

AI-native Control for Aukua Ethernet & Fibre Channel Test Platforms

Tell your AI assistant what you need done; configure ports, start generators, apply impairments, capture traffic, or pull stats. And with Aukua's MCP Server, your AI assistant translates intent into precise instrument API calls for control or for generating repeatable automation scripts. Numerous and repetitive manual GUI clicks and deep product expertise are no longer prerequisites for advanced Ethernet and Fibre Channel testing.

SOLUTION HIGHLIGHTS

Natural language control

Configure ports, generate traffic, apply impairments, capture packets, and pull stats by asking your AI assistant.

Runs on the instrument

Delivered with Aukua firmware. No separate server stack to install or maintain. No curl, Python, or CLI required for day-to-day control.

One MCP endpoint for the lab

Drive the local box or other Aukua test systems on the network (MGA2510 / MGA8410 / XGA4250).

Progressive API disclosure

Loads only the endpoints and schemas needed, keeping assistant context lean, accurate, and lower cost.

From conversation (chat) to automation

Use AI for interactive setup and inspection, then, without learning the REST API, ask it to generate what worked into repeatable scripts (e.g., python) for integration into a regression suite.

Instrument-aware triage

Read live stats, configurations, and analyze L1 / PCS and L2 packet captures with help from an assistant that can also have context into the device or system under test.

Open MCP standard

Works with Cursor, Claude, Microsoft Copilot, and other assistants without vendor lock-in.

The Challenge

Modern development and validation labs must move faster than ever to keep pace with today's innovation. Test tools need to enable that speed, not slow it down. Teams must build test plans, qualify functionality, prove performance, characterize latency, and troubleshoot hard problems, all under tight schedules. There is little time left to become proficient at operating or automating each instrument. So teams are turning to AI assistants to help get their work done.

How you connect those assistants to the equipment is critical. Manually feeding the assistant screenshots, stats dumps, or pasted config files is tedious, slow, and biased by what the user chooses to share, so the assistant never gets full, instrument-aware context. Dumping an entire OpenAPI schema into an LLM burns context, increases cost, and still leaves the model guessing which calls to make. And building and maintaining a custom CLI for every tool is even more complex and costly.

The Aukua MCP Solution

Aukua's MCP Server solves this by running on the instrument itself and giving your AI assistant a direct, structured way to work with the live box. Instead of pasting screenshots or dumping a full OpenAPI schema into chat, the assistant uses focused discovery and execution tools to find the right call, understand what it needs, then run it with progressive schemas that keep context lean and accurate. No custom CLI, curl, or extra software is required for day-to-day control: the assistant talks to Aukua MCP, and MCP drives the instrument API.

The same connection works across the Aukua portfolio. The server understands product-specific port pairs, API variants, and optional authentication, so one MCP endpoint can operate the local system or other Aukua test systems on the network. Teams use it to configure, run, and troubleshoot in plain language, then turn a successful session into a repeatable script, all through the open MCP clients they already use, such as Cursor, Claude, and Copilot.

Conversational Control:

"Change my Aukua box at 192.168.7.117 to 10GBASE-R Generator mode." - No extra software to install or maintain. Works from your AI assistant



AUKUA MCP SERVER

From Conversation to Controlled Automation

Business Benefits

Cut time-to-test

Configure, run, and inspect complex generator, capture, and impairment scenarios in natural language instead of multi-step GUI clicks.

Lower the expertise barrier

Network and validation engineers get productive on Aukua platforms without memorizing REST API, and leverage extensive feature sets without indepth product training.

Turn chats into scripts

Use conversational control for exploration, then generate Python scripts from the same assistant for repeatable automation and building regression suites.

Troubleshoot on the box

Pull live stats, configs, then analyze L1 / PCS and L2 packet captures with instrument-aware help in real-time.

Unify the lab

One MCP server understands the entire Aukua portfolio and can operate the local box or other reachable MGA2510 / MGA8410 / XGA4250 test systems. No extra tools to install or maintain.

Stay open and future-ready

Built on the open standard Model Context Protocol (MCP) so Aukua instruments work with the AI tools your teams already chose.

Example Use Cases

Conversational Control

"Configure my Aukua box at .145 for 10GBASE-R traffic generator and start line rate traffic with IMIX packet sizes and a PRBS payload."

Conversation to Automation

"Create a Python script that sets up Port Pair A traffic as we discussed, then iterates the delay, impairment, and bandwidth cases we defined on Port Pair B saving bandwidth and latency results in a pdf report."

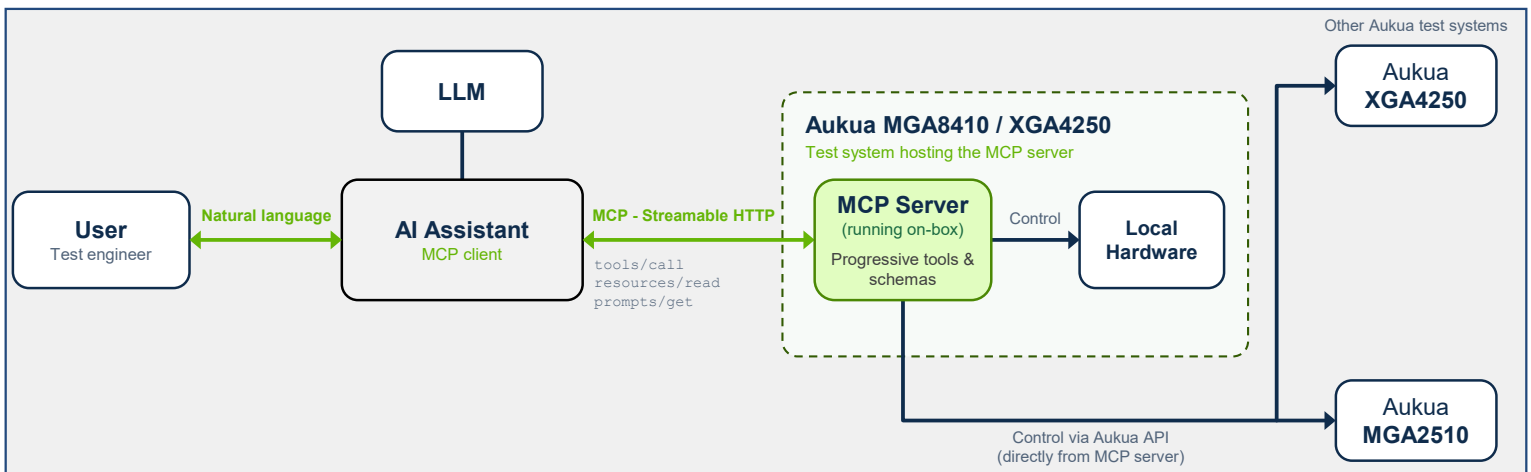
Live Triage

"Port 2 RX stats looks wrong. Pull live stats and config, help me find the problem, and suggest the next steps."

Capture Analysis

"Run an L1 PCS capture on Port 1 and summarize it, calling out any PCS or link issues."

How It Works



Contact Aukua to enable MCP, see a live demonstration, or discuss conversational control in your lab.