

AiCAN User Manual

CAN Analysis, Measurement, and Test Automation Guide

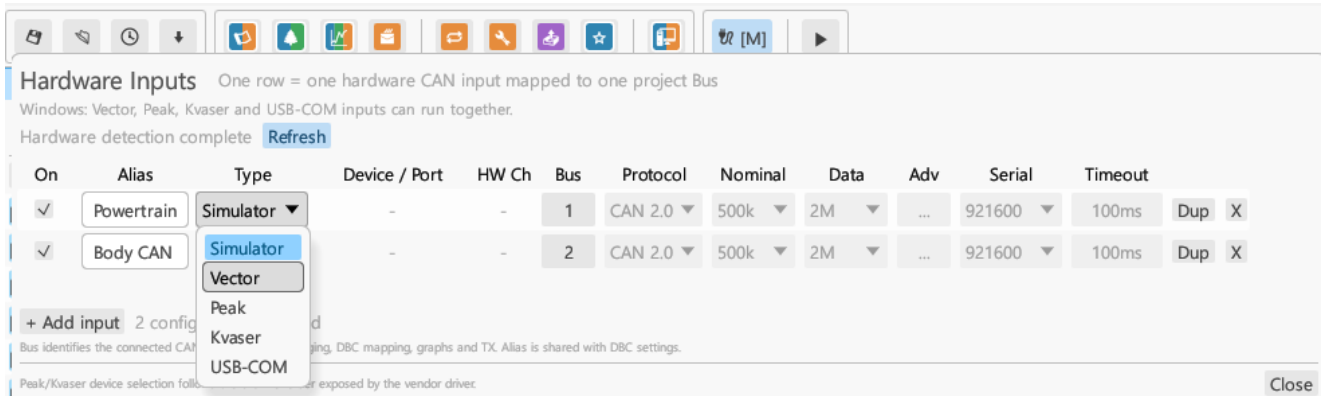
Table of Contents

Quick Start	2
1. Projects and Workspaces	3
1.1 Save Project	3
1.2 Load Project	3
1.3 Recent Projects	3
1.4 Workspaces	4
2. Logs and Signal Analysis	5
2.1 Export Log	5
2.2 Log Monitor	6
2.3 Signal Tree	6
2.4 Signal Graph	8
3. Transmission and Automation	10
3.1 Transmit	10
3.2 Log Stream	11
3.3 Script	11
3.4 App Builder	12
3.5 AI Assistant	13
4. System and Measurement	13
4.1 Console	13
4.2 Hardware Setup	14
4.3 Start Measurement	14
4.4 Notification	15
4.5 License	15
4.6 Information	15
Troubleshooting	16
A project or log does not open	16
Hardware measurement does not start	16
A feature button is locked	16

AiCAN combines CAN log analysis, DBC-based signal inspection, hardware measurement, and test automation in one workspace. This manual follows the application menu so that each feature is easy to find.

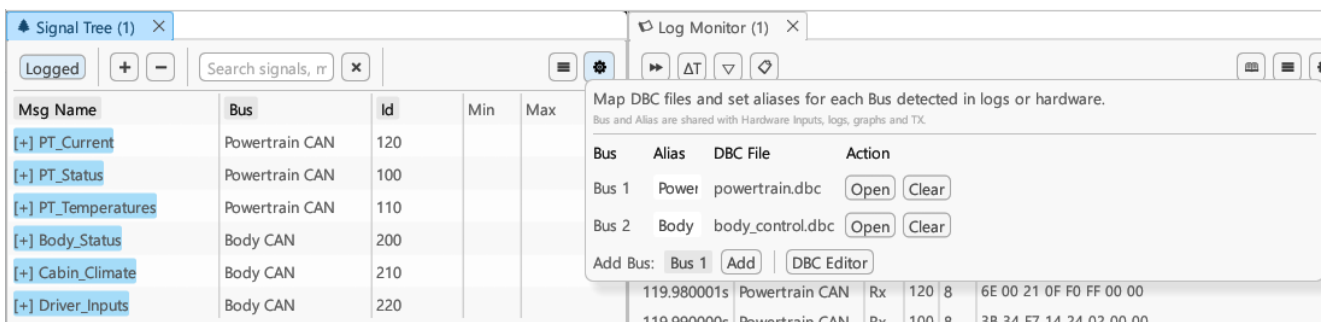
Quick Start

- 1 Configure a CAN input and Bus in Hardware Setup.



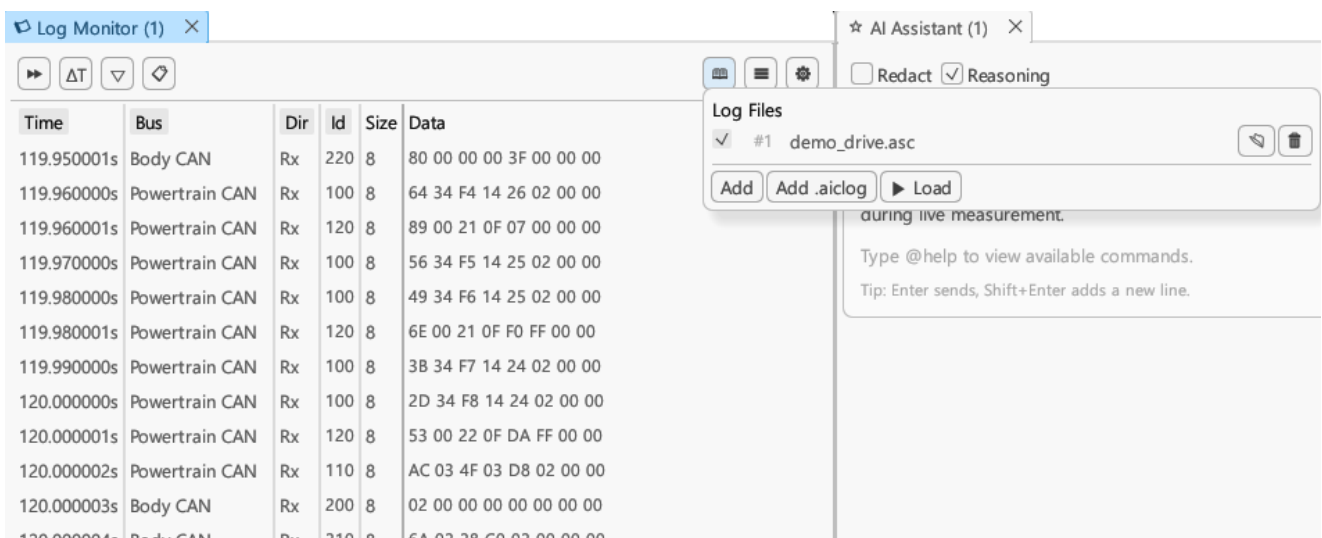
Configuring CAN inputs and Buses in Hardware Setup

- 1 If needed, assign a DBC to each Bus in Signal Tree.



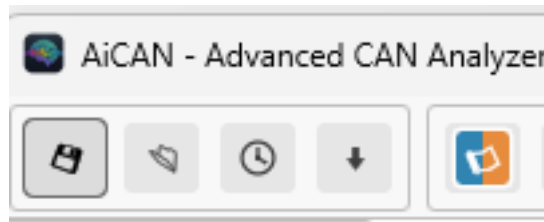
Assigning DBC files and aliases to each Bus in Signal Tree

- 1 Use Start Measurement for live capture or Log Monitor → Log Files for an existing log.



Adding and loading an existing log file in Log Monitor

- 1 Save the analysis layout and automation settings with Save Project.



Save Project button in the AiCAN toolbar

1. Projects and Workspaces

1.1 Save Project

Save the current work configuration as an AiCAN project (.aicfg).

- Saves workspaces, window layout, log and DBC paths, filters, and column settings.
- Also saves graph signals and colors, Transmit, Log Stream, Script, App Builder, and hardware settings.
- Original log and DBC files are not copied into the project. Move referenced files together when transferring a project.
- During measurement, AiCAN preserves the project's previous log references instead of saving the active live-log path.

