MEMORY / generation 1 — User manual

Library: `lib-set-01/src/lib2/memory/1/` · Member source folder

Status: SOURCE-ADMITTED / EXPERIMENTAL — 23 November 2026. This generation is accepted as exact source from Chapter 08 R012C, within the donor's bounded owner-built and owner-reported functional scope. This does **not** certify a generic caller-safe ABI, independent second client, exhaustive PB-2000C/AI-1000 target parity or physical-machine behavior.

Admission receipt and source SHA-256 ledger · Chapter 08 donor.

What this member is — and is not

Three small, display-independent **address/byte utilities extracted from the limon monitor**. The caller owns its display, command policy, interpretation of memory spaces and hardware safety. **This is not a 32/64/96/128 KiB memory-capacity detector, non-destructive RAM-probe guarantee, bank-alias tester or memory allocator.** It cannot distinguish valid RAM from ROM, an aliased bank or an open bus by a successful read alone.

Four exact donor files: `mem_read.s`, `mem_addr.s`, `mem_hex.s`, `memory.inc`; all byte- and SHA-matched to Chapter 08 R012C. There is no internal `ORG` or private RAM workspace.

Public entry contracts

Call	Inputs	Outputs / clobbers
`MEM_READ8`	`$0` = raw UA selection; `$1:$2` = little-endian address	`$15` = one raw byte; modifies `$2`, `$3`, IZ, flags; restores the original UA and interrupt-enable state. No RAM writes.
`MEM_HEX4`	Low nibble of `$16`	ASCII uppercase hexadecimal character in `$16`; changes flags only otherwise.
`MEM_BANKUA`	`$0` = **already validated** ASCII bank `'0'..'3'`	Corresponding monitor UA value in `$0`; changes flags, not the physical bank or installed-capacity state.

Caller safety and system context

`MEM_READ8` requires valid stack and established `$SX`/index conventions. Its one-byte fetch is inline while the UA context is temporarily changed, with **no JP/CAL/RTN inside that critical interval**. This matters because HD61700 PC-bank changes are delayed until a control transfer. The caller must pass a meaningful accessible memory map and avoid unmapped, protected and peripheral addresses. In the language-card environment, an attempted ROM1 address may resolve to ROM2 under an overlay. **Do not use this as a physical detection or destructive probing routine.**

`MEM_BANKUA` encodes limon's historical monitor bank-digit convention. Its digits are not established physical CS0–CS3 bank identities or proof of installed memory. Higher-level validation, error handling and target-specific mapping belong to the consumer.

Minimal source-level example

```
        LD      $0,&H54       ; known, permitted RAM1 context
        LDW     $1,&H3000     ; little-endian address pair
        CAL     MEM_READ8
        ; $15 now contains one observed byte
        ; $2/$3, IZ and condition flags are clobbered
```

This is an illustrative call, **not** permission to read unknown physical systems or an independently assembled second client. Code that needs a normalized hex display may pass a nibble through `MEM_HEX4`; a disassembler would provide its own instruction decoder and UI.

Donor evidence and remaining work

R012C's accepted owner native HD61 build contains the three primitives in its included source closure, with a reported working ROM. Earlier limon targeted tests are donor Experience, not independent consumer proof. **A separate disassembler, negative UA/IE/stack cases, bank-alias handling and hardware target matrix are not accepted.** Source admission permits controlled investigation and integration; any new caller must qualify its own bank/context, register-liveness and safety limits.

Praeteritum ad futurum servamus.