

10 — Logging

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

Copyright © 2025–2026 Embedya. All rights reserved. · <https://embedya.com>

Binary, deferred-format logging: the firmware sends a TID plus raw argument values; Traxcope reconstructs the full text using the format-string metadata — transmitted once by the firmware at session start (`TRAX_META_IN_FLASH`, default) or read from the `.elf` file set in the probe configuration (`TRAX_META_ELF_ONLY`); see [Chapter 02 §4](#). A log call costs microseconds and a dozen bytes on the wire — cheap enough to leave in tight loops and ISRs.

On the device

Basic logging

```
#include <trax.h>

/* TID registry (trax_log_tid.h) - auto-increment, no manual values */
enum trax_log_tid_t {
    LOG_TID_BOOT = TRAX_TID_RANGE_LOG_USER_START,
    LOG_TID_TEMP,
    LOG_TID_MOTOR_FAULT,
};

/* Call sites - signature: (tid, tag, format, args...) */
TRAX_LOG_INFO (LOG_TID_BOOT,          "boot",          "System up, fw %d.%d",
major, minor);
TRAX_LOG_DEBUG (LOG_TID_TEMP,         "sensors",        "Temperature: %.1f C",
TRAX_32F(temp_c));
TRAX_LOG_ERROR (LOG_TID_MOTOR_FAULT,  "motor_ctrl",    "Motor failed: 0x%08x",
err);
```

No separate "define" step is needed — the format string, severity level, and tag are captured into ELF metadata automatically at the call site. Each TID identifies one log call site/message.

Severity levels

Nine syslog-style macros, most to least critical:

TRAX_LOG_EMERGENCY , TRAX_LOG_ALERT , TRAX_LOG_CRITICAL , TRAX_LOG_ERROR ,
TRAX_LOG_WARNING , TRAX_LOG_NOTICE , TRAX_LOG_INFO , TRAX_LOG_DEBUG , TRAX_LOG_TRACE

(TRAX_LOG(. . .) is shorthand for INFO.)

Compile-time filtering: logs above TRAX_LOG_COMPILE_LEVEL (global) or TRAX_FILE_LOG_LEVEL (per-file override, define before including trax.h) are compiled out completely — zero code, zero metadata. These two macros have **no** TRAX_CFG_ **prefix**.

Tags

The tag argument ("boot", "motor_ctrl", ...) is a free filter string shown in the Log View's context column. Use consistent tags per subsystem — they drive the per-tag filtering in Traxcope. The library itself emits diagnostics under the tag diag .

Argument types

Arguments are packed as 32-bit words. Integers up to 32 bits pass as-is; for wider/floating types use the packing helpers:

Type	How to pass
int , uint32_t , pointers, chars	directly
float	TRAX_32F(value)
double	TRAX_64F_HI(value), TRAX_64F_LO(value) (two consecutive args)
int64_t / uint64_t	TRAX_64_HI(value), TRAX_64_LO(value)

All standard printf float conversions are supported on the Traxcope side — fixed (%f , %.2f), scientific (%e , %E), and shortest (%g , %G) — the firmware only ships the raw bits via TRAX_32F / TRAX_64F_* .

Format specifiers

Formatting happens entirely in Traxcope, driven by the format string in the metadata. Standard C flags, width, and precision are honored (e.g. %-08.3f); default float precision is 6. The length modifier decides how many 32-bit wire words the argument consumes: none or l → one word, ll → two words (high word first, then low — TRAX_64_HI/LO).

One wire word:

Specifier	Renders as	Pass on the MCU
%d / %i	int32_t decimal	directly
%u	uint32_t decimal	directly
%o	uint32_t octal	directly
%x / %X	uint32_t hex, lower/upper case	directly
%p	pointer — 0x + 8-digit upper hex	directly
%c	character (value = code point)	directly
%b	uint32_t binary — Traxcope extension	directly
%f / %F	float fixed-point	TRAX_32F(x)
%e / %E	float scientific	TRAX_32F(x)
%g / %G	float shortest of f/e	TRAX_32F(x)
%s	falls back to uint32_t decimal — C strings are not transmitted	—

Two wire words (ll):

Specifier	Renders as	Pass on the MCU
%lld / %lli	int64_t decimal	TRAX_64_HI(x), TRAX_64_LO(x)
%llu	uint64_t decimal	TRAX_64_HI(x), TRAX_64_LO(x)
%llo	uint64_t octal	TRAX_64_HI(x), TRAX_64_LO(x)
%llx / %llX	uint64_t hex, lower/upper case	TRAX_64_HI(x), TRAX_64_LO(x)
%llf / %llF	double fixed-point — Traxcope extension	TRAX_64F_HI(x), TRAX_64F_LO(x)
%lle / %llE	double scientific — Traxcope extension	TRAX_64F_HI(x), TRAX_64F_LO(x)
%llg / %llG	double shortest — Traxcope extension	TRAX_64F_HI(x), TRAX_64F_LO(x)

Notes:

- %llf / %lle / %llg are Traxcope extensions (standard printf spells the double conversions %f / %e / %g) — the ll tells the decoder to reassemble a true IEEE-754 double from the two words shipped by TRAX_64F_HI/LO .

- Substitution is conservative: if the frame carries fewer words than the format string needs, the line is shown with the format string unsubstituted rather than misaligned values.

Hex dumps

Dump binary buffers with a label, at any severity:

```
TRAX_HEX_INFO (LOG_TID_SPI_TX, "spi_driver", tx_buf, 16, "SPI TX frame");
TRAX_HEX_ERROR (LOG_TID_I2C_ERR, "i2c_driver", err_data, 32, "I2C%d error
dump", bus);
```

Traxcope renders these as formatted hex/ASCII blocks attached to the log line.

