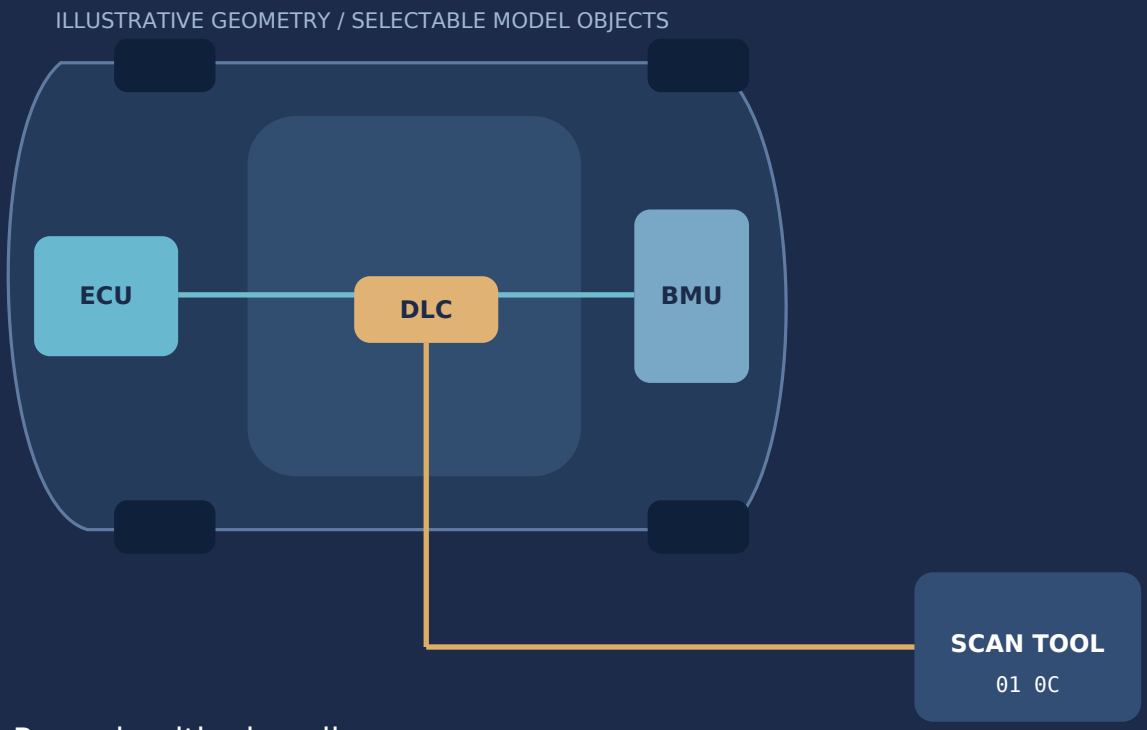
Inside the diagnostic loop.

A visual workbench for sockets, wires, signals and the questions a scan tool asks.



01 Run a healthy baseline

02 Disconnect a single path

03 Observe, repair and recheck

Reconstructed from the attached source. Generic teaching models.

Python 3.10+ • No additional runtime packages • 9 November 2026

01 / OPEN THE LAB

Start with one question

Use the visual views and instruments to examine the same deterministic experiment.

ONE-FILE LAUNCH

Save **run.sh** in any folder. In that folder, run `bash run.sh`. Open `http://127.0.0.1:8765` if needed. Keep the terminal open; **Ctrl+C** stops the lab.

Python 3.10 or newer is required. The lab runs offline without pip, Node.js or a CDN. The standalone launcher extracts its verified bundle into a temporary directory and normally removes it on shutdown.

View	Use it to...
Training panel	Select a component, socket, pin or wire; compare its live properties.
Transparent vehicle	Locate the same parts and see physical motion and logical activity.
Protocols & scan tool	Ask a named OBD/UDS question and inspect the reply.
Object layers	Follow stable identifiers from component to socket, wire and signal.
Learning book / Instruments	Read explanations, compare more channels and retain exports.

First pass: start a healthy coolant lesson. Read the sensor value. Disconnect the sensor, then advance time. Compare physical temperature, measured input, controller fallback and fault memory. Reconnect and run again.

Launch options: `--port 8766` selects another port; `--no-browser` leaves browser opening to you. The optional source ZIP has `run.sh`, `launcher.py` and `run.py` at the top level.

Timing rule: a connector action changes topology immediately. Sampling, fault detection and control change on subsequent simulation steps. Watch simulation time, not elapsed wall time.

