The Hruschka & Wahl 512 KB RAM disk: what the patched PX-8 OS ROM changes

For Fred Jan Kraan, from Yannis Doukakis. 28 October 2026.
Source: `ramdisk.tgz` from your page electrickery.nl/comp/px8/512kByteRAMDisk.html (files `px-8-org.epr`, `px060688.epr`, `pxdiff`, `tt.txt`).

Hi Fred,

Here is what I found when I compared the two ROM images in your file set. The short answer: the patch changes the disk parameters (DPBs) of drives A: and I:, replaces Epson's RAM disk drivers with a new driver for the Hruschka/Wahl board, and adds a Centronics printer driver. Below are the details, with the original and patched code side by side.

1. Summary

- **The original image is M25030CB.** `px-8-org.epr` is byte-identical to the M25030CB OS ROM I have been disassembling (SHA-1 `ac929f50…`), so every change could be read against a fully annotated listing.
- **`pxdiff` is incomplete.** It lists 10 differences only because DOS `COMP` gives up after 10. The real difference is **597 bytes in 15 places** (table in section 2).
- **Two drives of up to 256 KB each.** A: is the upper bank (92h = 4–7), I: the lower one (92h = 0–3). Each is filled in 32 KB steps (up to 8 × 32 KB SRAMs per drive, 16 chips in total, as on the board photo).
- **New DPBs.** Drive A: keeps 1 KB blocks, with DSM up to 255 and 128 directory entries. Drive I: changes to 2 KB blocks, with DSM up to 127 and 64 directory entries. Both sizes are measured at the first reset and stored in a hidden directory entry on each drive.
- **New driver.** The board has no auto-incrementing low-address register. During `IN r,(C)` / `OUT (C),r` the Z80 puts C (the port number, 90h) on CPU address lines A0–A7 and register B on CPU address lines A8–A15. The board decodes 90h as its data port and feeds CPU A8–A15 to the RAM chips' address pins A0–A7. So B is the low byte of the RAM address, and `INC B` steps through a 256-byte page.
- **Removed support.** Epson's intelligent and non-intelligent RAM disk drivers are gone, and so is the old read-only drive I: (the expansion unit's ROM capsule). This matches section 5 of `tt.txt`.
- **Centronics printer.** The default IOBYTE changes from A9h to E9h (LST: = UL1:). LIST and LIST status for UL1: go to port 93h, and CTRL/PF5 screen dump works with it.

2. The changed bytes

Ranges are merged where fewer than three equal bytes separate two changes. Addresses are ROM addresses (bank 0, 0000–7FFF).

ROM range	Length	Bytes changed	What it is
02FB–0332	56	55	Reset: detect the new unit (93h), flag F6A8 = E5h
0398–039D	6	5	Reset: call the new size/format routine instead of DSM = F6A8 - 1
1C28–1C2D	6	5	Drive I: DPB image (copied to F310): 2 KB blocks, 64 entries
2C96–2C96	1	1	Default IOBYTE A9h -> E9h (LST: = UL1: = Centronics)
2F64–2F70	13	13	SELDSK I: valid if its DSM (F315) is not 0
2FAF–2FDC	46	42	Free space after detect_drive_i reused: printer status/output, screen dump test
2FF5–2FF6	2	2	Drive I write: error routine -> new WRITE 329D
2FFF–3000	2	2	Drive I read: ROM-capsule read -> new READ 329C
301A–3026	13	11	Drive A read/write: new unit -> new driver 32A1
30A5–30B1	13	11	Drive A boot check: new unit -> new size/format routine 31A1
31A1–330A	362	360	Epson intelligent + non-intelligent drivers replaced by the new driver
348D–34DD	81	77	Old drive I read code replaced by the marker entry template (+ zeros)
36BE–36C2	5	5	LIST output: UL1 -> Centronics routine 2FC2
36F7–36FB	5	5	Screen dump (CTRL/PF5): also allowed with LST = UL1
4183–4185	3	3	LIST status: UL1 -> Centronics status 2FAF
15 ranges		**597**	

3. The hardware as the software sees it

Port	Write	Read
90h	RAM data	RAM data
91h	RAM address A8–A15	–
92h	RAM address A16–A18 (bit 2 = 1: drive A:, 0: drive I:)	–
93h	Centronics data	Status: D7 BUSY, D6 paper empty, D5–D2 = 0 when the board is fitted, D1 write protect A:, D0 write protect I:

