

PCM Telemetry HMI Tools: AEROPGLGo13

The **GLG Toolkit** (Generic Logic) is a cross-platform graphical framework used to build real-time HMI displays for PCM (Pulse Code Modulation) telemetry, SCADA, and aerospace monitoring. It is frequently utilized to process, decode, and visualize high-speed, low-latency telemetry data streams.

Core Capabilities for Telemetry

- **Real-Time Data Visualization:** Capable of rendering data from high-frequency telemetry streams using dynamic charts, dials, meters, and status indicators.
- **No-Code HMI Editor:** The **GLG Graphics Builder** allows engineers to design complex visual interfaces and instrumentation panels using a point-and-click interface without manual coding.
- **Cross-Platform Support:** Deployable across environments including C/C++, Java, C#/.NET, and JavaScript/React/Angular for web browsers and mobile devices.
- **Data Integration:** Interfaces natively with backend data acquisition systems, supporting custom data sources and math engines for real-time decommutation and derived parameter calculations.

Target Industries

Because of its ability to bypass "black box" limitations, the toolkit is a standard in industries with mission-critical requirements:

- **Aerospace & Defense:** Used for tracking satellite data, rocket launches, and aircraft telemetry.
- **Industrial Automation & SCADA:** Employed for process control, network monitoring, and traffic management.

If you're evaluating the GLG Toolkit, would you like information on:

- **Specific supported platforms** (e.g., integrating with C# vs. React)
- Downloading and testing the **Free Community Edition**
- Connecting the HMI to **your specific data source**

The GLG Toolkit by Generic Logic is a premier multi-platform graphical framework engineered specifically for building low-latency, mission-critical **PCM (Pulse Code Modulation) Telemetry and HMI (Human-Machine Interface)** visualization systems. In telemetry and aerospace environments, where decommutated PCM streams supply thousands of data points per second, the GLG Toolkit serves as the graphical rendering engine that transforms raw stream data into highly responsive instrumentation dashboards.

Core Components for Telemetry Display

The toolkit delivers an end-to-end environment to build and deploy complex ground station or test cell mimics:

- **GLG Graphics Builder:** A point-and-click vector graphics editor used to build telemetry display screens without manual coding. Developers can build custom telemetry panels using pre-built dials, multi-pen strip charts, meters, and standard aviation/engineering gauges.
- **GLG HMI Configurator:** A specialized, simplified version of the editor designed for end-users or test engineers. It allows field operators to easily rearrange dashboard layouts, change visual styles, and map incoming data channels to specific dashboard indicators on the fly.
- **High-Performance Charts:** Optimized rendering engines capable of displaying real-time strip charts and time-history plots tracking tens of thousands of telemetry data points with minimal CPU overhead.

Integration Architecture

Rather than acting as a standalone telemetry "black box," the toolkit provides an open API architecture. It drops directly into custom telemetry decommutation pipelines and data distribution systems:

[Raw PCM Stream] —> [Decommutator / Math Engine] —> [GLG Open Data API / Custom Browser] —> [Real-Time HMI Display]

- **Open Data API:** Easily bridges the graphics layer to backend data acquisition hardware, software buses, or telemetry databases.
- **Custom Data Browsers:** Developers can embed their own parameter search tools directly into the GLG editor. This allows test engineers to browse thousands of decoded PCM parameters and assign them to specific graphical elements by dragging and dropping.

Cross-Platform Deployment Flexibility

Telemetry architectures often combine rugged real-time operating systems with web-based remote viewing rooms. The GLG object engine allows you to build an interface once and deploy it across any stack:

Environment**Supported Technologies & Formats****Desktop Software**

Native C/C++ DLLs, C#/.NET Windows Forms, Java

Web & Control Rooms

Pure HTML5, JavaScript, React, Angular, Next.js

Embedded & Field Kits

Windows, Linux, Unix, ARM boards, RTOS

The web-based components leverage a lightweight HTML5 canvas rendering architecture to maintain resolution-independent vector graphics on remote display walls, tablets, or mobile test equipment.

Are you planning to build a **desktop-based control application** (e.g., C++ or C#) or a **web-based monitoring dashboard** (React/JavaScript) for your PCM telemetry data? If you can share details about your **data backend or streaming protocol**, I can provide more specific implementation steps

OS & HARDWARE SUPPORT**INDUSTRY APPLICATION**

Note: ** Specifications are copied from manufacturer data. Please contact before buying.