

vsRAD Real-time SDK displays Radar echoes : AEROPVS013

The vsRAD Real-time SDK is a high-performance, C-based visualizer that works directly with vsTASKER to display dynamic radar echoes. When vsTASKER generates tactical scenarios (like maritime or coastal surveillance), it sends the real-time detection data to vsRAD to render the raw dB per angle and range data using OpenGL.

How the Integration Works

- Scenario Generation (vsTASKER): Sets up the simulation environment, defines entity behaviors, and computes sensor detections.
- Raw Data Processing: Data like distances, speed (Doppler), and radar cross sections are calculated and piped to vsRAD via shared-memory.
- Real-time Rendering (vsRAD): Converts these raw parameters into adjustable, displayable radar images (PPI) and moving echo returns seamlessly.

Key Features

- Published API: Allows users to add custom callbacks to modify echo displays or overlay S-57 maritime charts and terrain models.
- Open C-Code & OpenGL: Runs on pure OpenGL without shader reliance, delivering high frame rates and avoiding flickering.
- Real-Time Kernel (RTC): Can be run at fixed frequencies or completely unthrottled ("as-fast-as-possible") depending on simulation needs.

You can read more about this on the VirtualSim vsRAD Product Page and review the maritime use case on the vsTASKER Maritime Radar Simulation Portfolio.

Would you like to know more about:

- Integrating vsRAD into a custom OpenGL or Visual Studio application?
- Generating specific radar cross-section (RCS) data in vsTASKER?

- Displaying 5GHz or C-Band radar raw data?

vsRAD operates as a real-time radar visualizer toolkit that receives and displays radar echoes, detections, and raw data generated by the vsTASKER simulation engine.

Both tools are developed by VirtualSim and are deeply integrated to provide comprehensive radar and tactical scenario simulations.

How the Integration Works

- Scenario Setup: vsTASKER acts as the core simulation engine. It setups the entire tactical environment, controls the terrain/charts, positions the radar, and simulates the real-time movement of targets (such as aircraft or marine vessels).
- Data Transmission: When the simulated radar in vsTASKER detects an entity, it computes data such as azimuth, distance, and orientation. This data is streamed directly to vsRAD in real-time.
- Real-time Echo Display: vsRAD utilizes a shared-memory segment to ingest these commands and data inputs instantly. It converts the incoming simulation signals into visual, adjustable radar echoes or merges target images over background clutter at runtime.

Key Technical Specs of vsRAD

- C & OpenGL Architecture: Written in pure C code and relying on native OpenGL without shaders to guarantee maximum rendering performance.
- White Box API: Features an open API that allows developers to add custom callbacks to drawings and integrate the visualizer directly into larger, custom user HMI panels.
- HMI Builder Integration: In vsTASKER v7, vsRAD comes natively integrated into the HMI builder, enabling developers to build console logic and system panels from a unified environment.

If you are setting up a specific simulation project, let me know:

- Are you simulating maritime surveillance or airborne/airport radar?

- Do you need to display raw data (IQ/dB levels) or processed track detections?
- What version of Visual Studio or development stack are you compiling with?

vsRAD Radar visualizer toolkit in C code - Simulation Software

Real-time SDK displays Radar echoes * White Box – vsRAD is designed, to allow the user to add his own callbacks to almost any draw...

vsTASKER download page - Simulation Software

vsRAD. vsRAD can be installed without dependencies. Its C code can be compiled and linked with any Visual Studio. Solutions for 20...

vsRAD

Real-time SDK displays Radar echoes

vsRAD mimics a radar display with data coming from any synthetic environment or simulation sources. Version 4 embeds real-time RTC and support incrustation layers for Navigation charts (S57), coastlines, Arc/Info, elevations (DTED) with embedded line-of-sight. Real-time data input (acquisition card) can be added on the display using bespoke user-defined drivers (or under consultation).

Written in C for the main engine, it comes with an optional user interface (GUI), a full programmable API, documentations and samples.

Designed to be easy to use and expand, vsRAD is an affordable COTS for simple OpenGL radar display. However, the software does not simulate waves propagation. The result of such analysis must be done by another software, offline or concurrently and displayed inside vsRAD.

The engine can be used standalone or embedded into any GUI: OpenGL/GLUT window, Qt, VAPS or GL-Studio panels, VegaPrime or OpenSceneGraph windows. It supports LAN or shared-memory remote control.

vsRAD is a **multi-functional, multi-modes, remote-controllable** and **customizable** software used to display radar echoes.

Version 4 brings a lot of improvements over the last builds, from customers requests and feedback: Raw-Data import, speed optimizations, coloring, aspect, Coastline, Arc/Info, S57 charts and terrain incrustations are available for large area radar coverage.

vsRAD can be connected to any real-time application through shared

memory or embedded into a custom-made GUI/HMI using the documented programmable API and Visual Studio compatible libraries.

The product is available as a standalone application as well as an SDK for custom integration. A GUI controller is also provided as a demo and can be replaced by any other.

It provides three different modes (PPI, AR & B). The engine can run in sync with the clock (synchronous) or as fast as the CPU can (asynchronous).

-
- **White Box** – vsRAD is designed, to allow the user to add his own callbacks to almost any drawings, using a published API.
 - **C code** – vsRAD is written in C for maximum performances.
 - **OpenGL** – vsRAD uses pure OpenGL and does not rely on shaders.
 - **RTC** – vsRAD provides an optional real-time kernel to fix the frequency or can run as-fast-as-possible for less flickering effects.
 - **Shared-Memory** – vsRAD is using a shared-memory segment for data inputs and commands.
 - **Connectivity** – vsRAD is also a standalone library that can be embedded into a graphical application that supports OpenGL.
 - **API** – vsRAD comes with a complete and documented API.
 - **Expandability** – vsRAD provides out of the box PPI, A/R and B modes. Other modes can be added by the user.

Easy Integration with VAPS™, GL-Studio™, OSG™, Qt™, vsTASKER

Operating Systems Windows XP/7/10

Compatibility Visual Studio C++ 2008 / 2010 / 2015

System Requirements Accelerated OpenGL graphic card

OS & HARDWARE SUPPORT	INDUSTRY APPLICATION

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