

# TRS-80\* COLOR BASIC AND EXTENDED

## SYSTEM REFERENCE CARD

© 1982, PHILIP R. FASSNACHT, and, SHIRLEY A. & PAUL P. NANOS

\*TRS-80 is a Trademark of Tandy Corp.

### MAGIC NUMBER IS 271

143	140	131	142	141	139	135	138	133	134	137	132	136	129	130	128

### MAGIC NUMBER IS 303

159	156	147	158	157	155	151	154	149	150	153	148	152	145	146	144

### MAGIC NUMBER IS 335

175	172	163	174	173	171	167	170	165	166	169	164	168	161	162	160

### MAGIC NUMBER IS 367

191	188	179	190	189	187	183	186	181	182	185	180	184	177	178	176

### MAGIC NUMBER IS 399

207	204	195	206	205	203	199	202	197	198	201	196	200	193	194	192

### MAGIC NUMBER IS 431

223	220	211	222	221	219	215	218	213	214	217	212	216	209	210	208

### MAGIC NUMBER IS 463

239	236	227	238	237	235	231	234	229	230	233	228	232	225	226	224

### MAGIC NUMBER IS 495

255	252	243	254	253	251	247	250	245	246	249	244	248	241	242	240

## SYSTEM COMMANDS

EXAMPLE FORMATS	DESCRIPTION
CLOAD ["A"]	Load Basic program from cassette.
CLOADM ["A"][,2000]	Load mach lang program from cassette.
CONT	Continue after BREAK or STOP.
CSAVE ["A"][,A]	Write Basic pgm to cassette. (A=ASCII).
CSAVEM"A",start,end,ctl	Write mach lang program to cassette.
★DEL [10] [-] [50]	Delete program lines.
★DLOAD "A"[,0[1]	Load mach lang pgm from other computer.
★EDIT 10	Edit a program line.
EXEC [32400]	Transfer control to mach lang program.
LIST [10] [-] [50]	Display program lines on screen.
LLIST [10] [-] [50]	Print program lines on printer.
NEW	Clear pgm from memory.
★RENUM [100] [,10] [,2]	Renummer program lines.
RUN [30]	Execute Basic program at specified line.
SKIPF ["A"]	Skip to end of program on cassette.
TROFF	Turn off trace.
TRON	Turn on trace.

### EDIT SUBCOMMANDS

FORMAT	DESCRIPTION
nSPACEBAR	Moves cursor n spaces to the right.
n←	Moves cursor n spaces to the left.
nC	Change n characters.
nD	Delete n characters.
E	Stop the edit and enter the changes.
H	Delete remainder of line and enter insert mode.
I	Enter insert mode.
nKc	Delete chars up to the nth occurrence of "c".
L	Display rest of line and go back to first character.
nSc	Move cursor to nth occurrence of "c".
X	Move cursor to end-of-line and enter insert mode.
Shift ↑	Escape from any edit subcommand.
ENTER	Enter the line as edited.

### BASIC "PRINT USING" EDIT PATTERN CHARACTERS

#	Represents a digit.
.	Decimal point.
,	Comma
**	Asterisk-fill indicator.
\$	Places dollar-sign in front.
\$\$	Floating dollar-sign indicator.
**\$	Asterisk-fill plus floating dollar-sign indicator.
↑↑↑↑	Exponential format indicator.
+	Forces plus or minus sign.
-	Always forces minus sign at front. (Forces minus sign at back if negative.)
%blanks%	Select chars=to no. of blanks+2 for each string.
!	Displays only first character of string.
char	String literals display as entered.

EXAMPLES:	(Δ=Space)	RESULTS (Auto-Round)
DATA	PATTERN SPECIFIED	
45678.910	"#####.###"	45678.91
45678.91	"#####.##"	45678.90
	"#####"	45679
	"###.##"	%45678.9
	"##,###.##"	45,678.90
453.36	"**##,###.##+"	*****453.36+
-364.59	"\$\$##,###.##+"	ΔΔΔΔ\$364.59-
45.23	"**\$##,###.##-"	*****\$45.23
-20.10	"+#.##"	-20.10
20.10	"-#.##"	-20.10
	"AMOUNTΔ##.##+"	AMOUNT 20.10+
"SAM"	"!"	S
"SAMMY","JOSEPH"	"!"	SJ
	"%Δ%"	SAMJOS

## BASIC SPECIAL CHARACTERS

" " ( : ; , ? '	Denotes string constant. Denotes priority in math expression. Separates BASIC statements. Keeps cursor at end of PRINTed data. Separates items, or, tabs after PRINT. Used instead of "PRINT". Indicates comments. Use this instead of "REM".
-----------------------------------	---

\$ E	Declares string data type. (AA\$,BB\$,etc). Single-precision notation. (9.87654E+15).
---------	--

+ - * / ↑	Add, concatenate, or sharp. Subtract, or flat. Multiply. Divide. Exponent (4↑2 = 16).
-----------------------	---

< > = <= or =< >= or => <> or ><	Less than or Precedes. Greater than or Follows. Equals. Less than or equal to (precedes or equals). Greater than or equal to (follows or equals). Not equal.
---	---

Order of operations: ↑,+,\*,/,+- (Left to Right, parens first)  
 Order of relations: NOT,AND,OR (Left to Right, parens first)

## BASIC LOGICAL OPERATORS

AND  OR  NOT	All conditions must be met. ALGEBRAIC: When ANDing, if one digit is zero, then the result is zero.  Only one set of conditions must be met. ALGEBRAIC: When ORing, if one digit is one, then the result is one.  Condition must not be met. ALGEBRAIC: Reverses 0 to 1, or 1 to 0.
--------------------------	---

