

[BULLETIN 10 MISSING]

BULLETIN NO. 11
DECEMBER 22, 1982

TO: DISTRIBUTION

FROM: ARD SOFTWARE ENGINEERING *Daily*

SUBJECT: RELEASE OF COLECOVISION PROGRAMMERS
MANUAL REV. 5

cc: Eric Bromley
Robert Schenck
Marshall Caras
Tom Helner

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	K. Lagace
R. Dionne	J. Michaels
A. Godfrey	M. Minto
L. Gray	A. Nguyen
C. Hager	L. Olbrych
R. Harris	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
R. Dionne
A. Godfrey
L. Gray
C. Hager
R. Harris
L. Henderson
R. Jepson
D. Jonker

K. Lagace
J. Michaels
M. Minto
A. Nguyen
L. Olbrych
R. Rizzo
D. Schulze
D. Stern
D. Thompson
B. Zawislak

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

CONFIDENTIAL



INDUSTRIES, INC.

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

Executive Office

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

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

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

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 be 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 . . . . .
U 0. . . . .
    . . . . .
    .B.C. . . . .
    .E.F. . . . .
    . . . . .
```

(diagram of upper left corner of pattern plane)

CONFIDENTIAL



INDUSTRIES, INC.

Executive Office

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

ColecoVision 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

```

```

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

```

List of access points to the ColecoVision Operating system ROM.
Only these 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	Added location: PX_TO_PIRN_PUS PUT_FRAME
2	13Apr1114p	Rob	Added documentation specific to Zaxxon Development. GET BKGRND CALC OFFSET
1	2Apr11153p	Ken Louque	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:
END

CONFIDENTIAL

CATION OBJECT CODE LINE SOURCE LINE

59	Symbol	Absolute Address	Partial Xref of routines used by other OS routines
60	!Name		
61			
62			
63			Start of defined reference points
64	ACTIVATE	EQV 01FF7H	
65	ACTIVATEP	EQV 01F64H	
66	ADDB16	EQV 001R1H	
67	AMERICA	EQV 00069H	
68	ASC11_TABLE	EQV 0006AH	
69	ATM_SHEEP	EQV 0012FH	
70	CALC_OFFSET	EQV 000C0H	
71	CARTIDGE	EQV 00000H	
72	CONTROLLER_MAP	EQV 00008H	!CONTROLLER:OS
73	DECLSN	EQV 00190H	
74	DECMN	EQV 0019BH	
75	DECODEP	EQV 01F79H	
76	DEFER_WRITES	EQV 073C6H	!PUT_OBJEC:OS
77	EFXOVER	EQV 002EEH	
78	ENLARGE	EQV 01F73H	
79	ENLRC	EQV 01D6CH	!GAME_OPT:OS
80	FILL_VRAM	EQV 01F82H	
81	FREE_SIGNAL	EQV 01FCAH	
82	FREE_SIGNALP	EQV 01F9DH	
83	FREQ_SWEEP	EQV 000FCH	
84	GAME_NAME	EQV 00024H	!LOGO:OS
85	GAME_OPT	EQV 01F7CH	
86	GET_BKGRND	EQV 0089BH	
87	GET_VRAM	EQV 01F8BH	!PUT_MOBILE:OS PUT_SPR:OS
88	GET_VRAMP	EQV 01F8EH	
89	INIT_SPR_ORDER	EQV 01FC1H	
90	INIT_SPR_ORDERP	EQV 01F94H	
91	INIT_TABLE	EQV 01F8DH	
92	INIT_TABLEP	EQV 01F8BH	!GAME_OPT:OS LOGO:OS
93	INIT_TIMER	EQV 01FC7H	
94	INIT_TIMERP	EQV 01F9AH	
95	INIT_WRITER	EQV 01FL5H	
96	INIT_WRITERP	EQV 01FAFH	
97	IRQ_INT_VECT	EQV 0001CH	
98	LEAVE_EFFECT	EQV 001D5H	
99	LOAD_ASCII	EQV 01F7FH	!GAME_OPT:OS
100	LOCAL_SPR_TBL	EQV 00002H	!TABLE_MA:OS
101	MODE_1	EQV 01FB5H	!GAME_OPT:OS
102	MSNTOL9H	EQV 001A6H	!TABLE_MA:OS
103	MUX_6PRITEB	EQV 073C7H	
104	NMI_INT_VECT	EQV 00021H	
105	NUMRER_TABLE	EQV 0006CH	
106	PLAY_IT	EQV 01FF1H	
107	PLAY_ITP	EQV 01F85H	
108	PLAY_SONGS	EQV 01F61H	
109	POLLER	EQV 01FL1H	
110	PUTFRAME	EQV 0000BH	
111	PUTOBJ	EQV 01CFAH	!PUT_CHPLX:OS
112	PUTORIP	EQV 01F67H	
113	PUT_FRAME	EQV 00000H	
114	PUT_VRAM	EQV 01F8EH	!GAME_OPT:OS !LOGO:OS PUT_MOBILE:OS PUT_SPR:OS
115	PUT_VRAMP	EQV 01F91H	

