

# **Satellite GPS Safety Beacon System**

Autonomous Localization and Tracking Solution

Melisa Khacef  
Lina Belaid

Master 2 ESTEL  
Université Côte d'Azur

Academic Year 2025–2026

# Contents

<b>1 Project Overview</b>	<b>2</b>
<b>2 System Architecture</b>	<b>3</b>
<b>3 Embedded System Design</b>	<b>4</b>
3.1 Coordinate Encoding . . . . .	4
<b>4 Node-RED Processing</b>	<b>5</b>
4.1 Flow Overview . . . . .	5
4.2 Debug Validation . . . . .	6
<b>5 Distance Computation</b>	<b>7</b>
<b>6 InfluxDB Storage</b>	<b>8</b>
6.1 Database Configuration . . . . .	8
6.2 Stored Distance Data . . . . .	9
6.3 Longitude Data Example . . . . .	9
<b>7 Visualization with Grafana</b>	<b>10</b>
7.1 Trajectory Visualization . . . . .	10
7.2 Detailed Map View . . . . .	11
7.3 Data Inspection . . . . .	12
<b>8 Node-RED Dashboard Interface</b>	<b>13</b>
<b>9 Energy Consumption Analysis</b>	<b>15</b>
<b>10 Experimental Results</b>	<b>16</b>
<b>11 Limitations and Improvements</b>	<b>17</b>
11.1 Current Limitations . . . . .	17
11.2 Future Improvements . . . . .	17
<b>12 Conclusion</b>	<b>18</b>

# Chapter 1

## Project Overview

This project presents the design and implementation of a fully autonomous satellite-based GPS tracking system intended for isolated or emergency environments where GSM and WiFi infrastructures are unavailable.

The objective is to create a compact and energy-efficient safety beacon capable of:

- Acquiring precise GPS coordinates
- Encoding them efficiently for satellite transmission
- Transmitting them via satellite uplink
- Processing and decoding them remotely
- Storing them in a time-series database
- Visualizing them in real time

The complete system forms an end-to-end IoT architecture.

# Chapter 2

## System Architecture



Figure 2.1: Global system architecture

The system pipeline is structured as follows:

1. GPS module acquisition
2. Microcontroller processing and encoding
3. Satellite uplink transmission
4. MQTT reception
5. Node-RED decoding and processing
6. InfluxDB time-series storage
7. Grafana visualization

Each component has been optimized for energy efficiency and reliability.

# Chapter 3

## Embedded System Design

The embedded device performs the following operations:

### 3.1 Coordinate Encoding

GPS coordinates (latitude and longitude) are:

1. Multiplied by  $10^6$  to preserve microdegree precision
2. Converted to signed 32-bit integers
3. Encoded in Big-Endian format
4. Converted into an 8-byte binary frame
5. Transformed into a 16-character HEX string

Final satellite command:

```
AT+SEND=1,0,16,1,<HEX_PAYLOAD>
```

This method reduces bandwidth usage while maintaining high positional precision.