## SPECIAL KEYBOARD KEYS      ⊗ = Shift key

DATA ENTRY			COMMAND ENTRY	
CHAR	DEC	HEX	KEY	RESULT
ETX	03	03	BREAK	Interrupt exec. until CONT entered.
FF	12	0C	CLEAR	Clear screen. Move cursor to home.
CR	13	0D	ENTER	Enter data or command.
HT	09	09	→	Nothing.
BS	08	08	←	Backspace and delete character.
LF	10	0A	↓	Nothing.
↑	94	5E	↑	↑.
DC3	19	13	⊗@	Pause exec. or list until key pressed.
DC2	18	12	⊗0	Toggle upper / lower case, normal / reverse graphics.
\	92	5C	⊗CLEAR	\.
]	93	5D	⊗→	].
NAK	21	15	⊗←	Move cursor to start of line. Erase line.
[	91	5B	⊗↓	[.
←	95	5F	⊗↑	← .

## CASSETTE LOADING ERROR MESSAGES

- I/O—Input/Output error.
- OM—Out of memory. Too much data for memory.
- TM—Type mismatch. Tried to CLOAD mach lang pgm, or vice versa.

## CHARACTER/STRING FUNCTIONS

FUNCTION	RETURNS
CHR\$(num)	Character value of "num". (65="A")
★HEX\$(num)	Hex value of "num". (11="B")
★INSTR(num,str,substr)	Position of "substr" within "str" relative to byte "num" in "str".
LEFT\$(str,num)	Substring of "num" chars from left side of string.
LEN(str)	Length of string.
MID\$(str,start,len)	Substring of "str" at "start" position for length "len".
★MID\$(str,start,len)="c"	Replace portion of string with new string.
RIGHT\$(str,num)	Substring of "num" chars from right side of string.
STR\$(num)	Character string of "num", including sign.
★STRING\$(num,char)	String of "num" repetitions of "char".

## NUMERIC FUNCTIONS

FUNCTION	RETURNS
ABS(num)	Absolute value of num. (-3.25=3.25)
ASC(str)	Decimal value of first character. ("A"=65)
★FIX(num)	Number with decimals removed. (5.25=5)
INT(num)	Whole number. (2.5=2) (-2.5=-3)
RND(0)	Random number between 0 and 1.
RND(num)	Random whole number greater than zero.
SGN(num)	0(zero), 1(positive), or, -1(negative).
VAL(str)	Number extracted from character string.

## MATH FUNCTIONS

FUNCTION	RETURNS
★ATN(angle)	Arctangent of angle(degrees), in radians.
★COS(angle)	Cosine of angle(radians), in radians.
★EXP(num)	Inverse LOG of num which is less than 87.3365.
★LOG(num)	Logarithm of "num".
SIN(angle)	Sine of angle(radians), in radians.
★SQR(num)	Square root of "num".
★TAN(angle)	Tangent of angle(radians), in radians.

Radians=Degrees/57.29577951

Degrees=Radians\*57.29577951

## INPUT/OUTPUT FUNCTIONS

FUNCTION	RETURNS
EOF(dev)	TRUE(-1) if end-of-file, else, FALSE(0).
INKEY\$	Character value of pressed keyboard key.
JOYSTK(num)	Position of joystick (0-63). num '0'=Right joystick, horizontal. num '1'=Right joystick, vertical. num '2'=Left joystick, horizontal. num '3'=Left joystick, vertical.
PEEK(65280)	127 or 255 -No button pressed. 126 or 254-Right joystick button pressed. 125 or 253-Left joystick button pressed. 124 or 252-Both joystick buttons pressed.
★POS(dev)	If dev=0, Cursor position(0-32) on screen line. If dev=-2, Printer carriage position.

## GRAPHICS FUNCTIONS

FUNCTION	RETURNS
POINT(x,y)	TRUE(-1) if graphics block=Character mode. FALSE(0) if graphics block=off. 1-8 if graphics block is a color. (x=0-63,y=0-31).
★PPOINT(x,y)	Same as POINT, except for a point instead of a block. (x=0-255,y=0-191).

## MEMORY FUNCTIONS

FUNCTION	RETURNS
MEM	Number of available bytes in memory.
PEEK(addr)	Decimal value of byte at location "addr".
★TIMER	Current timer value.
★VARPTR(var)	Address of variable "var" in memory.

## BASIC STATEMENTS (★=EXTENDED-ONLY)

### STATEMENT • DESCRIPTION

#### INPUT-OUTPUT

- CLOSE [#device] • Close an open device.
  - ★CLOSE [device] • Close an open device.
  - INPUT [msg;]var[,var . . . ] • Prompt, read keyboard response.
  - INPUT #device, var[,var . . . ] • Read data from device.
  - ★LINE INPUT [msg;]var • Read raw data string from keyboard.
  - OPEN "I"["O"], #device, "filename"[" "] • Open a closed device.
  - ★OPEN "I"["O"], device, "filename"[" "] • Open a closed device.
  - PRINT [#device],var[,var . . . ] • Write data to a device.
  - PRINT item;[,][item . . . ] • Display data.
  - PRINT@ loc,item;[,][item . . . ] • Display data at location.
  - PRINT TAB (num)item;[,][TAB(num)item . . . ] • Display data w/tabs.
  - ★PRINT USING spec;item;[spec;item . . . ] • Display and edit data.
- (DEVICES: Screen/Keyboard =0, Cassette =-1, Printer =-2, ALL = Blank)

#### MEMORY/DATA

- CLEAR [bytes][,][addr] • Reserve bytes for string storage, clear variables, set BASIC high-memory.
- DATA value[,value . . . ] • Store values into memory table.
- ★DEF FN name (var)=exp • Define your own numeric function.
- ★DEFUSRn=addr • Define address of user program "n". (n=0-9)
- DIM var(num[,num . . . ]) • Define array and dimensions.
- [LET] var=value • Assign value to a variable.
- POKE addr,num • Store number at address specified.
- READ var[,var . . . ] • Read values from DATA table into "var"(s).
- REM -or' • Identify following data as remarks.
- RESTORE • Reset pointer to beginning of DATA table.
- ★TIMER=num • Set timer to value (0-65535).

