Prajnaana Technologies

... embedded system solutions

SIMPLE GUIDE

AI-Assisted, MAP-Integrated CAN Data Simulator

Contents

AI Vehicle CAN Simulator

Who this is for: Anyone with computer knowledge but not necessarily car knowledge needed. **What it explains:** what this program is, what it does, and how it works (There is a separate, detailed *Technical Document* for engineers; this one is the plain-English version.)

Test Coverage: The system has been tested with **Classical CAN (CAN 2.0)** only. **CAN FD has not been tested or validated. Testing on the Steering Wheel Control (SWC) has also not been performed; therefore, SWC-related CAN functionality is not covered by the current validation.**

1. What is this program?

It is a **pretend car that drives on real roads, inside your computer.**

You tell it a starting place and a destination — like *"Bengaluru"* to *"New Delhi"*. The program finds the **real road route** between them (the same roads a map app would show) and then **drives an imaginary car** along that route, from start to finish, all by itself.

As the car drives, the screen shows everything a real car's dashboard and computers would show: **how fast it's going, engine heat, which gear it's in, how much it's turning the wheel, the weather outside, and more** — updating live, second by second.

Think of it as a **simulator, but for a car** — except instead of you driving, the computer drives, and you watch all the numbers.

2. What is it for?

It answers one question, completely:

> *"If a car drove from A to B on real roads, what would all its sensors, engine values, and internal messages look like — the whole way?"*

This is useful for people who **build and test car electronics**. Real cars have many small computers talking to each other. To test those, engineers need a stream of realistic car data. This program produces that data — a believable, moving car — without needing an actual car on an actual road.

3. How do you use it?

1. **Open the program.** A window appears.
2. **Type a start and end place.** As you type, it suggests real place names.
3. (**Optional) Pick how it decides speed** — with a smart AI, or a simple rule. You can also pick a car brand/model and a top-speed slider.
4. **Press START.**
5. **Watch.** A map appears with your route drawn on it and a little car icon that moves along the road. A second tab shows the full dashboard of numbers.
6. When the car reaches the destination, it **slows down, stops, and the trip ends** on its own.

That's it. You mostly watch.

4. What do you see on screen?

- **A map** with the route line, a start marker, an end marker, and a **moving car icon** showing where the car is right now.
- **An info bar** at the top: state (parked/city/highway), speed, time left, distance left, the next turn, and the current road.
- **A "Weather (live)" box** showing the weather where the car currently is.
- **A Telemetry tab** with the *complete* list of numbers — engine, climate, steering, navigation, weather, and the car's internal messages.
- **Buttons** to open a door, flash the lights, sound the horn, or stop/start the car by hand.

5. How does the car drive itself?

The car isn't controlled by one thing — it's a **team of helpers**, each doing one job. They work at the same time and share notes on a common "whiteboard."

Helper 1 — The Navigator (finds the road and tracks position)

First, it asks a free online map service for the **actual roads** between your two places, and saves that route. Then, as the trip runs, it figures out **where the car is** — not with real GPS, but by simple math: *speed* × *time* = *distance travelled*, then it finds that spot on the route. Like counting your steps to know how far you've walked along a marked trail.

Helper 2 — The Decider (chooses the target speed)

Every few seconds it decides **how fast the car should aim to go**, based on the road ahead. If you gave it a smart AI, the AI makes this choice like a thoughtful driver. If not, it simply uses a sensible speed for that road. Either way, it only picks a **target** — a goal to aim for, not the actual speed.

Helper 3 — The Driver (makes it feel real)

This is the busy one, updating about **100 times a second**. It presses the imaginary accelerator and brake to chase the target speed, and a small **trained "brain" (a neural network)** turns those pedal presses into believable engine behaviour — how quickly the car speeds up, engine heat, and so on. This is what makes the car feel like a real car instead of a number jumping around.

Helper 4 — The Announcer & Listener (the car's messaging)

These send all the car's values out onto the **car's internal network** (see section 8), and listen for any commands coming back in.