RAM address A0–A7. Every RAM access is `IN A,(C)` or `OUT (C),A` with C = 90h. During these instructions the CPU address bus carries C on A0–A7, which selects the port (90h), and B on A8–A15. An ordinary I/O device ignores A8–A15, but this board apparently passes them to the RAM chips as their address A0–A7. Ports 91h and 92h supply the higher RAM address bits A8–A18. The driver counts B up after each byte (`INC B`) instead of relying on an auto-increment register. This is the "driver code to compensate for the missing auto-increment" mentioned on your page.

Detection. Everywhere the ROM used to test port 94h (Epson non-intelligent unit) or 81h (intelligent unit), it now tests `IN A,(93h) / AND 3Ch`. Zero means the board is present. Without a board I would expect the port to read FFh; I haven't checked this on hardware.

Memory layout. A drive address is `(track | E) × 8 KB + sector × 128`, where E = 00h for I: and 20h for A:, with track < 32 and sector < 64. So 92h bits 0–1 = track bits 3–4, 92h bit 2 = drive A:, 91h = track bits 0–2 and sector bits 1–5, and B bit 7 = sector bit 0.

The ports and signal names agree with the port list in `tt.txt` (OCR, partly garbled there).

4. Disk parameter blocks

	Drive A: original	Drive A: patched	Drive I: original	Drive I: patched
Where	F28C (built in RAM; DSM/DRM set at reset)		F310, copied from ROM 1C26 (patched 1C28–1C2D)	
SPT	64	64	64	64
BSH / BLM	3 / 7 (1 KB)	3 / 7 (1 KB)	3 / 7 (1 KB)	**4 / 15 (2 KB)**
EXM	0	0	0	**1**
DSM	size in KB – 1	**KB – 1, up to 255** (F291)	128	**KB/2 – 1, up to 127** (F315)
DRM	31 (15 below 30 KB)	**127** (F293)	31	**63**
AL0	80h	**F0h** (F295, 4 KB directory)	80h	80h (one 2 KB block)
CKS / OFF	0 / 0	0 / 0	0 / 0	0 / 0

Why 256 KB per drive? This is my interpretation, not stated in the documents. The limit appears to come from the allocation vectors the ROM already reserves:

- Drive A:'s DPH points its ALV at F5D9, and the next DPH field at F5F9. That leaves 32 bytes, so 256 blocks of 1 KB.
- Drive I:'s ALV is ALV8 at FD7F–FD8F (17 bytes), and the BIOS jump table BIOTBL starts right after it at FD90. With 2 KB blocks, 256 KB needs 128 bits = 16 bytes, which fits.

That would explain why only drive I: was switched to 2 KB blocks, and why they split the 512 KB into two drives instead of making one large one.

5. Start-up: sizing and the hidden directory entry

Reset (02FA). If the board is present, the internal RAM disk is set to 0 KB (F009). F6A8, the RAM disk size the system display shows, is set to E5h as an "external disk" flag. Later, instead of computing DSM = F6A8 – 1 (0398), the ROM calls the new routine at 31A1. The display reads F6A8 directly (12FE), so it will show **229** as the RAM disk size. That is probably why `tt.txt` warns that the size in the system display is not the size of drive A:. Changing the RAM disk size with CONFIG would overwrite this flag, which fits the warning that the external disk is then no longer recognised.

Routine 31A1, per drive (I: first, then A:):

1. Compare directory entry 0 of the drive with the template at 348D: user 15, name starting with `RFL0`, then byte 5 = drive letter (`I` / `A`), byte 6 = a size byte, type `SYS`, everything else 0 (so no blocks allocated). In the template, bytes 5 and 6 are placeholders `y` and `x`.
2. If the entry is found, the size byte becomes the drive's DSM, and nothing on the drive is touched. This is why the drive can be write-protected after the first initialisation: from then on, a reset only reads it.
3. If it isn't found, the routine:
   - writes the template as entry 0 and fills the rest of the directory with E5h (I: 2 KB, A: 4 KB);
   - measures the RAM by reading the first byte of each 32 KB step, writing its complement, reading back, and restoring it, stopping at the first failure;
   - writes the drive letter and the DSM into bytes 5 and 6 of the entry, and reads the entry back to set the DPB.

