MATH / generation 1 — User manual

Library: `src/lib2/math/1/`
Edition: 2026-11-18, source/document review (documentation only)
Qualification: THIN-BETA-1 research holding, with bounded historical owner smoke tests; current modular sources and generic calling ABIs are experimental, not stable/multi-machine releases.

What this member does

Expose 23 PB-2000C resident MathROM methods via thin, tail-jump `MATH_*` adapters; they are not software mathematical implementations.

Public interfaces

Entry point or group	Inputs, outputs and behavior
`MATH_ADD / MATH_SUB / MATH_MUL / MATH_DIV`	Resident arithmetic wrappers; operands/results depend on low-ROM resident ABI.
`MATH_SIN / MATH_COS / MATH_TAN / MATH_ASN / MATH_ACS / MATH_ATN`	Resident trigonometric and inverse-trigonometric wrappers.
`MATH_EXP / MATH_LOG / MATH_LGT / MATH_POWER / MATH_SQR / MATH_SQ`	Resident power, logarithm and square/root wrappers; check per-entry source.
`MATH_PI / MATH_FIX / MATH_INC / MATH_RND12 / MATH_CNVR / MATH_B2BCD / MATH_CMP`	Resident constant, conversion, rounding and compare wrappers.

How to integrate

Select exactly the `math_<name>.inc` needed; for example `#INCLUDE (lib2/math/1/math_add.inc)` and `CAL MATH_ADD`, after establishing the exact resident-ABI operands and environment described in that module. A wrapper tail-jumps into firmware; it does not prepare numerical state.

Composition sketch (not an independently assembled client):

```
#INCLUDE (lib2/math/1/math_acs.inc)
; Supply the dependencies, RAM, bank context and symbols described above.
        CAL     MATH_ADD
```

Environment, dependencies and memory

PB-2000C low-ROM addresses and correct runtime numeric state, UA/banks/stack/firmware tables. Expanded B9 register operands and return flags vary by method; individual module header is the first authority.

Limits and evidence

No AI-1000 qualification. Clobbers, domain and full ABI still open. Do not substitute this for software `mathsoft` or claim all methods share one uniform operand convention.

This documentation does not introduce a new build or runtime PASS. Source equivalence, native assembler parity, emulator execution and physical-machine evidence remain distinct.

Actual source and further reference

- This exact member folder in Google Drive
- Source modules: `math_acs.inc`, `math_add.inc`, `math_asn.inc`, `math_atn.inc`, `math_b2bcd.inc`, `math_cmp.inc`, `math_cnvr.inc`, `math_cos.inc`, `math_div.inc`, `math_exp.inc`, `math_fix.inc`, `math_inc.inc`, `math_lgt.inc`, `math_log.inc`, `math_mul.inc`, `math_pi.inc`, `math_power.inc`, `math_rnd12.inc`, `math_sin.inc`, `math_sq.inc`, `math_sqr.inc`, `math_sub.inc`, `math_tan.inc`.
- Further contracts/source notes: API.md, API.json, README.md. These may include dated donor paths or older acceptance statements.
- For a new consumer, pin exact source files and hashes, verify full include closure, inspect register/bank/resource conflicts and test the resulting application on each claimed target.

Praeteritum ad futurum servamus.