02 / KEEP THE LAYERS DISTINCT

The socket, network and question

A shared connector can carry several kinds of communication. Name each layer before diagnosing it.

APPLICATION **OBD / UDS**	What does the message ask or report?
TRANSPORT **ISO-TP**	How does a longer message fit into CAN frames?
NETWORK **Classical CAN**	Which identified frames carry the bytes?
PHYSICAL ACCESS **DLC, pins and wires**	Which route connects the tester to the vehicle?

OBD centers on emissions-related reporting. **UDS** provides a broader diagnostic service vocabulary, independent of one data link. A generic engine OBD query does not read every brake or battery observation. [1,4]

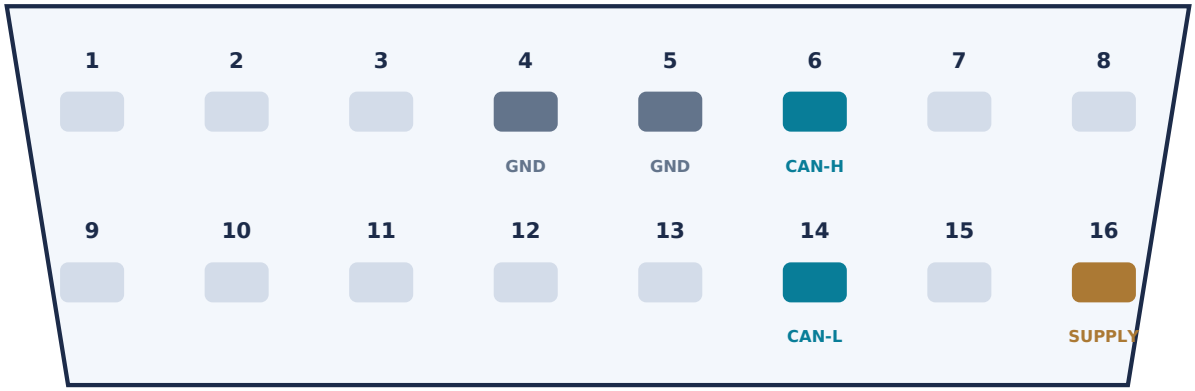
Family	Role in this recovered lab
OBD over CAN / ISO-TP	Runnable selected requests; 11-bit teaching IDs; synthetic frames.
UDS over CAN / ISO-TP	Runnable identification and session examples at 7E0.
ISO 9141-2 / KWP2000	K-line reference cards; no initialization or message engine.
J1850 PWM / VPW	Legacy network reference cards; neither is CAN.
DoIP / LIN	IP diagnostic route / local network reference cards.

The simulated trace does not model CAN arbitration, bit timing or analog waveforms. The primary references define the real protocol families; the lab declares a much smaller executable subset. [2,3,5-10]

03 / CONNECTIONS YOU CAN FOLLOW

Inspect a socket and every wire

Select an endpoint, then follow its wire to a measured quantity and its validity.



Vehicle-side DLC, looking into the contacts; wide edge at top. Highlighted cavities are the CAN lesson route. Legacy roles and populated pins vary with protocol and vehicle. [5]

Object	What to inspect
Component	Stable ID, physical purpose, sockets and position.
Socket / Pin	Owner, contact number, contact function and connection state.
Wire	Source and destination contacts, model properties, signal binding and unit.
Signal	Value, availability and any stale or invalid state. Zero is not the same as missing.

A WIRE CAN DISPLAY DIFFERENT QUANTITIES

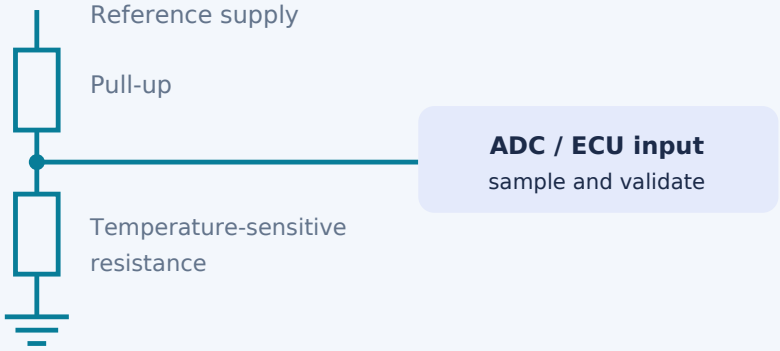
The coolant wire can show its modeled node voltage. A CAN wire can show a logical payload value or message age. Moving dots and scopes are **not measured CAN-H/CAN-L electrical waveforms**.

