Monitor Extender

REFERENCE CARD

ALL COMMANDS

Items with a ✓ are Monitor Extender

Format	Example	Description
ADR	*300(cr)	Displays the contents of the memory location ADR in hexadecimal.
✓ ADR(ctrl-Y)A	*300(ctrl-Y)A(cr)	As above but displays in ASCII (character codes).
✓ ADR(ctrl-Y)B	*300(ctrl-Y)B(cr)	As above but displays in binary.
ADR1.ADR2	*300.310(cr)	Displays the contents of the memory locations ADR1 through ADR2 in hexadecimal.
✓ ADR1.ADR2(ctrl-Y)A	*300.310(ctrl-Y)A(cr)	As above but displays in ASCII.
✓ ADR1.ADR2(ctrl-Y)B	*300.310(ctrl-Y)B(cr)	As above but displays in binary.
(return)	*(cr)	Displays the contents of the next 8 locations in hexadecimal.
.ADR2	*.340	Displays the contents of the memory range from the current address through ADR2 in hexadecimal.
✓ .ADR2(ctrl-Y)A	*.360(ctrl-Y)A(cr)	As above but displays in ASCII.
✓ .ADR2(ctrl-Y)B	*.360(ctrl-Y)B(cr)	As above but displays in binary.
ADR:DATA DATA DATA	*300:F3 AB 04(cr)	Enters the hexadecimal byte values into memory starting at ADR.
:DATA DATA DATA	*:C0 00 20(cr)	Enters the values into memory starting after the last ADR used for despositing.
✓ ADR (ctrl-Y)"TEXTSTRING(cr)	*300(ctrl-Y)"TEXTSTRING(cr)	Enters the ASCII values of the characters following the quote (") into memory starting at ADR. The carriage return is the last value stored ($8D). All bytes have the high bit set.
✓ ADR(ctrl-Y)"TEXT2(ctrl-@) (cr)	*300(ctrl-Y)"TEXT2(ctrl-@) (cr)	As above except stores a null byte (00) instead of a CR($8D) as the last byte.
✓ ADR(ctrl-Y)"TEXT3(ctrl-Z) (cr)	*300(ctrl-Y)"TEXT3(ctrl-Z) (cr)	As above except only the text bytes are stored; no delimiter is stored (no "00 or "8D").

Format	Example	Description
(ctrl-Y)"NEXT	*(ctrl-Y)"NEXT	Continues storing text values after the last location used when entering data.
VAL<ADR1.ADR2(ctrl-Y)F	*FF<300.310(ctrl-Y)F(cr)	Fills the memory locations ADR1 through ADR2 with the byte VAL.
ADR1<ADR2.ADR3M	300<310.320M(cr)	Move the contents of memory locations ADR2 through ADR3 to the locations starting at ADR1. (ADR1 should be less than ADR2 or greater than ADR3 to move correctly.)
ADR1<ADR2.ADR3(ctrl-Y)M	*310<300.320(ctrl-Y)M(cr)	As above but with no restrictions on the address ranges.
ADR1<ADR2.ADR3V	*300<A00.A57V(cr)	Verify that the contents of the memory range from ADR2 through ADR3 match the contents of the same size block beginning at ADR1. Any differences are displayed.
ADR1.ADR2(ctrl-Y)C	*300.333(ctrl-Y)C(cr)	Computes and displays a single byte checksum of the contents of the memory range ADR1 through ADR2.
I	*I(cr)	Sets inverse display mode - black characters on white background.
N	*N(cr)	Sets normal display mode - white characters on black background.
DATA1+DATA2	*78+36(cr)	Computes and displays the hexadecimal sum of DATA1 and DATA2.
DATA2–DATA2	*B8–5F(cr)	Computes and displays the hexadecimal difference of DATA1 and DATA2.
ADR1<ADR2.ADR3(ctrl-Y)S	*300<F800.FFFF(ctrl-Y)S(cr)	Searches the memory range ADR2 through ADR3 for the pattern of bytes at ADR1. The pattern contains the number of bytes to be searched for (1-63) in byte 1 (ADR1) with the actual bytes following (ADR1+1, ADR1+2,... ADR1+n).
ADRL	*800L(cr)	Disassembles 20 instructions starting at ADR into 6502 mnemonics.
L		Disassembles the next 20 instructions beginning at the current address.
ADR1.ADR2(ctrl-Y)Z	*800.850(ctrl-Y)Z(cr)	Disassembles the instructions from ADR1 through ADR2 and creates a labelled source text file in memory. Any existing text is lost.

