MCS Alpha1 and BETA 8 Firmware

Oct 25, 2026 · @Hans Otten

The MCS Alpha 1 and BETA 8 are 6502 development systems made by Micronic Computer Systeme in 1978. Its firmware is a 2K monitor called MONA at $F800, a floppy disk operating system at $B800, and a printer and disk utility ROM at $E800. All three have been reconstructed as TASM32 source that rebuilds the original ROM images byte for byte.

Memory map

The three ROMs sit at the top of the address space; the disk system keeps its buffers just below its own ROM.

Address	Contents
`$00DA–$00FF`	MONA zero-page variables (pointers, display buffer, saved registers)
`$B400–$B4FF`	loader41 workspace: command block, file table, drive state
`$B500–$B5FF`	loader41 directory sector buffer
`$B600–$B6FF`	loader41 read-channel sector buffer
`$B700–$B7FF`	loader41 write-channel sector buffer
`$B800–$BFBF`	loader41 ROM (disk operating system)
`$BFC0–$BFC4`	Floppy disk controller and drive-select latch
`$C000`	Load address of the BASIC file from disk
`$E000`	Load address of the DOS file from disk
`$E800–$EFFF`	drdisv4 ROM (printer and disk utilities)
`$F500–$F502`	Serial port (used by MONA4 only)
`$F600–$F603`	Second I/O port: printer port with MONA2, main 6530 with MONA4
`$F680–$F697`	6530 RRIOT: display, keypad, tape, teletype and timers (MONA 1.10.77 and MONA2)
`$F750–$F757`	drdisv4 variables (printer mode, baud, echo flag)
`$F7EF–$F7FF`	MONA RAM: tape and dump parameters, key vector, NMI and IRQ vectors
`$F800–$FFFF`	MONA ROM, including the 6502 vectors at `$FFFA`

Firmware versions

Three MONA versions are known. MONA2 is the one the disk ROMs were written for; MONA4 is a later teletype-only system.

Version	Main 6532	Serial port	Notable changes
MONA 1.10.77	`$F680`	6532 port B	Original version. Teletype speed set by hand with the `S` command; 110 baud default; clock routine at `$FFB2`
MONA2	`$F680`	6532 port B	Automatic baud-rate detection replaces `S` and the clock; user start vector at `$7FFE` checked at reset; tape load and save return as subroutines; 163 bytes differ from 1.10.77
MONA4	`$F600`	New port at `$F500`–`$F502`	Display and keypad cycle bypassed, always teletype; serial routines rewritten for the new port; small fix in paper-tape loading; 150 bytes differ from MONA2

The entry points that other software calls (`TAB1` `$F80B`, `STP` `$F868`, `ASCIIN` `$F9F1`, `CARRET` `$FA4F`, `TTYOUT` `$FA86`) are at the same address in all three versions.

MONA monitor

MONA runs either from a hex keypad with a six-digit 7-segment display or from a teletype; the PA6 switch picks between them (MONA 1.10.77 and MONA2). After reset it sets up the ports, prints its sign-on ("MONA") on the teletype and waits for commands.

Keypad. Digit keys shift into the address or the data byte, depending on the display mode. The function keys are:

Key code	Routine	Function
`$10`	`TDOWN`	Next address
`$11`	`TUP`	Previous address
`$12`	`TMNEMO`	Show the opcode at the address as a 3-letter mnemonic on the display
`$13`	`TPROGC`	Return to the program counter saved by the last stop
`$14`	`TAKKU`	Show the saved accumulator (`$E7`)
`$15`	`TSTATU`	Show the saved status register (`$E9`)

Key code	Routine	Function
$16	TADRES	Address entry mode
$17	TDATA	Data entry mode
$18	GO	Restore the saved registers and run from the address shown

The key codes and routine names come from the listing; the legends printed on the keys are not known.