1.2 Load Project

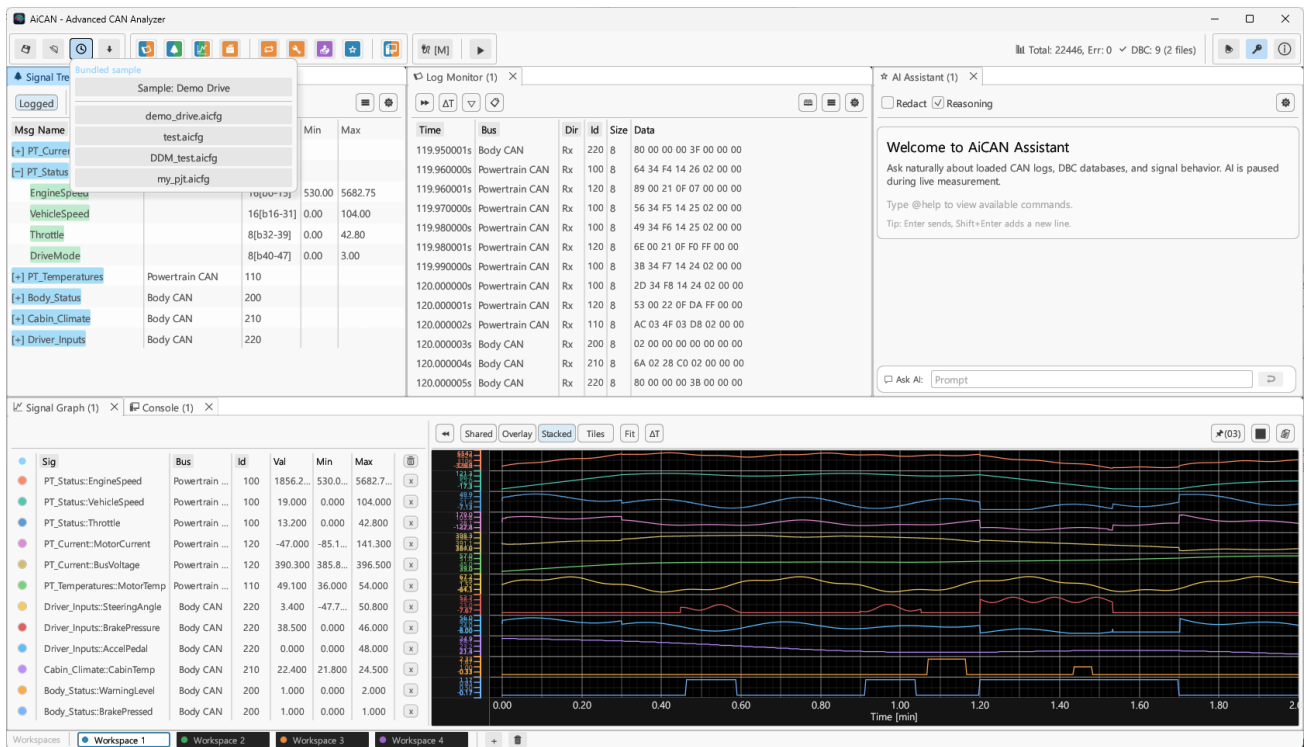
Open an .aicfg file and restore the working environment.

- Restores workspaces, layout, DBCs, graphs, filters, automation, and hardware settings.
- Relative paths are resolved from the project location, so keep the project folder structure intact.
- Projects cannot be opened during measurement. Run Stop Measurement first.
- Missing log or DBC references are reported in the Console.

1.3 Recent Projects

Quickly reopen a recently saved or loaded project.

- Shows up to 8 recent projects, newest first.
- Hover over a filename to see its full path.
- Files that no longer exist are removed from the list automatically.
- The bundled Sample: Demo Drive project is available from the same menu.
- Recent Projects is disabled during measurement.

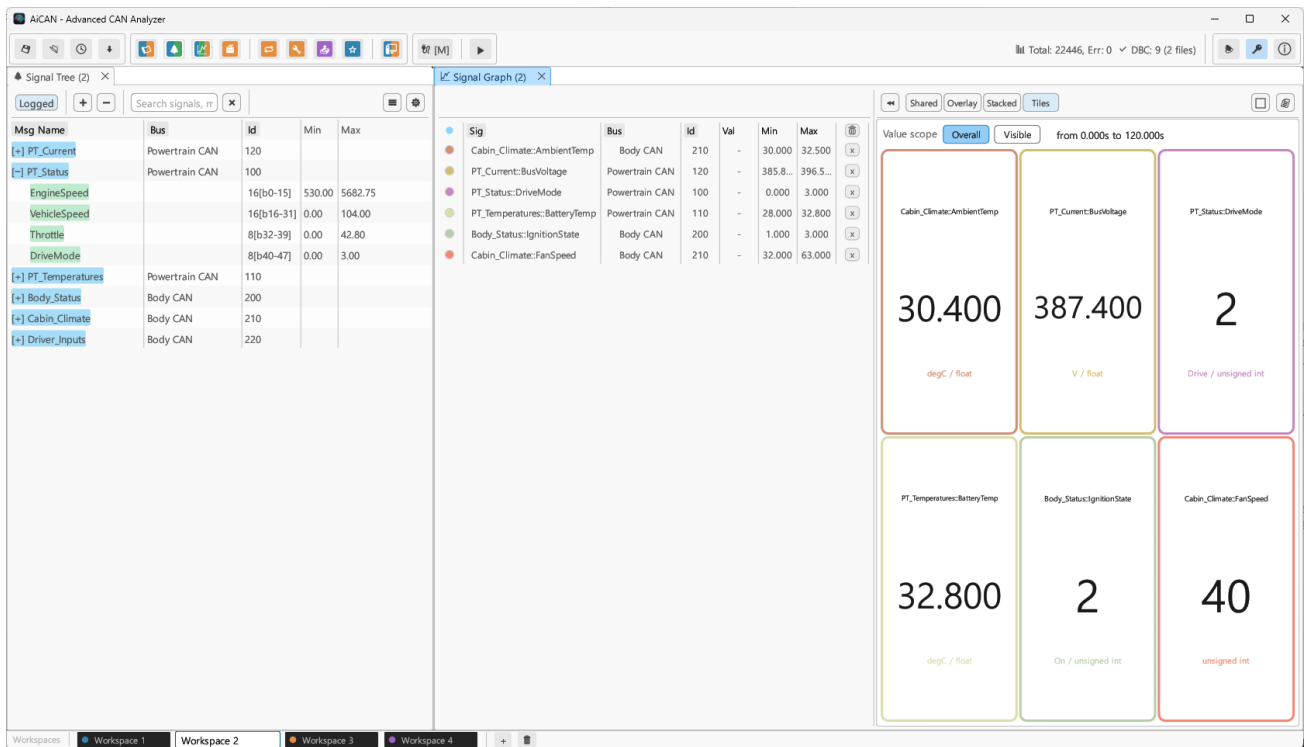


Demo Project and recent project list

1.4 Workspaces

Organize analysis windows into separate task-oriented work areas.

- Up to 8 workspaces can be created.
- Up to 8 instances of each tool window, such as Log Monitor, Signal Tree, and Signal Graph, can be opened.
- Click a Workspace tab in the bottom bar to switch work areas.
- Double-click a Workspace tab to rename it. Confirming an empty name restores its default name.
- Use + to add a workspace and the remove button to delete the active workspace. At least one is always kept.
- Drag a tool tab to dock it in the center, left, right, top, or bottom area.
- Toolbar button colors indicate the workspaces where that tool is currently open.
- Workspace names and layouts are included when the project is saved.



