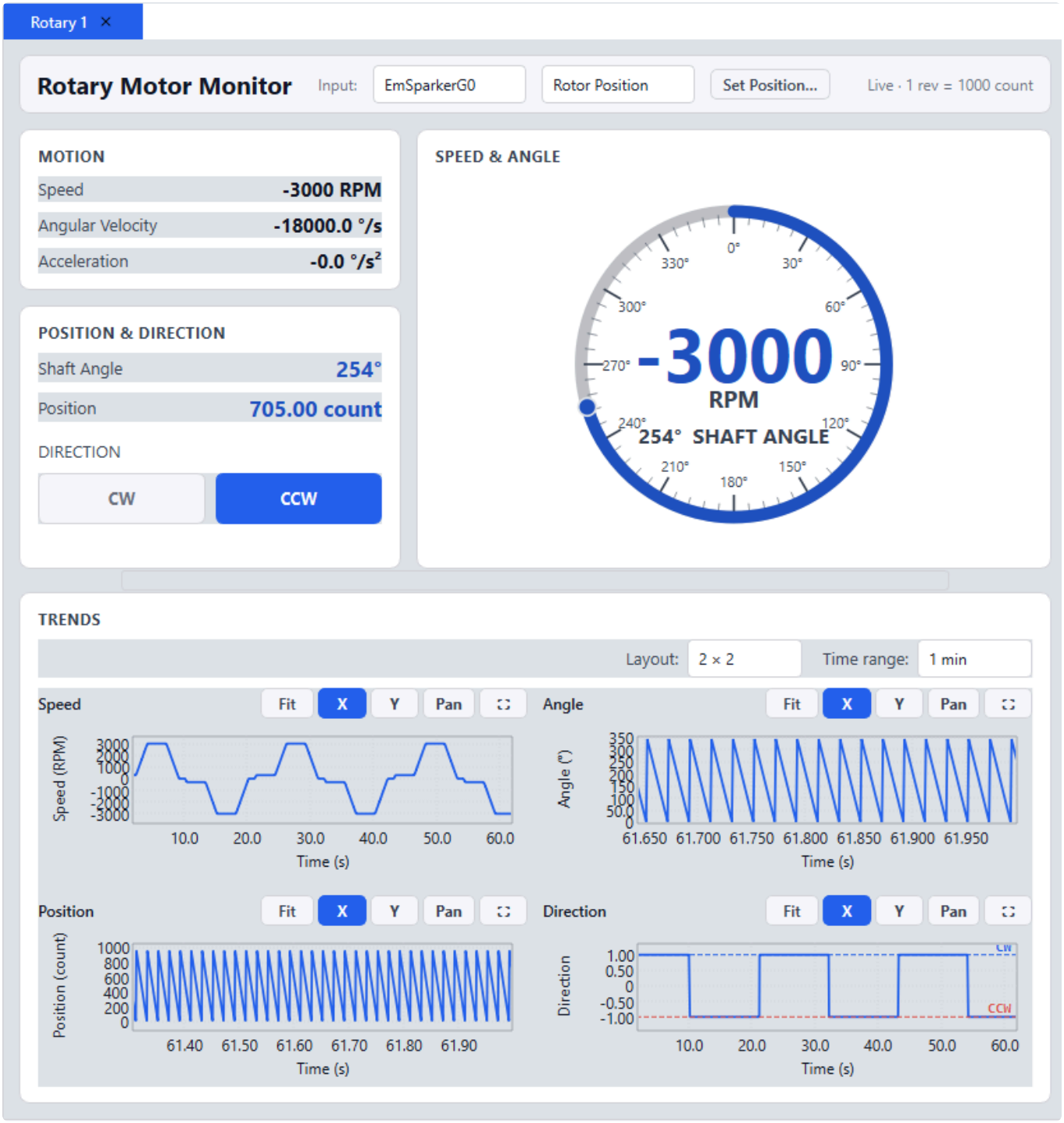
19 — Rotary Motor Monitor

TraxProbe & Traxcope User Manual · Version 1.0.0 · Updated 2026-07-28

A dashboard view for rotating machinery: bind it to **one rotor stream** and it derives and displays the full kinematic picture — shaft angle, speed (RPM), angular velocity, acceleration, and direction — live.

Open via Toolbox → **Rotary**.



Screenshot: Rotary Motor Monitor — Motion and Position & Direction cards, the shaft-angle dial, and the four trend plots

---

On the device

The monitor consumes a **rotor stream** — a dedicated variable-stream kind (Chapter 12) declared with `TRAX_STREAM_ROTOR_DEFINE`. Define the position variable as usual, then bind it to a rotor stream that declares the encoder geometry:

```
/* Position VAR — min/max is only a plotting hint for the Scope view */
TRAX_VAR_DEFINE(VAR_TID_ROTOR_POS, "Rotor Position", TRAX_DATA_TYPE_UINT16,
                0, 1000, "count", TRAX_COLOR_MAGENTA, TRAX_PLOT_LINE);

/* ROTOR stream: 1 kHz sampling, 1000 encoder counts = one revolution */
TRAX_STREAM_ROTOR_DEFINE(STREAM_TID_ROTOR, "Rotor Position 1 kHz",
    1000, 1,                    /* sample rate: 1000/1 Hz              */
    TRAX_BYTE_ORDER_LITTLE,
    1000u,                      /* counts_per_rev                      */
    VAR_TID_ROTOR_POS);
```

The runtime is the standard stream pattern — fill a block of position samples, then `TRAX_STREAM_START` / `TRAX_STREAM_UPDATE` / `TRAX_STREAM_SEND_ALL` on `STREAM_TID_ROTOR`, exactly as for an ADC group.