# Chapter 4

## Node-RED Processing

### 4.1 Flow Overview

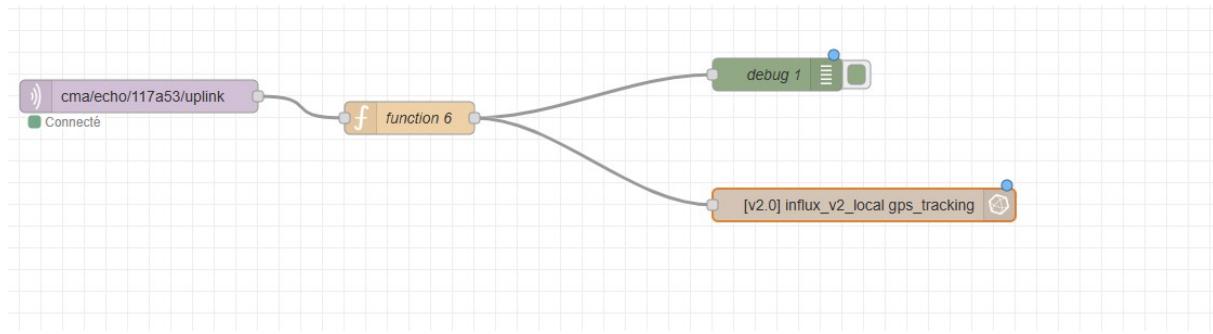
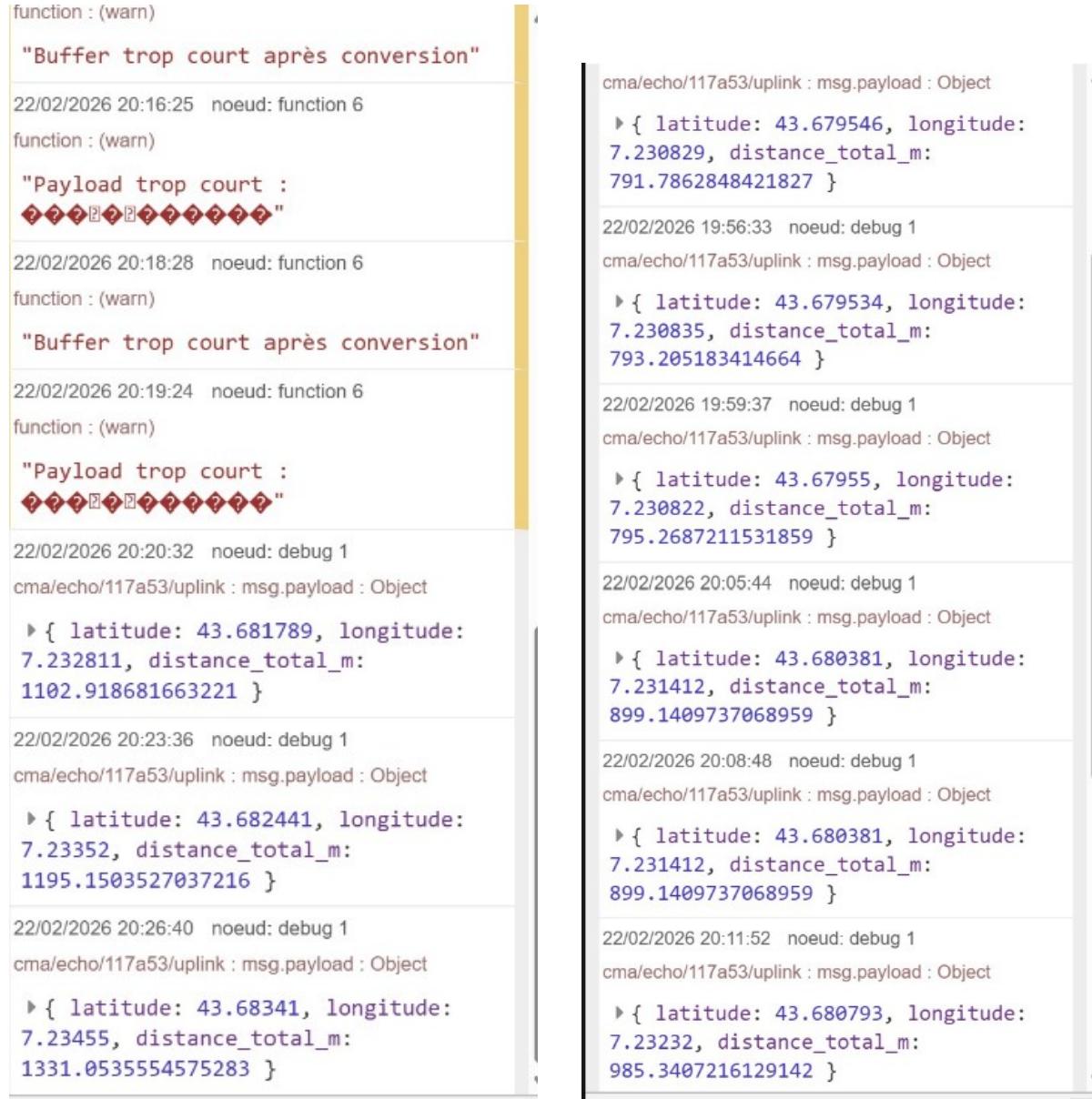


Figure 4.1: Node-RED processing flow

The Node-RED workflow performs:

- MQTT subscription to satellite uplink topic
- Base64 decoding
- HEX to binary conversion
- Big-endian integer extraction
- Distance computation
- Data forwarding to InfluxDB