Teletype. Hex digits build a number; the following character says what to do with it:

Character	Function
Space	Set the current address to the number typed
.	Store the number in the current cell, go to the next
Return	Next address
Line feed	Previous address
Rubout	Print the sign-on message again
G	Go: run from the current address
L	Load paper tape
Q	Punch paper tape from STRT to ENDE ($F7F0–$F7F3)
S	Set the teletype speed (MONA 1.10.77 only)

Each printed line shows the address, the byte and its mnemonic. The STOP key (NMI) and BRK both save all registers and return to the monitor.

Paper tape format. Records start with ;, then the byte count ($18 = 24 bytes), the address, the data bytes and a 16-bit checksum, all as hex characters. A final record with count $00 carries the number of records.

Cassette tape. Recording uses two tones, 3600 Hz and 2400 Hz, generated with the 6532 timer. A recording is 100 sync characters ($16), *, an ID byte ($F7F4), the start address, the data as hex characters, /, the checksum and two end-of-tape characters ($04). The start and end addresses come from STRT and ENDE; loading checks the ID and shows $FFFF in the address display on error.

I/O addresses

Addresses are for a MONA2 system; with MONA4, subtract $80 from the 6530 addresses (PA becomes $F600, T_1 becomes $F614, and so on).

Address	Name	Use
$F680	PA	6532 port A: keypad columns, display segments, PA6 teletype/keypad switch, PA7

Address	Name	Use
		teletype input
$F681	PADD	Port A direction
$F682	PB	6530 port B: display digit select, keypad rows, teletype output, tape output and motor
$F683	PBDD	Port B direction
$F686	TREAD	Timer read
$F687	VORBEI	Timer status (bit 7 = timed out)
$F694	T_1	Timer, ÷1 clock
$F695	T_8	Timer, ÷8 clock (used by drdisv4)
$F696	T_64	Timer, ÷64 clock
$F697	T_1024	Timer, ÷1024 clock
$F600–$F603	PA2…PBDD2	Second port (MONA2 systems): parallel printer data on port B, strobe and busy on port A; bit-banged serial printer
$F500, $F501, $F502	SERIN, SERFLG, SEROUT	Serial port (MONA4): input bit 7, start-up flag bit 7, output
$BFC0	FDCCMD	Disk controller command (write) and status (read)
$BFC1	FDCTRK	Disk controller track register
$BFC2	FDCSEC	Disk controller sector register
$BFC3	FDCDAT	Disk controller data register
$BFC4	DRVSEL	Drive select and motor latch (drive 0 = $62, drive 1 = $A2)

The disk controller is a Western Digital WD1771/179x-type chip; the code uses restore ($08), seek ($14), read sector ($88), write sector ($A8), read address ($C4) and write track ($F4). The $F600 port names (PA2…PBDD2, T_8) and the serial port names were added during the reconstruction; they do not appear in the original listing.

Disk operating system (loader41)

loader41 is a 2K file system in ROM at $B800. It handles two drives with 40 tracks of 16 sectors of 128 bytes, about 80 KB per disk. A program can have one file open for reading and one for writing at the same time, and moves data a byte at a time.

Disk layout. The directory is on track 0, read into `$B500`. Each entry is 12 bytes, ten to a sector: a 6-character name, a flag byte (bit 7, probably write protect) and the start and size of the file. Files are stored in consecutive sectors.

Entry points.

Address	Name	Function
`$B800`	`COLD`	Reset: close all files
`$B803`	`DOS`	Execute the command in `$B480`
`$B806`	`EOFTST`	Test for end of file
`$B809`	`RDBYTE`	Read the next byte from the open file into A
`$B80C`	`WRBYTE`	Write the byte in A to the open file

Calling a command. Put the command code in `$B480`, the drive number (0 or 1) in `$B481` and the 6-character file name in `$B482`–`$B487`, then `JSR $B803`. In most codes an even value works on the read channel and an odd value on the write channel.