Renaming a Workspace by double-clicking its tab

2. Logs and Signal Analysis

2.1 Export Log

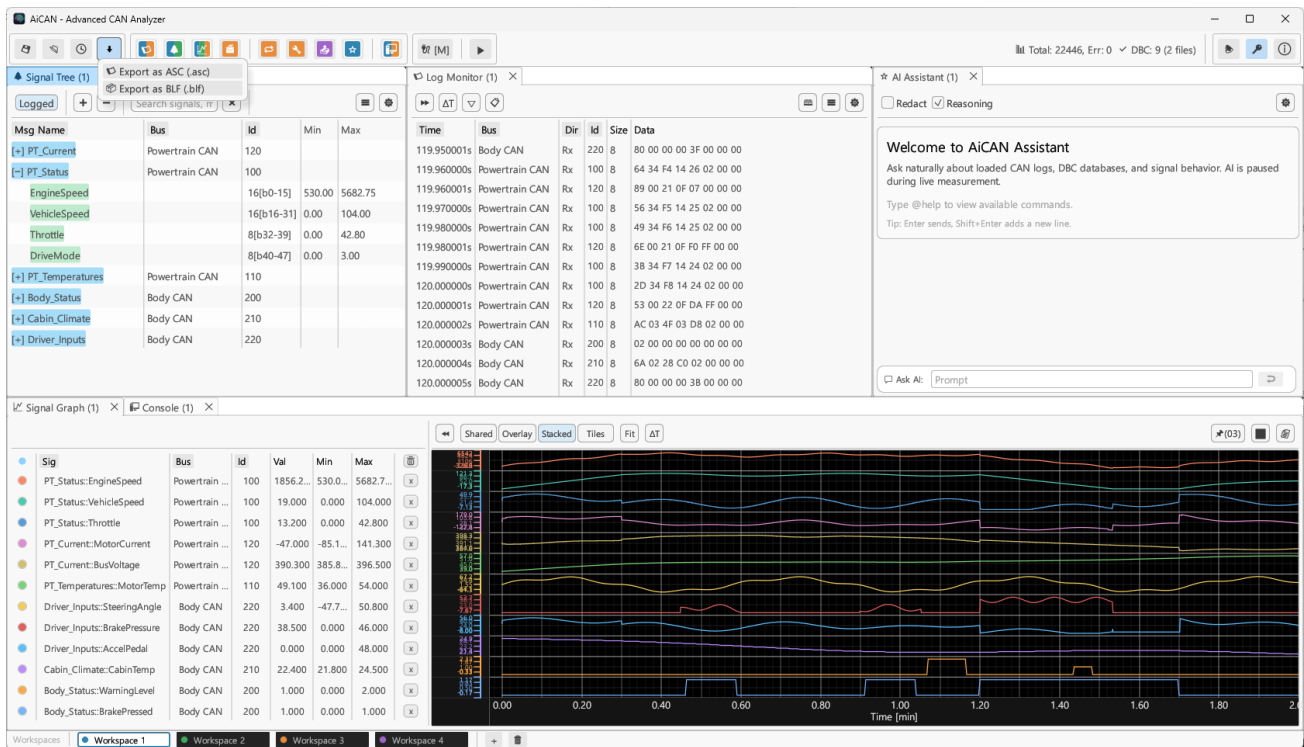
Save the current log as ASC (.asc), BLF (.blf), or a large AiCAN log.

- Select ASC or BLF from the Export button in the top toolbar.
- After measurement stops, the complete session can be exported.
- If the configured large-log threshold is exceeded, the session can be saved as a chunked .aiclog folder.
- Export is unavailable during measurement.

Keeping Selected .aiclog Ranges

The .aiclog/chunks folder contains .aicbin files split by time or size. Unneeded chunk files can be deleted, and the remaining files will still load normally.

For example, if a four-hour log is split into one-hour chunks and only the last hour is needed, delete the first three .aicbin files. Keep manifest.json and the chunks folder, and edit an exported .aiclog copy only after measurement and loading have stopped.



ASC and BLF log export menu

2.2 Log Monitor

Inspect live or loaded CAN frames in timestamp order.

- Default columns are Time, Bus, Dir, Id, Size, and Data.
- Add multiple ASC, BLF, or TRC files and load only the selected files.
- Add a chunked log folder with Add .aio log.
- Use Pass/Stop message filters or combine Bus, ID, direction, and data conditions in the advanced filter.
- ΔT shows the average cycle time for frames with the same Bus, direction, and ID.
- Copy the currently visible rows to the clipboard.
- Change column visibility, order, width, and sorting.
- Select a frame to inspect DBC-decoded message and signal values.
- Log Storage settings control the large-log threshold, chunk time and size, RAM range, temporary folder, and automatic recovery.

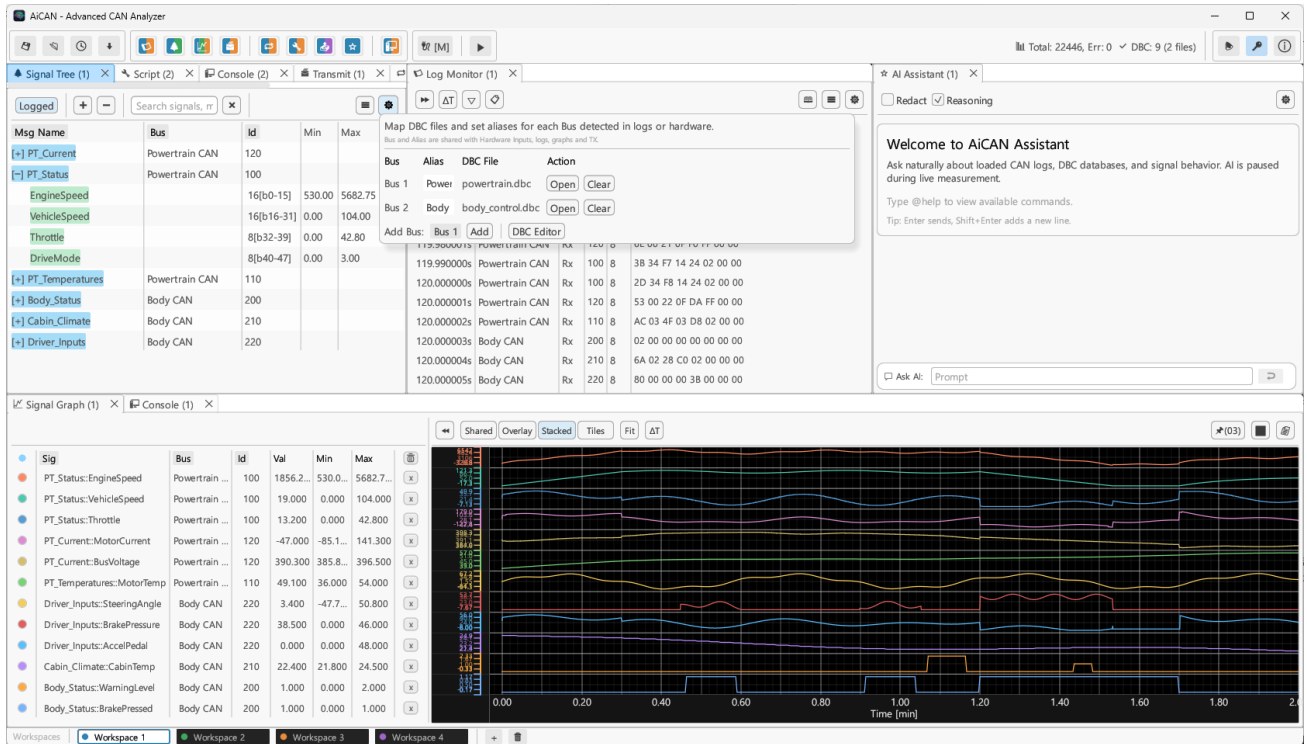
2.3 Signal Tree

Browse DBC messages and signals for each Bus together with current log data.

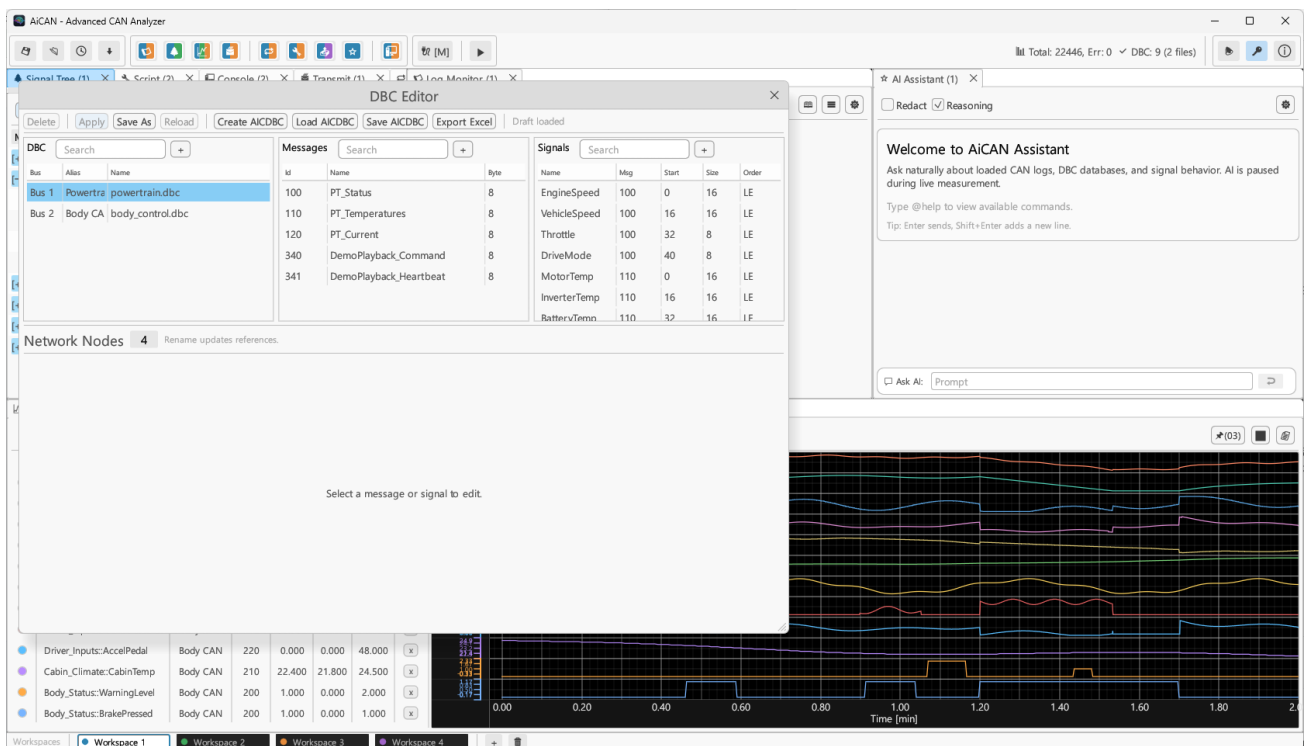
- Displays DBC content as Bus → Message → Signal.
- Search by message name, signal name, or ID.
- Enable Logged to focus on messages present in the current log.
- Inspect current signal values and observed Min/Max values.
- Add or remove signals from Signal Graph windows in the current workspace.
- Change column visibility, order, width, and sorting.

