

[BULLETIN 10 MISSING]



BULLETIN NO. 11  
DECEMBER 22, 1982

TO: DISTRIBUTION  
FROM: ARD SOFTWARE ENGINEERING *Daidy, King*  
SUBJECT: RELEASE OF COLECOVISION PROGRAMMERS  
MANUAL REV. 5

cc: Eric Bromley  
Robert Schenck  
Marshall Caras  
Tom Helmer

---

The ColecoVision Programmer's Manual Rev. 5 has been released. This manual is written for the applications programmer and is intended as both a day-to-day reference source as well as a training document for programmers new to ColecoVision.

This new edition contains the overview for both hardware and software. Subsequently, detail descriptions are given in the areas of:

- Graphics Generation Software
- Interrupt Handling
- Timing
- Controller Software
- Sound Generation Software
- Boot up Software and Utilities
- Defined Reference Locations

The Rev. 5 manual pertains to the current production OS\_7. Fundamental knowledge of the OS is presented in the manual without elaborating on application examples and design approaches. These materials will be documented in the proposed ColecoVision Applications Manual, scheduled to be released in second quarter 1983.

In the Appendix B you will find the graphics documentation (Rev. 1.0) has been updated with addition of materials describing PUT\_SPRIT and PUT\_COMPLEX.

The Sound documentation also received updates in the form of Notes and Errata attached at the end of Appendix C.

User feedback should be addressed to the Manager of Software Engineering of Coleco ARD. All adopted changes will be brought to your attention via ColecoVision Bulletin announcements.

This manual is confidential and should not be copied. All releases have to be signed out through the ARD Engineering secretary S. Kakowski.

---

DISTRIBUTION: C. Baldyga  
R. Dionne  
A. Godfrey  
L. Gray  
C. Hager  
R. Harris

K. Lagace  
J. Michaels  
M. Minto  
A. Nguyen  
L. Olbrych  
D. Schulze

CONFIDENTIAL





INDUSTRIES, INC.

845 ASYLUM AVENUE HARTFORD, CT 06105 (203) 278-0280

Executive Office

ColecoVision Software Bulletin

BULLETIN NO. 0012  
March 17, 1983

TO: DISTRIBUTION  
FROM: ARD SOFTWARE ENGINEERING DKH KAL  
RE: CORRECTIONS IN REGARD TO BULLETIN NO. 0004

- (1) The statement that "Sound Data Areas are off limits to programmers" is not true.
- (2) The "Null Song" method wastes CROM space. Writing OFFH to the first byte of the song's sound area IS recommended.

Since the ColecoVision Operating System turns off sounds by placing OFFH into the first byte of the Sound Data Areas anyway and changing the data structures of the Sound Data Areas would entail changing the operating system. It has been proven that the above method is the fastest and most direct way to abort sounds.

The "null song" method may still be used, but each additional song uses at least five bytes of CROM; four for the LST\_OF\_SND\_ADDRS and one for the END code.

DISTRIBUTION:

- |              |             |
|--------------|-------------|
| C. Baldyga   | K. Lagace   |
| R. Dionne    | J. Michaels |
| A. Godfrey   | M. Minto    |
| L. Gray      | A. Nguyen   |
| C. Hager     | L. Olbrych  |
| R. Harris    | R. Rizzo    |
| L. Henderson | D. Schulze  |
| R. Jepson    | D. Stern    |
| D. Jonker    | D. Thompson |
|              | B. Zawislak |

- CC: E. Bromley  
R. Schenck  
C. M. Caras  
T. Helmer

CONFIDENTIAL





Executive Office

845 ASYLUM AVENUE HARTFORD, CT 06105 (203) 278-0280

ColecoVision Software Bulletin

Bulletin No. 0013  
April 4, 1983

TO: Distribution  
FROM: ARD Software Engineering *DKH RFS*  
RE: Release of Additional ColecoVision OS Entry Points

The following is a list of additional entry points to the ColecoVision OS ROM.

PX_TO_PTRN_POS	EQU	07E8H
PUT_FRAME	EQU	080BH
GET_BKGRND	EQU	0898H
CALC_OFFSET	EQU	08C0H

Attached is a brief description of the routines which correspond to the entry points released.

DISTRIBUTION

CC: E. Bromley  
R. Schenck  
D. Hwang  
C. M. Caras  
T. Helmer  
File

C. Baldyga	J. Michaels
R. Dionne	M. Minto
L. Gray	A. Nguyen
C. Hager	L. Olbrych
R. Harris	R. Rizzo
L. Henderson	B. Rochette
R. Jepson	D. Schulze
D. Jonker	D. Stern
K. Lagace	D. Thompson
	B. Zawislak

CONFIDENTIAL

Here are the graphic subroutines which would be useful to have access to, along with a brief description of what each one does.

