OSI

6502 8K BASIC

USERS MANUAL

OSI 6502 8K BASIC VERSION 1.0

INTRODUCTION:

OSI BASIC was written by Microsoft, Inc., the company which wrote Altair 8080 and 6800 BASIC. BASIC is written in an interpretive marco code, which is then installed on each microprocessor of interest. The result is that 6502 8K BASIC is identical to 8080 8K BASIC except where deliberate changes were made to the macro-code. Thus 6502 8K BASIC benefits from years of debugging of 8080 BASIC.

OSI 6502 BASIC runs faster than 8080 BASIC because of the 6502's superior instruction set and due to higher clock speed operation of 6502 systems.

The following manual simply provides reference material for those familiar with BASIC. It is not intended to be a teaching aid, or to even fully characterize the language.

OS 6502 BASIC is a full ANSI Standard BASIC with additional features. Th s BASIC will run programs written for Altair®, IMSAI®, and SWTP® BASIC without modification. Speed improvements from 4 to 100 times can be expected over these machines.

PDP-8®, PDP-11®, and NOVA® BASIC programs will also run with very little modification.

Persons unfamiliar with BASIC can study any of the tutorial books on BASIC available at any college or university book store, hobby computer store, and through most of the hobby computer magazines. Schaum's outline series Programing with BASIC , published by McGraw-Hill, is an excellent tutorial book which is compatible with OSI BASIC.

Additional specific information about OSI 8K BASIC can be obtained from the MITS Altair® BASIC Reference Manual. The BASIC has been tested extensively with popular BASIC programs and is believed to be reasonably bug free. However, no warranty is made for its accuracy. If you think you have found a bug, please send a documented example to Ohio Scientific, Inc.

License Terms:

The purchase of 8K BASIC by itself or as part of a package with an OSI Challenger gives the purchaser a single user license for BASIC. The user may make back up copies for his own use but cannot sell, lend, or otherwise redistribute BASIC in any form.

The single user license is for one version of BASIC., i.e., if the user has 8K BASIC for cassette, he will still have to pay full price for 8K Disk BASIC.

The cost of the "license" is $30.00. The additional charges are for the media and copying. Licensed users can purchase additional back up copies or updated copies of BASIC at the current retail price minus $30.00.

Each copy of BASIC carries the copyright label as well as internal labels and serial numbers.

Persons, dealers, and companies who violate the copyright laws will be prosecuted!

SPECIAL CHARACTERS

Character	Use
@	Erases line being typed
(shift 0)	Erases last character being typed
Carriage return	Must be used after each line typed
Control C	Interrupts program execution or listing returns to command mode.
: (colon)	Allows multiple statements per line
Contol 0	Typing a control 0 once surpresses output until another control 0 is typed.
?	? can be used instead of print.

OSI 8K BASIC is a "standard" BASIC with additional string handling capability and I/0 commands, as well as the following features.

OSI BASIC allows multiple statements per line via ":". Next without a variable can be used when FOR-NEXT statements are not nested. END statements are not necessary. Question marks can be used instead of "PRINT". "LET" is optional. No spaces are required in BASIC. These features allow highly efficient memory usage when necessary.

Variables can be two characters long. Longer variables can be used but only the first two characters will be utilized. The first character must be alphabetic, the second can be alphabetic or numeric. Long variables can not contain words used by BASIC such as NEW, SIN, and so on. Since spaces are not necessary BASIC would interpert a variable such as "ANEW" as a variable A and the command "NEW" and would erase the program.

EXAMPLES:

LEGAL	ILLEGAL
A	IA
A1	#B
AZ	TOO
BEQ	RGOTO
APPLE	NEW 1
TUESDAY	FREQUENCY

Note: that variables AZ1 and AZ2 would be treated the same since BASIC looks only at the first two characters.

COMMANDS

NAME	EXAMPLE	COMMENTS
LIST	LIST LIST 100	Lists program Lists program from line 100. Control C stops program listing at end of current line.
NULL	NULL 3	Inserts 3 nulls at the start of each line to eliminate change return bounce problems. Null should be 0 when entering paper tapes from Teletype® readers. When punching tapes Null >=3. Higher settings are required on faster mechanical terminals.
RUN	RUN	Starts program execution at first line. All variables are reset. Use an immediate GOTO to start execution at a desired line.
	RUN 200	GOTO 200 with variables reset.
NEW	NEW	Deletes current program.
CONT	CONT	Continues program after Control C or STOP if the program has not been modified. For instance a STOP followed by manually printing out variables and then a CONT is a useful procedure in program debugging.
LOAD	LOAD	Used in cassette and Disk BASIC only.

