Comprehensive Feature Parity Checklist for TunerStudio Ultra

Introduction

TunerStudio Ultra, the flagship edition of one of the most widely adopted engine management tuning suites, sets a high bar for features, integration, and user productivity. Beyond its ability to handle Megasquirt, rusEFI, and other aftermarket ECUs, TunerStudio Ultra distinguishes itself through an expansive toolkit for real-time tuning, logging, dashboard customization, extensibility, and deep configuration support. For a developer or open-source project aiming to achieve true feature parity, understanding these capabilities-down to fine-grained keyboard shortcuts, configuration model intricacies, and extensibility points-is critical.

This exhaustive report provides a detailed, structured, and paragraph-driven checklist covering every major UI element, core and advanced functionality, extensibility mechanisms, configuration nuances, and user experience shortcuts found in TunerStudio Ultra as of its latest releases through mid-2025. Each section not only lists functional modules and their subfeatures but also delves into how these contribute to TunerStudio Ultra's renowned workflow, flexibility, and data-driven tuning power, referencing authoritative documentation, user guides, forums, and tutorials to ensure no critical detail is overlooked.

1. User Interface Elements

1.1 Main Application Window & Project Model

TunerStudio Ultra organizes its workflow around a **project-based model**, where every ECU configuration is contained in its own project directory. Upon launch, users are greeted with a project selection interface featuring:

- Quick Access: Recent project list for rapid switching.
- Project Wizards: Step-by-step dialogs for new project setup, ECU detection, INI file association, and communication configuration.
- Contextual Help: Hoverable help bubbles for virtually all UI elements.
- Menubar: Providing access to File, Edit, Data Logging, Communications, Tools, View, Window, and Help menus.

This structured approach ensures users can segregate, back up, or transport entire tuning environments cleanly-a crucial feature for developers to accurately replicate for managing complex tuning portfolios[1].

1.2 Dashboard Tabs and Layout

- **Tabbed Dashboards**: Users can create unlimited dashboard tabs, each potentially housing custom gauge layouts and clusters. Dashboards are switchable via icons or keyboard shortcuts.
- **WYSIWYG Dashboard Designer**: Entered via right-click/context menu on dashboards. Offers drag-and-drop gauge placement, direct attribute editing, gauge resizing, and indicator/warning light placement[2].
- **Full-Screen Mode**: Maximizes the current dashboard for distraction-free monitoring. Triggered by double-click or shortcuts.
- **Gauge Styles and Skins**: Multiple analog, digital, modern, and themed gauge styles provided. Custom styling editable per-gauge.

Developers should support a similarly flexible, multi-tab dashboard paradigm with both WYSIWYG designer mode and theme-able skins[3].

1.3 Tuning Views, Dyno View Designer & Table Dialogs

- **Tuning/Dyno Views**: Ultra users can craft custom “views” combining settings panes, live tuning tables (2D/3D), crosshair cell targeting widgets, gauge clusters, and burn-to-flash buttons on a single tab. These views are particularly optimized for specific tuning sessions (e.g., dyno pulls, acceleration setups)[5].
- **2D and 3D Table Editors**: TunerStudio supports smooth switching between 2D/3D modes for all tables, with real-time crosshairs tracking active cells during engine operation.

Reproducing this UI structure-including extensible tuning views and context-aware editors-is essential for completeness.

1.4 Curve Graphs, Baselines, and Multi-Select

- **Curve Graphs**: Used for 1D tables (e.g., cold-start enrichment) and for overlaying actual vs. target curves.
- **Baselining**: Users can set baselines for comparisons.
- **Multi-Select Editing**: Holding CTRL allows selecting multiple points on curves/tables for batch operations-a highly productive feature for rapid tuning changes.

1.5 Notes Log, Difference Reports, and Restore Points

- **Notes Log**: Project-level notes field for logging changes, tuning reasoning, and session outcomes.
- **Graphical Difference Reports**: Visual, side-by-side diff between current, saved, and historical tune files (msq).
- **Automatic Restore Points**: Auto-saved configuration snapshots at key events (on project close, manual save, etc.), with the ability to revert all or parts of the configuration.

Such historic tracking and visual diffing are important for safe, iterative tuning development.