DBC Settings and Editor

- Assign or clear an Alias and .dbc file for each Bus.
- Create, search, edit, or delete DBCs, Messages, Signals, and Network Nodes in DBC Editor.
- Edit Message ID/DLC/Transmitter and Signal bit layout, byte order, factor, offset, unit, and value descriptions.
- Use AICDBC metadata to organize messages by Category and export Excel documentation.



Signal Tree settings for assigning DBC files and aliases to each Bus



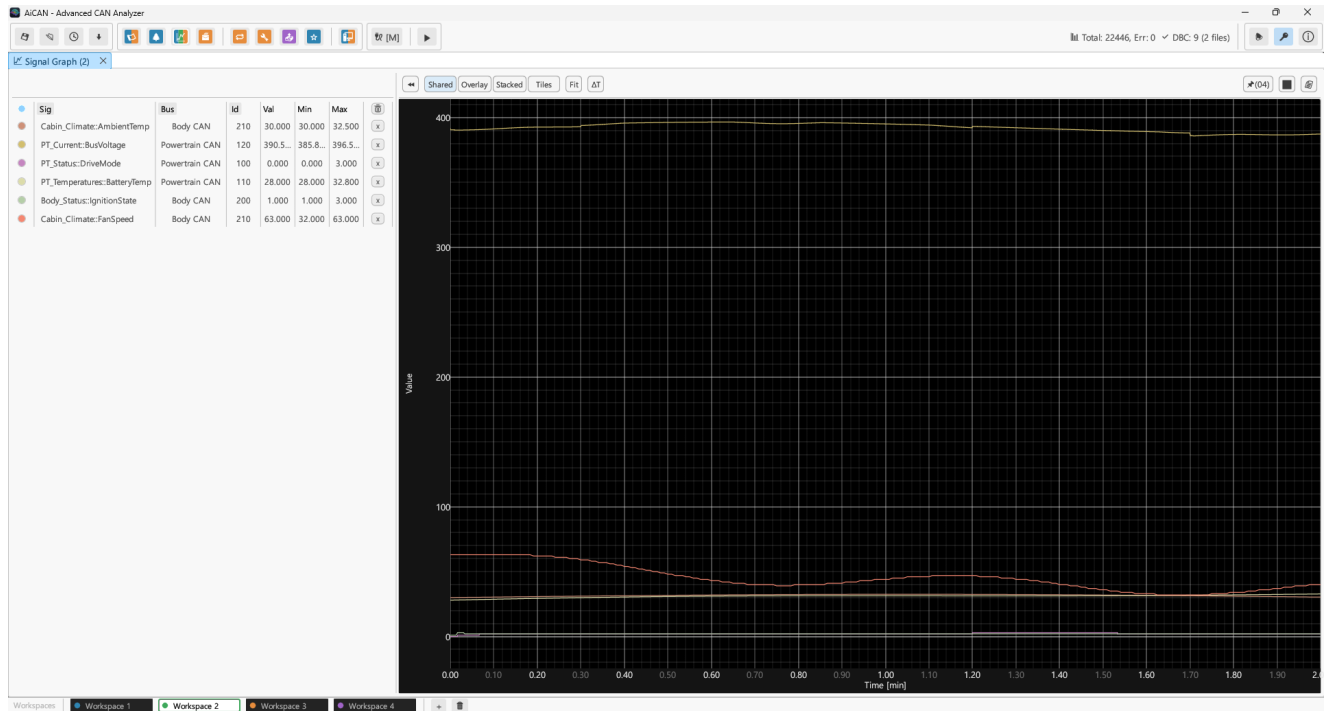
DBC Editor for searching and editing DBCs, Messages, and Signals

2.4 Signal Graph

Display selected physical signal values as time plots or value tiles.

Shared — Compare absolute values on one axis

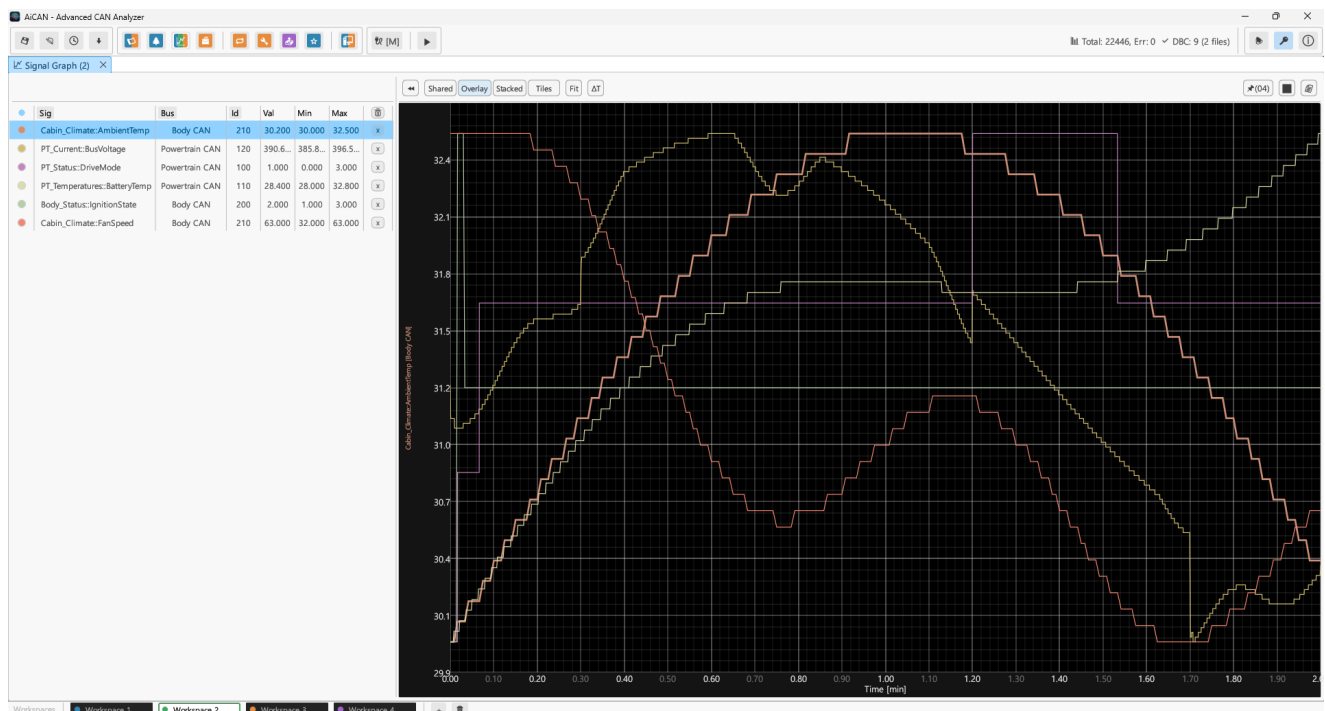
Displays every signal on one common Y axis. Use it to compare the actual magnitude and movement of signals with similar units and value ranges.