OPERATORS

SYMBOL	EXAMPLE	COMMENTS
=	A=10 LET B=10	LET is optional
-	C=-B	Negation
↑ (Shift/n)	X↑4	X to the 4th power
		C↑D with C negative and D not an integer gives an FC error.
*	C=A*B	Multiplication
/	D=L/M	Division
+	Z=L+M	Addition

	J=255.1-X	Subtraction
<>	10 IF A<>B THEN 5	Not equal
>	B>A	B greater than A
<	B<A	B less than A
<=, =<	B<=A	B less than or equal to A
=>,>=	B=>A	B greater than or equal to A
AND	IF B>A AND A>C THEN 7	If both expressions are true then--.
OR	IF B>A OR A>C THEN 7	If either expression is true then--.
NOT	IF NOT B>A THEN 7	If B<=A then--.

AND, OR, and NOT can also be used in Bit manipulation mode for performing Boolean operations of 16 bit 2s complement numbers (-32768 to +32767)

EXAMPLES

EXPRESSION	RESULT
63 AND 16	16
-1 AND 8	8
4 OR 2	6
10 OR 10	10
NOT 0	-1
NOT 1	-2

OPERATOR EVALUATION RULES: Math statements evaluated from left to right with * and / evaluated before + and - Parentheses explicitly determine order of evaluation.

Precedence for evaluation

1) By parenthese
2) ↑
3) Negation
4) * /
5) + -
6) =,<>, <, > , <=, >=
7) NOT
8) AND
9) OR

STATEMENTS

In the following examples

V or W is a numeric variable, X is a numeric expression, X$ is a string expression, I or J is a truncated integer.

NAME	EXAMPLE	COMMENTS
DATA	10 DATA 1,3,7	Data for READ statements must be in order to be read. Strings may be read in DATA statements.
DEF	10 DEF FNA (V)=V*B	User defined function of one argument.
DIM	110 DIM A (12)	Allocates space for Matrices and sets all matrix variables to zero. Non dimensioned variables default to 10.
END	999 END	Terminates program (optional)
FOR, NEXT	10 FOR x=.1 to 10 STEP.1 20_____ 30 NEXT X	STEP is needed only if X is not incremented by 1. NEXT X is needed only if FOR NEXT loops are nested if not NEXT alone can be used variables and functions can be used in FOR statements.
GOTO	50 GOTO 100	Jumps to line 100
GOSUB, RETURN	100 GOSUB 500 500 . . . . 600 RETURN	Goes to subroutine, RETURN goes back to next line number after the GOSUB
IF...THEN		
	10 If X=5 THEN 5	If the statement is true
	10 If x=5 THEN PRINT X	
	10 If X=5 THEN PRINT X:Y=Z	Then the following will be executed including multiple statements of that line.
IF...GOTO		
	10 IF X=5 GOTO5	Same as if THEN with line number
ON... GOTO		
	100 ON I GOTO 10, 20, 30	Computed GOTO If I=1 then 10 If I=2 then 20 If I=3 then 30

PRINT	10 PRINT X 20 PRINT "Test"	Prints value of expression Standard BASIC syntax with , ; " formats
READ	490 READ V, W	Reads data consecutively from DATA statements in program
REM	10 REM	This is a comment for non-executed comments.
RESTORE	500 RESTORE	Restores Intial values of all DATA statements
STOP	100 STOP	Stops program execution reports a BREAK. Program can be restarted via CONT.

FUNCTIONS

Function		Comment
ABS (X)		For X=>0 ABS(X)=X For X<0 ABS(X)=-X
INT (X)		INT (X) = largest integer less than X
RND (X)		Generates a random number between 0 and 1 RND (0) generates the same number always RND (X) with the same X always generates the same sequence of random numbers NOTE (B-A)* RND (1)+A generates a random number between B and A
SGN (X)		IF X>0 SGN(X)=1 IF X=<0 SGN(X)=0
SIN (X) COS (X) TAN (X) ATN (X)		Sine of X where X is in radians Same for COS, TAN, and ATN (ARC TAN)
SQR (X)		Square root
TAB (I)		Spaces the print head I.
USR (I)	See I/O section	
EXP (X)		E↑X where E is 2.71828
FRE (X)		Gives number of Bytes left in the workspace.
LOG (X)		Natural LOG to obtain base 10 logs use LOG(X)/LOG (10)

POS (I)	Gives current location of terminal print head.
SPC (I)	Prints I spaces, can only be used in print statements.

STRINGS

Strings can be from 0 to 255 characters long. All string variables end in $ ex. A$, B9$, HELLO$.