## 4.2 Debug Validation



The screenshot shows two panels of Node-RED logs. Panel (a) on the left displays the decoded GPS payload, showing messages like "Buffer trop court après conversion" and "Payload trop court" followed by binary data. Panel (b) on the right shows progressive distance accumulation over time, with messages like "cma/echo/117a53/uplink : msg.payload : Object" containing JSON objects with latitude, longitude, and distance values.

```

function : (warn)
"Buffer trop court après conversion"
22/02/2026 20:16:25 noeud: function 6
function : (warn)
"Payload trop court :
? ? ? ? ? ? ? ? ? ? ? ?"
22/02/2026 20:18:28 noeud: function 6
function : (warn)
"Buffer trop court après conversion"
22/02/2026 20:19:24 noeud: function 6
function : (warn)
"Payload trop court :
? ? ? ? ? ? ? ? ? ? ? ?"
22/02/2026 20:20:32 noeud: debug 1
cma/echo/117a53/uplink : msg.payload : Object
▶ { latitude: 43.681789, longitude:
7.232811, distance_total_m:
1102.918681663221 }

22/02/2026 20:23:36 noeud: debug 1
cma/echo/117a53/uplink : msg.payload : Object
▶ { latitude: 43.682441, longitude:
7.23352, distance_total_m:
1195.1503527037216 }

22/02/2026 20:26:40 noeud: debug 1
cma/echo/117a53/uplink : msg.payload : Object
▶ { latitude: 43.68341, longitude:
7.23455, distance_total_m:
1331.0535554575283 }

cma/echo/117a53/uplink : msg.payload : Object
▶ { latitude: 43.679546, longitude:
7.230829, distance_total_m:
791.7862848421827 }

22/02/2026 19:56:33 noeud: debug 1
cma/echo/117a53/uplink : msg.payload : Object
▶ { latitude: 43.679534, longitude:
7.230835, distance_total_m:
793.205183414664 }

22/02/2026 19:59:37 noeud: debug 1
cma/echo/117a53/uplink : msg.payload : Object
▶ { latitude: 43.67955, longitude:
7.230822, distance_total_m:
795.2687211531859 }

22/02/2026 20:05:44 noeud: debug 1
cma/echo/117a53/uplink : msg.payload : Object
▶ { latitude: 43.680381, longitude:
7.231412, distance_total_m:
899.1409737068959 }

22/02/2026 20:08:48 noeud: debug 1
cma/echo/117a53/uplink : msg.payload : Object
▶ { latitude: 43.680381, longitude:
7.231412, distance_total_m:
899.1409737068959 }

22/02/2026 20:11:52 noeud: debug 1
cma/echo/117a53/uplink : msg.payload : Object
▶ { latitude: 43.680793, longitude:
7.23232, distance_total_m:
985.3407216129142 }

```

(a) Decoded GPS payload      (b) Distance accumulation over time

Figure 4.2: Node-RED decoding and distance computation results

The decoded output confirms:

- Correct latitude extraction
- Correct longitude extraction
- Progressive distance calculation

# Chapter 5

## Distance Computation

Distance between two consecutive GPS points is computed using the Haversine formula:

$$d = R \cdot 2 \cdot \arctan(\sqrt{a}, \sqrt{1-a})$$

with:

$$a = \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\Delta\lambda}{2}\right)$$

Where:

- $R = 6,371,000$  meters
- $\phi$  = latitude
- $\lambda$  = longitude

The cumulative distance is stored using Node-RED flow context memory.

# Chapter 6

## InfluxDB Storage

### 6.1 Database Configuration

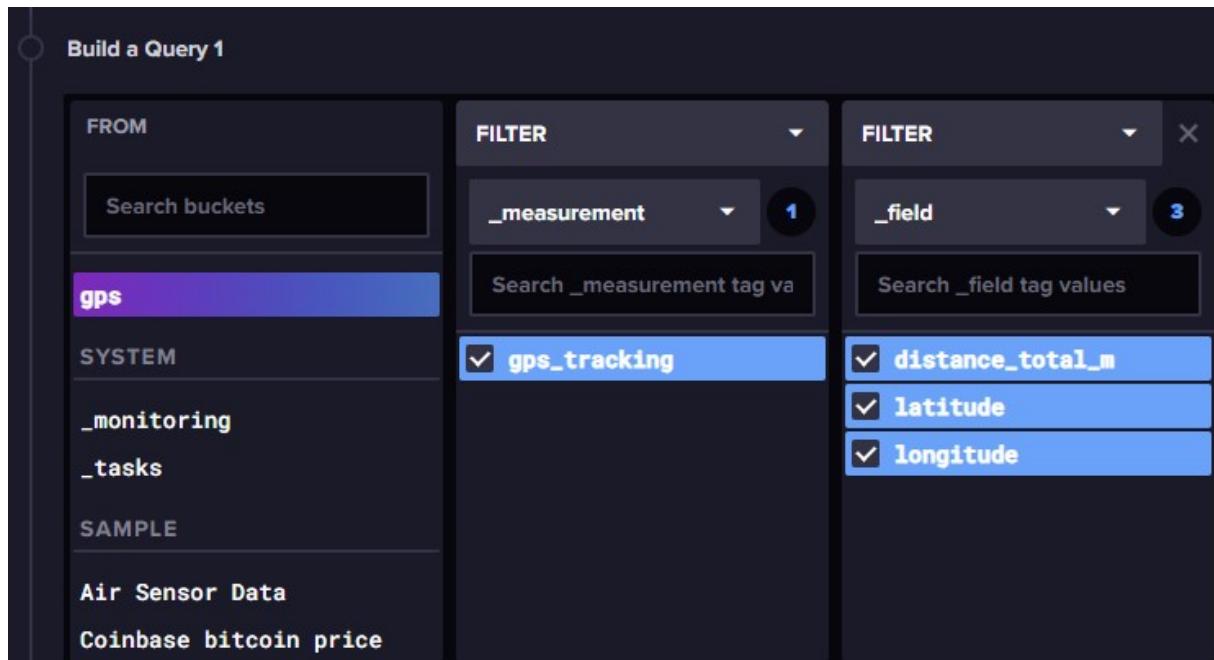


Figure 6.1: InfluxDB bucket configuration

Data structure:

- Bucket: gps
- Measurement: gps\_tracking
- Fields:
  - latitude
  - longitude
  - distance\_total\_m

## 6.2 Stored Distance Data

2026-02-22 15:29:56 G...	2026-02-22 21:29:56 G...	2026-02-22 19:21:14 GM...	<b>4,77</b>	distance_total_m	gps_tracking
2026-02-22 15:29:56 G...	2026-02-22 21:29:56 G...	2026-02-22 19:53:29 G...	<b>791,79</b>	distance_total_m	gps_tracking
2026-02-22 15:29:56 G...	2026-02-22 21:29:56 G...	2026-02-22 19:56:33 G...	<b>793,21</b>	distance_total_m	gps_tracking
2026-02-22 15:29:56 G...	2026-02-22 21:29:56 G...	2026-02-22 19:59:36 G...	<b>795,27</b>	distance_total_m	gps_tracking
2026-02-22 15:29:56 G...	2026-02-22 21:29:56 G...	2026-02-22 20:05:44 G...	<b>899,14</b>	distance_total_m	gps_tracking
2026-02-22 15:29:56 G...	2026-02-22 21:29:56 G...	2026-02-22 20:08:48 G...	<b>899,14</b>	distance_total_m	gps_tracking
2026-02-22 15:29:56 G...	2026-02-22 21:29:56 G...	2026-02-22 20:11:52 G...	<b>985,34</b>	distance_total_m	gps_tracking
2026-02-22 15:29:56 G...	2026-02-22 21:29:56 G...	2026-02-22 20:20:32 G...	<b>1 102,92</b>	distance_total_m	gps_tracking
2026-02-22 15:29:56 G...	2026-02-22 21:29:56 G...	2026-02-22 20:23:36 G...	<b>1 195,15</b>	distance_total_m	gps_tracking
2026-02-22 15:29:56 G...	2026-02-22 21:29:56 G...	2026-02-22 20:26:40 G...	<b>1 331,05</b>	distance_total_m	gps_tracking
2026-02-22 15:29:56 G...	2026-02-22 21:29:56 G...	2026-02-22 20:29:44 G...	<b>1 394,48</b>	distance_total_m	gps_tracking

Figure 6.2: Distance accumulation table

## 6.3 Longitude Data Example

2026-02-22 15:29:56 G...	2026-02-22 21:29:56 G...	2026-02-22 19:21:14 GM...	<b>7,22</b>	longitude	gps_tracking
2026-02-22 15:29:56 G...	2026-02-22 21:29:56 G...	2026-02-22 19:53:29 G...	<b>7,23</b>	longitude	gps_tracking
2026-02-22 15:29:56 G...	2026-02-22 21:29:56 G...	2026-02-22 19:56:33 G...	<b>7,23</b>	longitude	gps_tracking
2026-02-22 15:29:56 G...	2026-02-22 21:29:56 G...	2026-02-22 19:59:36 G...	<b>7,23</b>	longitude	gps_tracking
2026-02-22 15:29:56 G...	2026-02-22 21:29:56 G...	2026-02-22 20:05:44 G...	<b>7,23</b>	longitude	gps_tracking
2026-02-22 15:29:56 G...	2026-02-22 21:29:56 G...	2026-02-22 20:08:48 G...	<b>7,23</b>	longitude	gps_tracking
2026-02-22 15:29:56 G...	2026-02-22 21:29:56 G...	2026-02-22 20:11:52 G...	<b>7,23</b>	longitude	gps_tracking
2026-02-22 15:29:56 G...	2026-02-22 21:29:56 G...	2026-02-22 20:20:32 G...	<b>7,23</b>	longitude	gps_tracking
2026-02-22 15:29:56 G...	2026-02-22 21:29:56 G...	2026-02-22 20:23:36 G...	<b>7,23</b>	longitude	gps_tracking
2026-02-22 15:29:56 G...	2026-02-22 21:29:56 G...	2026-02-22 20:26:40 G...	<b>7,23</b>	longitude	gps_tracking
2026-02-22 15:29:56 G...	2026-02-22 21:29:56 G...	2026-02-22 20:29:44 G...	<b>7,23</b>	longitude	gps_tracking