XX

PX\_TO\_PTRN\_POS (Pixel to pattern plane position)  
(entry point xxxxH)

This routine divides the 16 bit signed value in the DE register pair by 8. An 8 bit signed result is returned in register E. Results of less than -127 are returned as -128, results of greater than +126 are returned as +127.

If this routine is passed the X(or Y) pixel coordinate position of a point on the pattern plane, the X(or Y) coordinate in pattern positions will be returned.

INPUT: DE = N (16 bit signed number)

OUTPUT: N/8 < -128 E = -128  
-128 <= N/8 <= 127 E = N/8  
N/8 > +126 E = +127

REGISTERS AFFECTED:

FLAGS  
DE

XX

PUTFRAME  
(entry point xxxxH)

**CONFIDENTIAL**

PUTFRAME moves data from cpu RAM to the Pattern Name Table in URAM. The data is assumed to be an array of Pattern Generator Names which when moved to the Pattern Name Table, will produce a rectangular graphic, or frame, composed of the patterns specified by these Pattern Generator Names. The array must be arranged in row major order.

The dimensions of array are passed to the routine in the BC register pair. These dimensions also define the height and width (in pattern plane positions) of the frame when displayed.

The upper left corner of the frame will appear on the pattern plane at a position determined by Y\_PAT\_POS and X\_PAT\_POS which are passed in the DE register pair. Y and X\_PAT\_POS are row and column coordinates in pattern plane positions as measured from the upper left corner of the pattern plane. Y and X\_PAT\_POS are interpreted as 8 bit signed values and, therefore, the corner of the frame may placed anywhere within or outside the boundaries

of the pattern plane. Therefore, the frame itself may be placed partially off screen in any direction.

The HL register pair must contain the address of the start of the array of pattern names.

INPUT:            HL = Address of array in CPU RAM  
                   B = Y dimension of array and Y\_EXTENT of frame  
                   C = X dimension of array and X\_EXTENT of frame  
                   D = Y\_PAT\_POS of upper left corner of frame  
                   E = X\_PAT\_POS of upper left corner of frame

OUTPUT:            Modifies URAM name table

REGISTERS AFFECTED:

                  All registers used

-----  
 As an example, if an array exists in CPU memory space which looks like...

ARRAY: DB        0,1,2,3,4,5

and the first six pattern generators in URAM have been initialized with the following patterns...

Pattern Generator #	Graphic
0	A
1	B
2	C
3	D
4	E
5	F

Then the following code sequence...

```
LD HL,ARRAY
LD B,2            ;B := Y_EXTENT
LD C,3            ;C := X_EXTENT
LD D,2            ;D := Y_PAT_POS
LD E,-1           ;E := X_PAT_POS
CALL PUT_FRAME
```

will produce this display...

```

          0 X_PAT_POS ->
Y_PAT_POS . . . . .
V D. . . . .
          .B.C. . . . .
          .E.F. . . . .
          . . . . .
```

(diagram of upper left corner of pattern plane)

**CONFIDENTIAL**

Note: Patterns A and D are not seen, since they would be to the left of the left-hand edge of the pattern plane.

\*\*\*\*\*

GET\_BKGRND  
(entry point xxxxH)

This routine is the inverse of the PUT\_FRAME routine described above. GET\_BKGRND moves an array of names from the pattern name table in VRAM into CPU RAM. The dimensions of the array and the position of the upper left corner of the frame it defines, are passed to the routine in same manner as in PUT\_FRAME. The names are moved to the location in CPU RAM specified by the contents of the HL register pair.

If part of the frame extends beyond the pattern plane, the names that correspond to positions which are not on the pattern plane will not be defined.

INPUTS:           HL = Destination address in CPU RAM to which  
                  names will be moved  
                  B = Y\_EXTENT of frame  
                  C = X\_EXTENT of frame  
                  D = Y\_PAT\_POS of upper left corner of frame  
                  E = X\_PAT\_POS of upper left corner of frame

OUTPUTS:           CPU RAM from HL to HL+(BxC)-1 filled with names  
                  from pattern name table

REGISTERS AFFECTED:

    All registers used

\*\*\*\*\*

CALC\_OFFSET  
(entry point xxxxH)

This routine calculates the offset from the start of the pattern name table corresponding to a pattern plane position specified by the coordinates Y\_PAT\_POS and X\_PAT\_POS.

The coordinates are passed to, and the result is passed back in the DE register pair.