Strings can be dimensioned equated, printed, read from Data statements, etc.

STRING FUNCTIONS

ASC (X$)	Returns ASCII value of first character in string.
CHR$ (I)	returns a I character string equivalent the ASCII value above.
LEFT$ (X$,I) RIGHT$(X$,I)	Gives left most I characters of string X$ Gives right most I character of string X$
MID $ (X$,I,J)	Gives string subset of string X$ starting at I^{th} character for J characters. If J is omitted, goes to end of string.
LEN (X$)	Gives length of string in bytes.
STR$ (X)	Gives a string which is the character representation of the numeric expression of X. Example X=3.1 X$=STR$(X) X$="3.1"
VAL (X$)	Returns string variable converted to number. Opposite of STR$(X)

I/O

The following features of OSI 8K BASIC are usefull primarily for I/O control. The user should be extremely careful with these statements and functions since they manipulate the memory of the computer directly. An improper operation with any of these commands can cause a system crash, wiping out BASIC and the users program, thus requiring a complete reload of the computer.

STATEMENT/FUNCTION	COMMENT
PEEK (I)	Returns the decimal value of the specified memory or I/O location. (Decimal) Example: X=PEEK (64256) Loads variable X with the 430 Board's A/D converter output. ($FB00_{hex}$)
POKE I,J	Loads memory location I (decimal) with J (Decimal) I must be between 0 and 65536 and J must be between 0 and 255 Example: 10 Poke 64256, 255 loads FB00 with FF (Hex) thus loads the 430 Board's D/A port 0 such that its output is +2 volts.
WAIT I,J,K	Reads status of memory location I (Decimal) exclusive OR's with K then AND's the result with J until a non zero result is obtained. If K is omitted, it is zero. Wait is used for fast service of input status flags. Example: Wait X,1 will wait until Bit zero of memory location X goes low then BASIC will continue.

The high speed servicing of flags via the WAIT command allows the programmer to service medium speed devices such as line printers or industrial equipment directly in BASIC.

USR: The USR function allows linkage to machine language routines such as ultra-fast device handlers, etc. The USR function calls only one machine language routine and can pass one integer value to the machine language routine so that 65,000 actual user routines are possible.

The beginning of the user subroutine must be poked into $23E_{hex}$ (low) and $23F_{hex}$ (high). The USR routine can use up to 8 levels of subroutines (16 stack locations) without page swapping.

The USR function can obtain the argument of the function by calling the routine pointed to by 6 (low) and 7 (high). This routine will

place the value of the argument in $AE_{(hex)}$ (high part) and $AF_{(hex)}$ (low part). To pass a value back to BASIC, the high part is placed in A and the low part is placed in Y and the subroutine pointed to by 8 and 9 should be called. If this function is not called USR (X) will equal X. An RTS returns from USR to BASIC. All registers can be modified by the user routine without affecting BASIC, however, no page zero locations can be modified! The POKE instruction can also be used to change the USR function call.

INTERRUPTS

For Interrupting routines of any significant length, page zero and page one should be swapped out to higher memory, or memory partitioning (A_{16} and A_{17} on late model OSI memory boards) should be used.

CONVERTING OTHER BASICS TO RUN ON OSI 6502 8K BASIC

MATRIX subscripts: Some BASICS use [ ]
OSI BASIC used ( ).

Strings:

OTHER	OSI
DIM A$(I,J)	DIM A$ (J)
A$ (I)	MID$ (A$,I,1)
AS (I,J)	MID$ (A$,I,J-I+1)

Multiple assignments: B=C=0 must be rewritten as B=0:C=0. Some BASICs use / to delimit multiple statements per line.Use ":". Some BASICs have MAT functions which will have to be rewritten with FOR-NEXT loops.

OSI 6502 8K BASIC ERROR MESSAGES

ERROR CODE	MEANING
BS	Bad Subscript: Matrix outside DIM statement range, etc.
DD	Double Dimension: Variable dimensioned twice. Remember subscripted variables default to dimension 10.
FC	Function Call error: Parameter passed to function out of range.
ID	Illegal Direct: Input or DEFIN statements can not be used in direct mode.
NF	NEXT without FOR:
OD	Out of Data: More reads than DATA
OM	Out of Memory: Program too big or too many GOSUBs, FOR NEXT loops or variables.
OV	Overflow: Result of calculation too large for BASIC
SN	Syntax error: Typo, etc.
RG	RETURN without GOSUB.
US	Undefined Statement: Attempt to jump to non-existant line number.
/0	Division by Zero.
CN	Continue errors: Attempt to inappropriately continue from BREAK or STOP.
LS	Long String: String longer than 255 characters.
OS	Out of String Space: Same as OM
ST	String Temporaries: String expresssion too complex.
TM	Type Mismatch: String variable mismatched to numeric variable.
UF	Undefined Function.