#### PROGRAM CONTROL

- END • End program execution normally.
- FOR var=num1TONum2[STEP Incr] • Loop thru NEXT until var > num2.
- GOSUB line • Perform routine at "line" until RETURN.
- GOTO line • Branch to "line".
- IF cond THEN action1ELSEaction2 • Do action(s) based on condition(s).
- NEXT var • Indicate end of FOR loop.
- ON num GOSUB line[,line . . . ] • Perform routine "num" in line list.
- ON num GOTO line[,line . . . ] • Branch to routine "num" in line list.
- RETURN • Return from a GOSUB.
- STOP • Interrupt program execution until "CONT" entered.
- A=USR(0) • Branch to machine-language program.
- ★X=USRn(arg) • Branch to machine-language program "n". (n=0-9)

#### MUSIC-SOUND

- AUDIO ON[OFF] • Connect/disconnect cassette to speaker.
- MOTOR ON[OFF] • Turn cassette on or off.
- ★PLAY "string" • Play music. (See: PLAYING MUSIC)
- SOUND tone,length • Sound a tone (1-255) for length (1-255).

#### GRAPHICS

- CLS[(n)] • Clear screen to green, or, color n.
- RESET (x,y) • Turn off graphics block.
- SET (x,y,n) • Turn on graphics block to color "n" (0-8).

#### EXTENDED GRAPHICS

- ★CIRCLE (x,y),width, [color],[height],[start],[end] • Make a circle.  
(See: MAKING A CIRCLE)
- ★COLOR foreground,background • Set default colors (0-8).
- ★DRAW "string" • Draw a line or many lines or figures. (See: DRAWING)
- ★GET (x1,y1)-(x2,y2),dlm,[G] • Save rectangle into array "dim".  
(G=Save all graphics details also)
- ★LINE (x1,y1)-(x2,y2),PSET [PRESET][,][B[BF] • Draw a line.
- ★PAINT (x,y),color, stopcolor • Paint a shape with a color.
- ★PCLEAR n • Reserve "n" 1.5K pages for graphics. (n=1-8, default=4).
- ★PCLSn • Clear screen to color n. (n=0-8). Default=Background.
- ★PCOPYnTON1 • Copy page n to page n1.
- ★PMDEN,page • Select PMODEN(0-4) and starting page (1-8). (Default=2,1)
- ★PRESET (x,y) • Resets point to background color.
- ★PSET (x,y,n) • Sets point to color n (0-8).
- ★PUT (x1,y1)-(x2,y2),dlm, [,][PRESET][PSET][AND][OR][NOT] • Move rectangle from array "dim" using optional action.
- ★SCREENn,color • Set text or graphics screen (n=0 or 1), and select color.

## PLAYING MUSIC

### ★PLAY "string"[var]

"string" = " [note[ ;On[ ;Ln[ ;Tn[ ;Vn[ ;Pn[ ;Xa\$]"

.....  
**note** = 1-12(C-B), including sharps(#,+), or flats(-).

**On** = Octave. n=1-5. Default=O2.

**Ln** = Note length. n=1-255. Default=curr. len.

**Tn** = Tempo. n=1-255. Default=T2.

**Vn** = Volume. n=1-31. Default=V15.

**Pn** = Pause length. n=1-255.

**Xa\$** = Execute string variable a\$.

**+** = Suffix for O,L,T,V. Adds 1 to curr. value.

**-** = Suffix for O,L,T,V. Subtracts 1 from curr. value.

**>** = Suffix for O,L,T,V. Doubles current value.

**<** = Suffix for O,L,T,V. Halves current value.

**.** = Suffix for "note". Holds note 1/2 time longer.

## MAKING A CIRCLE

### ★CIRCLE(x,y),width,[color ], [heighth], [start], [end]

**(x,y)** = Center point of circle.

(x=0-255,horiz) (y=0-191,vert)

**width** = Number of points from center, horizontally.

**color** = 0-8. Default=foreground color.

**heighth** = Number of measure from center, vertically. Default=1.

(Fatter circle if less than 1. (.25, etc.))

(Skinnier circle if greater than 1)

**start** = Start location at point on circle.

**end** = End location at point on circle.

Using a clock for ref., start and end points:

0=3:00 .25=6:00 .50=9:00 .75=12:00

**Largest circle is: CIRCLE(128,96),95**

## DRAWING

### ★DRAW "string"[var]

"string" parameters (separate with ";"):

.....  
**BMx,y** = Set start position. (x=0-255, y=0-191)

**BM±p,±p1** = Set start position plus or minus  
"p" points(x), and "p1" points(y).

**Un** = Draw "n" points upward.

**Dn** = Draw "n" points downward.

**Ln** = Draw "n" points to the left.

**Rn** = Draw "n" points to the right.

**En** = Draw "n" points at a 45° angle.

**Fn** = Draw "n" points at a 135° angle.

**Gn** = Draw "n" points at a 225° angle.

**Hn** = Draw "n" points at a 315° angle.

**Xa\$** = Execute variable string a\$.

**Cn** = Color. (n=0-8). Default=foreground color.

**An** = Set angle for subsequent lines. Default=0.

0=0°, 1=90°, 2=180°, 3=270°.