LOCATION ADDRESS CODE LINE SOURCE LINE

```

(07E8) 116 PX_TO_PTRN_POS EQU 007E8H
(1FFD) 117 RAND_GEN EQU 01F1D0H
(73C8) 118 RAND_NUM EQU 073C8H
(1FDC) 119 READ_REGISTER EQU 01FDC0H
(1FE2) 120 READ_VRAM EQU 01FE20H
(1FAC) 121 READ_VRAM EQU 01FAC0H
(1F6D) 122 REFLECT_HORIZONTAL EQU 01F6D0H
(1F6A) 123 REFLECT_VERTICAL EQU 01F6A0H
(1FCD) 124 REQUEST_SIGNAL EQU 01FCD0H
(1FA0) 125 REQUEST_SIGNALP EQU 01FA00H
(1F70) 126 ROTATE_90 EQU 01F700H
(800F) 127 RST_10H_RAM EQU 0800F0H
(8012) 128 RST_18H_RAM EQU 080120H
(8015) 129 RST_20H_RAM EQU 080150H
(8018) 130 RST_28H_RAM EQU 080180H
(801B) 131 RST_30H_RAM EQU 0801B0H
(800C) 132 RST_8H_RAM EQU 0800C0H
(1FEE) 133 SOUND_INIT EQU 01FEE0H
(1FB2) 134 SOUND_INITP EQU 01FB20H
(1FF4) 135 SOUND_NAN EQU 01FF40H
(8004) 136 SPRITE_ORDER EQU 080040H
(7349) 137 STACK EQU 073490H
(800A) 138 START_GAME EQU 0800A0H
(1FD0) 139 TEST_SIGNAL EQU 01FD00H
(1FA3) 140 TEST_SIGNALP EQU 01FA30H
(1FD3) 141 TIME_MGR EQU 01FD30H
(1FD6) 142 TURN_OFF_SOUND EQU 01FD60H
(1F88) 143 UPDATE_SPINNER EQU 01F880H
(73C3) 144 VDP_MODE_WORD EQU 073C30H
(73C5) 145 VDP_STATUS_BYTE EQU 073C50H
(8016) 146 WORK_BUFFER EQU 080160H
(1FEB) 147 WRITER EQU 01FEB0H
(1FD9) 148 WRITE_REGISTER EQU 01FD90H
(1FA6) 149 WRITE_REGISTERP EQU 01FA60H
(1FDF) 150 WRITE_VRAM EQU 01FDF0H
(1FA9) 151 WRITE_VRAM EQU 01FA90H
(1FCA) 152 WR_SPR_NH_TBL EQU 01FCA0H
(1F77) 153 WR_SPR_NH_TBLP EQU 01F770H
154

```

LOC0:05
PUT_MOBIL:05

TABLE_MA:05
LOC0:05

GRAPHICS:05 VD_DRIVER:05 TABLE_MA:05 PUT_MOBIL:05 ACT2:05
GRAPHICS:05 PUT_MOBIL:05 PUT_SPR:05 PUTSEM:05 ACT2:05
GAME_OPT:05 LOC0:05
GAME_OPT:05 LOC0:05 PUT_MOBIL:05