INPUTS:           D = Y\_PAT\_POS  
                  E = X\_PAT\_POS

OUTPUTS:           DE = Offset from start of pattern name table

REGISTERS AFFECTED:  
    FLAGS  
    DE

**CONFIDENTIAL**



Executive Office:

245 ASYLUM AVENUE HARTFORD, CT 06105 (203) 278-0280

ColecoVison Software Bulletin

Bulletin No. 0014  
April 12, 1983

TO: Distribution  
FROM: ARD Software Engineering  
RE: OS\_SYMBOLS Rev.4

DKH  
RFJ

Attached please find a listing of OS\_SYMBOLS Rev. 4. This listing holds all ColecoVision OS reserved data entry points released to date.

Attachment

Distribution:

C. Baldyga  
R. Dionne  
L. Gray  
C. Hager  
R. Harris  
L. Henderson  
R. Jepson  
D. Jonker  
K. Lagace  
J. Michaels

J. Milano  
M. Minto  
A. Nguyen  
L. Olbrych  
R. Rizzo  
B. Rochette  
D. Schulze  
D. Stern  
D. Taylor  
D. Thompson  
B. Zawislak

CC:

E. Bromley  
C. M. Caras  
T. A. Helmer  
D. Hwang  
R. Schenck  
File



LOCATION OBJECT CODE LINE SOURCE LINE

```

1 "Z80"
3 NAME "Rev 4 - RFJ"
4
5 DESCRIPTION MACRO
6 .CUTO ENDESCRIPTION
7
8 Author: Zac Smith
9 User: OS
10 Starting date: 13May1982
11 Header Rev: 1
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

OS BY MAIL: S  
 Coleco Modification Operation By mail  
 Software Release 1.0  
 Advanced Research and Development  
 Coleco Industries

CONFIDENTIAL

List of access points to the ColecoVision Operating system ROM.  
 Only those points listed in this file have been approved as absolute  
 locations of which the cartridge developer can access the OS rom.  
 Additionally, access to any memory locations indirectly, or by  
 offset to locations defined herein is denied except where defined by  
 the ColecoVision Programmer's Manual ( current rev 15 ).

List of OS symbols in alphabetical order with defining and referencing  
 modules (if any).

Rev History (one line note indicating the change)

Rev.	Date	Name	Change
4	13Apr1359	Rob	Remove Zaxxon related documentation in preparation for re-release of this file for general distribution
	11Apr1626	Rob	Added PUTFRAME (no underline) to match label in OS listing. Kept PUT_FRAME due to Software Bulletin released.
	11Apr 900	Rob	Updated Header to expand the description of this file.
3	05Apr1444	Rob	GLRD location added in rev 3 Added location: PX TO PTRN_PUS PUT_FRAME GET BKGRND CRIC OFFSET Zaxxon Development.
2	13Apr1114p	Rob	Added documentation specific to Zaxxon Development.
1	2Sep1153p	Ken Logate	Added 9 SOUND OS equates
0	13May	Zac Smith	Initial Jump table equates DATE 1 5/13/82 FOR REV 1 5 (OS 5:05)

ENDESCRIPTION:  
 HEAD

CATION OBJECT CODE LINE SOURCE LINE

59	iSymbol	Absolute Address	Partial Xref of routines used by other OS routines
60	iName		
61			
62			
63			Start of defined reference points
(1FF7)	64 ACTIVATE	EQV 01FF7H	
(1F64)	65 ACTIVATER	EQV 01F64H	
(0181)	66 ADDR16	EQV 00181H	
(0069)	67 AMERICA	EQV 00069H	
(006A)	68 ASCII_TABLE	EQV 0006AH	
(012F)	69 ATM_SWEEP	EQV 0012FH	
(00C0)	70 CALC_OFFSET	EQV 000C0H	
(0000)	71 CARTRIDGE	EQV 00000H	
(8000)	72 CONTROLLER_MAP	EQV 08000H	iCONTROLLERIOS
(0190)	73 DECLSN	EQV 00190H	
(019B)	74 DECM5N	EQV 0019BH	
(1F79)	75 DECODEP	EQV 01F79H	
(73C6)	76 DEFER_WRITES	EQV 073C6H	iPUT_OBJECTIONS
(02EE)	77 EFXOVER	EQV 002EEH	
(1F73)	78 ENLARGE	EQV 01F73H	
(1D6C)	79 ENLRC	EQV 01D6CH	
(1F82)	80 FILL_VRAM	EQV 01F82H	
(1FCA)	81 FREE_SIGNAL	EQV 01FCAH	
(1F9D)	82 FREE_SIGNALP	EQV 01F9DH	
(00FC)	83 FREQ_SWEEP	EQV 000FCH	
(8024)	84 GAME_NAME	EQV 00024H	
(1F7C)	85 GAME_OPT	EQV 01F7CH	
(8098)	86 GET_BKGRND	EQV 00898H	
(1F8A)	87 GET_VRAM	EQV 01F8AH	
(1F8E)	88 GET_VRAMP	EQV 01F8EH	
(1FC1)	89 INIT_SPR_ORDER	EQV 01FC1H	
(1F94)	90 INIT_SPR_ORDERP	EQV 01F94H	
(1F8B)	91 INIT_TABLE	EQV 01F8BH	
(1F89)	92 INIT_TABLEP	EQV 01F89H	
(1FC7)	93 INIT_TIMER	EQV 01FC7H	
(1F9A)	94 INIT_TIMERP	EQV 01F9AH	
(1FES)	95 INIT_WRITER	EQV 01FESH	
(1FAF)	96 INIT_WRITERP	EQV 01FAFH	
(801E)	97 IRQ_INT_VECT	EQV 0001EH	
(01D5)	98 LEAVE_EFFECT	EQV 001D5H	
(1F7F)	99 LOAD_ASCII	EQV 01F7FH	
(8082)	100 LOCAL_SPR_TBL	EQV 00082H	
(1F85)	101 MODE_1	EQV 01F85H	
(01A6)	102 MSNTOLSH	EQV 001A6H	
(73C7)	103 MIX_6PRITEB	EQV 073C7H	
(8021)	104 MNI_INT_VECT	EQV 00021H	
(006C)	105 NUMRER_TABLE	EQV 0006CH	
(1FF1)	106 PLAY_IT	EQV 01FF1H	
(1FB5)	107 PLAY_ITP	EQV 01FB5H	
(1F61)	108 PLAY_SONGS	EQV 01F61H	
(1FER)	109 POLLER	EQV 01FEH	
(0108)	110 PUTFRAME	EQV 000108H	
(1FAA)	111 PUTOBJ	EQV 01FAAH	
(1F67)	112 PUTORIP	EQV 01F67H	
(0808)	113 PUT_FRAME	EQV 00808H	
(1F6E)	114 PUT_VRAM	EQV 01F6EH	
(1F71)	115 PUT_VRAMP	EQV 01F71H	
			iGAME_OPT108 iOC0105 iPUT_MOBILIOS iPUT_SPR105
			iGAME_OPT109 LOGO105
			iGAME_OPT105
			iTABLE_MA105
			iTABLE_MA108
			iTABLE_MA109
			iPUT_CMPLX105

CONFIDENTIAL

LOCATION UN-CT CODE LINE SOURCE LINE