**Sn** = Set size of drawing to scale. (n=1-62). Default=4.

**EXAMPLES:**

S1=1/4    S5=1 1/4    S9=2 1/4

S2=1/2    S6=1 1/2    S10=2 1/2

S3=3/4    S7=1 3/4    S11=2 3/4

S4=Full    S8=Double    S12=Triple

**N** = Return control to start position.

**B** = Make next line a blank (invisible) line.

DERIVED FUNCTIONS (X is in radians.)	
FUNCTION	EXPRESSION
SECANT	1/COS(X)
COSECANT	1/SIN(X)
COTANGENT	1/TAN(X)
INVERSE	
SINE	ATN(X/SQR(-X*X + 1))
COSINE	-ATN(X/SQR(-X*X + 1)) + 1.5708
SECANT	ATN(SQR(X*X - 1)) + (SGN(X) - 1) * 1.5708
COSECANT	ATN(1/SQR(X*X - 1)) + (SGN(X) - 1) * 1.5708
COTANGENT	-ATN(X) + 1.5708
HYPERBOLIC	
SINE	(EXP(X) - EXP(-X))/2
COSINE	(EXP(X) + EXP(-X))/2
TANGENT	-EXP(-X)/(EXP(X) + EXP(-X))*2 + 1
SECANT	2/(EXP(X) + EXP(-X))
COSECANT	2/(EXP(X) - EXP(-X))
COTANGENT	EXP(-X)/(EXP(X) - EXP(-X))*2 + 1
INVERSE HYPERBOLIC	
SINE	LOG(X + SQR(X*X + 1))
COSINE	LOG(X + SQR(X*X - 1))
TANGENT	LOG((1 + X)/(1 - X))/2
SECANT	LOG((SQR(-X*X + 1) + 1)/X)
COSECANT	LOG((SGN(X)*SQR(X*X + 1) + 1)/X)
COTANGENT	LOG((X + 1)/(X - 1))/2

BASIC MESSAGES AND CODES	
MESSAGE	REASON
/0	Divided into zero, or, number too small.
AO	Tried to open an open file. Power off and on.
BS	Addressed array with invalid subscript.
CN	Tried to continue, but continuation not possible.
DD	Re-DIMensioned array. Not allowed.
DN	Wrong device number. Should be 0, -1, or -2.
DS	Statement without line number found during execution.
FC	Invalid parameter for a function.
FD	Bad or out-of-sequence data was read.
FM	Tried to read output or write input.
ID	Tried to enter a BASIC-only statement as a command.
IE	Reading past end-of-file.
IO	Input/output error.
LS	String longer than 255 character maximum.
NF	Executed "NEXT" before "FOR".
NO	Tried to access an unopened file.
OD	"READ" or "INPUT#" cannot find data. Out of data.
OM	Out of memory. All used or reserved.
OS	Out of allocated string space. Allocate more if needed.
OV	Number too large for computer. Overflow.
RG	Executed "RETURN" before "GOSUB".
SN	Syntax error. (Punctuation, spelling, format, etc).
ST	String operation too complex. Break into smaller units.
TM	Used wrong datatype in statement. (Char or Num).
UF	Undefined function called. (Bad or missing DEF).
UL	You referenced a line which does not exist.

# MUSIC

L1	L2	L4	L8	L16	L32	L64	L128
WHOLE	HALF	QTR	1/8	1/16	1/32	1/64	1/128
P1	P2	P4	P8	P16	P32	P64	P128

**NOTE:** F F# G G# A A# B

<b>PLAY</b>	6	7	8	9	10	11	12
<b>SOUND OCTAVE</b>	5 (O2)	19	32	45	58	69	78

**NOTE:** C C# D D# E F F#

<b>PLAY</b>	1	2	3	4	5	6	7
<b>SOUND OCTAVE</b>	89 (O3)	99	108	117	125	133	140

**NOTE:** G G# A A# B C C#

<b>PLAY</b>	8	9	10	11	12	1	2
<b>SOUND OCTAVE</b>	147	153	159	165	170	176	180 (O4)

**NOTE:** D D# E F F# G G#

<b>PLAY</b>	3	4	5	6	7	8	9
<b>SOUND</b>	185	189	193	197	200	204	207

**NOTE:** A A# B C C# D D#

<b>PLAY</b>	10	11	12	1	2	3	4
<b>SOUND OCTAVE</b>	210	213	216	218	221	223	225 (O5)