Figure 6.3: Longitude field values in InfluxDB

# Chapter 7

## Visualization with Grafana

### 7.1 Trajectory Visualization

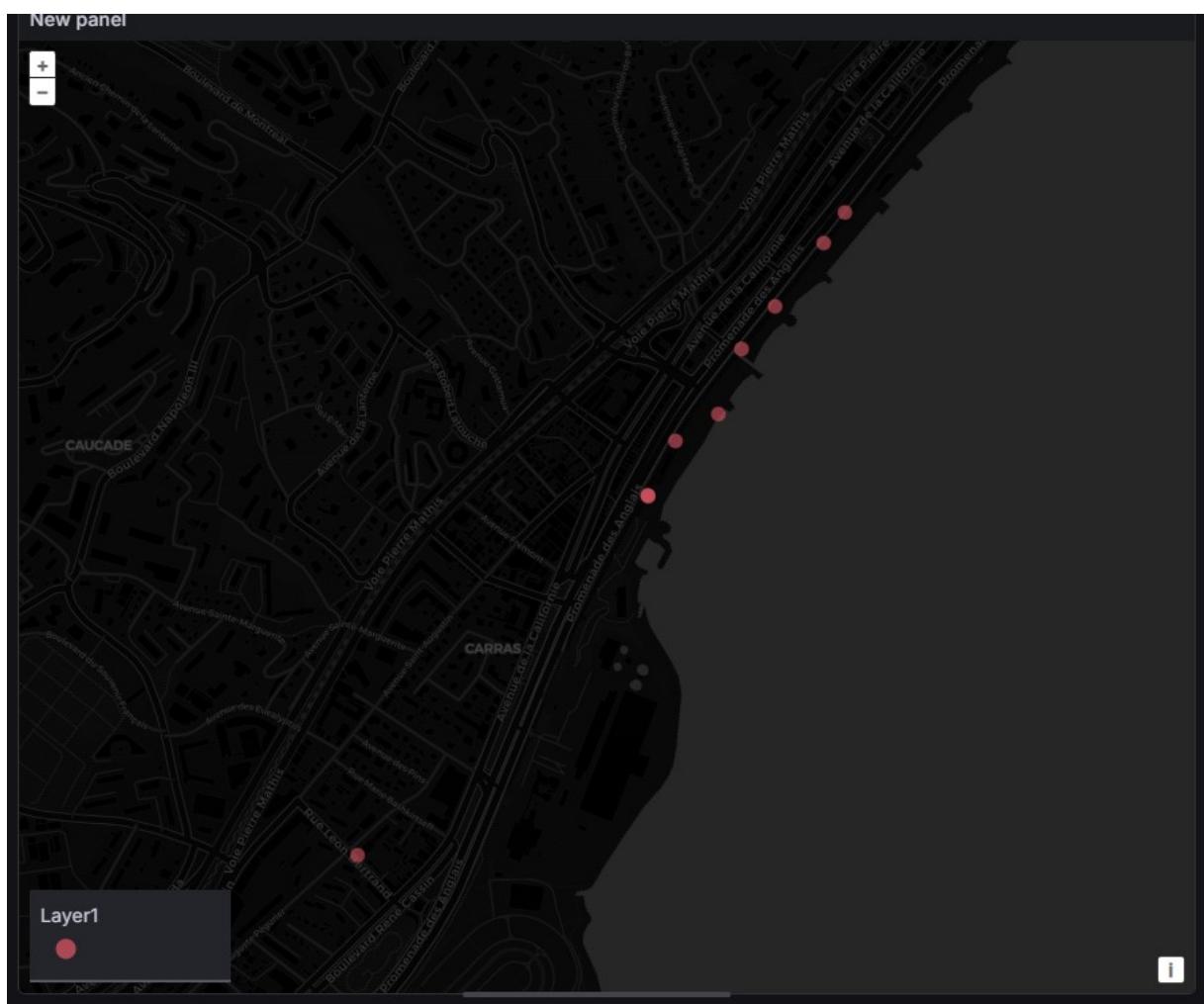


Figure 7.1: Grafana real-time trajectory