```

(07E8) 116 PX_TO_PTRN_POB EQU 007E8H
(1FFD) 117 RAND_GEN EQU 01F1D8H
(73C8) 118 RAND_NUM EQU 073C8H
(1FDC) 119 READ_REGISTER EQU 01FDCH
(1FE2) 120 READ_VRAM EQU 01FE2H
(1FAC) 121 READ_VRAM EQU 01FACH
(1F6D) 122 REFLECT_HORIZONTAL EQU 01F6DH
(1F6A) 123 REFLECT_VERTICAL EQU 01F6AH
(1FCD) 124 REQUEST_SIGNAL EQU 01FCDH
(1FA0) 125 REQUEST_SIGNALP EQU 01FA0H
(1F78) 126 ROTATE_90 EQU 01F70H
(800F) 127 RST_10H_RAM EQU 800FH
(8012) 128 RST_18H_RAM EQU 8012H
(8015) 129 RST_20H_RAM EQU 8015H
(8018) 130 RST_28H_RAM EQU 8018H
(801B) 131 RST_38H_RAM EQU 801BH
(800C) 132 RST_8H_RAM EQU 800CH
(1FEE) 133 SOUND_INIT EQU 01FEEH
(1FB2) 134 SOUND_INITP EQU 01FB2H
(1FF4) 135 SOUND_NAN EQU 01FF4H
(8004) 136 SPRITE_ORDER EQU 8004H
(7349) 137 STACK EQU 07349H
(800A) 138 START_GAME EQU 800AH
(1FD0) 139 TEST_SIGNAL EQU 01FD0H
(1FA3) 140 TEST_SIGNALP EQU 01FA3H
(1FD3) 141 TIME_MGR EQU 01FD3H
(1FD6) 142 TURN_OFF_SOUND EQU 01FD6H
(1F88) 143 UPDATE_SPINNER EQU 01F88H
(73C3) 144 VDP_MODE_WORD EQU 073C3H
(73C5) 145 VDP_STATUS_BYTE EQU 073C5H
(8006) 146 WORK_BUFFER EQU 8006H
(1FEB) 147 WRITER EQU 01FEBH
(1FD9) 148 WRITE_REGISTER EQU 01FD9H
(1FA6) 149 WRITE_REGISTERP EQU 01FA6H
(1FDF) 150 WRITE_VRAM EQU 01FD FH
(1FA9) 151 WRITE_VRAM EQU 01FA9H
(1FCA) 152 WR_SPR_NH_TBL EQU 01FCAH
(1F77) 153 WR_SPR_NH_TBLP EQU 01F77H
154

```

!LOCO:05  
!PUT\_MOBIL:05

!TABLE\_MA:05

!LOCO:05

!GRAPHICB:05 VD\_DRIVER:05 TABLE\_MA:05 PUT\_MOBIL:05 ACT2:05  
!GRAPHICB:05 PUT\_MOBIL:05 PUT\_SPR:05 PUTSEM:05 ACT2:05  
!GAME\_OPT:05 LOCO:05  
!GAME\_OPT:05 LOCO:05 PUT\_MOBIL:05

!End of defined reference points

LOCATION OBJECT CODE LINE SOURCE LINE