2. Keyboard Shortcuts and Keybinds

2.1 Default Keyboard Shortcuts

Key/Combo	Function
Tab	Cycle selection among stacked or overlapping gauges/dash elements
CTRL + Click	Multi-select gauges/graph points/dash elements
F1	Open context-sensitive help for selected dialog
F2	Rename table or dashboard
CTRL + S	Save project/session
F5	Start data logging
F6	Stop data logging
F7	Mark data log
CTRL + Z	Undo (in Dash Designer/Table editor, where supported)
Arrow keys	Move cell selection in tables/graph points
Number keys	Directly type in values for selected cells
Enter	Confirm/commit change in cell or dialog
ESC	Cancel current edit/dialog
Shift + Click	Range select multiple cells/points
ALT + Arrow	Adjust gauge size/position (in Designer mode)
CTRL + D	Duplicate selected gauge or dashboard tab
CTRL + T	Open Table Tune dialog
F10	Open Tuning & Dyno View Editor (Ultra only)

Note: Some shortcuts, particularly those involving direct cell value entry, multi-selection, and table movement, are not customizable-the software relies on a consistent, productivity-focused keymap. Efficient support for these same keybinds in an alternative implementation is necessary for feature parity.

2.2 Shortcuts for Quick Navigation and Designer Mode

Users can quickly switch dashboards, access favorite dialogs, and jump between stacked UI elements using the keyboard-a substantial productivity accelerator for advanced users and critical for data-heavy tuning sessions.

3. Tuning Capabilities

3.1 Table Tuning (2D & 3D) and CurveGraphs

- **Full-Feature 2D/3D Table Editing**: Fuel (VE), spark, AFR, trim, and specialty tables all support both grid and graphical tuning. Hotspots indicate live engine position; crosshair trackers show active cell.
- **Value Interpolation**: Interpolate values across a selected range (multi-cell selection), smoothing out abrupt changes.
- **Batch Editing**: Through multi-select and drag, users can make batch modifications (raise/lower, scale, apply formulas) to targeted table regions.
- **Baseline Comparison**: Ability to overlay a baseline for reference or comparison against current edits.
- **Cell Lock/Authority Limit**: Prevent changes to certain cells or limit the magnitude of auto-tuned adjustments per session.

These features empower both novice and professional tuners to quickly iterate and maintain safe, effective tables while ensuring comprehensibility and version tracking[5].

3.2 VE Analyze Live (Auto Tune)

A critical automation innovation, “VE Analyze Live” (registered and Ultra versions) provides:

- **Automated Table Adjustment**: Real-time adjustment of VE tables by analyzing live data against AFR targets.
- **Customizable Filters**: Users can set standard (RPM, CLT, MAP, load) and custom filters to improve data quality used in analysis.
- **Lambda Delay Compensation**: Corrects AFR readings by accounting for sensor/exhaust transport delays.
- **Heat Maps/Histogram**: Visual representation of coverage and degree of cell changes for easy progress tracking.
- **Authority Limiting**: Restrict max permissible changes per session for safety.

Automated tuning, with reporting/authority safeguards, is a standout capability that must be deeply integrated in any open-source competitor aiming for parity[6].

3.3 Warm-Up Enrichment (WUE) Analyze

Warm-up auto-tuning helps optimize cold-start enrichment tables:

- **Live Cold Start Logging**: Captures engine warming and recommends optimal WUE table adjustments.
- **Filter and Authority**: Similar customizable filters and safety limits as VE Analyze.
- **1D/2D Table Editor with Real-Time Crosshair**: For precise warm-up curve refinement.

3.4 Trim Table Auto-Tune

Unique to Ultra, this module offers:

- **Individual Cylinder Tuning**: Automated adjustment of trim tables for each cylinder using data from per-cylinder or shared wideband sensors (with sensor rotation support).
- **Flexible Sensor Mapping**: Works with setups from 1 to n widebands (moves sensor as needed), supporting both MS3 and MS2-Extra firmwares.

This feature extends the automation paradigm to granular individual-cylinder optimization, improving both performance and drivability.