Signal Graph Shared mode displaying all signals on one common Y axis

Overlay — Compare change patterns on one plot

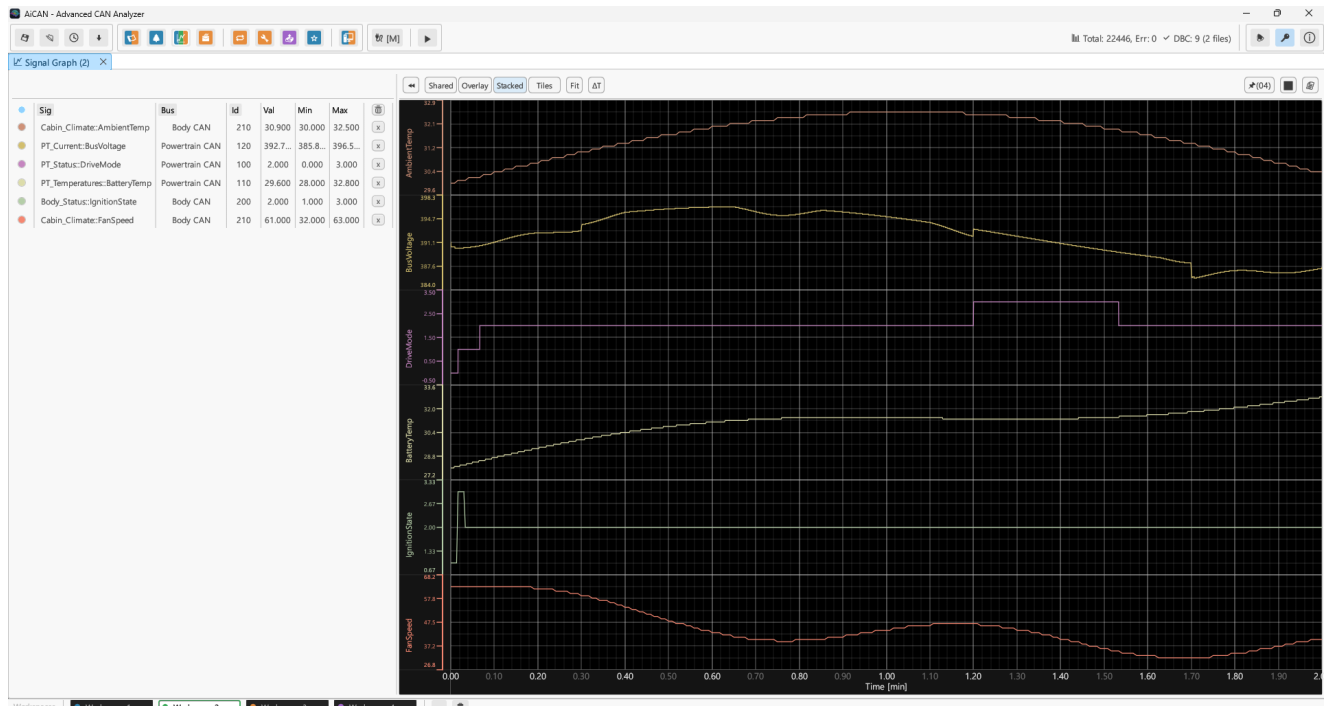
Normalizes each signal's value range and overlays the results on one plot. This makes rising and falling points or similar waveform shapes easy to compare even when units and magnitudes differ.



Signal Graph Overlay mode displaying signals with different ranges on one plot

Stacked — Separate signals into independent axes

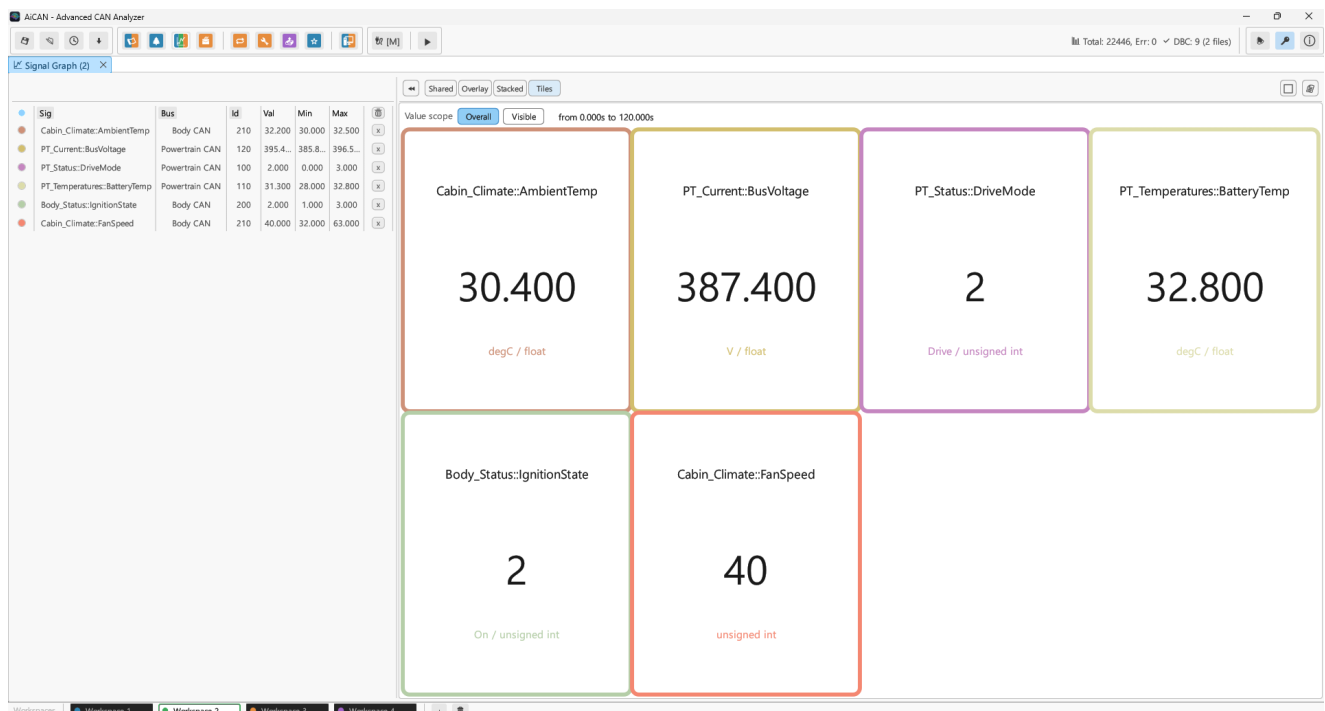
Places each signal in its own vertical lane with an independent Y axis. Use it to preserve waveform detail while avoiding overlapping lines.



Signal Graph Stacked mode separating signals into independent Y-axis lanes

Tiles — Monitor current values as large cards

Shows each signal's current value, unit, and data type in a large tile instead of a time plot. Use it when the current state matters more than the change over time.