```

156
157
158 GLB ACTIVATE
159 GLB ACTIVATIEP
160 GLB ADDR16
161 GLB AMERICA
162 GLB ASCII_TABLE
163 GLB ATN_SWEEP
164 GLB CALC_OFFSET
165 GLB CARTRIDGE
166 GLB CONTROLLER_MAP
167 GLB DECLSN
168 GLB DECM5H
169 GLB DECODER
170 GLB DEFER_WRITEB
171 GLB EF_XOVER
172 GLB ENLARGE
173 GLB ENLRC
174 GLB FILL_VRAM
175 GLB FREE_SIGNAL
176 GLB FREE_SIGNALP
177 GLB FREQ_SWEEP
178 GLB GAME_NAME
179 GLB GAME_OPT
180 GLB GET_BKGRND
181 GLB GET_VRAM
182 GLB GET_VRAMP
183 GLB INIT_SPR_ORDER
184 GLB INIT_SPR_ORDERP
185 GLB INIT_TABLE
186 GLB INIT_TABLEP
187 GLB INIT_TIMER
188 GLB INIT_TIMERP
189 GLB INIT_WRITER
190 GLB INIT_WRITERP
191 GLB IRQ_INT_VECT
192 GLB LEAVE_EFFECT
193 GLB LOAD_ASCII
194 GLB LOCAL_SPR_TBL
195 GLB MSGTOLSN
196 GLB MDWE_1
197 GLB NUX_SPRITES
198 GLB NINI_INT_VECT
199 GLB NUMBER_TABLE
200 GLB PLAY_IT
201 GLB PLAY_ITP
202 GLB PLAY_GONGS
203 GLB POLLER
204 GLB PUTFRAME
205 GLB PUTDHIJ
206 GLB PUTDHIJP
207 GLB PUTFRAME
208 GLB PUT_VRAM
209 GLB PUT_VRAMP
210 GLB PX_TO_PTRN_POS
211 GLB RAND_GEN
212 GLB RND_NUM

```

The following defines each access point  
; as Global.

CONFIDENTIAL

LOCATION	OBJECT CODE LINE	SOURCE LINE
	213	GLB READ_REGISTER
	214	GLB READ_VRAM
	215	CLR READ_VRAM
	216	GLB REFLECT_HORIZON
	217	GLB REFLECT_VERTICAL
	218	GLB REQUEST_SIGNAL
	219	GLB REQUEST_SIGNALP
	220	GLB ROTATE_90
	221	GLB RST_10H_RAM
	222	GLB RST_18H_RAM
	223	GLB RST_20H_RAM
	224	GLB RST_28H_RAM
	225	GLB RST_30H_RAM
	226	GLB RST_0H_RAM
	227	GLB SOUND_INIT
	228	GLB SOUND_INITP
	229	GLB SOUND_HAM
	230	GLB SPRITE_ORDER
	231	GLB STACK
	232	GLB START_GAME
	233	GLB TEST_SIGNAL
	234	GLB TEST_SIGNALP
	235	GLB TIME_MGR
	236	GLB TURN_OFF_SOUND
	237	GLB UPDATE_SPINNER
	238	GLB VDP_MODE_WORD
	239	GLB VDP_STATUS_BYTE
	240	GLB WORK_BUFFER
	241	GLB WRITER
	242	GLB WRITE_REGISTER
	243	GLB WRITE_REGISTERP
	244	GLB WRITE_VRAM
	245	GLB WRITE_VRAMP
	246	GLB WR_SPR_NM_TEL
	247	GLB WR_SPR_NM_TELP

Errors: 0

CONFIDENTIAL



LOCATION OBJECT CODE LINE SOURCE LINE