**NOTE:** E F F# G G# A A#

<b>PLAY</b>	5	6	7	8	9	10	11
<b>SOUND</b>	227	229	231	232	234	236	237

**NOTE:** B C C# D D# E

<b>PLAY</b>	12					
<b>SOUND</b>	238	239	241	242	243	244



## MEMORY MAP

ADDRESS		DESCRIPTION
DEC	HEX	
0	0000	DIRECT PAGE RAM, USEABLE BY MACH LANG PGMS.
111	006F	<b>DEVNUM</b> —DEVICE NUMBER.
112	0070	DIRECT PAGE RAM, UNUSEABLE IF M/L CALLS BASIC RTN.
124	007C	<b>BLKTYP</b> —CASSETTE BLOCK TYPE.
125	007D	<b>BLKLEN</b> —CASSETTE BLOCK LENGTH.
126	007E	<b>CBUFAD</b> —CASSETTE BUFFER ADDR IS LOCATED HERE.
129	0081	<b>CSRERR</b> —CASSETTE READ-ERR FLAG. (0=NONE, 1=CKSUM, 2=MEM ERR).
149	0095	<b>LPTBTD</b> —LINE PRINTER BAUD RATE.
151	0097	<b>LPTLND</b> —LINE PRINTER LINE DELAY.
153	0099	<b>LPTCFW</b> —LINE PRINTER COMMA FIELD WIDTH.
154	009A	<b>LPTLCF</b> —LINE PRINTER LAST COMMA FIELD.
155	009B	<b>LPTWID</b> —LINE PRINTER LINE WIDTH.
156	009C	<b>LPTPOS</b> —LINE PRINTER CURRENT POSITION.
255	00FF	DIRECT PAGE RAM.
256	0100	INTERRUPT VECTORS.
274	0112	<b>USRJMP</b> —BRANCH TO BASIC USR ROUTINE. (POKE ADDR INTO 275 & 276).
277	0115	USEABLE BY MACH LANG PROGRAMS.
282	011A	KEYBOARD LOCK FLAG. (00=UNLOCKED, FF=LOCKED)
283	011B	KEYBOARD DELAY CONSTANT.
285	011D	USEABLE BY MACH LANG PROGRAMS.
338	0152	KEYBOARD ROLLOVER TABLE.
346	015A	<b>POTVAL</b> —LEFT JOYSTICK UP/DOWN POSITION.
347	015B	LEFT JOYSTICK LEFT/RIGHT POSITION.
348	015C	RIGHT JOYSTICK UP/DOWN POSITION.
349	015D	RIGHT JOYSTICK LEFT/RIGHT POSITION.
350	015E	RESERVED FOR INTERNAL USE.
1023	03FF	EXTENDED PAGE RAM.
1024	0400	<b>TEXT MODE VIDEO MEMORY.</b>
1536	0600	BEGINNING OF USER PROGRAM RAM.
4095	0FFF	END OF 4K SYSTEMS PROGRAM RAM.
1536	0600	EXTENDED GRAPHIC SCREEN MEMORY, PAGE 1.
3072	0C00	EXTENDED GRAPHIC SCREEN MEMORY, PAGE 2.
4608	1200	EXTENDED GRAPHIC SCREEN MEMORY, PAGE 3.
6144	1800	EXTENDED GRAPHIC SCREEN MEMORY, PAGE 4.
7680	1E00	<b>BASIC STATEMENT STORAGE (Condensed format).</b>
7680	1E00	EXTENDED GRAPHIC SCREEN MEMORY, PAGE 5.
9216	2400	EXTENDED GRAPHIC SCREEN MEMORY, PAGE 6.
10752	2A00	EXTENDED GRAPHIC SCREEN MEMORY, PAGE 7.
12288	3000	EXTENDED GRAPHIC SCREEN MEMORY, PAGE 8.
13824	3600	EXTENDED COLOR USER PROGRAM RAM.
16383	3FFF	END OF 16K SYSTEMS PROGRAM RAM.
16384	4000	UNUSED.
32768	8000	EXTENDED COLOR BASIC ROM.
40960	A000	COLOR BASIC ROM.
40960	A000	<b>POLCAT</b> —SCAN KEYBOARD FOR PRESSED KEY.
40962	A002	<b>CHROUT</b> —WRITE CHARACTER TO DEVICE SPECIFIED AT 006F.
40964	A004	<b>CSRDON</b> —START CASSETTE AND ENTER READ MODE.
40966	A006	<b>BLKIN</b> —READ CASSETTE BLOCK.
40968	A008	<b>BLKOUT</b> —WRITE CASSETTE BLOCK.
40970	A00A	<b>JOYIN</b> —STORE JOYSTICK VALUES IN 015A-015D.
42760	A708	<b>WRTLDR</b> —TURN ON CASSETTE, WRITE LEADER.
46061	B3ED	<b>INTCNV</b> —BASIC INTEGER CONVERSION ROUTINE.
46324	B4F4	<b>GIVABF</b> —RETURN TO BASIC.
49152	C000	CARTRIDGE MEMORY.
65280	FF00	I/O BUFFERS. (Joystick button, byte 1).
65314	FF22	CONTROL REGISTER.
65472	FFC0-1	VDG REG., BIT 0, OFF/ON.
65474	FFC2-3	VDG REG., BIT 1, OFF/ON.
65476	FFC4-5	VDG REG., BIT 2, OFF/ON.
65478	FFC6-7	PAGE SELECT REG., BIT 0, OFF/ON.
65480	FFC8-9	PAGE SELECT REG., BIT 1, OFF/ON.
65482	FFCA-B	PAGE SELECT REG., BIT 2, OFF/ON.
65484	FFCC-D	PAGE SELECT REG., BIT 3, OFF/ON.
65486	FFCE-F	PAGE SELECT REG., BIT 4, OFF/ON.
65488	FFD0-1	PAGE SELECT REG., BIT 5, OFF/ON.
65490	FFD2-3	PAGE SELECT REG., BIT 6, OFF/ON.
65536	FFFF	END OF 64K.

### BASIC RESERVED WORDS

↑	CLS	FN	MEM	POS	SKIPF
'	COLOR	FOR	MID\$	PPOINT	SOUND
*	CONT	GET	MOTOR	PRESET	SQR
+	COS	GOSUB	NEW	PRINT	STEP
-	CSAVE	GOTO	NEXT	PSET	STOP
<	CSAVEM	HEX\$	NOT	PUT	STR\$
=	DATA	IF	OFF	READ	STRING\$
>	DEF	INKEY\$	ON	REM	SUB
ABS	DEL	INPUT	OPEN	RENUM	TAB
AND	DIM	INSTR	OR	RESET	TAN
ASC	DLOAD	INT	PAINT	RESTORE	THEN
ATN	DRAW	JOYSTK	PCLEAR	RETURN	TIMER
AUDIO	EDIT	LEFT\$	PCLS	RIGHT\$	TO
CHR\$	ELSE	LEN	PCOPY	RND	TROFF
CIRCLE	END	LET	PEEK	RUN	TRON
CLEAR	EOF	LINE	PLAY	SCREEN	USING
CLOAD	EXEC	LIST	PMODE	SET	USR
CLOADM	EXP	LLIST	POINT	SGN	VAL
CLOSE	FIX	LOG	POKE	SIN	VARPTR