Helper 5 — The Weather Watcher

Checks the **real weather** where the car currently is and reports it (see section 7).

The big idea: these helpers never talk to each other directly. They all read and write one shared "whiteboard" of information. That keeps everything simple and in sync.

6. How fast does the car go, and why?

The car obeys the **slowest of several limits** at any moment — whichever is most careful wins. Here are the limits, in plain terms:

- **The road type.** A tiny lane is slow; a highway is fast. The map service already knows this and suggests a sensible speed for each road.

- **Curves.** The car **slows down for bends** — the sharper the turn, the slower it goes, so it doesn't slide. On a straight road there's no such limit.
- **The road's tilt.**
  - Camber — every road is tilted slightly sideways so **rain drains off**. It's always there (about 1.4°).
  - Banking — on a curve, the road is tilted **into the bend** (like a race track) so cars can corner faster. Straight roads have no banking.
- **Weather.** In rain or snow the road is slippery, so the car takes curves **slower** and eases off a bit overall (see section 7).
- **Legal speed signs.** Where the map knows the posted limit, the car obeys it.
- **Your slider.** The top speed you chose is the overall ceiling.
- **The destination.** As it nears the end, it **gently slows to a stop**.

For safety, the car actually drives at about **85%** of the fastest-safe curve speed — a comfortable margin, not right at the edge.

In one line: *straight open road → go near your chosen top speed; sharp bend, small road, rain, or near the destination → slow down.*

7. The weather (it's real, and it changes)

The program looks up the **actual current weather** for the exact area the car is driving through, using a free weather service. On a long trip across different regions, the weather **updates as the car travels** (dry inland, rainy near the coast, and so on). On a short trip it's just one reading.

What the weather changes:

- **Rain/snow → slower cornering** and a slightly more cautious speed overall (like a careful driver).
- **Hot or cold outside → the engine and cabin temperatures shift** to match.
- If the smart AI is driving, it's **told the weather** so it drives more carefully in bad conditions.

The weather box on screen shows the condition (Clear, Rain, Fog…), temperature, wind, and a "grip" number (1.00 = dry roads; lower = more slippery).

(Note: the air-conditioning, fan, and set-temperature are controlled by a separate car computer, not by the weather.)

8. The "CAN bus" — the car's nervous system

Real cars have a small internal network so all their computers can talk. It's called a **CAN bus**. Think of it as the car's **nervous system**: the speed sensor, engine, brakes, and dashboard all send little messages to each other over it, constantly.

This program **reproduces that network faithfully.** Every value the car produces — speed, engine heat, steering, and so on — is packed into these little messages and sent out at realistic timing, exactly as a real car would. If real car-network hardware is plugged into the computer, the messages go out on it; if not, the program uses a **pretend network** so everything still works.

This is the whole point of the tool: it gives car engineers a **realistic stream of car messages** to test their equipment against.

9. Stopping at the destination

As the car gets close to the end, it **eases off smoothly** — like a real driver lifting off the accelerator early — and comes to a **complete stop right at the destination**, showing a speed of exactly 0. Then the trip ends automatically with a "destination reached" message. It won't stop short of the destination, and it won't freeze while still rolling.

10. What if something is missing? (It keeps working.)

The program is built to **never crash just because something is unavailable:**

- **No internet for weather or road details?** It uses sensible defaults and keeps driving.
- **No smart-AI key?** It falls back to simple road-speed rules.
- **No real car-network hardware?** It uses the pretend network.
- **A required piece of the "brain" is blocked by the computer's security?** It switches to a simpler backup and still runs.
- **The map service is busy?** The free map services the program uses have daily limits. If one is temporarily too busy, the program says so and asks you to wait a minute and try again — rather than showing an error you can't do anything about.

So, a demo will keep going even if the Wi-Fi drops or a device is unplugged.

Licence

One thing it is **not** for: this is a desk simulator and test tool. It should never be connected to a vehicle that is moving, or used where a fault could cause injury or damage.

This is the simple, everyday-language guide. For the full engineering detail — formulas, code structure, and exact behaviour — see the separate Technical Document.