End of defined reference points

CONFIDENTIAL

LOCATION OBJECT CODE LINE SOURCE LINE

```

156
157 GLB ACTIVATE
158 GLB ACTIVATIEP
159 GLB ADDB16
160 GLB AMERICA
161 GLB ASCII_TABLE
162 GLB ATN_SWEEP
163 GLB CALC_OFFSET
164 GLB CARTRIDGE
165 GLB CONTROLLER_MAP
166 GLB DECLSN
167 GLB DECM5H
168 GLB DECM5H
169 GLB DECODE
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 MSMTOLSN
196 GLB MDWE_I
197 GLB MUX_SPRITES
198 GLB NINI_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_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	CLB READ_REGISTER	
214	CLB READ_VRAM	
215	CLB READ_VRAM	
216	CLB REFLECT_HORIZONTAL	
217	CLB REFLECT_VERTICAL	
218	CLB REQUEST_SIGNAL	
219	CLB REQUEST_SIGNALP	
220	CLB ROTATE_90	
221	CLB RST_10H_RAM	
222	CLB RST_18H_RAM	
223	CLB RST_20H_RAM	
224	CLB RST_28H_RAM	
225	CLB RST_30H_RAM	
226	CLB RST_0H_RAM	
227	CLB SOUND_INIT	
228	CLB SOUND_INITP	
229	CLB SOUND_RAM	
230	CLB SPRITE_ORDER	
231	CLB STACK	
232	CLB START_GAME	
233	CLB TEST_SIGNAL	
234	CLB TEST_SIGNALP	
235	CLB TIME_MGR	
236	CLB TURN_OFF_SOUND	
237	CLB UPDATE_SPINNER	
238	CLB VDP_MODE_WORD	
239	CLB VDP_STATUS_BYTE	
240	CLB WORK_BUFFER	
241	CLB WRITER	
242	CLB WRITE_REGISTER	
243	CLB WRITE_REGISTERP	
244	CLB WRITE_VRAM	
245	CLB WRITE_VRAMP	
246	CLB WR_SPR_NN_TEL	
247	CLB WR_SPR_NN_TELP	

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 #####
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_SYMBOLS
Colecovision-Operating System
Software Engineering
Advanced Research and Development
Coleco Industries

*** Confidential ***

List of access points to the Colecovision Operating system ROM.
Only these 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 05).

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	GLBed locations added in rev 3 Added locations PX_TO_PTRN_POS 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)

ENDESCRIPTION:
NEND

LOCATION OBJECT CODE LINE SOURCE LINE