### BASIC INTERNAL CODES

DEC.	HEX	BASIC KYWD	DEC.	HEX	BASIC KYWD	DEC. (255+)	HEX (FF+)	BASIC KYWD
128	80	FOR	167	A7	THEN	128	80	SGN
129	81	GO	168	A8	NOT	129	81	INT
130	82	REM	169	A9	STEP	130	82	ABS
131	83	'	170	AA	OFF	131	83	USR
132	84	ELSE	171	AB	+	132	84	RND
133	85	IF	172	AC	-	133	85	SIN
134	86	DATA	173	AD	*	134	86	PEEK
135	87	PRINT	174	AE	/	135	87	LEN
136	88	ON	175	AF	↑	136	88	STR\$
137	89	INPUT	176	B0	AND	137	89	VAL
138	8A	END	177	B1	OR	138	8A	ASC
139	8B	NEXT	178	B2	>	139	8B	CHR\$
140	8C	DIM	179	B3	=	140	8C	EOF
141	8D	READ	180	B4	<	141	8D	JOYSTK
142	8E	RUN	181	B5	DEL	142	8E	LEFT\$
143	8F	RESTORE	182	B6	EDIT	143	8F	RIGHT\$
144	90	RETURN	183	B7	TRON	144	90	MID\$
145	91	STOP	184	B8	TROFF	145	91	POINT
146	92	POKE	185	B9	DEF	146	92	INKEY\$
147	93	CONT	186	BA	LET	147	93	MEM
148	94	LIST	187	BB	LINE	148	94	ATN
149	95	CLEAR	188	BC	PCLS	149	95	COS
150	96	NEW	189	BD	PSET	150	96	TAN
151	97	CLOAD	190	BE	PRESET	151	97	EXP
152	98	CSAVE	191	BF	SCREEN	152	98	FIX
153	99	OPEN	192	C0	PCLEAR	153	99	LOG
154	9A	CLOSE	193	C1	COLOR	154	9A	POS
155	9B	LLIST	194	C2	CIRCLE	155	9B	SQR
156	9C	SET	195	C3	PAINT	156	9C	HEX\$
157	9D	RESET	196	C4	GET	157	9D	VARPTR
158	9E	CLS	197	C5	PUT	158	9E	INSTR
159	9F	MOTOR	198	C6	DRAW	159	9F	TIMER
160	A0	SOUND	199	C7	PCOPY	160	A0	PPOINT
161	A1	AUDIO	200	C8	PMODE	161	A1	STRING\$
162	A2	EXEC	201	C9	PLAY			
163	A3	SKIPF	202	CA	DLOAD			
164	A4	TAB(	203	CB	RENUM			
165	A5	GOTO	204	CC	FN			
166	A6	GOSUB	205	CD	USING			

### TO VIEW A SECTION OF A LONG PROGRAM

Enter LIST followed by ⊗@. Enter any key to continue list. Enter BREAK to completely get out of LIST. (This procedure can also be used to pause program execution.)

### TO MAKE MOST STATEMENTS EXECUTE TWICE AS FAST

Do a POKE 65495,0. To return to normal speed, do a POKE 65494,0. Some I/O operations to cassette, printer, etc., may not work at higher speed.

### TO FREE THE MAXIMUM MEMORY FOR PROGRAM USE

Enter: POKE 25,6: NEW: CLEAR 0:

### TO LEARN ALL THE FILES ON A TAPE

Enter: SKIPF"X"  
(Where "X" is a name not expected to be on the tape).

### TO GET THE BEST VISUAL EFFECTS ON GRAPHICS

Use two separate screen areas. Build graphics in the second while displaying the first, then flip and build graphics in the first while displaying the second:

```
10 PG=1:X=10:COLOR 7,5
20 PMODE 1,PG 'POINT TO SCREEN AREA
30 PCLS: LINE (X,90)-(X+15,105),PSET,BF
50 SCREEN 1,1 'FLASH CURRENT SCREEN AREA.
60 X=X+15:IF X>225 THEN 10
70 IFPG=1THENPG=3ELSEPG=1 'FLIP AREAS
80 GOTO 20
```

### PLAY WITH PMODE 4

It is documented as a 2-color mode but other colors can be gotten by PSET on every other point, every third point, etc.

### HEX/DEC CONVERSION CHART

8 4 2 1		8 4 2 1		8 4 2 1		8 4 2 1	
HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC
0	0	0	0	0	0	0	0
1	4096	1	256	1	16	1	1
2	8192	2	512	2	32	2	2
3	12288	3	768	3	48	3	3
4	16384	4	1024	4	64	4	4
5	20480	5	1280	5	80	5	5
6	24576	6	1536	6	96	6	6
7	28672	7	1792	7	112	7	7
8	32768	8	2048	8	128	8	8
9	36864	9	2304	9	144	9	9
A	40960	A	2560	A	160	A	10
B	45056	B	2816	B	176	B	11
C	49152	C	3072	C	192	C	12
D	53248	D	3328	D	208	D	13
E	57344	E	3584	E	224	E	14
F	61440	F	3840	F	240	F	15
65535		4095		255		15	