Log → variable binding (optional)

A log argument can double as a plotted variable. Bind argument positions to VAR TIDs once:

```
/* arg[0] → VAR_TEMP, arg[1] → plain, arg[2] → VAR_SPEED */
TRAX_LOG_DEFINE_VARS (LOG_TID_MOTOR_STATUS, VAR_TID_TEMP, 0, VAR_TID_SPEED);

TRAX_LOG_INFO (LOG_TID_MOTOR_STATUS, "motor", "T=%d err=%d v=%d", temp, err,
speed);
```

One log call now produces both a log line **and** scope samples for `VAR_TID_TEMP` and `VAR_TID_SPEED` — no separate `TRAX_VAR_SET` calls needed. See [Chapter 11](#) for the variable side.

Cost & cautions

- Runtime cost: one atomic frame write — header + timestamp + TID + args (≈12 bytes + 4 per argument). Safe from ISRs.
- The TID, not the format string, is the runtime identity: **one TID per call site**. Reusing a TID for two different format strings is a compile error in the same file (duplicate symbol) and a decode hazard across files — let the enum auto-increment.

In Traxcope — the Log View

Open via Toolbox → **Log**. Multiple instances allowed.

Time	Delta	Probe	Level	Tag	Context	Message
0:00:00:04 001.029.156	+000.001.735	Probe 1	ERROR	PWR	logshow	bus=48.27 V, current=0.0125 A
0:00:00:04 001.030.921	+000.001.765	Probe 1	CRITICAL	RF	logshow	impedance=4.700e+03 ohm, leakage=1.23E-07 F
0:00:00:04 001.032.656	+000.001.735	Probe 1	ALERT	CAL	logshow	gain_min=1.23e-05, gain_max=1.23457E+06
0:00:00:04 001.034.421	+000.001.765	Probe 1	EMERGENCY	FAULT	logshow	fault_pc=0x08001234, stack_top=0x20023FFC
0:00:00:04 001.036.156	+000.001.735	Probe 1	INFO	SYS	logshow	active grid phases=5
0:00:00:04 001.037.750	+000.001.594	Probe 1	ERROR	GEO	logshow	latitude=41.015137000000003 deg
0:00:00:04 001.039.484	+000.001.734	Probe 1	CRITICAL	PHYS	logshow	avogadro=6.022141e+23 1/mol
0:00:00:04 001.041.187	+000.001.703	Probe 1	ALERT	PHYS	logshow	euler e=2.718281828
0:00:00:04 001.043.015	+000.001.828	Probe 1	INFO	FRAME	logshow	[HEX] rx frame, len=16 bytes: AA 55 10 02 34 12 78 56 DE AD BE EF C0 FF EE 00
0:00:00:06 001.020.859	+1 999.977.844	Probe 1	TRACE	PERF	logshow	latency probe — start (no args)
0:00:00:06 001.022.374	+000.001.515	Probe 1	DEBUG	PERF	logshow	latency probe — end (no args)
0:00:00:06 001.023.890	+000.001.516	Probe 1	INFO	REG	logshow	status=0xAB, ctrl=0x1234, base=0x40012800
0:00:00:06 001.025.671	+000.001.781	Probe 1	NOTICE	FLAGS	logshow	status=0b10100101, nibble=0b1010, alarms=0b111
0:00:00:06 001.027.421	+000.001.750	Probe 1	WARNING	ENERGY	logshow	net metered energy=-12345678901 mWh
0:00:00:06 001.029.156	+000.001.735	Probe 1	ERROR	PWR	logshow	bus=48.27 V, current=0.0125 A

Screenshot: Log View, live mode, lines of multiple severities and tags, color-coded levels

Live mode

New lines append continuously and the view follows the tail. Each row shows timestamp, probe, severity, tag, and the fully formatted message (rendered from the metadata).

Filtering

The probe filter popup (toolbar → **All Probes**) gives, per probe:

- **Level chips** — toggle individual severities on/off (or **All**); each chip carries its severity color.
- **Tag chips** — include/exclude specific tags (this is where disciplined firmware-side tags pay off; e.g. show only `motor_ctrl`, or hide the library's `diag` tag).
- **Probe visibility** — which probes' logs this view instance shows (useful in multi-probe setups).

Screenshot: Log View filter popup — per-probe level chips (Emergency ... Trace) and tag chips

Review mode: search & navigation

Switch to Review to navigate history:

- **Text search** scans the entire store (not just loaded rows), with result navigation.

- The navigation rail jumps **Oldest / Older / Newer / Latest** through pages of history.
- A virtual scrollbar covers the full session, even when it is far too large to hold in memory.

Sync with other views

With **Sync** enabled (Review mode):

- Selecting a row broadcasts its timestamp — every synced view (Trace, Scope, ...) jumps to that instant. This is the fastest way to answer *"what was the system doing when this error printed?"*
- Cursor/span set elsewhere is reflected in the Log View, scrolling it to the corresponding lines.

Troubleshooting

Symptom	Likely cause
Logs show raw TIDs / LOG_0x0112 instead of text	No ELF set in probe config, ELF doesn't match firmware (check build ID), or metadata mode mismatch
A log never appears	Compiled out by TRAX_LOG_COMPILE_LEVEL / TRAX_FILE_LOG_LEVEL , or filtered in the view
ERROR lines tagged diag you didn't write	Library diagnostics (ring overflow, frame corruption) — see Chapter 02 §6