Tokens in OSI Basic

Barry L. Beal RFD #1 Box 160
Machias, ME 04654

As a OSI C1P owner, I'm glad that you have decided to support the OSI systems. Enclosed is some BASIC stuff that might interest other frustrated OSI owners like myself. My version of the Micro-soft BASIC is ver 1.0 rev 3.2. Upon disassembling the BASIC ROM, I found that it is written by Richard W. Weiland.

OSI BASIC USES RAM memory from $0300 to the end of RAM available (unless the answer to MEMORY SIZE? is not the standard cr.). Note: that P-DOS gives you 8K from $2100 to $22FA and 65DV3. 0 uses from $317E to the end of RAM. The Maximum address being $7FFF, although I don't see why it can't go to $A000, the start of the BASIC*IN*ROM. The BASIC ROM makes use of four fields in storing each BASIC STATEMENT: (1) a 00 byte to mark the start of the next line; (2) two byte field indicating the start of the next line (this address is actually the address of the next forward reference in the BASIC forward linked storage scheme); (3) a two byte line number field where the line number is stored as its hex equivalent (low, high format); and (4) n bytes containing the BASIC statement expressed as tokens, ASCII CODES, etc. The codes $01 to $1F are the appropriate OSI graphics characters when they are listed (they would only be in the BASIC statement field if someone mistakenly put them there). The codes $20 to $7F are the corresponding ASCII CODES, they are used to represent the line numbers, variable names, etc. that are not BASIC tokens. The codes $80 to $FE correspond to the tokens list here on a separate page. $FF is the corresponding graphics character. For example 1 REM**EXAMPLE would be represented as 00 XX XX 01 00 8E 2A 2A 45 58 41 4D 50 4C 45. The following is the vector table for my C1P:

Monitor	JUMP $FE00	$FFEB	JMP ($0218)
WARM START	JUMP $0000	$FFEF	JMP ($021A)
COLD START	JUMP $BD11	$FFF1	JMP ($021C)
DISK BOOT	JUMP $FC00	$FFF4	JMP ($021E)
		$FFF7	JMP ($0220)

RESTART VECTOR	$FF00	
NMI VECTOR	$0130	
IRQ VECTOR	$01C0	(also break vector)

Note: NMI AND IRQ VECTOR locations should point to the appropriate routine. (Remember these locations are in the stack so watch the stack ptr.)

More BASIC info of interest. The following addresses might come of handy:

0000	jump for warm start (should be 4C 74 A2)
0003	Ok, input command vector (should be 4C C3 A8)
000B	low, high USR VECTOR
000F	TERMINAL WIDTH (standardly 72 decimal of hex 48)
0011-005F	72 bytes for storage of BASIC statements
0079,007A	appears to be start of RAM memory used by BASIC
0081,0082	appears to be end of useable RAM

Note: that by making the changes for memory locations $0000 to $0005 that one can exit the cold-start routine when one doesn't want to destroy what's in memory because of accidently pressing (break) C; instead of W.

I hope this information inspires other OSI owners to write in about the various secrets about the C1P and C2-4P machines that OSI forgot to mention. SUCH lack of documentation is apt to scare away many potential buyers of this very useful computer.

OSI BASIC VER 1.0 REV 3.2 TOKENS

TOKENS	EQUIVALENT	TOKENS	EQUIVALENT	TOKENS	EQUIVALENT
$80	END	$81	FOR	$82	NEXT
$83	DATA	$84	INPUT	$85	DIM
$86	READ	$87	LET(opt)	$88	GOTO
$89	RUN	$8A	IF	$8B	RESTORE
$8C	GOSUB	$8D	RETURN	$8E	REM
$8F	STOP	$90	ON	$91	NULL
$92	WAIT	$93	LOAD	$94	SAVE
$95	DEF	$96	POKE	$97	PRINT
$98	DONT	$99	LIST	$9A	CLEAR (variables)
$9B	NEW	$9C	TAB(	$9D	TO
$9E	FN	$9F	SPC	$A0	THEN
$A1	NOT	$A2	STEP	$A3	+
$A4	-	$A5	*	$A6	/
$A7	^	$A8	AND	$A9	OR
$AA	>	$AB	=	$AC	<
$AD	SGN	$AE	INT	$AF	ABS
$B0	USR	$B1	FRE	$B2	POS
$B3	SQR	$B4	RND	$B5	LOG
$B6	EXP	$B7	COS	$B8	SIN
$B9	TAN	$BA	ATN	$BB	PEEK
$BC	LEN	$BD	STR$	$BE	VAL
$BF	ASC	$C0	CHR$	$C1	LEFT$
$C2	RIGHT$	$C3	MID$	$C4	(

FROM $C4 TO $D3 BASIC ERROR CODES

TOKENS	EQUIVALENT	TOKENS	EQUIVALENT	TOKENS	EQUIVALENT
$C5	SN	$C6	RG	$C7	OD
$C8	FC	$C9	OV	$CA	OM
$CB	US	$CC	BS	$CD	DD
$CE	/Ø	$CF	ID	$D0	TM
$D1	LS	$D2	ST	$D3	CN

FROM D TO $FE repeats $80 on