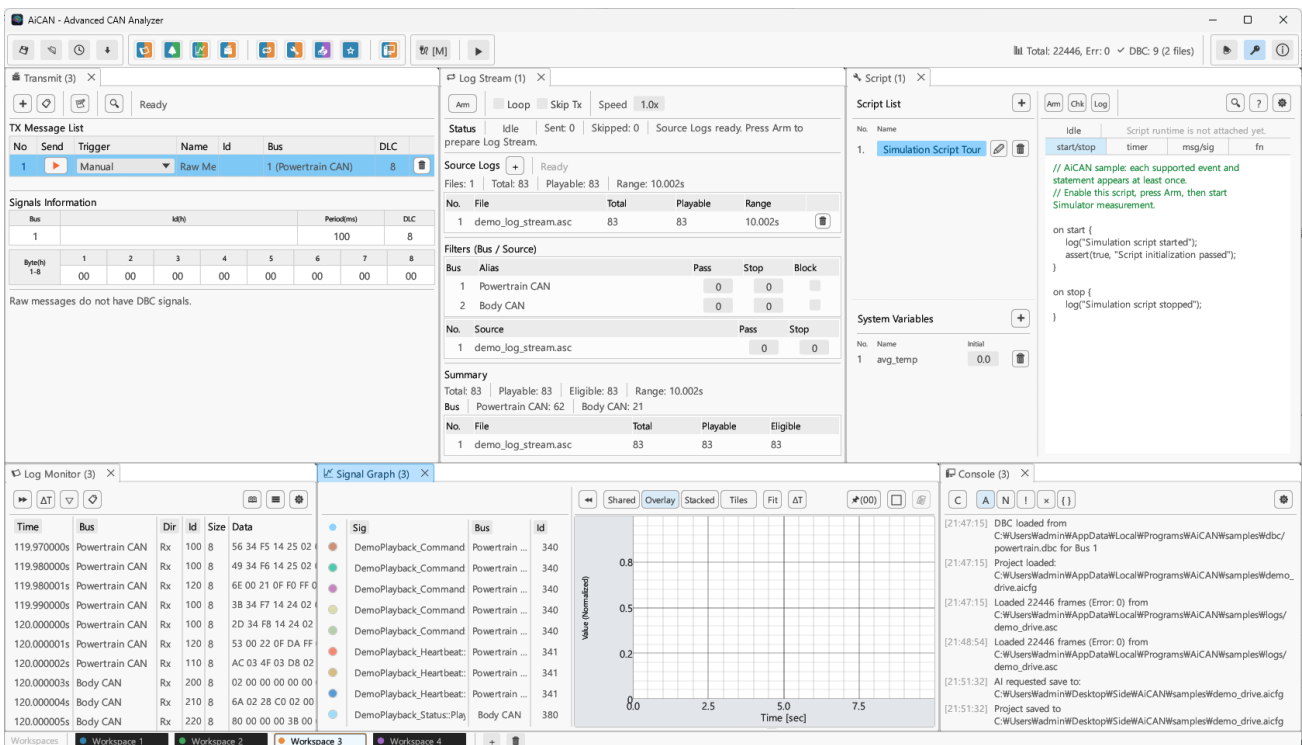


Signal Graph Tiles mode displaying current signal values as cards

Main tools:

- Use Fit to fit the X/Y data range to the view.
- Use T1/T2 cursors in ΔT to measure the time between two points.
- Double-click the plot to add a Bookmark and memo.
- Jump to or delete an item from the Bookmark list.
- Switch the plot background between Light and Dark.
- Export signal data from the visible time range as CSV.
- Pan and zoom with dragging and the mouse wheel over the plot axes.
- At a close zoom level, actual received samples are shown as dots.

3. Transmission and Automation



Automation workspace combining Transmit, Log Stream, and Script

3.1 Transmit

Send Raw or DBC messages to a CAN Bus manually or periodically.

- Create a new message or copy the selected message.
- Search and add a DBC message by name, ID, or Bus.
- Edit Bus, ID, Extended ID, DLC, Period, and Data bytes.
- Send a Manual message once with the Send button.
- Arm a Periodic message and start cyclic transmission with Start Measurement.
- Editing physical values in a DBC message automatically repacks the Data bytes.
- Add a note of up to 50 characters to each message and review notes in Notes Summary.

3.2 Log Stream

Replay ASC, BLF, TRC, or .aiclog data to a CAN Bus using the original timing.

- Add multiple Source Logs and inspect per-file information.
- Use Arm to prepare the sources; streaming starts with Start Measurement.
- Loop repeats from the beginning after the final frame.
- Skip Tx excludes frames marked as transmitted in the source.
- Speed scales source timing from 0.1x to 20.0x.
- Apply Pass/Stop filters by Bus, source file, and message ID.
- Review Total, Playable, Eligible, Sent, and Skipped counts.
- Filters must be re-armed after they are changed.

3.3 Script

Write test logic using a dedicated event-driven measurement script.

- Create, copy, rename, delete, and enable multiple Scripts.
- Source is organized into start/stop, timer, msg/sig, and fn tabs.
- Arm prepares a validated Script for the next measurement start.
- Chk checks syntax, events, functions, and references.
- Log shows Runtime events and the Action Queue.
- Find DBC messages and signals in DBC Library and insert code snippets.
- Review supported syntax and examples in API Reference.
- System Variables reset to their configured initial values when measurement starts.

Supported events: on start, on stop, on timer, on message, on signal

Main API: log, assert, mark_event, get_signal, set_signal, get_sysvar, set_sysvar, send_message, stop, abort