   Drive A: also gets DRM = 127 and AL0 = F0h.

This matches section 3 of `tt.txt`: to re-size a drive, go to user 15 and erase that entry, then do an initial reset. Byte 6 of the name is a binary size value (for example FFh for a full 256 KB drive A:), so the name is not printable. My guess is that a wildcard such as `ERA RFL0*.SYS` in user 15 is needed; I haven't tried it.

6. The new READ / WRITE (329C / 329D)

The entry trick is the same as Epson's own at 3014: `LD A,0AFh` at 329C stores AFh (read), and entering at 329D executes `XOR A` (write). Drive A: comes in at 32A1 from the patched check at 301D. Drive I: comes in through the READ/WRITE tables at 2FF4/2FFE. It used to be read-only: writes went to the "R/O" error at 3316.

- **Range check:** track ≥ 32 or sector ≥ 64 returns an error.
- **Read:** 128 × (`IN A,(C)`, store, `INC B`).
- **Write:** each byte is written with `OUT (C),A`, then the driver reads 93h (value discarded), then reads the RAM byte back and compares. A mismatch returns error A = 1.
- **Success** exits through the old common exit at 3070 (BIOSERROR = 0, A = 0).

> I did not find any code that reads the write-protect bits D0/D1 of port 93h. My reading is that a protected bank simply doesn't store the byte, the read-back fails, and CP/M reports a BDOS error. The dummy read of 93h between write and read-back is probably there so the read-back can't see the value just written still floating on the data bus; that is a guess.

7. Centronics printer

- Initial reset sets IOBYTE = E9h instead of A9h, so LST: = UL1: (bits 7–6 = 11). This is why `tt.txt` says the parallel port is the default printer after an initial reset and can only be selected again with `STAT`, not CONFIG.
- **LIST status** (4183 → 2FAF): for UL1:, ready = 93h bit 7 (BUSY) low.
- **LIST** (36BE → 2FC2): for UL1:, wait while BUSY, then `OUT (93h),C`. The strobe must come from the board's hardware, since no separate strobe output is written.
- **Screen dump** CTRL/PF5 (36F7 → 2FD1): the original refused any LST: with bit 6 set (UL1: and the 11 setting); now only the 01 setting is refused.

The printer code sits in the bytes freed at 2FAF–2FDA, where the old drive I: ROM-capsule detection used to be.

8. Other observations

- The space freed by the removed drivers is not all used: 31A1–330A holds the new driver with a few NOPs, and 3499–34DD is zero-filled.
- The size probe tests only one byte per 32 KB step, so it assumes whole 32 KB chips, as the read.me says ("the software expects 32 KB steps"). 8 KB chips would need a different probe.
- I haven't traced what happens with no RAM at all in the drive A: bank. For drive I:, an empty bank gives DSM = 0, and SELDSK then refuses drive I: (2F64).

Appendix A. Original and patched code at the small patch sites

The left side is from my annotated M25030CB listing (label names are mine). The right side is the patched image, disassembled with z80dasm, with my comments.

Reset: RAM disk detection (02FA)

Original (M25030CB)

```
02fa  DB 94        IN A,(RD_CMD_STATUS)
02fc  CB 7F        BIT 0x7,A
02fe  20 0A        JR NZ,LAB_ram_030a
0300  06 40        LD B,0x40
0302  CB 57        BIT 0x2,A
0304  28 02        JR Z,LAB_ram_0308
0306  06 80        LD B,0x80
0308  18 1F        JR LAB_ram_0329
030a  DB 81        IN A,(IRD_STATUS)
030c  CB 7F        BIT 0x7,A
030e  20 23        JR NZ,LAB_ram_0333
0310  DB 81        IN A,(IRD_STATUS)
0312  CB 4F        BIT 0x1,A
0314  20 FA        JR NZ,LAB_ram_0310
0316  AF           XOR A
0317  D3 81        OUT (IRD_STATUS),A
0319  06 3C        LD B,0x3c
031b  DB 81        IN A,(IRD_STATUS)
031d  CB 47        BIT 0x0,A
031f  28 F8        JR Z,LAB_ram_0319
0321  DB 80        IN A,(IRD_DATA)
0323  CB 47        BIT 0x0,A
0325  28 02        JR Z,LAB_ram_0329
0327  06 78        LD B,0x78
0329  AF           XOR A
032a  32 09 F0     LD (ramdisk_kb),A
032d  78           LD A,B
032e  32 A8 F6     LD (ramdisk_kb_total),A
0331  18 06        JR LAB_ram_0339
0333  3A 09 F0     LD A,(ramdisk_kb)
0336  32 A8 F6     LD (ramdisk_kb_total),A
```

