

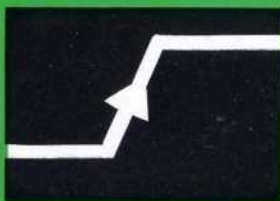


www.commodore.ca

May Not Reprint Without Permission

POCKET REFERENCE  
GUIDE TO COMMODORE'S

# 2001 PET™



LEADING EDGE  
COMPUTER PRODUCTS

DIRECT ORDERS & INQUIRIES TO  
**MELAD ASSOCIATES**

P. O. BOX 159

MILLTOWN, NEW JERSEY 08850

Phone: (201) 828-3682

COPYRIGHT © 1979, LEADING EDGE CO.  
ALL RIGHTS RESERVED

# MISC. INFORMATION

## USER I/O PORT

NAME	DEC.	HEX.	COMMENTS
Data Direction Register	59459	E843	1 = Output, 0 = Input PA0 - PA7
I/O Register w/o handshake	59471	E84F	
I/O Register with handshake	59457	E841	
Auxiliary Control Register	59467	E84B	Set to 16 for free running shift register
Read/Write Counter	59464	E848	Varies shift rate of data through CB 2
Serial I/O Shift Register	59466	E84A	Write or read shift register

Locations 59467, 59464, 59466 are used to generate "CB2" sound on the PET. NOTE: Location 59467 must be reset to zero for cassette to function properly.

## USEFUL MEMORY LOCATIONS

NAME OF LOC.	2001-8		2001-16, -32		BYTES	COMMENTS
	DEC	HEX	DEC	HEX		
Video Memory	32768	8000	32768	8000	1000	
Top of Memory	134	86	52	34	2	LO, HI BYTE
Start of Variables	124	7C	42	2A	2	LO, HI BYTE
Start of Strings	130	82	48	30	2	LO, HI BYTE
Start of Array Tbl.	126	7E	44	2C	2	LO, HI BYTE
IRQ RAM Vector	537	219	144	90	2	LO, HI BYTE
NMI RAM Vector	None	None	148	94	2	LO, HI BYTE
BRK RAM Vector	539	216	146	92	2	LO, HI BYTE
No. of Dropout Errors X 2	630	276	192	C0	2	Pass 1, Pass 2 Errors
Cassette Buffer 1	634	27A	634	27A	192	
Cassette Buffer 2	826	33A	826	33A	192	
USR Floating Point Acc.	176	60	94	5E	7	EMMMMS
Keystroke Value	515	203	151	97	1	255 = No key pressed
Keystroke Buffer	527	20F	623	26F	10	Keys encoded by row & col.
Index into Keystroke Buffer	525	20D	158	9E	1	# of characters in buffer

## Machine Language Routines & Important Hardware Locations

	HEX	
Print character in Register A to screen	FFD2	All PETS
Get a character from keyboard	FFE4	All PETS
Input a character from keyboard	FFCF	All PETS
Prints a carriage return, line feed (CR, LF)	FDD0	(not in 2001-8)
Print a space	FDCC	(not in 2001-8)
Print a byte in Register A	E775	(not in 2001-8)
Input a byte in Register A	E7B6	(not in 2001-8)
ASCII to Hex in Register A	E7E0	(not in 2001-8)
Entry for machine language monitor: SYS (1024) or SYS (64785) on 2001-16,-32		

Graphics/Lower Case Select: POKE 59468, 12 = Graphics  
POKE 59468, 14 = Lower Case

NOTE: PET Models 2001-16, -32 behave differently in the lower case mode than Model 2001-8 (shifted and unshifted alpha are reversed).

Cassette Motor Control: POKE 59411, 53 Turns Motor On  
POKE 59411, 61 Turns Motor Off

CRT blanking (2001-8 only): POKE 59409, 52 Turns Screen Off  
POKE 59409, 61 Turns Screen On

# GENERAL INFORMATION

Numeric Accuracy: 9 digits

Numeric Range:  $\pm 1.70141184E + 38$  (  $\pm 32767$  for integers)  
 $\pm 2.93873587E - 39$

Numeric, Integer, & Array Variable Range:

Valid variables are any alphabetic (A-Z) character optionally followed by other alphanumeric (A-Z, 0-9) characters. Keywords cannot be used within variable names, and only the first two characters are recognized.