---

4. Data Logging and Log Viewer

4.1 Data Log Profiles

- **Multi-Profile Logging**: Ability to define several logging profiles per project, selecting which runtime fields are logged. Profiles can be switched on the fly, aiding in targeted diagnostics or storage optimization.
- **Partial Field Logging for Speed**: With compatible ECUs/firmwares (e.g., MS2 Extra 3.3+, MS3 1.3+), logging fewer fields can boost logging frequency well above 15Hz.
- **Profile Editor UI**: Add, name, enable/disable fields with intuitive selection menus; visual confirmation of logging targets[7].

4.2 Triggered Logging

- **Auto Logging Triggers**: Data logging can be started/stopped automatically based on:
  - Engine RPM thresholds
  - Triggering specific digital inputs
  - Custom user-defined expressions (enabling conditional logging for critical events)
- **Manual Marking**: “Mark” feature allows tuner to tag points of interest in the log during operation, paving the way for subsequent granular analysis.

Flexible, automation-friendly logging is essential for advanced diagnostics and reproducible tests.

4.3 Integrated Log Viewer and Playback

- **Integrated Log Analysis**: Ultra contains a built-in log viewer (similar to MegaLogViewer), supporting standard log formats and:
  - Full cross-referencing-logs are linked with tune tables so users can click points in logs and see corresponding table/curve cells.
  - Real-time table tracing during log playback.

    - Support for both tabular and graphical displays, with annotations and overlayed event marks.

- **Full Application Log Playback**: Playback logs as if they are live engine inputs. All dashboards, tables, and crosshairs in TunerStudio follow playback, turning logs into dynamic simulation/diagnostic sessions.
- **Export/Import Capability**: Standard log export formats support downstream use (e.g., external visualization, statistical analysis)[9][10].

A direct, embedded log viewer offering synchronized data review and application-wide simulation is a hallmark Ultra feature.

---

5. Dashboard Customization & Designer Mode

5.1 WYSIWYG Dashboard Designer

- **Drag-and-Drop Design**: Add, remove, and arrange any dashboard element (gauges, indicators, backgrounds, images) in a live preview.
- **Custom Images**: Import custom graphics for backgrounds, borders, needles, and indicators (image packs freely available in the user community)[3].
- **Attribute Editing**: Every gauge/indicator supports detailed customization-colors, ranges, pointer/face style, tick marks, fonts, warning/critical limits, units (metric/imperial), and dynamic labels.
- **Tab Key Cycling/Selection**: The ability to select hard-to-reach gauges through keyboard navigation, crucial for designs with overlaid or stacked elements.
- **Multi-Select**: Simultaneously edit style attributes (color, font, range) and properties for multiple gauges via CTRL+Click grouping[2].

5.2 Interactive Indicators and Alerts

- **Indicator Placement**: Warning lights can be anywhere, sized to preference.
- **Indicator Logic Source**: Can be tied to any output channel, standard or user-defined.
- **Dynamic Alerts**: Ultra enables both passive (color changes, animations) and active (popup messages, sounds) alerting via configurable triggers.

5.3 Dashboard File Handling

- **Import/Export of Dashboard (.dash) files**: Copy between projects or for sharing online (many free and premium dashboards circulate in the community).
- **Support for Custom Dashboards**: Mix and match via simple file copy, with in-app assignment of output channels and unit adjustments to match firmware requirements[3].

A modern open-source alternative should aim for parity with this dynamic, highly customizable dashboard infrastructure.

6. Communication and Integration Options

6.1 Communication Protocols and Hardware

- **Serial (RS232/USB)**: The canonical communication channel, with auto-detection and variable baud rate support per project/firmware.
- **FTDI D2XX USB**: Direct USB driver support for low-latency, high-speed communication.
- **Direct Bluetooth**: Native Bluetooth device pairing/discovery in Windows (other OSes via virtual com ports); enables wireless tuning and diagnostics.
- **WiFi**: For ECUs or gateways supporting WiFi, direct connection via IP/TCP.
- **GPS Integration**: Uses any NMEA-compliant GPS dongle via serial, USB, or Bluetooth (with device and firmware-specific baud rate configuration). Offers up to 10Hz updates, supporting real-time speed and positioning data capture with configuration dialogs for COM port assignment, baud, and polling[12].