Script Syntax Quick Guide

Write the actions to run inside an event block. Check the source with Chk, select Arm, and then start measurement.

```
```text on signal PT_Status.EngineSpeed { let rpm = get_signal("PT_Status", "EngineSpeed");
if rpm >= 5000 { log("High RPM"); assert(false, "Engine speed limit exceeded"); } } ```
```

- Events: on start, on stop, on timer 100ms, on message 0x123, on signal Message.Signal
- Variables: Create a numeric local variable with let value = .... Function parameters and let variables are available only inside the current event or function.
- Arithmetic: +, -, \*, /, parentheses (), and unary +value or -value
- Comparison: ==, !=, >, <, >=, <=
- Logic: &&, ||, true, false
- Branches: if condition { ... } else { ... }
- Functions: Define one with fn name(a, b) { ... } and return a value with return expression.

- Main actions: Use `log()` to record information, `assert()` for Pass/Fail results, `mark_event()` for event markers, and `send_message()` to transmit a CAN message.
- Signals and shared values: Read or write DBC signals with `get_signal()/set_signal()`. Share a preconfigured System Variable with `get_sysvar()/set_sysvar()`.
- A trailing semicolon (;) is optional, and text after `//` is treated as a comment.

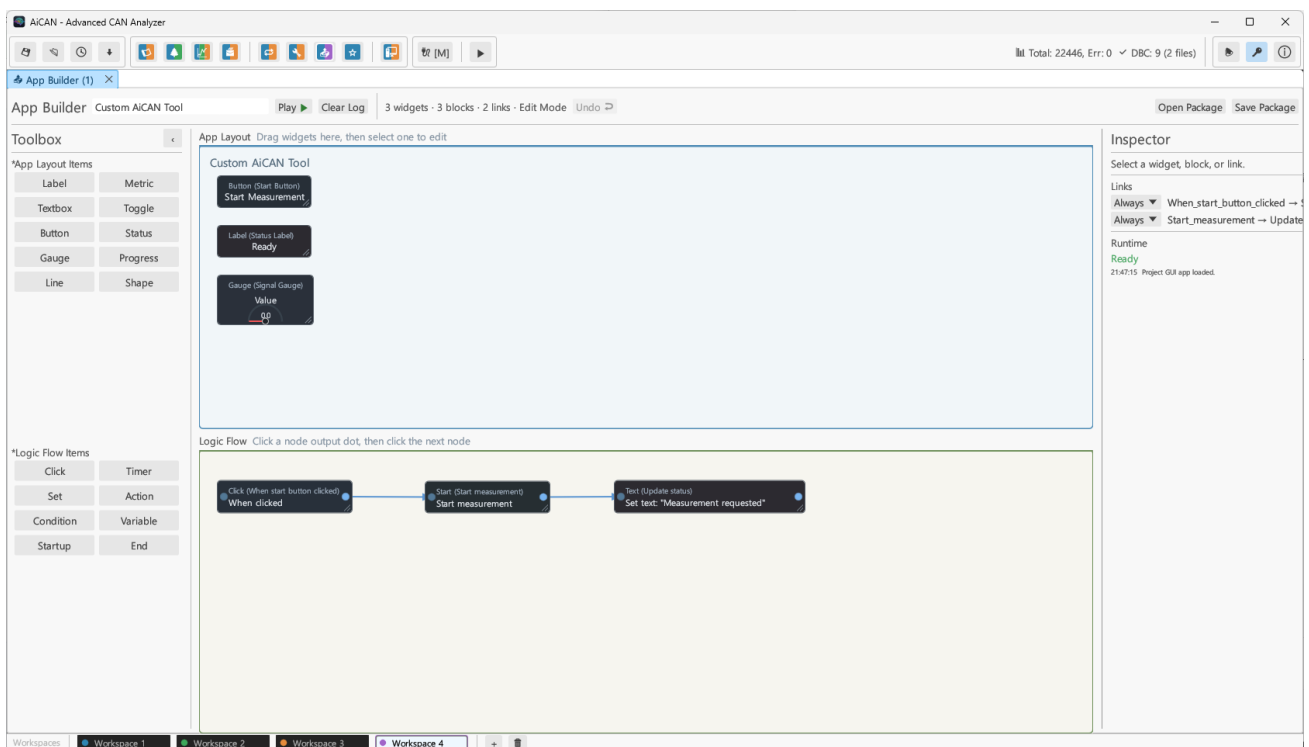
Important behavior:

- A failed `assert()` does not stop the Script. Use `abort("reason")` to fail and stop immediately.
- `return` ends the current event or function; `stop` stops and disarms the Script.
- Dividing a numeric expression by zero produces a Runtime Error.
- Nested event or function blocks and recursive function calls are not supported.

### 3.4 App Builder

Build a custom UI and Logic Flow without code, then save it as an .aicapp package.

- Layout widgets: Label, Metric, Textbox, Toggle, Button, Status, Gauge, Progress, Line, Shape
- Logic blocks: Click, Timer, Set, Action, Condition, Variable, Startup, End
- Click a Toolbox item or drag it onto the matching canvas.
- Configure names, size, style, DBC Signal binding, and behavior in Inspector.
- Connect blocks and select Always/Yes/No branches.
- Use Play to enter Preview Mode and Stop to return to Edit Mode.
- Use Undo or Ctrl+Z to revert the most recent edit.
- Use Open Package and Save Package to load or save an .aicapp file.

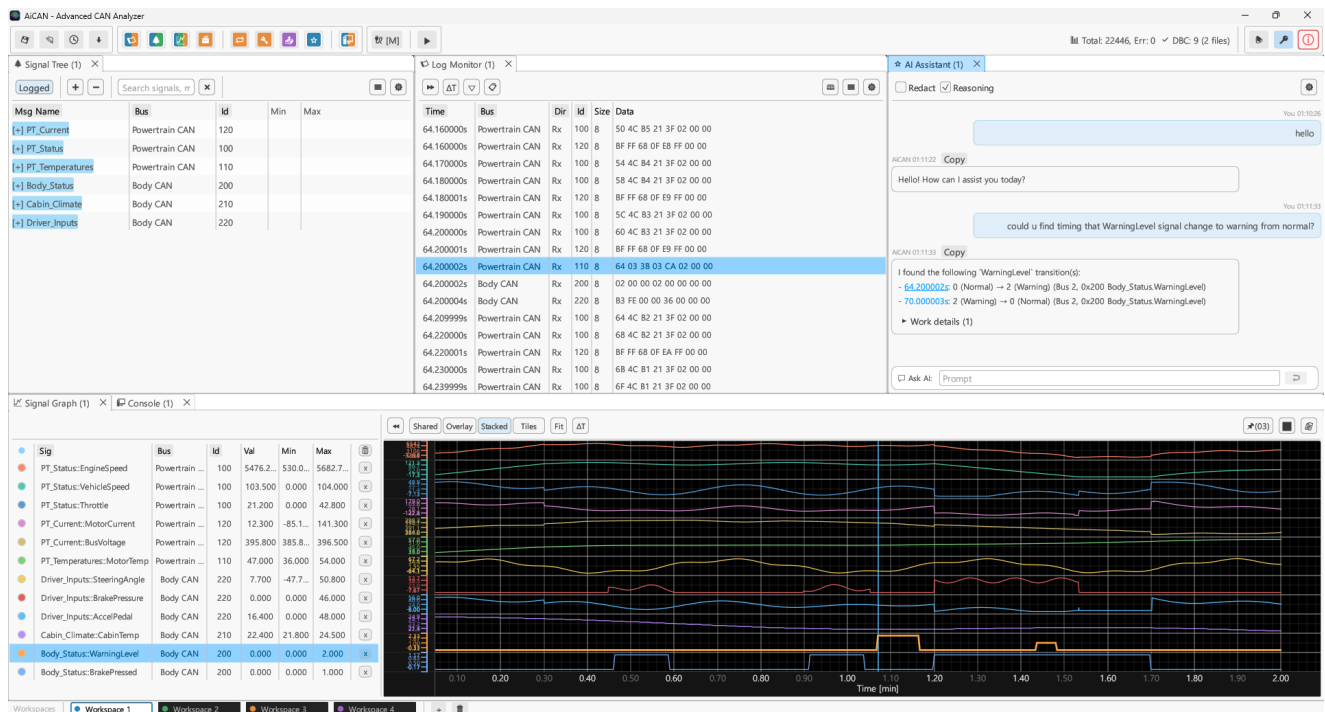


App Builder App Layout, Logic Flow, and Inspector

## 3.5 AI Assistant

Analyze the current log, DBC, signals, and project state using natural language.

- Providers: Local, ChatGPT, Claude, Gemini, Grok, and Custom.
- Configure Endpoint, Model, and API Key for an external provider.
- API Keys can be encrypted on the PC per provider and deleted when no longer needed.
- Download a supported Local AI model for use without an API Key.
- Redact reduces frame-payload exposure in external requests.
- Reasoning requests deeper reasoning from supported models.
- Max Context limits the amount of project context sent with a request.
- Use natural-language questions or AiCAN @ commands.
- AI requests are paused during measurement.



AI Assistant signal-transition analysis with linked log results

## 4. System and Measurement

### 4.1 Console

View project, file, hardware, and automation events in timestamp order.

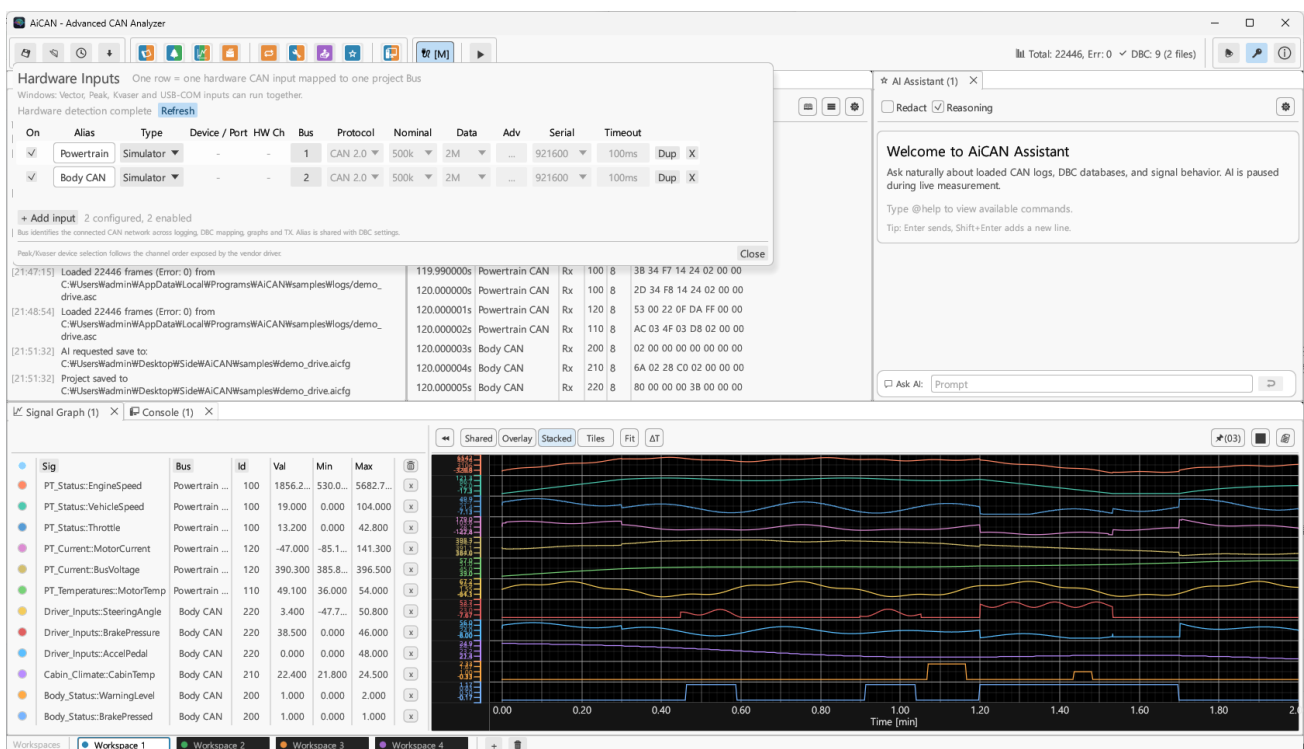
- Filter by All, Normal, Warning, Error, or Script.
- Colors distinguish information, warnings, errors, and Script messages.
- Use Clear Log to remove the current history.
