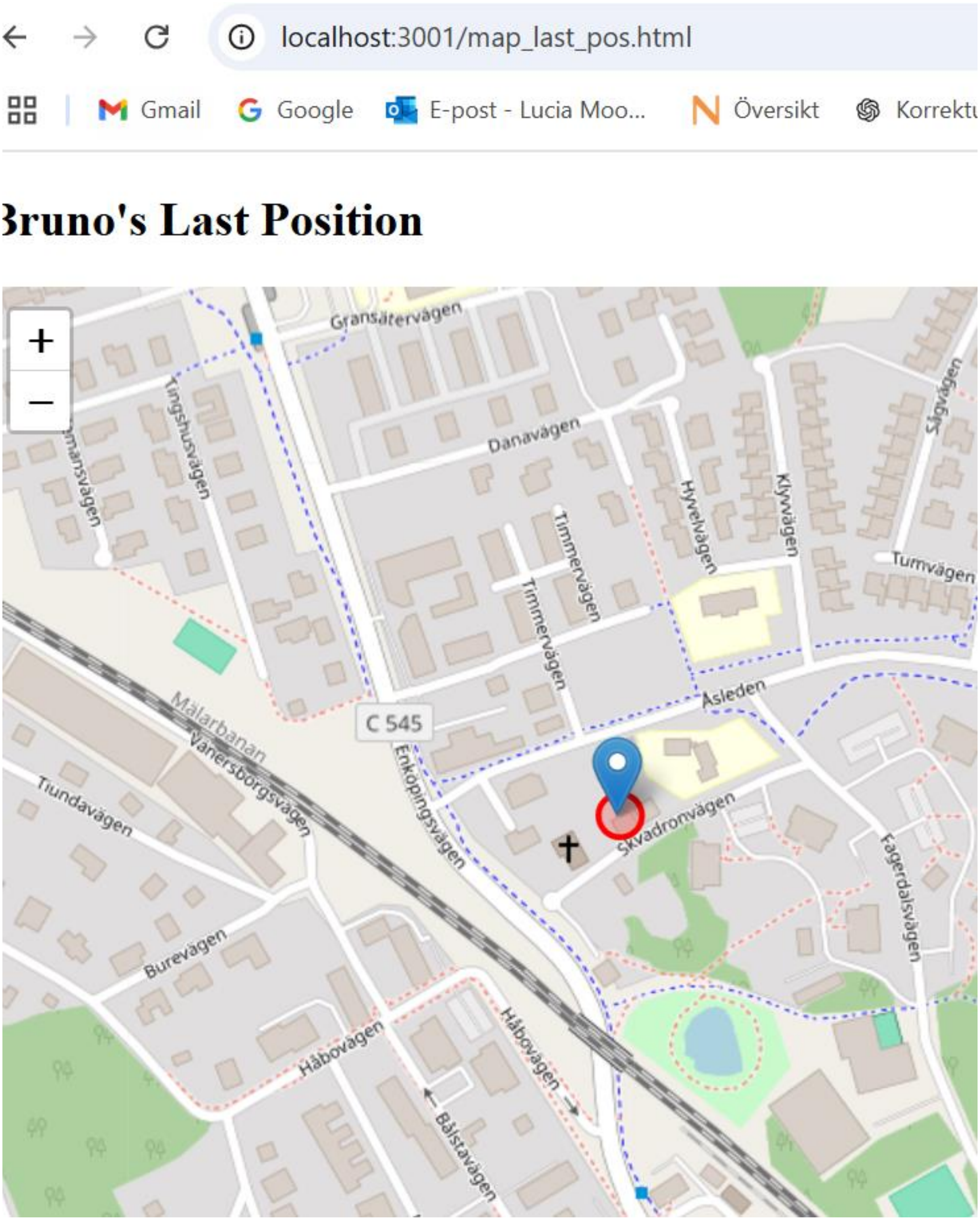
Results

- **Collecting metadata**: The device was able to successfully collect and send Wi-Fi metadata to Adafruit IO feed
- **Convert metadata**: ***server.js*** successfully retrieved data from Adafruit feed and converted them into location (latitude, longitude).
- **Visualisation**:

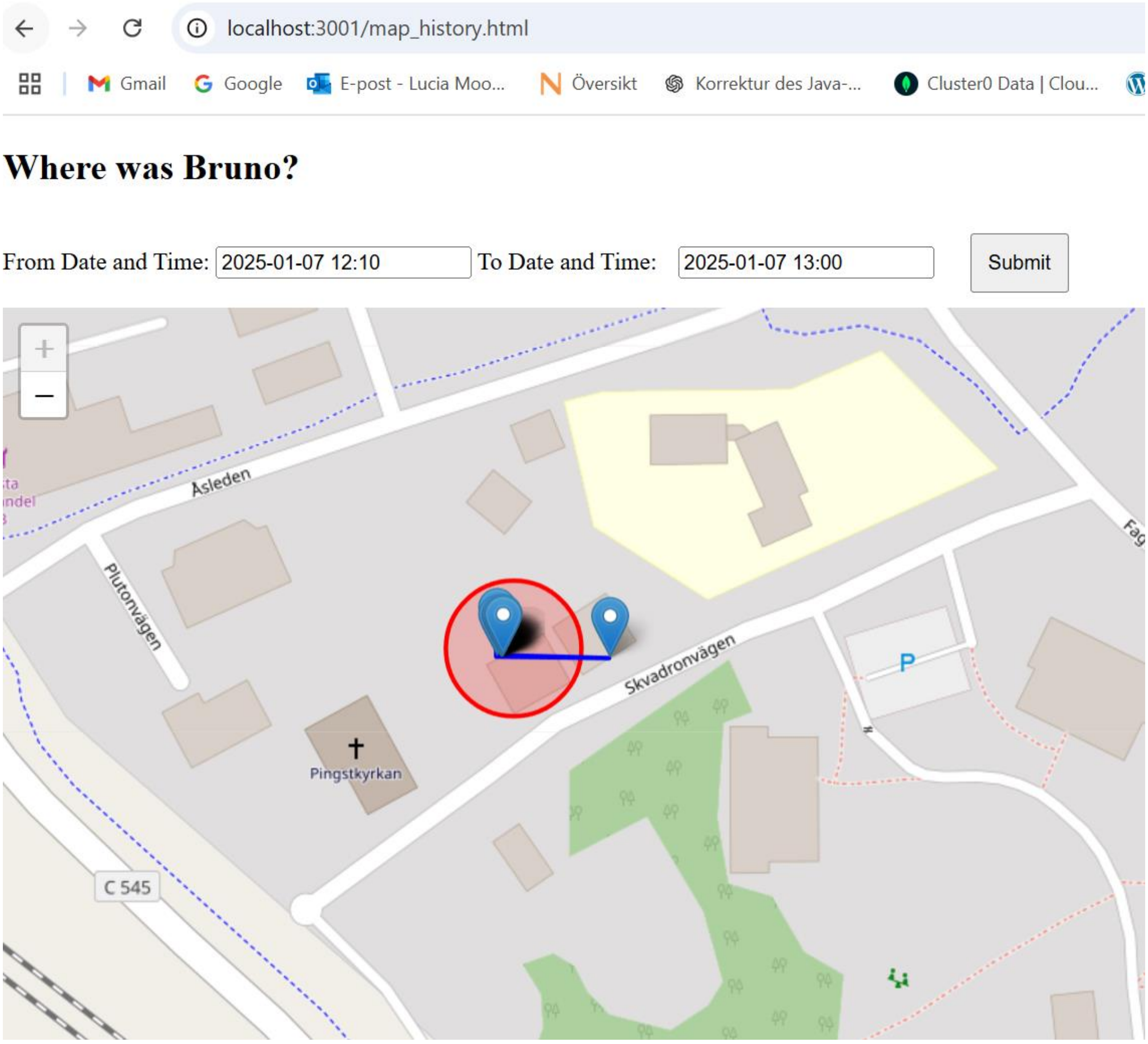
First comes index.html



If chosen Last Position button:



When chosen history:



Discussion and Conclusions

- **Challenges**:
    - GPS positioning was hindered by poor weather conditions, and the battery connection was unstable during some tests. Wi-Fi positioning was more reliable. Which resulted to remove GPS part from project completely.
    - The device's battery holder was malfunctioning (likely a factory fault), requiring a workaround using jumpers to connect the battery.
    - Cold weather caused malfunctions, affecting data transmission intervals and reducing battery life.
    - The map visualization needed adjustment to filter and show data based on specific date/time ranges to avoid constant updates.

Future Work:

- Combining GPS and Wi-Fi positioning in real-time could improve accuracy and reliability.
- Implementing geofencing capabilities and notifications would enhance user experience by alerting when the animal moves outside a predefined area.

Suggestions for Improvement:

- **Hybrid GPS/Wi-Fi**: A combination of GPS and Wi-Fi would improve the accuracy and reliability of the tracking system.
- **Front end:** This project aimed for functionality rather than user's experience. Nicer front end to make it more appealing can make

Sources:

https://io.adafruit.com

https://cloud.mongodb.com

https://cloud.google.com