## STRING FUNCTIONS

FUNCTION	EXAMPLE	DEFINITION
ASC	100 N = ASC (AS)	Returns the ASCII value of the first character in the string.
CHRS	100 IS = CHRS (N)	Returns the character equivalent of the ASCII value N.
LEFTS	100 IS = LEFTS (AS, N)	Returns N leftmost characters of the string.
LEN	100 N = LEN (AS)	Returns number of characters in the string.
MIDS	100 IS = MIDS (AS,X,Y)	Returns Y characters, starting from the Xth character of the string.
RIGHTS	100 IS = RIGHTS (AS,N)	Returns N rightmost characters of the string.
STRS	100 IS = STRS (N)	Returns string value of the number.
VAL	100 N = VAL (AS)	Returns numeric value of the string.

## SPECIAL SYMBOLS

SYMBOL	EXAMPLE	FUNCTION
"	100 IS = "ABC"	String Delimiter
\$	100 IS = "XYZ"	String Identifier
%	100 B% = INT (N)	Integer Identifier
:	100 X = 0 : Y = 4	Allows multiple statement lines.
:	100 PRINT X; Y	Prints numbers separated by 3 spaces.
:	100 PRINT AS; BS	Prints strings concatenated.
:	100 PRINT X, Y	Prints numbers at tab stops 10, 20, 30, and 40.

## BASIC ABBREVIATIONS

Most BASIC words can be abbreviated using the first letter of the word and the second letter shifted. For example: LIST = Li, RIGHT \$ = Ri, etc. In some cases, there can be abbreviations that could represent more than one word (e.g.: STEP and STOP). A list of these exceptions follows:

RESTORE	= REs	STEP	= STe
GOSUB	= GOs	LEFTS	= LEf
CLOSE	= CLo	STRS	= STr
RETURN	= REt	PRINT	= ?
INPUT#	= In	PRINT#	= Pr
TAB(	= Ta	SPC(	= Sp

INPUT HAS NO ABBREVIATION.



STATEMENT	EXAMPLE	DEFINITION
SAVE	SAVE "FILE", T	Saves current program in memory to logical file T and gives it the name "FILE".
SPC	10 PRINT SPC (A)	Prints A spaces.
TAB	10 PRINT TAB (A)	Tabs cursor to position A + 1 on the CRT.
VERIFY	VERIFY VERIFY "FILE", T	Compares next encountered program on built in cassette with program within memory. Compares specified file name on logical file T with program within memory.
WAIT	150 WAIT X,Y,Z	Stops BASIC program flow until contents of memory location X, anded with variable Y, and exclusive orred with variable Z is not equal to zero (Z is optional and defaults to zero).

## I/O DEFAULTS & PRE-ASSIGNED IEEE DEVICE ADDRESSES

### PRIMARY DEVICE ADDRESSES

- 0 = Keyboard
- \* 1 = Panel Mounted Cassette
- 2 = Optional External Cassette
- 3 = Video Screen (CRT)
- 8 = Floppy Disk Drive
- 4-30 = External IEEE488 Device Addresses

\* Indicates Default Value

### SECONDARY ADDRESSES FOR CASSETTES

- \* 0 = Tape being opened for a "Read"
- 1 = Tape being opened for a "Write"
- 2 = Tape being opened for a "Write" with an "end of tape" header being forced when the file is closed
- 3-31 = Other IEEE488 Secondary Addresses

## STATUS BYTE (ST) NOMENCLATURE

ST BIT POS.	ST NUM. VALUE	CASSETTE READ	IEEE R/W	TAPE VERIFY & LOAD
0	1		Timeout on Write (Listener)	
1	2		Timeout on Read (Talker)	
2	4	Short Block		Short Block
3	8	Long Block		Long Block
4	16	Unrecoverable Read Error		Any Mismatch
5	32	Checksum Error		Checksum Error
6	64	End of File	EOI Line went low on last byte trans.	
7	-128	End of Tape	Device not present	End of Tape

### BIT POSITIONS

STATUS BYTE:

7 | 6 | 5 | 4 | 3 | 2 | 1 | 0

# OPERATORS

RELATIONAL	OPERATOR	EXAMPLE	RESULT
EQUAL	=	9 = 7	FALSE
LESS THAN	<	5 < 10	TRUE
GREATER THAN	>	6 > 10	FALSE
LESS THAN OR EQUAL	<=	4 <= 4	TRUE
GREATER THAN OR EQUAL	>=	4 >= 3	TRUE
NOT EQUAL	<>	4 <> 8	TRUE

## BOOLEAN

AND	•	AND	1 AND 0	0
OR	+	OR	1 OR 0	1
NOT	.	NOT	NOT 1	0

## ARITHMETIC

ADDITION	+	4 + 3	7
SUBTRACTION	-	12 - 9	3
MULTIPLICATION	•	4 • 4	16
EXPONENTIATION	↑	4 ↑ 2	16
DIVISION	/	12 / 3	4
NEGATION (A=3)	-	-A	-3

## STRING

CONCATENATION	+	"A" + "B"	"AB"
---------------	---	-----------	------

**ORDER OF OPERATIONS:** Parenthesis, Exponentiation, Negation Multiplication & Division (from left to right), Addition & Subtraction (from left to right), Relational Operators, Not, And, Or.

# SPECIAL VARIABLES

POS (0)	Returns position of cursor on CRT
FRE (0)	Read only variable, ? FRE (0) returns free bytes of memory
TIS	Internal time of day clock "HHMMSS" (i.e. "024500" is 2:45 AM)
TI	Read only variable counting 60ths of seconds since power up
ST	Read only I/O status byte (see I/O section)
π	Mathematical constant PI = 3.1415927

# PET MONITOR COMMANDS

.M	.M addr1, addr2	Displays memory from addr1 to addr2.
.R	.R	Displays 6502 registers.
.G	.G addr1	Begin execution of machine language program at addr1.
.L	.L "NAME", #1	Loads a machine language program called "NAME" from cassette 1.
.S	.S "NAME", #1, addr1, addr2	Saves a machine language program to cassette 1 from addr1 to addr2 called "NAME".
::	::addr1xx xx xx	Modify memory in format that was displayed with .M command.
::	::addr1 xx xx xx	Modify registers in format that was displayed with .R command.

# BASIC COMMANDS

COMMAND	EXAMPLE	DEFINITION
CLR	CLR	Clears all references to variables and program control statements (all variables equal zero or null).
CONT	CONT	Continues program after a stop, or use of stop key. Invalid after an error, editing, CLR, or NEW.
LIST	LIST LIST - A LIST B - LIST X-Y	Lists entire program. Lists program up to line A. Lists program from line B. Lists program from line X to line Y.
NEW	NEW	Clears the present BASIC program from memory and clears all variables to zero or null.
RUN	RUN RUN W	Starts program execution from first line of program. Starts program execution from line W.

## STATEMENTS: DECLARES & VARIABLE ASSIGNMENTS

STATEMENT	EXAMPLE	DEFINITION
DATA	10 DATA 1,2,3,4  20 DATA UP, DOWN 30 DATA "1.1", "LOOK OUT"	Specifies data to be read via the "READ" command. Data is read from left to right. Strings need not be surrounded by quotes unless they contain spaces, commas, colons, or graphics.
DIM	25 DIM A(10), A%(2,3) 30 DIM A\$(4,11) 45 DIM L(4,20,2) 50 DIM QS(L,4,M)	Dimensions an array of a specified type of variable to a specified amount of elements (zeroth element included).
LET	10 LET U = 7  20 LET U% = 7  30 LET US = "7"  40 Y = X - 3	Assigns the number 7 to the floating point variable U. Assigns the number 7 to the integer variable U%. Assigns the string "7" to the string variable US. Assigns the value of the numeric expression to the variable Y (Note: LET is optional).

STATEMENT	EXAMPLE	DEFINITION
RESTORE	50 RESTORE	Resets data pointer to the first data statement in the program.
REM	10 REM VER 1.5	Defines the following characters in the line as a remark or comment which is non-executable.