Patched (PX060688)

```
02FA  DB 93        in a,(93h)          ; 93h: D5–D2 read 0 when the board is fitted
02FC  E6 3C        and 3Ch
02FE  20 33        jr nz,333h          ; no board -> old path at 0333
0300  AF           xor a
0301  32 09 F0     ld (F009h),a        ; internal RAM disk = 0 KB
0304  3E E5        ld a,E5h            ; F6A8 = E5h: "external disk" flag
0306  32 A8 F6     ld (F6A8h),a
0309  18 2E        jr 339h             ; -> 0339 (module addresses)
030B  3A A8 F6     ld a,(F6A8h)        ; entered from 0398 (patch below)
030E  FE E5        cp E5h
0310  28 06        jr z,318h           ; flag set -> 0318
0312  26 00        ld h,00h            ; no flag: same as the old 0398–039D
0314  6F           ld l,a
0315  C3 9E 03     jp 039Eh
0318  CD A1 31     call 31A1h          ; size + format both drives, set DPBs
031B  C3 AE 03     jp 03AEh            ; skip the old DSM/DRM setup
031E–0332  00 x 21     nop (free)
0333  3A 09 F0     ld a,(F009h)
0336  32 A8 F6     ld (F6A8h),a
```

Reset: drive A DSM setup (0398)

Original (M25030CB)

```
0398  2A A8 F6     LD HL,(ramdisk_kb_total)
039b  26 00        LD H,0x0
039d  7D           LD A,L
039e  B7           OR A
039f  28 01        JR Z,LAB_ram_03a2
03a1  2B           DEC HL
03a2  22 91 F2     LD (BYTE_ram_f291),HL
```

Patched (PX060688)

```
0398  C3 0B 03     jp 030Bh            ; -> 030B
039B-039D  00 x 3      nop (free)
039E  B7           or a
039F  28 01        jr z,03A2h
03A1  2B           dec hl
03A2  22 91 F2     ld (F291h),hl
```

SELDSK drive I (2F61)

Original (M25030CB)

```
2f61  32 43 F5     LD (bios_phys_drive),A
2f64  CB 43        BIT 0x0,E
2f66  CC AF 2F     CALL Z,detect_drive_i
2f69  21 00 00     LD HL,0x0
2f6c  D8           RET C
2f6d  21 00 F3     LD HL,DPH_DRIVE_I
2f70  C9           RET
```

Patched (PX060688)

```
2F61  32 43 F5     ld (F543h),a
2F64  3A 15 F3     ld a,(F315h)        ; F315 = DSM of drive I (set at reset)
2F67  21 00 00     ld hl,0000h
2F6A  B7           or a
2F6B  C8           ret z               ; DSM 0 -> HL = 0: select error
2F6C  21 00 F3     ld hl,F300h         ; DPH of drive I
2F6F  C9           ret
2F70  00           nop
```

Free space after SELDSK reused (2FAF)

Original (M25030CB)

```
2faf  AF           XOR A
2fb0  32 6C F3     LD (drvi_capacity),A
2fb3  3D           DEC A
2fb4  32 6B F3     LD (drvi_dir_entries),A
2fb7  DB 94        IN A,(RD_CMD_STATUS)
2fb9  17           RLA
2fba  D8           RET C
2fbb  AF           XOR A
2fbc  0E 02        LD C,0x2
2fbe  57           LD D,A
2fbf  5F           LD E,A
2fc0  CD 8D 34     CALL nramdisk_read_byte
2fc3  FE E5        CP 0xe5
2fc5  20 E5        JR NZ,LAB_ram_2fac
2fc7  DB 93        IN A,(RD_DATA)
2fc9  FE 37        CP 0x37
2fcb  20 DF        JR NZ,LAB_ram_2fac
2fcd  DB 93        IN A,(RD_DATA)
2fcf  32 6C F3     LD (drvi_capacity),A
2fd2  1E 16        LD E,0x16
2fd4  CD 8D 34     CALL nramdisk_read_byte
2fd7  32 6B F3     LD (drvi_dir_entries),A
2fda  AF           XOR A
2fdb  18 CF        JR LAB_ram_2fac
```