6.2 GPS & External Sensor Data

- **GPS Data Fields in Runtime/Logs**: Latitude, longitude, speed (MPH/KPH), heading, accuracy, altitude, X/Y position (from lock point), and synchronized time.
- **Performance Metrics**: With 10Hz GPS, enables accurate calculation of acceleration, HP, distance, and fuel consumption-integrated into gauges and logs.
- **User Gauge Assignment**: Right-click any gauge to assign a GPS data source.

Integration of GPS and other NMEA/serial sensors is a must for parity in motorsports-oriented open-source tuning suites.

6.3 Multi-Controller, Multi-Project Support

- **Multiple Project/Controller Handling**: Seamless switching and concurrent management of settings for multiple ECUs within same software install.
- **CAN Bus Integration**: In advanced firmwares (MS3/MS2-Extra), supports configuration and tuning of multiple CAN devices in a single project.

7. Plugin API and Extensions

7.1 Plugin System

TunerStudio has a Java-based plugin architecture since v1.34, allowing:

- **Dialog Plugins**: Add custom dialogs for settings, visualization, or configuration.
- **Data Access API**: Read runtime values, update controller settings, create new UI components, or run custom calculations on output channels or constants.
- **Tab-Based Plugins** (planned/in-progress): Full tab plugins could enable embedding of advanced widgets or entirely new view modules (e.g., MP3 player, external diagnostic tool).
- **Plugin Management UI**: Users can add or update plugins via Tools → TunerStudio Plugins → Add/Update, with hot-swapping for development. Manifest-based discovery/registration ensures plugins don't interfere with core stability.
- **Easy Data/Setting Binding**: Plugin API translates controller/ini communication for the developer.

A robust, documented, user-manageable plugin architecture is required for open extensibility.

---

8. Configuration Options and INI/Project Files

8.1 Firmware INI Specification

- **.ini File Parsing**: TunerStudio reads firmware definition files for each ECU/family, defining:
  - Metadata, communication parameters, channel and constant definitions
  - All settings dialogs, table structures, curve graphs, and available data fields
  - Conditional and multi-value settings groups ([SettingGroups] section), enabling customizable project/wizard dialogs for firmware options selection.
- **Conditional/Boolean Blocks**: Enables/Disables sections of the UI or configuration logic based on firmware settings or user selections.
- **Custom Controller Commands/Buttons**: Expose one-click actions for rarely used or advanced firmware capabilities.

Any competitor suite must fully support reading, parsing, and conditionally presenting all relevant .ini file constructs for flexibility with varied firmware.

8.2 Project and Session Configuration

- **Project Properties Dialog**: Settings for sensor type (narrowband vs. wideband O2), temperature units (C/F), ECU variant switches, and interface options.
- **Template and Default Importing**: Standard MSQ file structure handling, with auto-migration for basic/legacy settings.
- **Single Dialog/Batch Save/Load**: Save all settings or subsets (e.g., only fuel table) to MSQ or partial files.

Fine control over all settings and table persistence options is vital for reproducible, shareable configurations[1].

---

9. Channel Editor and CurveGraph Multi-Select

9.1 Add Channel Wizard & Custom Channel Editor

- **Add Channel Wizard**: Simple guided interface for new sensor channels-linear, table-mapped, or mathematical expression-based channels.
- **Advanced Custom Channel Editor**: Allows creation and editing of output channels, gauge templates, and log fields using advanced mathematical expressions, table lookups, or binary expressions.
  - Links to runtime values or constants, supports derived channel creation with custom names, units, and scaling.
  - Supports both graphical and direct .ini editing modes for power users.
- **Batch Data Log Field Mapping**: Choose which fields appear in logs and on dashboards with a right-click change.

The system is so advanced it nearly eliminates the need for manual .ini editing for custom sensor or channel support. Replicating this flexibility is crucial for competing products[14][15].

9.2 CurveGraph Multi-Select & X-Y Plotting

- **Drag/Box and CTRL Multi-Select**: Apply operations to batches of curve points, supporting mass smoothing, scaling, and editing.
- **X-Y Plotting**: Overlay log data points directly onto tuning curves, providing instant feedback (e.g., real-world vs. theoretical traction control graphs).

