OpsPilot

Meter Reading & Condition-Based Triggers — User Manual

Turning Measurements Into Action · ISO 17359 · AI Engineering Co-Pilot

What this guide covers — what condition-based triggers are, how the OpsPilot module designs the program, what to have ready, and the output you receive.

1. What are condition-based triggers?

Condition-based triggers are the discipline that turns a measurement into an action. Collecting readings — vibration, temperature, oil condition, pressure — is pointless unless each reading has a defined threshold and a pre-agreed response when it's crossed. The failure mode is the data graveyard: trends collected faithfully and watched by nobody, because no one decided in advance what number means “do something” and who does it.

OpsPilot designs the program to *ISO 17359 (the CBM framework), ISO 13379/13374 (data interpretation and processing), ISO 18436 (analyst qualifications) and the measurement standards (ISO 20816, ISO 4406, API RP 670)*. It is the threshold-monitoring counterpart to the pattern-based Anomaly Detection module.

2. What the OpsPilot module does

Role	Responsibility
🎯 **AI Coach — Senior Reliability Engineer (OpsPilot)**	Designs the program along the ISO 17359 path — identify equipment, determine criticality, identify failure modes, select techniques, establish baselines, set three-tier thresholds (Alert / Alarm / Trip), and — critically — define the response procedure with a named owner for each.
👷 **Reliability Engineer / Maintenance Manager (you)**	Provide the equipment scope, the criticality, the failure-mode inputs (from FMEA or RCM) and the operating context — and own the day-to-day execution.

3. How it works — the ISO 17359 path

#	Step
1	Identify equipment and determine criticality
2	Identify the failure modes to monitor (from FMEA / RCM)

#	Step
3	Select the technique per failure mode — vibration, oil, thermography, ultrasonics, MCSA, performance trending
4	Establish baselines — what “normal” looks like
5	Set three-tier thresholds — Alert / Alarm / Trip
6	Define the response procedure for each threshold — with a named role owner

4. What you will be asked — have this ready

- The equipment scope and criticality.
- The failure modes to monitor — ideally from your FMEA or RCM output.
- The operating context and any existing baseline data.
- Who will own the response when each threshold is crossed.

5. What you receive — the output

A complete condition-monitoring program design (Word): the technique selected per failure mode, the baselines, the three-tier Alert / Alarm / Trip thresholds, and the pre-defined response procedure for each threshold with a named owner — so a crossed threshold leads to an action, not a shrug.

6. Worked example (illustrative)

A critical pump bearing. The failure mode (from the FMEA) is bearing degradation; the technique is vibration trending. The baseline is established at 2.5 mm/s RMS when healthy. Three tiers are set: Alert at 4.5 mm/s (investigate at next round), Alarm at 7.1 mm/s (plan intervention within the week, named owner: the reliability engineer), Trip at 11 mm/s (stop and inspect now, named owner: the shift supervisor). The crucial part isn't the numbers — it's that each tier has a pre-agreed action and a person, so when the reading hits 7.1 the response is automatic and owned, not a debate about whether it matters. That's what turns a measurement into protection.

7. Getting the best result

- **Every threshold needs a response.** A number with no pre-agreed action is just data.
- **Name an owner per tier.** “Someone should look at it” means nobody does.
- **Drive technique from failure mode.** Monitor the way the asset actually fails — use your FMEA/RCM.
- **Establish the baseline first.** You can't set a meaningful threshold without knowing normal.