OPERATING INSTRUCTIONS: AUDIO CASSETTE VERSION

Minimum support Configuration: 6502 CPU, 65V PROM monitor, 12K RAM, 440 video board, 430 cassette interface.

Turn on Procedure: OSI 8K BASIC for cassette is in standard OSI Auto-Load^{tm} format and takes approximately 15 minutes to load.

1. Turn on computer, monitor.
2. Reset computer, turn on the cassette.
3. As soon as the tape advances past the white leader, type "L" on the keyboard.
4. The Auto-Load^{tm}-loader will now start loading in 65V format at 2130.
5. After a few seconds, a full checksum load will start. (15 minutes load time)
6. If an "ERR" is outputed at any time, immediately stop the tape, back it up for a few seconds, start it and type a "G".
7. At the end of the load BASIC will auto execute and output "MEMORY SIZE". If machine language routines are not anticipated, simply type "RETURN" and BASIC will utilize all available memory.
8. BASIC will then ask terminal width. Answer with three times your actual width -3. Example; displayable width is 25 answer "72" then return.

BASIC will now return its prompter "OK" and can be used as any standard on-line BASIC.

The display width can be adjusted by poking a new constant into the CRT output routine. Example: "POKE 8745, 22" reduces the line width to 22 characters. 8745 is the decimal location of the constant.

Dumping a Program to Cassette:

Everything you see on the screen will be outputed to the cassette if it is turned on. To store a program on cassette type "LIST" but do not return. Turn the cassette on to generate some leader then type return. The program will be outputed to cassette as well as the screen.

Inputing a Program from Cassette:

First clear the workspace with a "NEW". Then advance the cassette to the leader just before the program of interest. Type "LOAD", turn the tape on and return. Typing any key will now transfer command back to the keyboard, so when the load is complete type return. BASIC will report an SN error when it reads the "OK" at the end of the tape, so just ignore it.

OPERATING INSTRUCTIONS: PAPER TAPE VERSION

Minimum support configuration: 6502 CPU 65A monitor 12K or more RAM and an ASR33 teletype or equivalent.

Turn on Procedure:

The 8K BASIC paper tape has a checksum loader dumper program at the beginning followed by 8K BASIC in checksum format.

1. Turn on and reset the computer.
2. Place the beginning of the loader in the reader.
3. Type an "L" and turn the reader on.
4. A checksum loader starting at 2E70 similar to the 4K KIM load dump of 12S-1 should load in in 65A format.
5. After this load is complete, type "R" then "L 0129 00 00 00 00 28 2EE3" then "R" then "G".
6. Advance the tape to the beginning of 8K BASIC and turn the reader on.
7. You should see a checksum load occur. The records are so long that the checksum over strikes at the end of each line.
8. Watch carefully for an occurrence of "ERR" if an ERR occurs, stop the reader. Immediately back up the tape at least two full lines. Type "G" and start the reader again. If "ERR"s occur often, the serial interface baud rate is misadjusted or the teletype is in need of repairs.
9. After BASIC loads (in about 45 minutes) Type "R" to verify that you are in the monitor then type "L 012E 1F3F" then "G".
10. BASIC will ask memory size. If you do not require any memory reserved for machine language routines simply type return and BASIC will use all available memory. On teletypes type return when BASIC asks "TERMINAL WIDTH".

BASIC will now return its prompter "OK" and can be used as any standard on-line BASIC.

Punching out a Program!

When you are ready to punch a program, set null for 3 using the immediate mode by typing "Null 3" then return. Then generate some leader off line. With the punch off and the terminal on line type "LIST", then turn the punch on, then type return. BASIC will list out to paper tape.

Reading In a Program:

First Type "NEW" to clear the workspace, then set Null to 0 by "Null 0". Then place the BASIC program tape in the reader and turn it on.

You can optionally type control O just before turning on the reader to surpress output. BASIC will report an "SN error" when it reads the OK at the end of the tape. Simply ignore this.

If a reset occurs from BASIC you may be able to re-enter it by "L 012E 0000" then "R" then "G". If the stack has been disturbed by another program or extensive use of the monitor, this will not be possible and BASIC may be damaged by attempting to warm start it. It is always safe to cold start BASIC from the monitor at 1F3F by "L 012E 1F3F", then "R", then "G". However, any program in BASIC will be lost.