---

10. Advanced Math Functions and Action Triggers

10.1 Math Parser Functions

TunerStudio’s internal expression parser enables powerful calculations across:

- **Standard Functions**: sin(), cos(), tan(), log(), sqrt(), floor(), ceil(), round(), abs(), min(), max(), e^ x, etc.
- **Signal/Log Math**: All forms of data smoothing, historical record lookup, summing/accumulating fields for odometer/fuel used/trip computer scenarios, HP calculation (accelHp()), aerodynamic drag, rolling resistance, and more.
- **Table Lookup and Array Access**: For values stored in 1D/2D tables, with interpolation.
- **Conditional Logic**: Excel-style if() statements, toggle switches, selectExpressions.
- **Time and State Functions**: Time to Exceed, timeTrue, isTrueFor-powerful for creating conditional warnings/alerts and virtual actuators.

Full parity requires a robust, embedded math engine for live channels and configuration logic.

10.2 Action Triggers and User Actions

- **Action Trigger System**: Developers or users can define arbitrary conditional expressions to trigger built-in or user-generated actions, based on live engine data or digital inputs.
- **Built-in Actions**:
  - Start/stop/toggle data logging
  - Mark log points
  - Dashboard navigation
  - Project switching
  - Application shutdown
- **User Actions**: Custom parameterized actions-play sound, run script/program, reset accumulator, show warning/pop-up, load tune fragment, or send arbitrary controller command.

This automation supports high customizability and enables event-driven reactions crucial to advanced tuning and safety workflows[16].

---

11. Graphical Difference Reports

11.1 Visual Diff Tools

- **Tune-to-Tune Comparison**: Users can compare any two MSQ or restore point files, with differences graphically highlighted side-by-side.
- **Field-by-Field Tracking**: Every ECU setting and table entry compared for fast spotting of accidental or unexpected changes.
- **On-Connect Differential Alert**: Optionally, TS will prompt for diff analysis at ECU connection, warning of any unsynced project edits.

This visual approach, not just plain text diff, is crucial for both safety and user friendliness in high-stakes motorsports tuning environments.

---

12. GPS and External Sensor Integration

12.1 GPS Support Details

- **Device Compatibility**: Supports virtually all NMEA 0183 GPS dongles (USB/Bluetooth/serial), with easy configuration dialogs for port, baud, and update frequency (1Hz or recommended 10Hz).
- **Direct Dashboard Integration**: Assign GPS fields to any gauge via context menu; supports units conversion (KPH/MPH).

- **Performance Logging**: Log all GPS data fields, enabling derived metrics (e.g., time-to-speed, dragstrip timeslip auto-generation in logs).
- **External Sensor Expansion**: Custom channel integration allows virtually unlimited sensor addition (pressure, temperature, fuel, etc.) with runtime and logging visibility.

Any competing suite must natively support NMEA GPS and make sensor extension as frictionless as possible[12].

---

13. User Actions, Sound Alerts, and Notifications

13.1 Configurable Alerts

- **Visual Indicator Alerts**: Dashboard indicators light up/redline based on user-set thresholds or channel states, supporting both firmware-defined or user-constructed output channels.
- **Sound Alerts**: Ultra can play .wav or .mp3 files as user-defined triggers (critical temperature, low oil, etc.).
- **Pop-Up and Passive Messages**: Global warnings (blocking pop-up requiring dismissal) or passive toast-style notifications in the UI, all assignable to triggers.
- **Script/Program Execution**: Execute external apps or system scripts as reactions to engine events, supporting full workflow integration.
- **Log Marking and Comments**: Automated log marks with custom text, timestamped for rapid session analysis.

Customizable, event-driven alerting and automation dramatically improve both safety and user productivity, and are particularly valuable for high-performance and research applications[16].

---

14. Additional Notable Features

14.1 Contextual Help System

- **F1 Help**: Context-sensitive help for every dialog, table, and setting (incorporated into INI/user documentation).
- **Tooltip Bubbles**: Hover tooltips provide descriptions and tuning guidance for every UI control.