ECU, sensor and BMU socket assignments, dimensions and wire values are invented teaching properties. The diagnostic disconnect isolates the tester CAN branch; it leaves DLC supply contacts and the internal vehicle network in place.

04 / PHYSICS, MEASUREMENT AND CONTROL

Follow an open coolant circuit

A changed measurement does not instantly change the physical condition it was meant to describe.



Illustrative divider. Values come from the model parameters.

Physical truth	Sensor report	Control decision
Coolant continues evolving through the plant heat model.	An open input becomes invalid; this lesson reports -40 °C as its marker.	The ECU can substitute a separate fallback and record a detected fault.

- **Baseline:** connect the sensor, advance, then ask `01 05`.
- **Inject:** disconnect `ect` and advance in small steps. Read `07`, `03` and `01 01`.
- **Repair:** reconnect and advance. The signal can recover while P0118 remains stored.
- **Recheck:** use `04` after repair, run again, and read the warning lists.

P0118 is the model's circuit observation, not proof of a uniquely defective component. Detection timers, readiness and clearing semantics are simplified. Full drive cycles and permanent DTCs are outside the model.

05 / THREE MORE DIAGNOSTIC ROUTES

Separate missing information

Use the profile that contains the affected control loop, then compare the same evidence before and after.

A / NO SCAN-TOOL REPLY

Start with **01 0C**. Disconnect the **DLC CAN branch**, then repeat it. Observe no application reply while engine and vehicle control can continue. Reconnect and ask again.

The test isolates the diagnostic CAN branch. DLC supply and the internal network remain present.

B / A BATTERY MESSAGE GOES STALE

Use the **hybrid profile**. Disconnect **BMU CAN** and advance. Compare message age, supervisor mode and battery permission. Reconnect to restore communication.

Protective shutdown can remain latched. Cycle simulated ignition off/on, or reset, to clear that latch.

C / A WHEEL TURNS WITHOUT A READING

Use the **braking profile** with initial motion. Disconnect the **front-left wheel sensor**. Compare physical wheel speed with unavailable measured speed and controller degradation.

The wheel continues obeying the plant. Reconnection restores its measurement after the next sampling update.

One clean comparison: record profile, inputs, connection states and simulation time. The connector switches are independent. A separate active fault override can still hold a fault after a connector is repaired; inspect overrides or reset the experiment.

06 / ASK, DECODE, COMPARE

Read the bytes and the meaning

Named questions are the easiest entry point. Application bytes and transport frames are separate views of the exchange.

Request	Question / action	Address
01 05 / 01 0C / 01 0D	Coolant / engine rpm / vehicle speed	7DF
01 01 / 01 00	Warning lamp & monitor / supported PIDs	7DF
03 / 07 / 04	Stored / pending / clear teaching engine memory	7DF
09 02	Read invented VIN; see a longer reply	7DF
22 F1 90 / 22 F1 87	UDS synthetic VIN / lab part identifier	7E0
10 01 / 10 03 / 3E 00	UDS default / extended session / tester present	7E0

EXAMPLE / A 1000 RPM ANSWER

Application request: `01 0C`
Application response: `41 0C 0F A0`
Decode: `(256 x 15 + 160) / 4 = 1000 rpm`

Direction	CAN ID	Single-frame data
Tester to ECU	7DF	02 01 0C 00 00 00 00 00
ECU to tester	7E8	04 41 0C 0F A0 00 00 00

The first frame byte above is ISO-TP length information. It is **not** part of the OBD application request. Long replies use first (FF), flow-control (FC) and consecutive (CF) frames; short replies use a single frame (SF). [3,5]

Silence is different from rejection. No response can mean an isolated branch or an unsupported destination. A UDS negative response is an actual reply. Read the result explanation and frame trace together.

This build does not include full OBD/UDS coverage, programming, security access or actuator tests. The extended UDS session expires after 5 s of simulation time without a physical request. VIN and part identity are invented; one diagnostic ECU is modeled.

07 / A MODEL THAT CAN GROW

Connect objects to a 3D view

Keep physical behavior, observed signals, protocol messages and rendering separate.

Layer	Stable identity / responsibility
Component	coolant_sensor: purpose, sockets, position and model connection.
Socket and pin	coolant_socket: owner and numbered contacts.
Wire and signal	ect_signal: two endpoints, quantity, unit and validity.
Connection action	ect: the switch used by Sim.set_connection(id, connected).
Live observation	network_catalog() describes structure; observe_network(snapshot) describes live availability and values.