Code	Function
`$F0` / `$F1`	Open a file for reading / writing (probable)
`$E0` / `$E1`	Close the read / write file; closing for writing flushes the last sector
`$F2` / `$F3`	Open at a record position (probable)
`$F4`	Open from the start of the disk (probable)
`$F5`	Find or create a directory entry (probable)
`$F6` / `$F8`	Write a directory entry; `$F8` also sets the flag bit
`$E3`	Write out the current sector
`$F7`	Copy a file between drives (in drdisv4, `$EE4E`)
`$F9`	Set the file position for random access (in drdisv4, `$EE19`)
`$E2`, `$E5`–`$E9`	Accepted, do nothing

The functions marked "probable" are read from the code, not from documentation.

Errors. On an error the loader restores the caller's stack and returns with an error code in `$B48B`; 0 means no error.

Code	Meaning
1	File not found
2	Sector read or write failed

Code	Meaning
3	Position beyond the end of the file, or a restore or seek failed
5	Disk full
6	Directory full
7	Unknown command
8	Disk write-protected
9	Channel already open

Codes 4 and 15 can also come out of the controller error table for other failed commands; their exact meaning is unclear.

Printer and disk utilities (drdisv4)

drdisv4 is a 2K ROM at `$E800` that adds printer drivers and disk utilities. The name probably stands for **Dr**ucker (printer) and **Dis**k, version 4. It calls MONA2 and loader41 directly and uses MONA2's timer addresses.

Address	Function
`$E801`	Print a carriage return on the parallel printer
`$E803`	Print the character in A on the parallel printer (data on port B of `$F600`, strobe on PA1, busy on PA3/PA4)
`$E8A4`	Send a character on the bit-banged serial printer line
`$E91F`–`$E954`	Set the serial baud rate with the 6530 timers
`$E95A`	Receive a character on the bit-banged serial line
`$E9CD`	`TTYOUT` with printer echo
`$E9F0`	`CARRET` with printer echo
`$EA08`	`ASCIIN` with printer echo; Ctrl-P (`$10`) switches echo on and off (flag at `$F757`)
`$EB00`	Format a disk: write all 40 tracks, verify every sector, write an empty directory
`$EB02`	Copy a disk from drive 0 to drive 1, file by file (probable)
`$EE19`	Loader command `$F9`: set the file position for random access
`$EE4E`	Loader command `$F7`: copy a file from one drive to the other (probable)
`$EF20` / `$EF23`	Load the file `BASIC` from drive 0 / drive 1 into memory from `$C000`

Address	Function
$EF70	Save memory from STRT to ENDE to a file
$EFAA	Load the file DOS to $E000

Format and copy return to the monitor with $0000 in the address display on success or $FFFF on error, like MONA's own tape commands. During formatting the IRQ vector points to a short abort routine at $ED10.

The DOS file itself, which would load to $E000, has not been found.

Source files

All five sources assemble with TASM32 and rebuild their ROM image byte for byte (2048 bytes each).

File	ROM	Address	Based on
MONA_alpha1.ASM	MONA 1.10.77	$F800	The printed listing, comments and layout kept as printed
MONA2 from listing.ASM	MONA2	$F800	MONA.ASM, with only the changed parts rewritten
MONA4 from listing.ASM	MONA4	$F800	MONA2.ASM, with only the changed parts rewritten
LOADER41.ASM	loader41	$B800	Disassembly of the ROM dump
DRDISV4.ASM	drdisv4	$E800	Disassembly of the ROM dump, names matching MONA2.ASM and LOADER41.ASM

The original was written for the Micronic 650X assembler, which follows Motorola conventions. The changes made for TASM32:

- ORG, EQU, FCB, FCC and FDB became .ORG, .EQU, .BYTE, .TEXT and .WORD.
- FDB stores the high byte first, so every .WORD operand has its two bytes swapped to give the same memory contents.
- Labels with : or ? were renamed: T:1 became T_1, TASTE? became TASTE_, and so on.
- ASL AC became ASL A; character constants got a closing quote; the semicolon constant is written #$3B.
- MONA parts A and B share one symbol table, so they are one file and the END between them is commented out.

In the disassembled ROMs (loader41 and drdisv4), the labels were chosen during reconstruction; the original names are not known. Code that jumps into the middle of a BIT instruction to skip two bytes is written as .BYTE $2C (or $24) followed by the hidden instruction.