Two things make the rotor stream the *only* input the monitor accepts:

- **The geometry contract.** `counts_per_rev` declares how many counts (post-quadrature, as present in the data) make **one mechanical shaft revolution**. That drives the shaft-angle normalization (0–360°), the RPM conversion, and direction detection. A plain variable's min/max is a plotting hint, not a measurement contract, and a generic ADC group carries no geometry at all — so neither appears in the monitor's Input selector. Position samples are expected to wrap within `[0 .. counts_per_rev)`.
- **The sample rate.** The wrap-aware differencing resolves at most **half the encoder range per sample**, so the sample rate caps the measurable speed: `max RPM = (Fs / 2) × 60` for a full-range step. A 100 Hz `TRAX_VAR_SET` publisher would top out at 3 000 RPM on a 1000-count encoder — and cost a full var frame per sample. The stream path ships one timestamp anchor per block with implicit sample spacing, so kilohertz rates cost almost nothing on the wire (1 kHz ≈ 2 kB/s) and raise the ceiling to 30 000 RPM.

Initial position

Position samples on the wire are **absolute** — the monitor derives everything from the values the probe actually sends, starting with the very first sample. A device that persists its shaft position across power cycles (FRAM, EEPROM, backup RAM — persistence is entirely the application's responsibility, TraxProbe plays no part in it) simply restores the saved count at boot and starts the stream from there; the monitor then opens at that exact position instead of assuming zero.

In Traxcope

Binding the input

Pick the probe and input with the **Input** selectors in the header bar. The variable list offers **only rotor-stream position channels** — variables published with `TRAX_VAR_SET` or bound to generic ADC-group streams are not shown, because they carry no encoder-geometry contract. The binding is saved in the workspace, so the monitor reconnects to its channel on the next launch. The right side of the header shows the binding status (*No channel* until an input is selected, then *Live · 1 rev = N count* from the stream's `counts_per_rev`).

Derived kinematics

All readouts are computed host-side from the position samples by finite differences:

Readout	Meaning
Shaft Angle	Position normalized into one revolution, 0–360° (also the dial needle)
Speed	Revolutions per minute (RPM)
Angular Velocity	Degrees per second
Acceleration	Degrees per second²
Position	The raw input value, in the variable's own unit
Direction	CW / CCW pills — the active one lights up with the accent color

The differencing is **wrap-aware**: when the position jumps across the range boundary (e.g. 999 → 2 on a 0–1000 encoder), the monitor folds the step back instead of reporting a huge velocity spike, so a steadily spinning motor reads a steady speed. Velocity is lightly low-pass filtered to keep the dial readable; readouts refresh at ~30 Hz.

Setting the current position

Set Position… in the header bar re-references the display at any time: enter what the position readout should *currently* show, and the monitor applies the constant difference to every position-derived readout from then on — Position, Shaft Angle, the dial needle, and the Position/Angle trends (which restart at the new reference to avoid a fake step edge).

This is a **display-side reference only**:

- **Nothing is sent to the probe.** The device's own encoder count is untouched; the monitor keeps consuming the raw absolute samples and adds the reference on top.
- **Rates are unaffected.** Speed, angular velocity, acceleration and direction difference the raw samples; a constant offset cancels out.
- **The reference resets** when the binding changes or a new session starts — a (re)connected probe is always shown at the position *it* indicates (see *Initial position* above) until you re-adjust. The active reference is visible in the header status (*offset N count*).

The button enables once the monitor has a live position fix (two samples).

Layout

- **MOTION card** — the rate readouts (Speed, Angular Velocity, Acceleration).
- **SPEED & ANGLE dial** — hero gauge: the needle tracks the shaft angle, the center shows live RPM.
- **POSITION & DIRECTION card** — absolute shaft angle and raw position, plus the CW/CCW indicator pills.
- **TRENDS strip** — four time-series plots: Speed (RPM), Angle (0–360°), Position, and Direction (a CW/CCW state timeline with ±1 reference lines). A vertical splitter lets you trade space between the dial region and the trends.

Trend controls

- **Layout** (shared) — how the four plots are arranged: **Auto** (reflows from side-by-side to stacked as the view narrows), **1 × 4** (one row), **2 × 2**, or **4 × 1** (each plot full width on its own row).
- **Time range** (shared) — 10 s, 1 min, or 10 min of history.
- Per-plot toolbar: **Fit** (restore the configured time range and any pinned Y axis), **X** / **Y** (which axes the mouse wheel zooms), **Pan** (left-drag pans instead of rubber-band zoom), and ⛶ (maximize — that plot alone fills the whole strip; press again to restore the grid).

The plots **stay live while you wheel-zoom**: zooming the time axis just widens or narrows the follow window anchored at the live edge, and zooming the value axis pins your chosen Y range while time keeps rolling. Only deliberate trips into history — panning, or dragging a rubber-band zoom box — freeze the view; **Fit** or a double-click brings it back to live.

Troubleshooting

Symptom	Likely cause
Everything shows `--`	No input bound, or the stream hasn't produced samples yet (stream not started)
Input selector is empty	The firmware defines no rotor streams — the monitor only lists `TRAX_STREAM_ROTOR_DEFINE` inputs, not plain variables or ADC groups
RPM wildly wrong	`counts_per_rev` in `TRAX_STREAM_ROTOR_DEFINE` doesn't match the actual encoder resolution (remember: post-quadrature counts)
Speed spikes at each rotation	The position samples don't wrap at `counts_per_rev` — the value must stay within `[0 .. counts_per_rev)`