```

1 ^Z80^
3 NAME ^Rev 4 - RFJ^
4
5 DESCRIPTION MACRO
6 .GOTO ENDESCRIPTION
7
8 Author: Zac Smith
9 Userid: OS
10 Starting date: 13may1982
11 Header Rev: 1
12 #####
13 #####
14 ##### OS SYMBOLS
15 ##### ColecoVision_Operating_System
16 ##### Software_Engineering
17 ##### Advanced_Research_and_Development
18 ##### Coleco_Industries
19 #####
20 #####
21 ##### *** Confidential ***
22 #####
23 #####
24
25 List of access points to the Colecovision Operating system ROM.
26 Only these points listed in this file have been approved as absolute
27 locations of which the cartridge developer can access the OS rom.
28 Additionally, access to any memory locations indirectly, or by
29 offset to locations defined herein is denied except where defined by
30 the ColecoVision Programmer's Manual ( current rev 05 ).
31
32 List of OS symbols in alphabetical order with defining and referencing
33 modules (if any).

```

Rev History (one line note indicating the change)

Rev.	Date	Name	Change
4	13apr1359	Rob	Remove Zaxxon related documentation in preparation for re-release of this file for general distribution
	11apr1626	Rob	Added PUTFRAME (no underline) to match label in OS listing. Kept PUT_FRAME due to Software Bulletin released.
	11apr 900	Rob	Updated Header to expand the description of this file.
3	05apr1444	Rob	GLBed locations added in rev 3 Added locations PX TO PTRN_P05 PUT_FRAME GET_BKGRND_CALC_OFFSET
2	13sept1114p	Rob	Added documentation specific to Zaxxon Development.
1	2sept1153p	Ken Lagace	Added 9 SOUND OS equates
0	13may	Zac Smith	Initial Jump table equates DATE : 5/13/82 FOR REV : 5 (OS 5:05)

56 ENDESCRIPTION:  
57 NEND

LOCATION OBJECT CODE LINE SOURCE LINE