Patched (PX060688)

```
2FAF  FE C0        cp C0h              ; LIST status, entered from 4183
2FB1  CA B7 2F     jp z,2FB7h          ; C0h = LST: is UL1:
2FB4  3E FF        ld a,FFh            ; other devices: as the old 4183
2FB6  C9           ret
2FB7  DB 93        in a,(93h)          ; D7 = BUSY
2FB9  E6 80        and 80h
2FBB  20 03        jr nz,2FC0h
2FBD  3E FF        ld a,FFh            ; not busy: FF = ready
2FBF  C9           ret
2FC0  AF           xor a               ; busy: 0
2FC1  C9           ret
2FC2  FE 40        cp 40h              ; LIST output, entered from 36BE
2FC4  CA AE 36     jp z,36AEh          ; as the old 36BE-36C0
2FC7  DB 93        in a,(93h)          ; wait while BUSY
2FC9  E6 80        and 80h
2FCB  20 FA        jr nz,2FC7h
2FCD  79           ld a,c
2FCE  D3 93        out (93h),a         ; character to the Centronics latch
2FD0  C9           ret
2FD1  E6 C0        and C0h             ; screen dump test, entered from 36F7
2FD3  FE 40        cp 40h              ; only 40h (UL1 bits = 01) is refused now
2FD5  CA 2C 38     jp z,382Ch
2FD8  C3 FC 36     jp 36FCh            ; go on with the dump
2FDB  00           nop
2FDC  00           nop
```

READ / WRITE dispatch for drive I (2FF1)

Original (M25030CB)

```
2ff1  21 FC F1     LD HL,DISK_WRITE_TABLE
2ff4  11 16 33     LD DE,bios_err_readonly
2ff7  18 08        JR LAB_ram_3001
2ff9  3E 01        LD A,0x1
2ffb  21 EC F1     LD HL,DISK_READ_TABLE
2ffe  11 99 34     LD DE,drive_i_read
3001  32 36 F5     LD (BIOSERROR),A
```

Patched (PX060688)

```
2FF1  21 FC F1     ld hl,F1FCh
2FF4  11 9D 32     ld de,329Dh         ; drive I write -> 329D (was: R/O error 3316)
2FF7  18 08        jr 301h
2FF9  3E 01        ld a,01h
2FFB  21 EC F1     ld hl,F1ECh
2FFE  11 9C 32     ld de,329Ch         ; drive I read -> 329C (was: 3499)
3001  32 36 F5     ld (F536h),a
```

Drive A READ / WRITE (3014)

Original (M25030CB)

```
3014  3E AF        LD A,0xaf
3016  32 A3 F6     LD (ramdisk_rw_flag),A
3019  DB 81        IN A,(IRD_STATUS)
301b  CB 7F        BIT 0x7,A
301d  CA 2A 32     JP Z,iramdisk_rw
3020  DB 94        IN A,(RD_CMD_STATUS)
3022  CB 7F        BIT 0x7,A
3024  CA 8E 32     JP Z,nramdisk_rw
3027  3A 44 F5     LD A,(bios_track)
```

Patched (PX060688)

```
3014  3E AF        ld a,AFh
3016  32 A3 F6     ld (F6A3h),a
3019  DB 93        in a,(93h)          ; new unit present?
301B  E6 3C        and 3Ch
301D  CA A1 32     jp z,32A1h          ; yes -> new driver
3020  18 05        jr 327h             ; no -> internal RAM disk as before (3027)
3022-3026  00 x 5      nop (free)
3027  3A 44 F5     ld a,(F544h)
```

Drive A boot check (30A4)