The view sends an action, then paints the returned model state. Recoloring a wire does not repair a connection. Physical wheel speed and measured wheel speed must stay separate; that distinction should survive any future renderer.

TRANSPARENT VEHICLE / INCLUDED SUPPORT

The offline **WebGL2 renderer** has a Canvas 2D fallback. Rotate, zoom, change body opacity and select parts. Wheel motion and logical signal activity follow simulation state. The included asset is `web/models/training_vehicle.glb` with a separate bindings JSON.

LOCAL MODEL IMPORT

The small importer supports **uncompressed embedded triangle geometry, node transforms and base colors**. It does not load textures, skins, animation clips, Draco data or external buffers. Use named `wheel_fl/fr/rl/rr` nodes to bind the four physical wheels.

Three.js readiness: `web/three-adapter.js` supplies hooks for an installed Three.js scene graph. Three.js and GLTFLoader are not bundled. Imports are session-local; retain the GLB and bindings JSON for the next launch. The adapter preserves node IDs and simulation bindings. See `docs/VEHICLE_MODELS.md`.

08 / WHAT THIS BUILD ESTABLISHES

Record evidence and extend carefully

A visible animation is useful when you can trace it back to an explicit model state.

Runs in the lab	Explained or outside this build
Generic engine, hybrid and braking models	Production vehicle equations or original ECU firmware
Four independent connector experiments	Complete harness, transceiver or analog network simulation
Selected OBD services and small UDS subset	All services, multiple diagnostic ECUs or conformance testing
Synthetic Classical CAN / ISO-TP transcript	CAN arbitration, bit timing, real waveform measurement
Offline vehicle rendering and limited local GLB import	Full Three.js asset pipeline or vehicle-specific geometry
Legacy protocol, DoIP and LIN reference cards	Running K-line, J1850, DoIP or LIN protocol engines

KEEP A REPEATABLE RECORD

Export signal **CSV**, full run **JSON** and the diagnostic **trace**. Keep profile, initial settings, input overrides, connection states and simulation time with the evidence. A screenshot alone cannot explain the state that generated it.

ADD THE BEHAVIOR BEFORE THE ANIMATION

- Define a stable component ID, sockets, pins and wire endpoints.
- Implement the physical or sensor behavior; expose a quantity, unit and validity.
- Verify healthy input, loss of input and repair through the model API.
- Bind the renderer to the returned state and preserve separate physical and measured channels.

The recovered source includes its validation record. Runtime checks, packaging checks and browser inspection are distinct from testing the user's particular Linux Mint installation. The preserved EngDoc file is a static circuit worksheet; full editor import and live synchronization remain unverified.

09 / GO DEEPER

References and useful words

Primary references checked 9 November 2026. They describe real standards; this lab implements its stated teaching subset.

[1] OBD application - ISO 15031-5:2015 - Emissions-related diagnostic services
https://www.iso.org/standard/66368.html

[2] CAN diagnostic path - ISO 15765-4:2021 - Requirements for emissions-related systems
https://www.iso.org/standard/78384.html

[3] ISO-TP transport - Linux kernel documentation - ISO 15765-2
https://docs.kernel.org/networking/iso15765-2.html

[4] UDS application - ISO 14229-1:2026 - Application layer
https://www.iso.org/standard/87962.html

[5] Connector and legacy interfaces - Elm Electronics - ELM327 data sheet, v2.3; figures 8 and 9
https://elmelectronics.com/wp-content/uploads/2020/05/ELM327DSL.pdf

[6] Legacy ISO diagnostics - ISO 9141-2:1994 - Digital information interchange
https://www.iso.org/standard/16738.html

[7] K-line physical layer - ISO 14230-1:2012 - Diagnostic communication over K-Line
https://www.iso.org/standard/55591.html

[8] J1850 network - SAE J1850_202212 - Class B Data Communications Network Interface
https://saemobilus.sae.org/standards/j1850_202212-class-b-data-communications-network-interface

[9] Diagnostics over IP - ISO 13400-2:2025 - Transport protocol and network layer services
https://www.iso.org/standard/13400-2

[10] LIN physical layer - ISO 17987-4:2025 - Electrical physical layer specification
https://www.iso.org/standard/85128.html

SMALL GLOSSARY

Term	Meaning
ECU / BMU	Electronic control unit / battery monitor unit.
DLC / DTC	Diagnostic link connector / diagnostic trouble code.
PID / DID	OBD parameter identifier / UDS data identifier.
Invalid / stale / stored	Untrustworthy now / too old / retained as fault memory.