05 - RouteTrack Pi — Route Intelligence & Metrics Engine

Date: December 24th, 2025
Category: Raspberry Pi / GPS / Data Processing

Backlink: RouteTrack Pi — GPS Data Logging Service

Project Goal

This phase introduces the **route intelligence layer** for RouteTrack.

Raw GPS points alone are not useful for reporting or visualization. This page defines how RouteTrack transforms logged GPS data into meaningful metrics such as:

- Mileage
- Moving vs stopped time
- Stop events (time-on-site)
- Daily route summaries

These metrics will later power:

- The local web dashboard (Leaflet)
- Historical route review
- Daily summaries and exports

Data Inputs

This phase consumes GPS data already being logged into SQLite:

Table: `gps_points`

Key fields used:

- `ts` — GPS timestamp (UTC)
- `lat`, `lon` — geographic position
- `speed` — meters per second
- `mode` — GPS fix quality (0–3)

Only **`mode = 3`** records are considered trustworthy for route calculations.

Route Mileage Calculation

Method: Haversine Distance

Mileage is calculated using the **Haversine formula**, which computes the great-circle distance between two latitude/longitude points on Earth.

This approach is:

- Accurate for vehicle-scale distances
- Lightweight (no external geo libraries required)
- Suitable for Raspberry Pi hardware

Rules Applied

To avoid false mileage caused by GPS drift:

- Only include points where:
  - `mode = 3`
  - `speed >= movement_threshold`
- Distance is accumulated **only between consecutive valid points**

Movement Threshold

A minimum speed threshold is applied:

- **Default:** `0.5 m/s` (~1.1 mph)

This filters out:

- Stationary GPS jitter

- Minor antenna noise when parked

Stop Detection (Time-on-Site)

Stops are inferred from GPS behavior rather than ignition signals.

Stop Definition

A **stop event** occurs when:

- Speed remains below `movement_threshold`
- For longer than `stop_dwell_time`

Recommended defaults:

- `movement_threshold`: `0.5 m/s`
- `stop_dwell_time`: `120 seconds`

This prevents brief slowdowns (traffic, turns) from being classified as stops.

Stop Events Table (Planned)

Detected stops will be stored in a dedicated table.

Table: `stop_events` (planned)

Fields:

- `id`
- `start_ts`
- `end_ts`
- `duration_seconds`
- `lat`
- `lon`

Each stop represents a **single continuous stationary period**.

Daily Route Summaries

RouteTrack will generate daily summaries based on processed GPS data.

Metrics Tracked Per Day

- **Total mileage**
- **Total moving time**
- **Total stopped time**
- **Shift start time**
- **Shift end time**
- **Number of stops**

Daily Summary Table (Planned)

Table: `daily_summary`

Fields:

- `date`
- `start_ts`
- `end_ts`
- `total_distance_miles`
- `moving_time_seconds`
- `stopped_time_seconds`
- `stop_count`

Daily summaries allow:

- Fast dashboard loading
- Simple reporting
- Long-term trend analysis

Processing Strategy

Route intelligence will be computed using **post-processing scripts**, not in the logger itself.

Reasons:

- Keeps the logger lightweight and reliable
- Allows recalculation if thresholds change
- Makes testing and validation easier

Processing can be triggered:

- On demand
- On a schedule (cron)
- Before dashboard rendering

Relationship to Local Dashboard

The local dashboard will **not** calculate metrics in real time.

Instead, it will:

- Read pre-computed route data
- Display routes, stops, and summaries using Leaflet
- Query SQLite via a lightweight API

This keeps the UI responsive and the system scalable.

Current Status

At this stage:

- GPS data is logged continuously
- SQLite schema is stable
- Logger service is running reliably
- No route intelligence calculations are active yet

Next Steps

The next phase will implement:

1. **Route processing script**
   - Compute mileage

    - Detect stops
    - Populate summary tables
2. **Database schema extensions**
    - `stop_events`
    - `daily_summary`
3. **Local Web Dashboard**
    - Flask backend
    - Leaflet-based map
    - Live and historical views

Why This Page Matters

This page clearly separates:

- **Data collection**
- **Data interpretation**
- **Visualization**

It is just a thoughtful write up on my next page which will be integrating the data before bringing up the Flask + Leaflet dashboard so it will launch with meaningful data!

Revision #3
Created 25 December 2025 02:25:09 by Nate Nash
Updated 25 December 2025 18:15:28 by Nate Nash