Original (M25030CB)

```
30a4  DB 81        IN A,(IRD_STATUS)
30a6  CB 7F        BIT 0x7,A
30a8  CA B4 31     JP Z,iramdisk_boot_check
30ab  DB 94        IN A,(RD_CMD_STATUS)
30ad  CB 7F        BIT 0x7,A
30af  CA E3 32     JP Z,nramdisk_boot_check
30b2  3A 09 F0     LD A,(ramdisk_kb)
```

Patched (PX060688)

```
30A4  DB 93        in a,(93h)
30A6  E6 3C        and 3Ch
30A8  CA A1 31     jp z,31A1h          ; new unit -> size/format 31A1
30AB  18 05        jr 30B2h            ; no -> internal RAM disk as before (30B2)
30AD-30B1  00 x 5      nop (free)
30B2  3A 09 F0     ld a,(F009h)
```

LIST output (36BE)

Original (M25030CB)

```
36b9  FE 80        CP 0x80
36bb  CA A8 41     JP Z,list_out
36be  FE 40        CP 0x40
36c0  28 EC        JR Z,LAB_ram_36ae
36c2  C9           RET
```

Patched (PX060688)

```
36B9  FE 80        cp 80h
36BB  CA A8 41     jp z,41A8h
36BE  C3 C2 2F     jp 2FC2h            ; -> 2FC2
36C1  00           nop
36C2  00           nop
```

Screen dump test (36F4)

Original (M25030CB)

```
36f4  3A 34 F5     LD A,(iobyte_copy)
36f7  CB 77        BIT 0x6,A
36f9  C2 2C 38     JP NZ,LAB_ram_382c
```

Patched (PX060688)

```
36F4  3A 34 F5     ld a,(F534h)
36F7  C3 D1 2F     jp 2FD1h            ; -> 2FD1
36FA  00           nop
36FB  00           nop
```

LIST status (417F)

Original (M25030CB)

```
417f  FE 80        CP 0x80
4181  28 03        JR Z,LAB_ram_4186
4183  3E FF        LD A,0xff
4185  C9           RET
```

Patched (PX060688)

```
417F  FE 80        cp 80h
4181  28 03        jr z,4186h
4183  C3 AF 2F     jp 2FAFh            ; -> 2FAF
```

Default IOBYTE (2C95)

Original (M25030CB)

```
2c95  3E A9        LD A,0xa9
2c97  32 34 F5     LD (iobyte_copy),A
```

Patched (PX060688)

```
2C95  3E E9        ld a,E9h            ; E9h: LST bits 7-6 = 11 = UL1:
2C97  32 34 F5     ld (F534h),a
```

Appendix B. The new driver (31A1–3304) and the marker entry template (348D)