DEC	HEX	KEYS	POKE	PRINT
0	00		(@)	NUL
1	01		(A)	SOH
2	02		(B)	STX
3	03	BREAK	(C)	ETX
4	04		(D)	EOT
5	05		(E)	ENQ
6	06		(F)	ACK
7	07		(G)	BEL
8	08	←	(H)	BS
9	09	→	(I)	HT
10	0A	↓	(J)	LF
11	0B		(K)	VT
12	0C	CLEAR	(L)	FF
13	0D	ENTER	(M)	CR
14	0E		(N)	SO
15	0F		(O)	SI
16	10		(P)	DLE
17	11		(Q)	DC1
18	12	⊗0	(R)	DC2
19	13	⊗@	(S)	DC3
20	14		(T)	DC4
21	15	⊗←	(U)	NAK
22	16		(V)	SYN
23	17		(W)	ETB
24	18		(X)	CAN
25	19		(Y)	EM
26	1A		(Z)	SUB
27	1B		([	ESC
28	1C		(\)	FS
29	1D		(])	GS
30	1E		(↑)	RS
31	1F		(←)	US
32	20	SPACE	SPACE	SPACE
33	21	!	(!)	!
34	22	"	(")	"
35	23	#	(#)	#
36	24	\$	(\$)	\$
37	25	%	(%)	%
38	26	&	(&)	&
39	27	'	(')	'
40	28	(	((	(
41	29	)	)	)
42	2A	*	(*	*
43	2B	+	(+)	+
44	2C	,	(,)	,
45	2D	-	(-)	-
46	2E	.	(.)	.
47	2F	/	(/)	/
48	30	0	(0)	0
49	31	1	(1)	1
50	32	2	(2)	2
51	33	3	(3)	3
52	34	4	(4)	4
53	35	5	(5)	5
54	36	6	(6)	6
55	37	7	(7)	7
56	38	8	(8)	8
57	39	9	(9)	9
58	3A	:	(:)	:
59	3B	;	(;)	;
60	3C	<	(<)	<
61	3D	=	(=)	=
62	3E	>	(>)	>
63	3F	?	(?)	?

⊗=SHIFT

DEC	HEX	KEYS	POKE	PRINT
64	40	@	@	@
65	41	⊗A	A	A
66	42	⊗B	B	B
67	43	⊗C	C	C
68	44	⊗D	D	D
69	45	⊗E	E	E
70	46	⊗F	F	F
71	47	⊗G	G	G
72	48	⊗H	H	H
73	49	⊗I	I	I
74	4A	⊗J	J	J
75	4B	⊗K	K	K
76	4C	⊗L	L	L
77	4D	⊗M	M	M
78	4E	⊗N	N	N
79	4F	⊗O	O	O
80	50	⊗P	P	P
81	51	⊗Q	Q	Q
82	52	⊗R	R	R
83	53	⊗S	S	S
84	54	⊗T	T	T
85	55	⊗U	U	U
86	56	⊗V	V	V
87	57	⊗W	W	W
88	58	⊗X	X	X
89	59	⊗Y	Y	Y
90	5A	⊗Z	Z	Z
91	5B	⊗↓	[	[
92	5C	⊗CLEAR	\	\
93	5D	⊗→	]	]
94	5E	↑	↑	↑
95	5F	⊗↑	←	←
96	60		SPACE	@
97	61	A	!	a
98	62	B	"	b
99	63	C	#	c
100	64	D	\$	d
101	65	E	%	e
102	66	F	&	f
103	67	G	'	g
104	68	H	(	h
105	69	I	)	i
106	6A	J	*	j
107	6B	K	+	k
108	6C	L	,	l
109	6D	M	-	m
110	6E	N	.	n
111	6F	O	/	o
112	70	P	0	p
113	71	Q	1	q
114	72	R	2	r
115	73	S	3	s
116	74	T	4	t
117	75	U	5	u
118	76	V	6	v
119	77	W	7	w
120	78	X	8	x
121	79	Y	9	y
122	7A	Z	:	z
123	7B		;	[
124	7C		<	\
125	7D		=	]
126	7E		>	↑
127	7F		?	←

⊗=SHIFT

DEC	HEX	GRAPHICS	DEC	HEX	GRAPHICS
128	80		192	C0	
129	81		193	C1	
130	82		194	C2	
131	83		195	C3	
132	84		196	C4	
133	85		197	C5	
134	86		198	C6	
135	87		199	C7	
136	88		200	C8	
137	89		201	C9	
138	8A		202	CA	
139	8B		203	CB	
140	8C		204	CC	
141	8D		205	CD	
142	8E		206	CE	
143	8F		207	CF	
144	90		208	D0	
145	91		209	D1	
146	92		210	D2	
147	93		211	D3	
148	94		212	D4	
149	95		213	D5	
150	96		214	D6	
151	97		215	D7	
152	98		216	D8	
153	99		217	D9	
154	9A		218	DA	
155	9B		219	DB	
156	9C		220	DC	
157	9D		221	DD	
158	9E		222	DE	
159	9F		223	DF	
160	A0		224	E0	
161	A1		225	E1	
162	A2		226	E2	
163	A3		227	E3	
164	A4		228	E4	
165	A5		229	E5	
166	A6		230	E6	
167	A7		231	E7	
168	A8		232	E8	
169	A9		233	E9	
170	AA		234	EA	
171	AB		235	EB	
172	AC		236	EC	
173	AD		237	ED	
174	AE		238	EE	
175	AF		239	EF	
176	B0		240	F0	
177	B1		241	F1	
178	B2		242	F2	
179	B3		243	F3	
180	B4		244	F4	
181	B5		245	F5	
182	B6		246	F6	
183	B7		247	F7	
184	B8		248	F8	
185	B9		249	F9	
186	BA		250	FA	
187	BB		251	FB	
188	BC		252	FC	
189	BD		253	FD	
190	BE		254	FE	
191	BF		255	FF	