```

59 | Symbol | Absolute | Partial Xref of routines used
60 | Name | Address | by other OS routines
61 |-----|-----|-----|-----
62 | | | |
63 | | | | :Start of defined reference points
64 | (1FF7) | | |
65 | (1F64) | | |
66 | (01B1) | | |
67 | (0069) | | |
68 | (006A) | | |
69 | (012F) | | |
70 | (08C0) | | |
71 | (8000) | | |
72 | (8008) | | |
73 | (0190) | | |
74 | (019B) | | |
75 | (1F79) | | |
76 | (73C6) | | |
77 | (02EE) | | |
78 | (1F73) | | |
79 | (106C) | | |
80 | (1F82) | | |
81 | (1FCA) | | |
82 | (1F9D) | | |
83 | (00FC) | | |
84 | (8024) | | |
85 | (1F7C) | | |
86 | (0898) | | |
87 | (1F8B) | | |
88 | (1F8E) | | |
89 | (1FC1) | | |
90 | (1F94) | | |
91 | (1F88) | | |
92 | (1F8B) | | |
93 | (1FC7) | | |
94 | (1F9A) | | |
95 | (1FE5) | | |
96 | (1FAF) | | |
97 | (801E) | | |
98 | (01D5) | | |
99 | (1F7F) | | |
100 | (8002) | | |
101 | (1F85) | | |
102 | (01A6) | | |
103 | (73C7) | | |
104 | (8021) | | |
105 | (006C) | | |
106 | (1FF1) | | |
107 | (1FB5) | | |
108 | (1FG1) | | |
109 | (1FEB) | | |
110 | (800B) | | |
111 | (1FFA) | | |
112 | (1F67) | | |
113 | (8008) | | |
114 | (1FBE) | | |
115 | (1F91) | | |
64 | ACTIVATE | EQU 01FF7H |
65 | ACTIVATER | EQU 01F64H |
66 | ADDR16 | EQU 001B1H |
67 | AMERICA | EQU 00069H |
68 | ASCII_TABLE | EQU 0006AH |
69 | ATN_SWEEP | EQU 0012FH |
70 | CALC_OFFSET | EQU 008C0H |
71 | CARTRIDGE | EQU 08000H |
72 | CONTROLLER_MAP | EQU 08008H |
73 | DECLSN | EQU 00190H |
74 | DECM5N | EQU 0019BH |
75 | DECODER | EQU 01F79H |
76 | DEFER_WRITES | EQU 073C6H |
77 | EF_XOVER | EQU 002EEH |
78 | ENLARGE | EQU 01F73H |
79 | ENLARG | EQU 01D6CH |
80 | FILL_VRAM | EQU 01F82H |
81 | FREE_SIGNAL | EQU 01FCAH |
82 | FREE_SIGNALP | EQU 01F9DH |
83 | FREQ_SWEEP | EQU 000FCH |
84 | GAME_NAME | EQU 08024H |
85 | GAME_OPT | EQU 01F7CH |
86 | GET_BKGRND | EQU 00898H |
87 | GET_VRAM | EQU 01F8BH |
88 | GET_VRAMP | EQU 01F8EH |
89 | INIT_SPR_ORDER | EQU 01FC1H |
90 | INIT_SPR_ORDERP | EQU 01F94H |
91 | INIT_TABLE | EQU 01F88H |
92 | INIT_TABLEP | EQU 01F8BH |
93 | INIT_TIMER | EQU 01FC7H |
94 | INIT_TIMERP | EQU 01F9AH |
95 | INIT_WRITER | EQU 01FE5H |
96 | INIT_WRITERP | EQU 01FAFH |
97 | IRQ_INT_VECT | EQU 0801EH |
98 | LEAVE_EFFECT | EQU 001D5H |
99 | LOAD_ASCII | EQU 01F7FH |
100 | LOCAL_SPR_TBL | EQU 08002H |
101 | MODE_1 | EQU 01F85H |
102 | MSNTOLSN | EQU 001A6H |
103 | MUX_SPRITES | EQU 073C7H |
104 | NMI_INT_VECT | EQU 08021H |
105 | NUMBER_TABLE | EQU 0006CH |
106 | PLAY_IT | EQU 01FF1H |
107 | PLAY_ITP | EQU 01FB5H |
108 | PLAY_SONGS | EQU 01FG1H |
109 | POLLER | EQU 01FEBH |
110 | PUTFRAME | EQU 0800BH |
111 | PUTOBJ | EQU 01FFAH |
112 | PUTOBJP | EQU 01F67H |
113 | PUT_FRAME | EQU 08008H |
114 | PUT_VRAM | EQU 01FBEH |
115 | PUT_VRAMP | EQU 01F91H

```

:PUT\_CMPLX:05

:GAME\_OPT:05 LOGO:05 PUT\_MOBIL:05 PUT\_SPR:05

LOCATION OBJECT CODE LINE SOURCE LINE

```

(07E8) 116 PX_TO_PTRN_POS EQU 007E8H
(1FFD) 117 RAND_GEN EQU 01FFDH
(73C8) 118 RAND_NUM EQU 073C8H
(1FDC) 119 READ_REGISTER EQU 01FDCH
(1FE2) 120 READ_VRAM EQU 01FE2H
(1FAC) 121 READ_VRAM EQU 01FACH
(1F6D) 122 REFLECT_HORIZONTAL EQU 01F6DH
(1F6A) 123 REFLECT_VERTICAL EQU 01F6AH
(1FCD) 124 REQUEST_SIGNAL EQU 01FCDH
(1FA0) 125 REQUEST_SIGNALP EQU 01FA0H
(1F70) 126 ROTATE_90 EQU 01F70H
(800F) 127 RST_10H_RAM EQU 0800FH
(8012) 128 RST_18H_RAM EQU 08012H
(8015) 129 RST_20H_RAM EQU 08015H
(8018) 130 RST_28H_RAM EQU 08018H
(801B) 131 RST_30H_RAM EQU 0801BH
(800C) 132 RST_8H_RAM EQU 0800CH
(1FEE) 133 SOUND_INIT EQU 01FEEH
(1FB2) 134 SOUND_INITP EQU 01FB2H
(1FF4) 135 SOUND_MAN EQU 01FF4H
(8004) 136 SPRITE_ORDER EQU 08004H
(73B9) 137 STACK EQU 073B9H
(800A) 138 START_GAME EQU 0800AH
(1FD0) 139 TEST_SIGNAL EQU 01FD0H
(1FA3) 140 TEST_SIGNALP EQU 01FA3H
(1FD3) 141 TIME_MGR EQU 01FD3H
(1FD6) 142 TURN_OFF_SOUND EQU 01FD6H
(1F88) 143 UPDATE_SPINNER EQU 01F88H
(73C3) 144 VDP_MODE_WORD EQU 073C3H
(73C5) 145 VDP_STATUS_BYTE EQU 073C5H
(8006) 146 WORK_BUFFER EQU 08006H
(1FEB) 147 WRITER EQU 01FEBH
(1FD9) 148 WRITE_REGISTER EQU 01FD9H
(1FA6) 149 WRITE_REGISTERP EQU 01FA6H
(1FDF) 150 WRITE_VRAM EQU 01FDFH
(1FA9) 151 WRITE_VRAM EQU 01FA9H
(1FC4) 152 WR_SPR_NM_TBL EQU 01FC4H
(1F97) 153 WR_SPR_NM_TBL EQU 01F97H
154

```

:TABLE\_MA:05

:LOGO\_05

:GRAPHICS:05 VD\_DRIVER:05 TABLE\_MA:05 PUT\_MOBIL:05 ACT2:05

:GRAPHICS:05 PUT\_MOBIL:05 PUT\_SPR:05 PUTSEM12:05 ACT2:05

:GAME\_OPT:05 LOGO:05

:GAME\_OPT:05 LOGO:05 PUT\_MOBIL:05

:End of defined reference points

LOCATION OBJECT CODE LINE SOURCE LINE