This replaces Epson's intelligent RAM disk driver (31A1–328D) and non-intelligent driver (328E–330A), and the drive I: ROM-capsule read routine (348D–34DD).

```
31A1  0E 90        ld c,90h            ; C = 90h (CPU A0-A7: port); B (CPU A8-A15) = RAM A0-A7
31A3  CD 4E 32     call 324Eh          ; drive I: marker entry present?
31A6  28 1F        jr z,31C7h          ; yes: keep the disk, go to drive A
31A8  26 00        ld h,00h            ; bank 0 (92h = 0)
31AA  1E 08        ld e,08h            ; 8 pages = 2 KB directory
31AC  CD 25 32     call 3225h          ; write marker + E5 directory
31AF  26 00        ld h,00h
31B1  16 04        ld d,04h            ; probe 92h = 0..3
31B3  CD EB 31     call 31EBh          ; size probe -> L = DSM (2 KB blocks)
31B6  AF           xor a               ; address 0 of drive I
31B7  D3 92        out (92h),a
31B9  D3 91        out (91h),a
31BB  06 05        ld b,05h            ; byte 5 of the marker entry:
31BD  3E 49        ld a,49h            ; 'I'
31BF  ED 79        out (c),a
31C1  04           inc b               ; byte 6:
31C2  ED 69        out (c),l           ; size byte (DSM)
31C4  CD 4E 32     call 324Eh          ; read back -> F315
31C7  CD 5C 32     call 325Ch          ; drive A: marker entry present? (-> F291/F293/F295)
31CA  C8           ret z               ; yes: done
31CB  26 04        ld h,04h            ; bank 4 (92h = 4)
31CD  1E 10        ld e,10h            ; 16 pages = 4 KB directory
31CF  CD 25 32     call 3225h
31D2  26 04        ld h,04h            ; probe 92h = 4..7
31D4  16 08        ld d,08h
31D6  CD EB 31     call 31EBh
31D9  3E 04        ld a,04h            ; address 0 of drive A
31DB  D3 92        out (92h),a
31DD  AF           xor a
31DE  D3 91        out (91h),a
31E0  06 05        ld b,05h
31E2  3E 41        ld a,41h            ; 'A'
31E4  ED 79        out (c),a
31E6  04           inc b
31E7  ED 61        out (c),h           ; size byte = DSM (1 KB blocks)
31E9  18 71        jr 325Ch            ; read back, set drive A DPB
31EB  2E 00        ld l,00h            ; SIZE PROBE: H = first 92h value, D = end
31ED  06 00        ld b,00h
31EF  7C           ld a,h              ; address H:L:00
31F0  D3 92        out (92h),a
31F2  7D           ld a,l
31F3  D3 91        out (91h),a
31F5  ED 78        in a,(c)            ; read the byte
31F7  5F           ld e,a
31F8  2F           cpl                 ; write its complement
31F9  ED 79        out (c),a
31FB  F5           push af
31FC  DB 93        in a,(93h)          ; dummy read of 93h (bus discharge?)
31FE  F1           pop af
31FF  D5           push de
3200  ED 58        in e,(c)            ; read back
3202  BB           cp e
3203  D1           pop de
3204  ED 59        out (c),e           ; restore the original byte
3206  20 0A        jr nz,3212h         ; no RAM here: stop
3208  D5           push de
3209  11 80 00     ld de,0080h         ; next 32 KB (91h bit 7 = A15)
```

```
320C  19          add hl,de
320D  D1          pop de
320E  7A          ld a,d
320F  BC          cp h
3210  20 DD       jr nz,31EFh
3212  00          nop
3213  00          nop
3214  CB 3C       srl h              ; HL / 4 -> L = KB (n x 32)
3216  CB 1D       rr l
3218  CB 3C       srl h
321A  CB 1D       rr l
321C  7D          ld a,l
321D  B4          or h
321E  C8          ret z              ; drive I without RAM: Z
321F  65          ld h,l             ; H = KB - 1   (DSM, 1 KB blocks)
3220  25          dec h
3221  2D          dec l              ; L = KB/2 - 1 (DSM, 2 KB blocks)
3222  CB 3D       srl l
3224  C9          ret
3225  2E 00       ld l,00h           ; FORMAT: H = 92h, E = end page of directory
3227  06 00       ld b,00h
3229  7C          ld a,h
322A  D3 92       out (92h),a
322C  16 20       ld d,20h
322E  7D          ld a,l
322F  D3 91       out (91h),a
3231  E5          push hl
3232  21 8D 34     ld hl,348Dh        ; marker entry template
3235  7E          ld a,(hl)          ; 32 bytes -> directory entry 0
3236  23          inc hl
3237  ED 79       out (c),a
3239  04          inc b
323A  15          dec d
323B  20 F8       jr nz,3235h
323D  E1          pop hl
323E  16 E5       ld d,E5h           ; rest of the directory = E5h
3240  7D          ld a,l
3241  D3 91       out (91h),a