## STATEMENTS: PROGRAM CONTROL

STATEMENT	EXAMPLE	DEFINITION
END	9000 END	Ends program execution.
FOR • NEXT	10 FOR A = N TO P STEP R 50 NEXT A	Sets up a loop which executes the statements between lines 10 and 50 $((P-N) + 1) / R$ times.
GOSUB	10 GOSUB M	Transfers program flow to the subroutine at line M. Program flow resumes at line immediately after 10 when the subroutine executes a RETURN.
GOTO	10 GOTO Y	Transfers program flow to line Y.
IF • THEN	25 IF A ( ) B THEN A = A + 1  30 IF A ( ) B THEN 500 30 IF A ( ) B GO TO 500	If the condition of the statement between the IF and the THEN is true the statement following the THEN is executed. Otherwise the next line is executed.  Transfers program flow to line 500 if condition is true, otherwise executes next line.
ON • GOSUB	70 ON H GOSUB 100,200,300	Transfers program flow to a subroutine at lines 100, 200, or 300 depending on the value of the index variable H.
ON • GOTO	70 ON H GOTO 100, 200, 300	Transfers program flow to line number 100, 200, or 300 depending on the index variable H.
RETURN	100 RETURN	Returns program flow to the line immediately following the subroutine call; subroutine exit.
STOP	200 STOP	Stops program execution.
SYS	150 SYS (X) 150 SYSX	Program flow is transferred to a machine language program at decimal address X.
USR	150 USR (Y)	Passes a parameter Y in the floating point accumulator to a machine language routine pointed to by memory locations 1 & 2.



# STATEMENTS: I/O

STATEMENT	EXAMPLE	DEFINITION
CLOSE	100 CLOSE T	Closes logical file T.
CMD	10 CMD T	Commands logical file T to monitor IEEE Bus.
GET	20 GET AS	Accepts a single string character from the keyboard.
GET #	20 GET # T, X	Accepts a single character from logical file T.
INPUT	100 INPUT 100 INPUT Y,R\$,A%	Accepts a number for the variable Y from the keyboard. Accepts a number, a string, and an integer from the keyboard.
INPUT #	150 INPUT # T, Y,A,R\$	Accepts two numbers and a string from logical file T.
LOAD	40 LOAD  60 LOAD "ABC", T	Loads next file or program encountered on the built in cassette into memory. Loads program or file named "ABC" into memory from logical file T.
OPEN	100 OPEN T, D,S, "NAME"	Opens logical file T and assigns device address D, secondary address S as well as a file name "NAME" to it (See I/O default table).
PEEK	100 A = PEEK(X)	Returns decimal value of memory location X to A.
POKE	100 POKEX, A	Puts the quantity A into memory location X.
PRINT	100 PRINT Y 100 PRINT "Y" 100 PRINT Y\$, Y  100 PRINT Y\$: R\$: A;	Prints the variable Y on the CRT. Prints the character "Y" on the CRT. Prints the string, then tabs to the next tab stop (10, 20, 30, 40 ...) and prints the number. Prints Y\$ concatenated with R\$, and the number A separated by 3 spaces from R\$. The carriage return is suppressed.
PRINT #	150 PRINT # T, A,B	Prints variables A, B onto logical file T. NOTE: ? # is not valid.
READ	200 READ A, A\$,C,A%	Assigns these variables their values from the data statements.
SAVE	SAVE	Saves the current program in memory on the built in cassette, no name is specified.

64	80	32	48	192	208	160	176	CHRS
128	144	160	176	192	208	224	240	RVS
0	16	32	48	64	80	96	112	OFF
								+
@	P	SPACE	0	☐	☐	SPACE	☐	0
A	Q	!	1	♠	♣	♠	☐	1
B	R	”	2	☐	☐	☐	☐	2
C	S	#	3	☐	♥	☐	☐	3
D	T	\$	4	☐	☐	☐	☐	4
E	U	%	5	☐	☐	☐	☐	5
F	V	&	6	☐	☒	☐	☐	6
G	W	'	7	☐	☐	☐	☐	7
H	X	(	8	☐	♣	☐	☐	8
I	Y	)	9	☐	☐	☐	☐	9
J	Z	*	:	☐	♦	☐	☐	10
K	[	+	;	☐	☐	☐	☐	11
L	\	,	<	☐	☐	☐	☐	12
M	]	-	=	☐	☐	☐	☐	13
N	↑	.	>	☐	π	☐	☐	14
O	←	/	?	☐	☐	☐	☐	15