```
59
60 | Symbol      Absolute      Partial Xref of routines used
61 | Name         Address         by other OS routines
62 |-----|-----|-----|
63 |
64 1(F67)        64 ACTIVATE      EQU 01FF7H
65 1(F64)        65 ACTIVATER     EQU 01F64H
66 1(B11)        66 ADD816         EQU 001B1H
67 1(069)        67 AMERICA       EQU 00069H
68 1(06A)        68 ASCII_TABLE   EQU 0006AH
69 1(12F)        69 ATN_SWEEP      EQU 0012FH
70 1(0C0)        70 CALC_OFFSET    EQU 008C0H
71 1(000)        71 CARTRIDGE      EQU 00000H
72 1(008)        72 CONTROLLER_MAP EQU 00008H
73 1(090)        73 DECLSN        EQU 00190H
74 1(09B)        74 DECMN         EQU 0019BH
75 1(F79)        75 DECODER       EQU 01F79H
76 1(3C6)        76 DEFER_WRITES  EQU 073C6H
77 1(0EE)        77 EF_XOVER      EQU 002EEH
78 1(F73)        78 ENLARGE       EQU 01F73H
79 1(06C)        79 ENLGR         EQU 01D6CH
80 1(F82)        80 FILL_VRAM     EQU 01F82H
81 1(FCA)        81 FREE_SIGNAL   EQU 01FCAH
82 1(F9D)        82 FREE_SIGNALP  EQU 01F9DH
83 1(0FC)        83 FREQ_SWEEP    EQU 000FCH
84 1(024)        84 GAME_NAME     EQU 00024H
85 1(F7C)        85 GAME_OPT      EQU 01F7CH
86 1(098)        86 GET_BKGRND   EQU 00098H
87 1(F8B)        87 GET_VRAM     EQU 01F8BH
88 1(F8E)        88 GET_VRAM     EQU 01F8EH
89 1(FC1)        89 INIT_SPR_ORDER EQU 01FC1H
90 1(F94)        90 INIT_SPR_ORDERP EQU 01F94H
91 1(F88)        91 INIT_TABLE    EQU 01F88H
92 1(F8B)        92 INIT_TABLEP   EQU 01F8BH
93 1(FC7)        93 INIT_TIMER    EQU 01FC7H
94 1(F9A)        94 INIT_TIMERP   EQU 01F9AH
95 1(FE5)        95 INIT_WRITER   EQU 01FE5H
96 1(FAF)        96 INIT_WRITERP  EQU 01FAFH
97 1(01E)        97 IRQ_INT_VECT  EQU 0001EH
98 1(0D5)        98 LEAVE_EFFECT  EQU 001D5H
99 1(F7F)        99 LOAD_ASCII   EQU 01F7FH
100 1(002)       100 LOCAL_SPR_TBL EQU 00002H
101 1(F85)       101 MODE_I       EQU 01F85H
102 1(0A6)       102 MSNTOLSN     EQU 001A6H
103 1(73C7)     103 MUX_SPRITES  EQU 073C7H
104 1(021)     104 NMI_INT_VECT EQU 00021H
105 1(06C)     105 NUMBER_TABLE EQU 0006CH
106 1(FE1)     106 PLAY_IT     EQU 01FE1H
107 1(FB5)     107 PLAY_ITP    EQU 01FB5H
108 1(F61)     108 PLAY_SONGS  EQU 01F61H
109 1(FEB)     109 POLLER      EQU 01FEBH
110 1(00B)     110 PUTFRAME    EQU 0000BH
111 1(FFA)     111 PUTOBJ      EQU 01FFAH
112 1(F67)     112 PUTOBJP     EQU 01F67H
113 1(00B)     113 PUT_FRAME   EQU 0000BH
114 1(FBE)     114 PUT_VRAM    EQU 01FBEH
115 1(F91)     115 PUT_VRAM    EQU 01F91H

;Start of defined reference points

;CONTROLE:OS
;PUT_OBJEC:OS
;GAME_OPT:OS
;GAME_OPT:OS
;LOGO:OS
;LOGO:OS
;PUT_MOBIL:OS PUT_SPR:OS
;GAME_OPT:OS LOGO:OS
;GAME_OPT:OS
;TABLE_MA:OS
;GAME_OPT:OS
;TABLE_MA:OS
;PUT_CMPLX:OS
;GAME_OPT:OS LOGO:OS PUT_MOBIL:OS PUT_SPR:OS
```

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
(1F60) 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 01FAGH
(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_TBLP EQU 01F97H
154
```

:LOGO:OS
:PUT_MOBIL:OS

:TABLE_MA:OS
:LOGO_OS

:GRAPHICS:OS VD_DRIVER:OS TABLE_MA:OS PUT_MOBIL:OS ACT2:OS
:GRAPHICS:OS PUT_MOBIL:OS PUT_SPR:OS PUTSEMI2:OS ACT2:OS
:GAME_OPT:OS LOGO:OS
:GAME_OPT:OS LOGO:OS PUT_MOBIL:OS

:End of defined reference points

LOCATION OBJECT CODE LINE SOURCE LINE

```
156
157
158 GLB ACTIVATE
159 GLB ACTIVATEP
160 GLB ADD816
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 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_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 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_POS
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_VRAM	
			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_VRAM	
			246	GLB WR_SPR_NN_TBL	
			247	GLB WR_SPR_NN_TBLP	

Errors= 0