```
156
157
158 GLB ACTIVATE
159 GLB ACTIVATEP
160 GLB ADDR16
161 GLB AMERICA
162 GLB ASCII_TABLE
163 GLB ATM_SWEEP
164 GLB CALC_OFFSET
165 GLB CARTRIDGE
166 GLB CONTROLLER_MAP
167 GLB DECLSN
168 GLB DECSN
169 GLB DECODER
170 GLB DEFER_WRITES
171 GLB EF_XOVER
172 GLB ENLARGE
173 GLB ENLGR
174 GLB FILL_VRAM
175 GLB FREE_SIGNAL
176 GLB FREE_SIGNALP
177 GLB FREQ_SWEEP
178 GLB GAME_NAME
179 GLB GAME_OPT
180 GLB GET_BCKGRND
181 GLB GET_VRAM
182 GLB GET_VRAMP
183 GLB INIT_5PR_ORDER
184 GLB INIT_SPR_ORDERP
185 GLB INIT_TABLE
186 GLB INIT_TABLEP
187 GLB INIT_TIMER
188 GLB INIT_TIMERP
189 GLB INIT_WRITER
190 GLB INIT_WRITERP
191 GLB IRO_INT_VECT
192 GLB LEAVE_EFFECT
193 GLB LOAD_ASCII
194 GLB LOCAL_SPR_TBL
195 GLB MSNTOLSN
196 GLB MODE_1
197 GLB MUX_SPRITES
198 GLB NMI_INT_VECT
199 GLB NUMBER_TABLE
200 GLB PLAY_IT
201 GLB PLAY_ITP
202 GLB PLAY_SONGS
203 GLB POLLER
204 GLB PUTFRAME
205 GLB PUTOBJ
206 GLB PUTOBJP
207 GLB PUT_FRAME
208 GLB PUT_VRAM
209 GLB PUT_VRAMP
210 GLB PX_TO_PTRN_P05
211 GLB RAND_GEN
212 GLB RAND_NUM
```

```
:The following defines each access point
: as Global.
```

LOCATION	OBJECT CODE	LINE	SOURCE LINE
		213	GLB READ_REGISTER
		214	GLB READ_VRAM
		215	GLB READ_VRAMP
		216	GLB REFLECT_HORIZONTAL
		217	GLB REFLECT_VERTICAL
		218	GLB REQUEST_SIGNAL
		219	GLB REQUEST_SIGNALP
		220	GLB ROTATE_90
		221	GLB RST_10H_RAM
		222	GLB RST_18H_RAM
		223	GLB RST_20H_RAM
		224	GLB RST_28H_RAM
		225	GLB RST_30H_RAM
		226	GLB RST_8H_RAM
		227	GLB SOUND_INIT
		228	GLB SOUND_INITP
		229	GLB SOUND_MAN
		230	GLB SPRITE_ORDER
		231	GLB STACK
		232	GLB START_GAME
		233	GLB TEST_SIGNAL
		234	GLB TEST_SIGNALP
		235	GLB TIME_MGR
		236	GLB TURN_OFF_SOUND
		237	GLB UPDATE_SPINNER
		238	GLB VDP_MODE_WORD
		239	GLB VDP_STATUS_BYTE
		240	GLB WORK_BUFFER
		241	GLB WRITER
		242	GLB WRITE_REGISTER
		243	GLB WRITE_REGISTERP
		244	GLB WRITE_VRAM
		245	GLB WRITE_VRAMP
		246	GLB WR_SPR_NN_TBL
		247	GLB WR_SPR_NN_TBLP

Errors= 0

