openrcs-protocol user guide

This is **a reference for the TCP control protocol of Analog Way Midra and LiveCore series video processors** — the wire format, the variable model, and the complete variable tables.

It is documentation only. There is no software here. It exists so that anyone building control software for these processors has a single, precise description of how they talk on the network.

> **Before you rely on this:** it is **reverse-engineered, not vendor documentation.** The LiveCore side has been checked against device behaviour; the **Midra tables are complete but have had less exercise against hardware.**
>
> **Treat per-variable ranges as strong guidance rather than a guarantee — the device is always the final authority.**
>
> This reference was written with AI assistance, directed and reviewed by a human author.

Not affiliated with or endorsed by Analog Way. Product names are used only to describe compatibility.

At a glance

- **Transport:** TCP port **10500**, one control socket, **no handshake**.
- **Framing:** ASCII, and **asymmetric** — a command *ends* with its mnemonic, a reply *begins* with it, and a reply's last comma-separated field is the value.
- **Terminators: Midra sends `CRLF`, LiveCore sends `LF`** — and **the device replies in `CRLF` on both.**
- **Push:** the device sends value updates **unsolicited**, so state must be read continuously, not just polled.
- **Errors:** a rejected command is answered `E<code>` — `E10` unknown, `E12` wrong index count.

Three of those five are the ones that catch an implementation out. **The asymmetry** means a naive parser that assumes replies echo commands will never match anything. **The terminator mismatch** means splitting on the terminator you *send* leaves a stray carriage return on every line you *receive*. And **the unsolicited push** means a poll-only client will show stale state whenever somebody touches the front panel.

What is in here

framing.md	the wire format — commands, replies, terminators, errors, push
variable-model.md	variables, indices, values, layer geometry, memories
midra.md	Midra series — all 562 variables, grouped
livecore.md	LiveCore series — all 1014 variables, grouped
`data/`	the same tables as machine-readable JSON

Devices covered

Midra series: Pulse2, Eikos2, Saphyr, SmartMatriX2, QuickMatriX, QuickVu.

LiveCore series: Ascender 16/32/48, NeXtage 8/16, SmartMatriX Ultra.

Using it

Point an implementation at **framing.md** for the codec, and at the per-platform tables for the variables.

The JSON can be code-generated into a typed table — that is exactly what openrcs does, rather than transcribing 1,576 variables by hand into source that then drifts.

Two things to design around

Very few concurrent control sessions. The device accepts only a handful. If a vendor client or another controller already holds the connection, yours will not get one — plan for that as a normal state rather than an error.

No authentication of any kind. Anyone who can reach port 10500 can drive the switcher. That is the protocol, not a configuration choice, so the network is the only access control there is.