## 7.2 Detailed Map View



Figure 7.2: Detailed trajectory path

## 7.3 Data Inspection

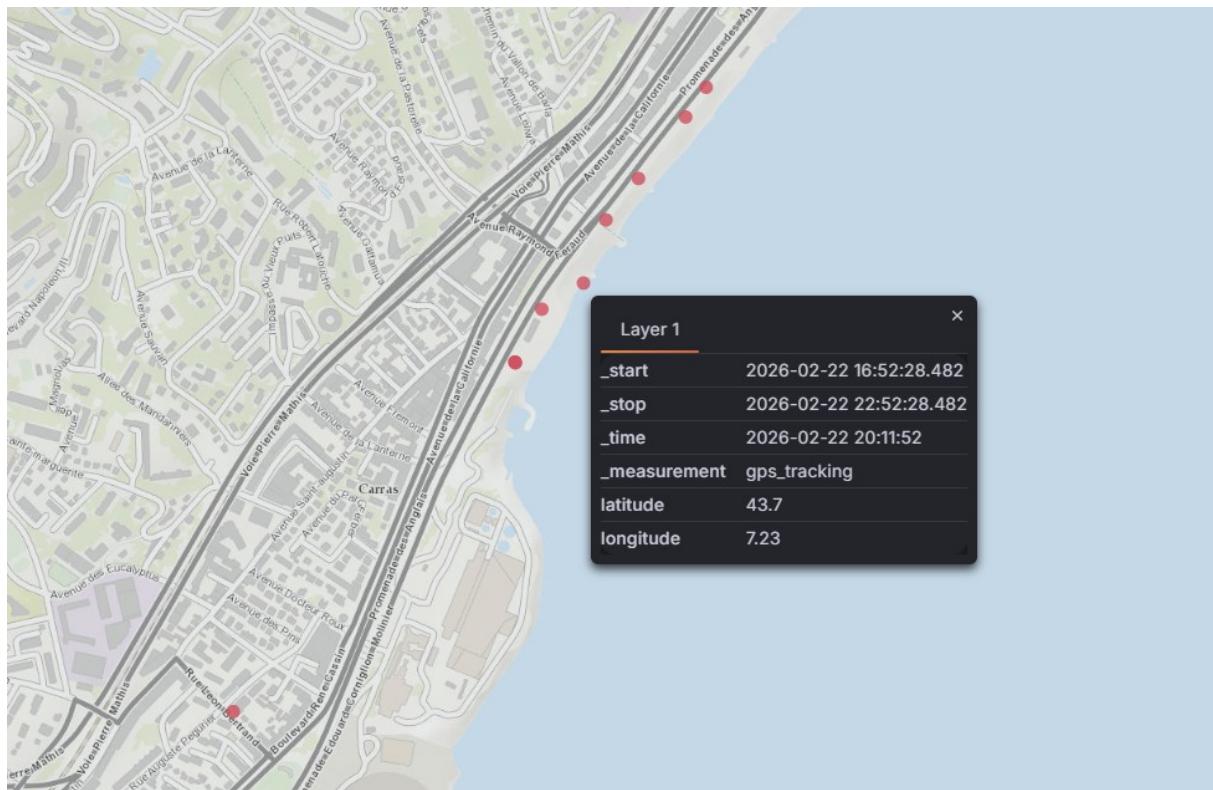


Figure 7.3: Detailed point information

# Chapter 8

## Node-RED Dashboard