3243  ED 51       out (c),d
3245  04          inc b
3246  20 F8       jr nz,3240h
3248  2C          inc l
3249  7B          ld a,e
324A  BD          cp l
324B  20 F3       jr nz,3240h
324D  C9          ret
324E  26 00       ld h,00h           ; drive I: find marker (byte 5 = I)
3250  16 49       ld d,49h
3252  CD 74 32     call 3274h
3255  7C          ld a,h
3256  32 15 F3     ld (F315h),a       ; size byte -> DPB drive I DSM
3259  AF          xor a              ; Z = found
325A  B5          or l
325B  C9          ret
325C  26 04       ld h,04h           ; drive A: find marker (byte 5 = A)
325E  16 41       ld d,41h
3260  CD 74 32     call 3274h
3263  7C          ld a,h
3264  32 91 F2     ld (F291h),a       ; size byte -> DPB drive A DSM
3267  3E 7F       ld a,7Fh           ; DRM = 127 (128 entries)
3269  32 93 F2     ld (F293h),a
326C  3E F0       ld a,F0h           ; AL0 = F0h (4 directory blocks)
326E  32 95 F2     ld (F295h),a
3271  AF          xor a
3272  B5          or l
3273  C9          ret
3274  06 00       ld b,00h           ; compare entry 0 with the template
3276  7C          ld a,h
3277  D3 92       out (92h),a
3279  3E 00       ld a,00h
327B  D3 91       out (91h),a
327D  21 8D 34     ld hl,348Dh
3280  ED 78       in a,(c)
3282  04          inc b
3283  BE          cp (hl)
3284  23          inc hl
3285  20 10       jr nz,3297h
```

```
3287  3E 05        ld a,05h            ; bytes 0-4: user 15, "RFLO"
3289  B8           cp b
328A  20 F4        jr nz,3280h
328C  ED 78        in a,(c)
328E  04           inc b
328F  BA           cp d                ; byte 5 = drive letter
3290  20 05        jr nz,3297h
3292  ED 60        in h,(c)            ; byte 6 = size -> H, L = 0 (found)
3294  2E 00        ld l,00h
3296  C9          ret
3297  2E FF        ld l,FFh            ; not found: L = FF, H = 0
3299  26 00        ld h,00h
329B  C9          ret
329C  3E AF        ld a,AFh            ; READ entry (329D = WRITE: XOR A)
329E  32 A3 F6     ld (F6A3h),a
32A1  3A 43 F5     ld a,(F543h)        ; entry from 301D (drive A)
32A4  1E 00        ld e,00h
32A6  FE 08        cp 08h              ; physical drive 8 = I:
32A8  28 02        jr z,32ACh
32AA  1E 20        ld e,20h            ; others (A:): 20h = upper 256 KB
32AC  3A 44 F5     ld a,(F544h)
32AF  FE 20        cp 20h              ; track < 32
32B1  30 4F        jr nc,332h
32B3  B3           or e
32B4  67           ld h,a
32B5  3A 46 F5     ld a,(F546h)
32B8  FE 40        cp 40h              ; sector < 64
32BA  30 46        jr nc,332h
32BC  2E 00        ld l,00h            ; address = (track | E) x 8 KB + sector x 128
32BE  CB 3C        srl h
32C0  CB 1D        rr l
32C2  CB 3C        srl h
32C4  CB 1D        rr l
32C6  CB 3C        srl h
32C8  CB 1D        rr l
32CA  06 00        ld b,00h
32CC  CB 3F        srl a
32CE  CB 18        rr b
32D0  B5           or l
32D1  6F           ld l,a
32D2  7C           ld a,h
32D3  D3 92        out (92h),a         ; RAM A16-A18
32D5  7D           ld a,l
32D6  D3 91        out (91h),a         ; RAM A8-A15 (B = RAM A0-A7)
32D8  2A 57 F5     ld hl,(F557h)       ; DMA buffer
32DB  0E 90        ld c,90h
32DD  16 80        ld d,80h
32DF  3A A3 F6     ld a,(F6A3h)
32E2  B7           or a
32E3  28 0A        jr z,32EFh          ; 0 = write
32E5  ED 78        in a,(c)            ; read 128 bytes
32E7  04           inc b
32E8  77           ld (hl),a
32E9  23           inc hl
32EA  15           dec d
32EB  20 F8        jr nz,32E5h
32ED  18 10        jr 32FFh
32EF  7E           ld a,(hl)           ; write 128 bytes
32F0  5F           ld e,a
32F1  23           inc hl
32F2  ED 79        out (c),a           ; write byte
32F4  DB 93        in a,(93h)          ; dummy read of 93h
32F6  ED 78        in a,(c)            ; read back
32F8  BB           cp e
32F9  20 07        jr nz,332h          ; mismatch (write-protected?) -> error
32FB  04           inc b
32FC  15           dec d
32FD  20 F0        jr nz,32EFh
32FF  C3 70 30     jp 3070h            ; BIOSERROR = 0, A = 0
3302  AF           xor a               ; error: A = 1, NZ
3303  3C           inc a
3304  C9           ret

348D  0F 52 46 4C 4F 79 78 20 20 53 59 53   ; marker entry template: user 15, "RFLO" y x "  " "SYS", rest 0
```

Best regards,
Yannis