- Set the maximum history from 50 to 50,000 in Console Settings.
- When a problem occurs, check Warning and Error messages first.

## 4.2 Hardware Setup

Map CAN hardware inputs to logical project Buses.

- Supported types: Simulator, Vector, Peak, Kvaser, SocketCAN, and USB-COM.
- Configure On, Alias, Device/Port, HW Channel, Bus, and Protocol for each input.
- Select CAN 2.0 or CAN-FD and set Nominal/Data bitrates.
- Configure Prescaler, TSEG1, TSEG2, and SJW in Advanced Bit Timing.
- Duplicate, add, or remove inputs and refresh driver detection.
- Aliases are shared with DBC, Log Monitor, Signal Graph, and Transmit.
- Hardware settings cannot be changed during measurement.

OS support: Vector is available on Windows, Peak/Kvaser on Windows and Linux, and SocketCAN on Linux. Simulator and USB-COM are common options.



Hardware Setup configured with two Simulator inputs

## 4.3 Start Measurement

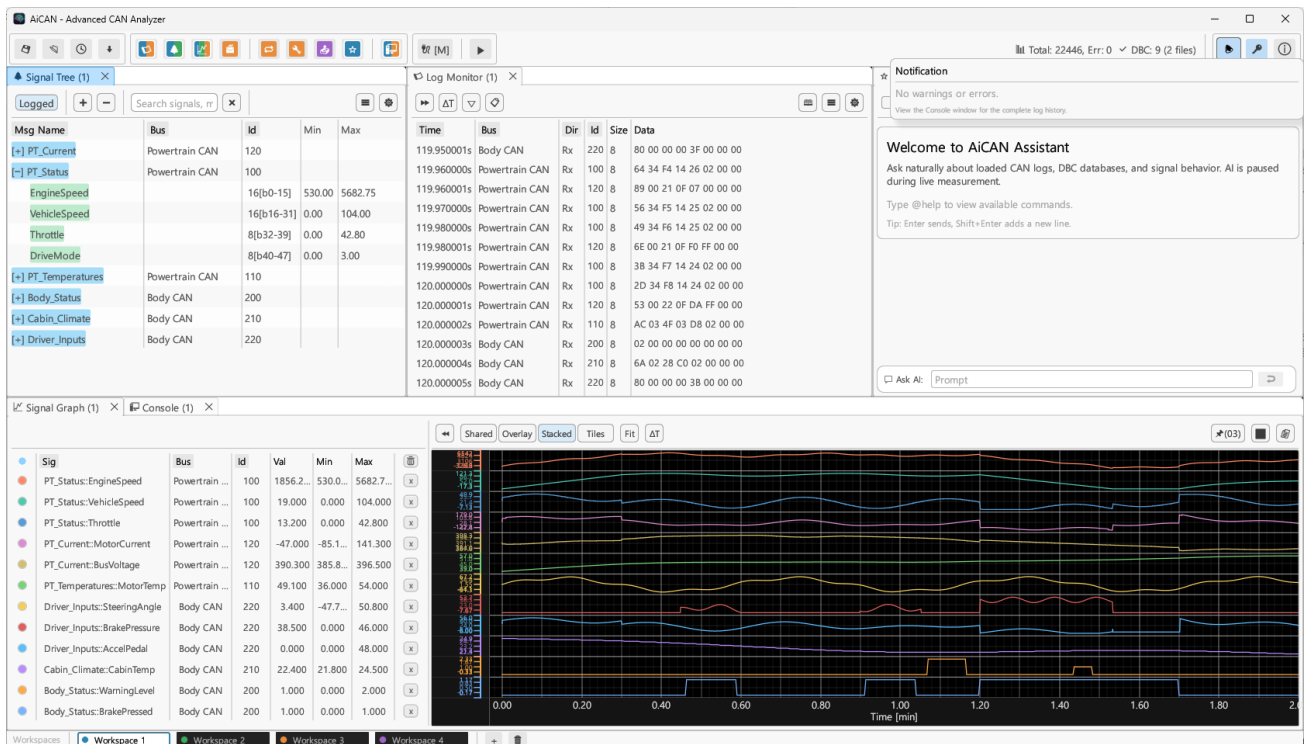
Open enabled Hardware Inputs and begin live CAN frame capture.

- AiCAN validates the configuration and license before mapping each input to its Bus.
- Live data is recorded to temporary binary chunks.
- Armed Periodic Transmit, Scripts, and Log Streams start together.
- [T], [S], and [P] on the start button indicate armed features.
- Enable Follow Latest to follow the newest frame during capture.
- Stop Measurement closes hardware and automation and finalizes the log index.
- Export any required previous session before starting a new measurement.

## 4.4 Notification

Quickly review recent Warnings and Errors from the bell icon.

- New Warnings are highlighted in orange and Errors in red.
- Up to 10 recent items are shown, newest first.
- Opening the popup marks the current items as seen.
- Use Console for the full history and Info/Script messages.
- Notification is an in-app view, not an OS or email notification.



Notification popup for recent warnings and errors

## 4.5 License

Review license status and activate a Trial or product key.

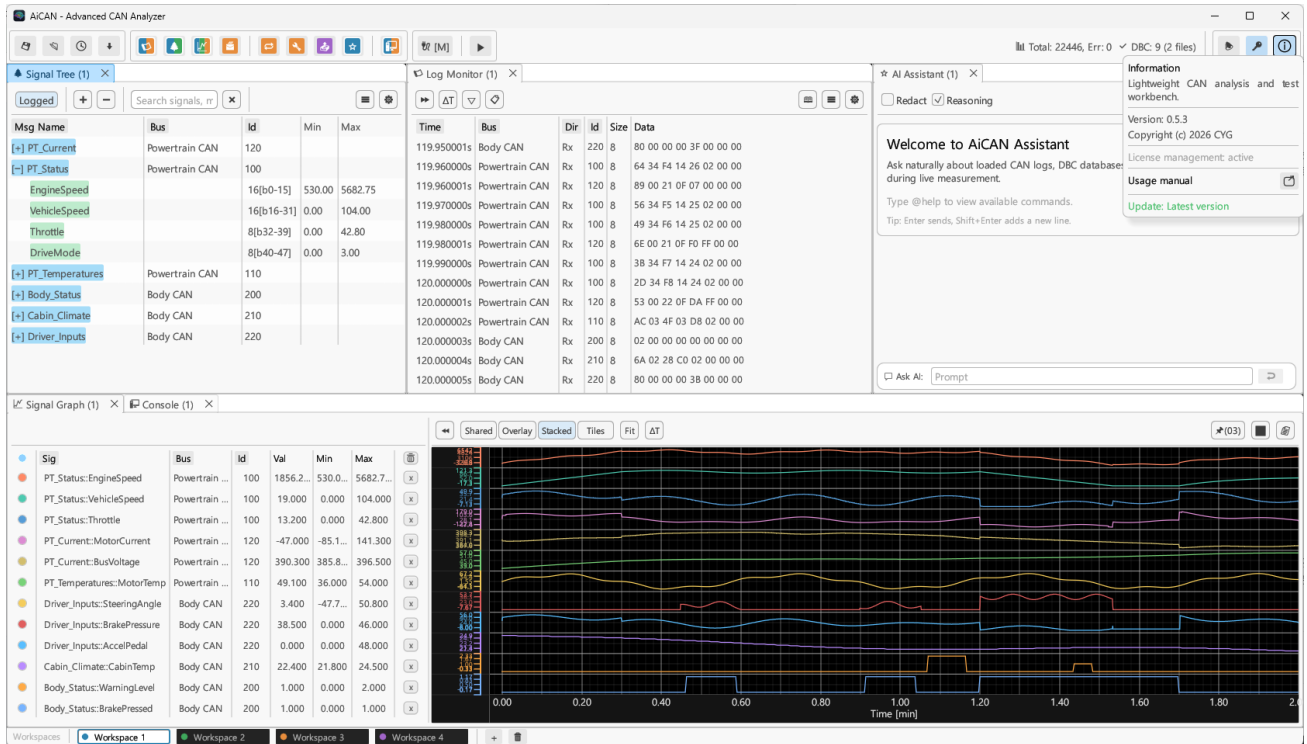
- Displays Status, Plan, Type, Licence ID, Device ID, and enabled Features.
- Review expiration, remaining time, Grace, and Offline deadlines.
- Use Refresh to update online validation.
- Enter a key under Register and select Activate.
- Features not included in the current license are locked with plan guidance.

## 4.6 Information

Review the app version, user manual, and update status.

- Displays the current AiCAN version and product information.
- Opens this web manual from the Usage Manual link.
- Checks for a new release and downloads the installer.
- After download, use Restart and Install to apply the update.

- Perpetual licenses show Renew guidance when maintenance renewal is required.
- Update failures are reported in Information and Console.



Information popup showing version, manual, and update status

## Troubleshooting

### A project or log does not open

- Stop Measurement before opening the file.
- Confirm that referenced files have not been moved or deleted.
- Review Warning and Error messages in Console.

### Hardware measurement does not start

- Confirm that the required input is On in Hardware Setup.
- Check Device/Port, HW Channel, Bus, and bitrate settings.
- Make sure physical channels and logical Buses are not duplicated.
- Refresh Vector, Peak, or Kvaser driver detection.

### A feature button is locked

- Open License and review the current Status, Plan, and Feature list.
- Connect to the internet and run Refresh.