Interface

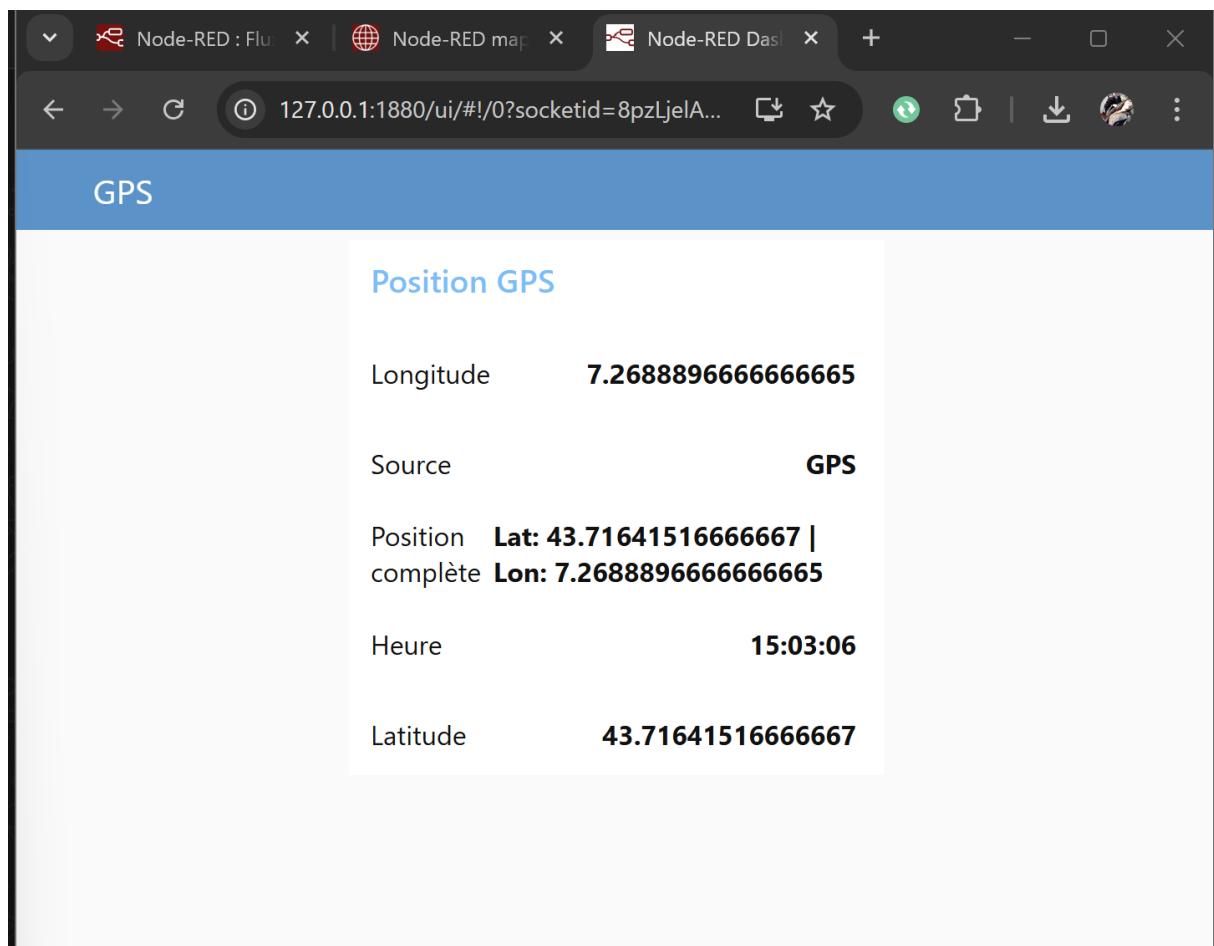


Figure 8.1: Live GPS dashboard interface

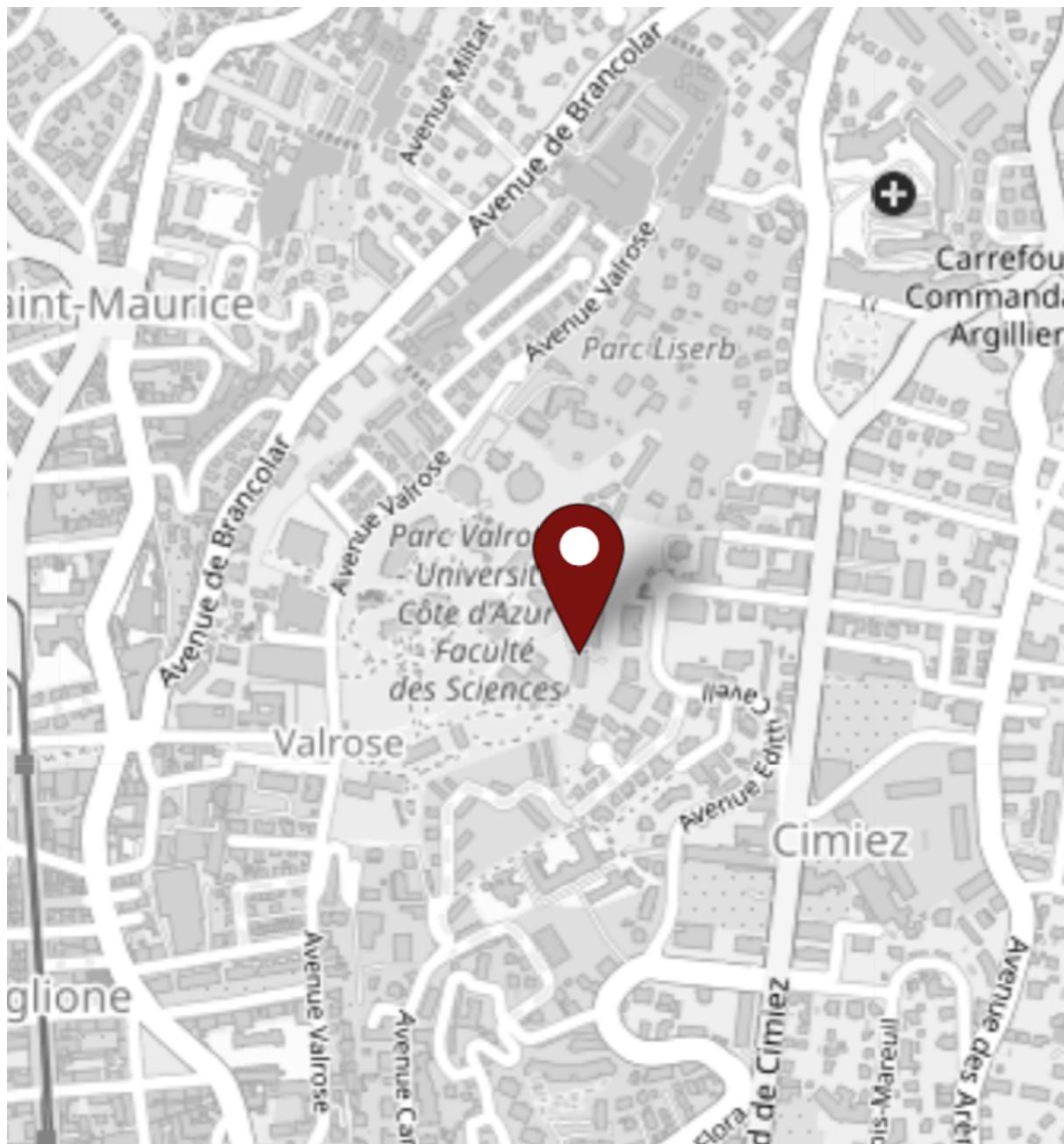


Figure 8.2: Position displayed on interactive map

The dashboard displays:

- Latitude
- Longitude
- Timestamp
- Source

# Chapter 9

## Energy Consumption Analysis

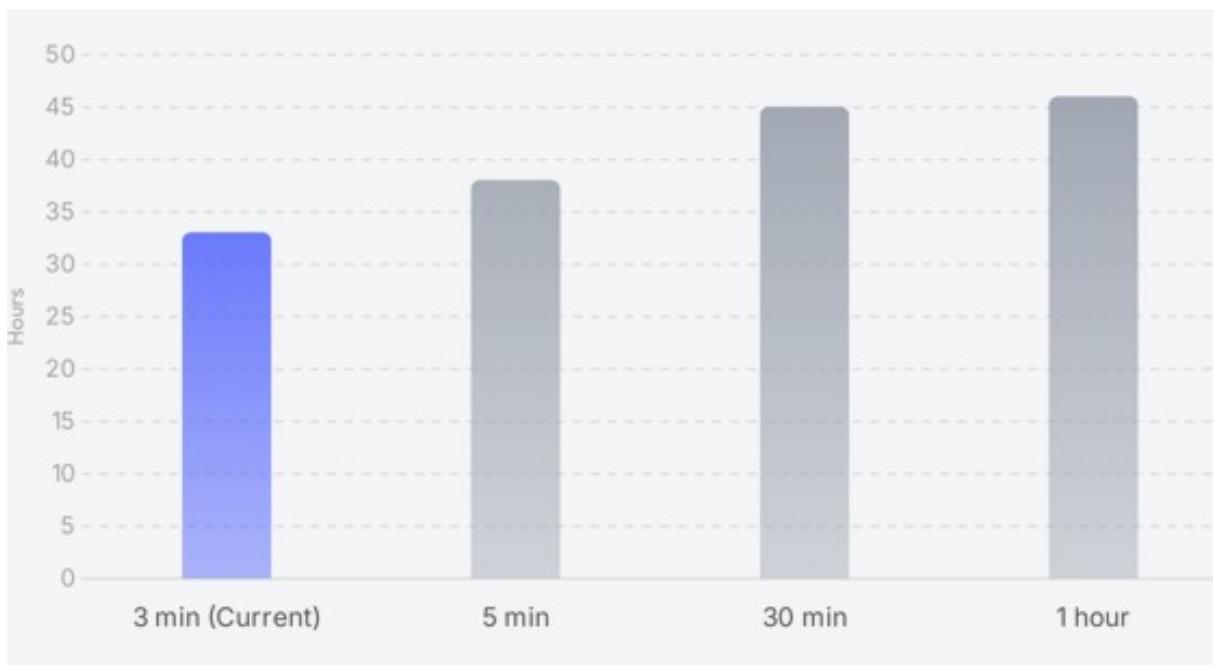


Figure 9.1: Battery autonomy depending on transmission interval

Battery characteristics:

- Capacity: 10,000 mAh
- Voltage: 3.7 V
- Total energy: 37 Wh

Measured autonomy:

- 3-minute interval: 33 hours
- 5-minute interval: 38 hours
- 30-minute interval: 45 hours
- 1-hour interval: 46 hours

Increasing the transmission interval significantly improves autonomy.

# Chapter 10

## Experimental Results

Field testing demonstrated:

- Reliable satellite transmission
- Accurate GPS precision
- Stable MQTT communication
- Correct database storage
- Real-time visualization

End-to-end chain validated:

*Satellite → MQTT → Node – RED → InfluxDB → Grafana*

# Chapter 11

## Limitations and Improvements

### 11.1 Current Limitations

- Satellite latency
- Energy consumption during frequent transmissions
- No emergency auto-trigger detection

### 11.2 Future Improvements

- SOS emergency button
- Accelerometer-based fall detection
- Adaptive transmission rate
- Compact PCB design
- Waterproof casing

# Chapter 12

## Conclusion

This project demonstrates a complete autonomous satellite GPS tracking solution combining:

- Embedded systems
- Efficient binary encoding
- IoT backend processing
- Time-series database management
- Real-time visualization

The system is suitable for rescue operations, remote monitoring, and extreme environment tracking applications.