## SCREEN LINE LAYOUT

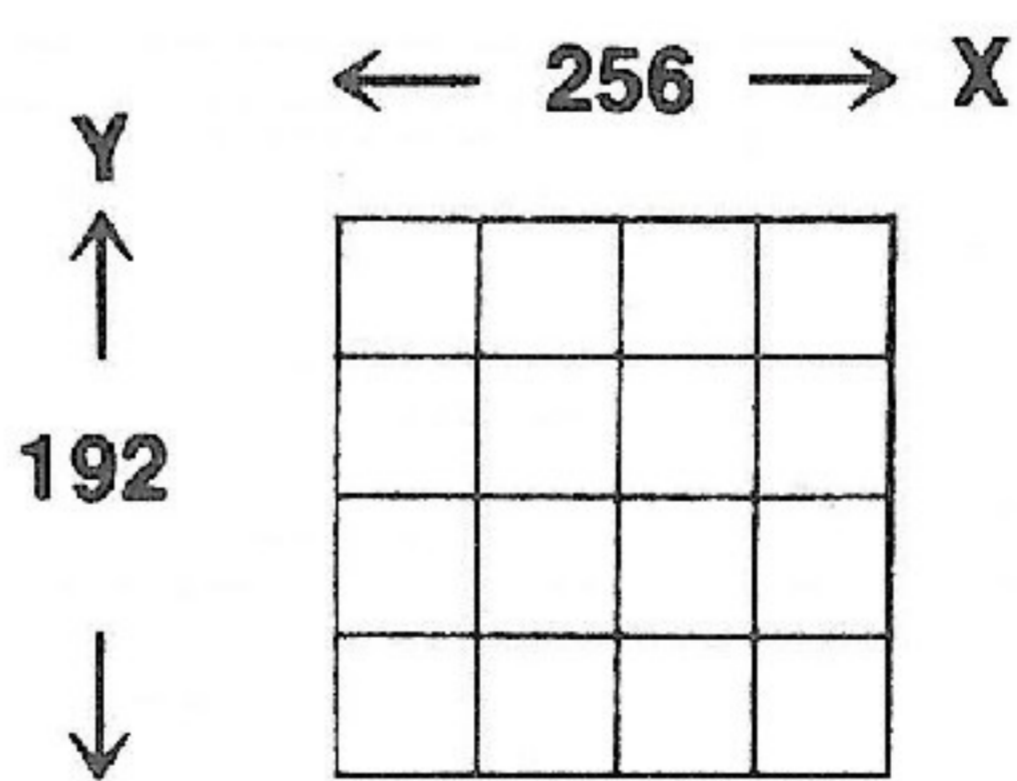
LINE	PRINT	SET	POKE	HEX
1	0	0 1	1024	0400
2	32	2 3	1056	0420
3	64	4 5	1088	0440
4	96	6 7	1120	0460
5	128	8 9	1152	0480
6	160	10 11	1184	04A0
7	192	12 13	1216	04C0
8	224	14 15	1248	04E0
9	256	16 17	1280	0500
10	288	18 19	1312	0520
11	320	20 21	1344	0540
12	352	22 23	1376	0560
13	384	24 25	1408	0580
14	416	26 27	1440	05A0
15	448	28 29	1472	05C0
16	480	30 31	1504	05E0
	511		1535	05FF

### EXTENDED GRAPHICS

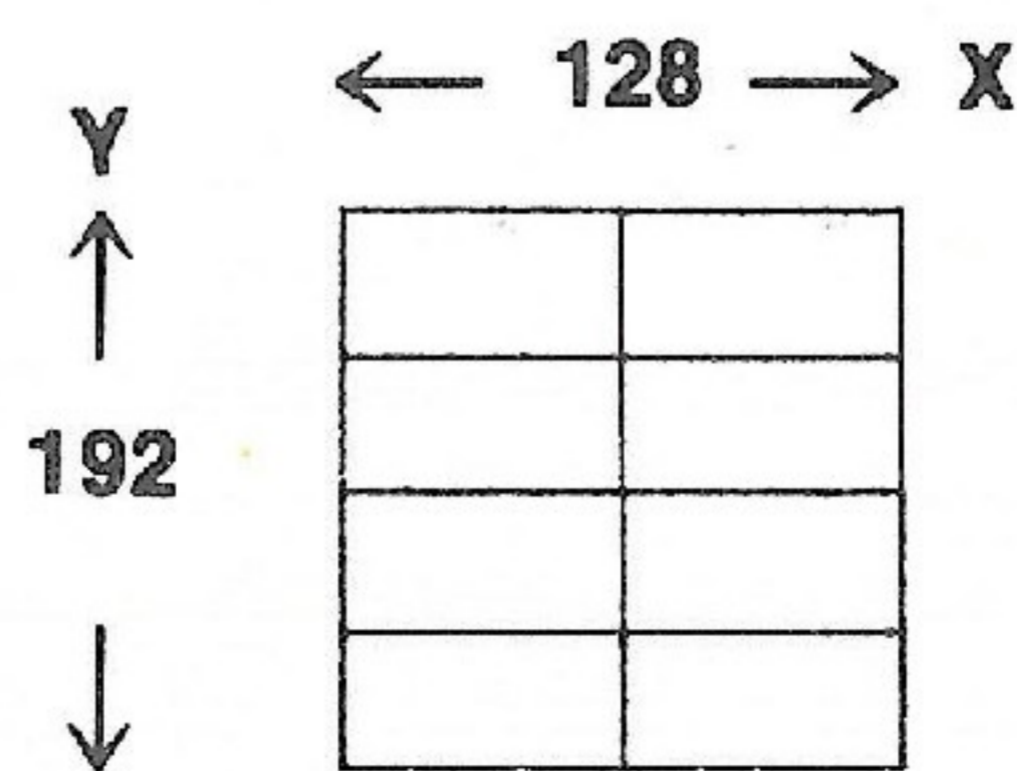
**PMODE**

Points per square

**4**



**3,2**



**1,0**