Format	Example	Description
ADR1.ADR2(ctrl-Y)+	*900.930(ctrl-Y)+(cr)	As above but appends the new text to any previously existing text in the file.
(ctrl-Y)L	*(ctrl-Y)L(cr)	Lists disassembled text.
(ctrl-Y)?	*(ctrl-Y)?(cr)	Displays the start and end +1 locations of the text file.
(ctrl-Y)<	*(ctrl-Y)<(cr)	Compresses the text file by converting all multiple spaces to 1 space.
(ctrl-Y)	*(ctrl-Y)	Executes the subroutine starting at address $3F8. This is taken over by the Monitor Extender and replaced by:
(ctrl-Y) &	*(ctrl-Y)&(cr)	Executes the subroutine starting at address $3F5. This is generally a jump (JMP) to the actual routine.
(ctrl-E)	*(ctrl-E) (cr)	Opens & displays the 6502 pseudo registers. The registers may be modified by typing in new data if the display line is the most recent display.
ex: A=8D X=FE Y=02 P=32 P=BA	*:F9 *:F9 FE 03	Changes A register Changes Y register - note that you must retype data you don't want to change.
ADRG	*800G(cr)	Runs a machine language program beginning at ADR.
ADRS	*800S(cr)	Singlesteps through a program beginning at ADR. Each executes one instruction. The pseudo registers are opened with each step (see (ctrl-E) command). (Not available in Autostart ROM).
or	*800S(cr) *S(cr) *S(cr) *SSS(cr)	
ADRT	*800T(cr)	Trace a program beginning at ADR and continue until encountering a "BRK" (00) or until "RESET" is pressed. A BRK returns control to the monitor. The pseudo registers are opened. (Not available in the Autostart ROM).
ADR1.ADR2R	*800.FFFR(cr)	Read data from cassette into the range ADR1 through ADR2. The record length of data on the tape must equal the memory range given.
ADR1.ADR2W	*800.FFFW(cr)	Write the data from ADR1 through ADR2 onto cassette tape.

Format	Example	Description
X(ctrl-P)	*6(ctrl-P) (cr)	Sets the output to go to slot #X(X=0-7). If X=0, the output goes to the video display.
X(ctrl-K)	*2(ctrl-K) (cr)	Sets the input to come from slot #X(X=0-7). If X=0, the input comes from the keyboard.
(ctrl-B)	*(ctrl-B) (cr)	Enter the lanquage on the ROMs (cold-start).
(ctrl-C)	*(ctrl-C) (cr)	Re-enter the current language in the ROMs (warmstart).
MonExt addr+6G	*806G(cr)	Enable the SLOLIST feature of the Monitor Extender. Pressing the space bar halts listing until another key is pressed. A (ctrl-C) will abort the listing.
F666G	*F666G	Enter the mini-assembler(not available in Apple-II Plus)
$FF69G	!$FF69G	Exit the mini-assembler.
(ctrl-Y)/I	*(ctrl-Y)/I(cr)	Initialize the Monitor Extender disk access mode.
S.D(ctrl-Y)/#	*6.1(ctrl-Y)/#(cr)	Set slot (S) and drive (D) numbers for disk access.
T.S(ctrl-Y)/R	*11.C(ctrl-Y)/R(cr)	Read the 256 bytes of the track (T) and sector (S) into the disk buffer.
T.S(ctrl-Y)/W	*12.3(ctrl-Y)/W(cr)	Write the disk buffer to the given track (T) and sector (S).
(ctrl-Y)/>	*(ctrl-Y)/>(cr)	Write the disk buffer to the last defined track and sector.
VOL(ctrl-Y)/F	*23(ctrl-Y)/F(cr)	Format the disk with the volume number ($1-$FE) given. **Any information on the disk is destroyed.** This does not INITIALIZE the disk!
(ctrl-Y)/?	*(ctrl-Y)?(cr)	Displays the disk access status-slot, drive, track ,sector, & buffer address.