14.2 Export and Import Ecosystem

- **Dashboards**: Wide library of free and premium downloadable dashboards (.dash files), with easy import/export.
- **Tune Files**: Interchangeable .msq and .ini files encourage sharing and collaboration.

Open platforms should both enable and actively nurture a rich user-contributed ecosystem for dashboards, tune templates, and extensions.

14.3 OS and Platform Support

- **Windows, Mac, Linux**: Cross-platform installer/distribution, with direct hardware and virtual com support for all major platforms.

Conclusion: Feature Parity and Open-Source Development Checklist

To **replicate true feature parity with TunerStudio Ultra**, an open-source project must implement not only the core tuning dialogs and logging mechanisms, but also the advanced productivity, extensibility, and safety features detailed above. This includes:

- Comprehensive, modular UI with WYSIWYG dashboard and view designers, deep attribute editing, and flexible import/export
- Full VE/WUE/Trim table autotune and authority controls
- Instantly accessible, in-app log viewer with synchronized playback
- Modular data logging with auto-trigger, marking, and profile management
- Advanced custom channel, math, and sensor integration with UI-based editors
- Robust action/automation triggers supporting scripts, sound, messages, and programmatic reactions
- Plugin API with runtime data, settings manipulation, and UI extension hooks
- Advanced diff, restore, and historic tracking tools for reproducibility and safety
- GPS and external sensor support natively, with performance metrics and live dashboard assignment
- Extensive, intuitive keyboard support and direct navigation/UI manipulation
- Built-in contextual help, community resource integration, and shareable assets/collaboration ecosystem

By replicating these integrated features-ranging from UI, logging, automation, and extensibility, to ecosystem and workflow-open-source developers can deliver a credible alternative to TunerStudio Ultra's dominant, productivity-focused platform. Omitting any of these areas may significantly detract from daily usability, advanced tuning capability, or project safety, and thus should be prioritized from the earliest development stages.

This extensive analysis is derived from official documentation, developer manuals, forums, video tutorials, community-contributed resources, and up-to-date changelogs to ensure every critical aspect of TunerStudio Ultra is accurately represented and deeply contextualized for the developer's checklist[4][7][9][3][1][14][16][17].

References (18)

1. *How To Setup TunerStudioMS - DIYAutoTune.com*. https://www.diyautotune.com/support/technical-articles/other/tunerstudio/
2. *Using TunerStudio Dashboard Designer*. https://tunerstudio.com/index.php/support/manuals/53-using-tunerstudio-dashboard-designer
3. *Welcome to TunerStudio Dashboards*. https://tunerstudiodashboards.com/
4. *Default Tuning & Dyno Views for TS3 beta - Megasquirt EFI*. https://www.msextra.com/forums/viewtopic.php?t=62389
5. *VE Analyze Live - Step by step instructions - Megasquirt EFI*. https://www.msextra.com/forums/viewtopic.php?t=60408
6. *Data Log Profiles - TunerStudio*. https://tunerstudio.com/index.php/products/tuner-studio/tsarticles/94-data-log-profiles
7. *TunerStudio Integrated LogViewer*. https://www.tunerstudio.com/index.php/products/113-tunerstudio-integrated-logviewer
8. *EFI Analytics News Letter*. https://www.efianalytics.com/email/2024-05-03-NewsLetter.html
9. *GPS and TunerStudio with a Pi Dash - Megasquirt EFI*. https://www.msextra.com/forums/viewtopic.php?t=66473
10. *Unable to use "Advanced Channel Editor" in TunerStudio*. https://www.msextra.com/forums/viewtopic.php?t=65785
11. *Custom advanced inputs for Tunerstudio gauges - YouTube*. https://www.youtube.com/watch?v=etn2Mx1MX3A
12. *Action Management - TunerStudio*. https://tunerstudio.com/index.php/products/tuner-studio/tsarticles/125-action-management
13. *TunerStudio MS Ultra - Tuning and Dyno Views*. https://tunerstudio.com/index.php/products/tuner-studio/tsarticles/114-tunerstudio-tuning-and-dyno-views
14. *Tunerstudio and Shadowdash custom alerts? - Megasquirt Support Forum ...*. https://www.msextra.com/forums/viewtopic.php?t=72539