HI-RES MULTI-PAGE TEST DEMO

8KB PACKAGE
ADDRESSED $2000-3FFF_H$

FOR MODIFIED LOW/HIGH-RES ASTROCADE
WITH EIGHT 16KB PAGES OF SCREEN SRAM (STATIC RAM)
TOTAL 128KB SCREEN RAM
320 x 204 PIXEL RESOLUTION EACH PAGE

MCM DESIGN 2020

8KB PACKAGE IS SELF CONTAINING.
ALL NECESSARY HI-RES ROUTINES
ARE INCLUDED WITHIN PACKAGE.

LOW-RES OR CUSTOM ROM AT $0000-1FFF_H$
IS REQUIRED ONLY TO JUMP TO 2000_H.

NO USER RAM $8000-FFFF_H$ IS REQUIRED,
HI-RES SCREEN RAM IS ADDRESSED $4000-7FFF_H$.
MAGIC RAM $0000-3FFF_H$.
REFER TO NUTTING MANUAL FOR SYSTEM DESCRIPTION,

(This page intentionally left blank.)

IMPORTANT PROGRAMMING NOTES

TV PAGE DISPLAY (SCAN) OUTPUT PORT 74_H

D3 D2 D1 D0

- D3 → DISPLAY EXPANSION TO 16 PAGES
- D2 D1 D0 → DISPLAY ONE OF 8 PAGES (0-7)

Z80 READ OR WRITE MAGIC HARDWARE READ OR WRITE OUTPUT 75_H

D7 D6 D5 D4 | D3 | D2 D1 D0

- D7 D6 D5 D4 → WRITE TO ONE OF 8 PAGES (0-7)
- D7 → WRITE EXPANSION TO 16 PAGES
- D3 → READ EXPANSION TO 16 PAGES
- D2 D1 D0 → READ ONE OF 8 PAGES (0-7)

NOTES:

WHEN EXECUTING MAGIC "XOR", MAGIC "OR" WRITES, THE PROGRAM MUST POINT THE Z80 CPU NOT ONLY TO THE PAGE BEING WRITTEN, BUT ALSO POINT TO THE SAME PAGE TO BE READ FROM. THE MAGIC HARDWARE READS AND WRITES LOGICALLY TO THE SPECIFIED PAGE FOR THESE 2 LOGICAL MAGIC XOR, OR WRITES.

EXAMPLE: WRITE A GRAPHIC PATTERN TO PAGE 2 USING A MAGIC "XOR" FUNCTION

1. SET UP MAGIC OUTPUT PORT $0C_H$ FOR A LOGICAL XOR MAGIC FUNCTION.
2. POINT THE Z80 CPU TO THE SPECIFIED PAGE FOR THE MAGIC XOR WRITE BEFORE THE ACTUAL Z80 WRITE INSTRUCTION, SUCH AS LDIR, IS EXECUTED. IF THE GRAPHIC PATTERN WRITE SUBROUTINE UTILIZES PUSH/POP INSTRUCTIONS, YOU MUST ALSO POINT THE Z80 TO THE PAGE THAT IS UTILIZING A STACK AREA. IF THE STACK AREA IS IN PAGE 0 AND YOU ARE WRITING A GRAPHICAL PATTERN TO PAGE 2, YOU MUST POINT THE Z80 TO PAGE 0 FOR THE PUSH/POP INSTRUCTIONS AND POINT THE Z80 TO PAGE 2 FOR THE PATTERN WRITE INSTRUCTION.

```
3E 22    LD A, 0010 0010
D3 75    OUT (75H), A
```

POINT THE Z80 CPU TO PAGE 2 SO THE MAGIC HARDWARE CAN LOGICALLY READ/WRITE A MAGIC XOR FUNCTION.

3. WRITE THE GRAPHIC PATTERN.
4. RE-POINT THE Z80 CPU TO APPROPRIATE PAGE, IF NECESSARY.

IMPORTANT NOTE!

REMEMBER, YOU HAVE TO REALLY PAGE ATTENTION TO WHERE YOU ARE POINTING THE Z80 CPU (OUTPUT PORT 75_H) WHEN WORKING WITHIN MULTIPLE PAGES. THIS INCLUDES WORKING A Z80 STACK AREA, VECTOR BLOCKS, VARIABLES AND FLAGS WITHIN ANY SPECIFIED PAGE.

MCM DESIGN 8KB PROGRAM PACKAGE ($2000-3FFF_H$) PAGE (ii)
HI-RES MULTI-PAGER TEST DEMO
EXECUTES ON MCM DESIGN'S MODIFIED HI-RES ASTROCADE
WITH 128KB, 8 PAGE STATIC SCREEN RAM (SRAM)

ENTRANCES:
MULTI-PAGE TEST DEMO PAGOA, P.41
CRITTER MOVE AND ERASE 8 PAGES COPYP, P.100
HI-RES FISH DEMO P.114
MOVE CRITTER, USING HAND CONTROL 1, WITHIN 3 CONNECTING SCENES P.106

ROUTINES / SUBROUTINES OF INTEREST:
CHECK FOR LEFT/RIGHT KEYPAD COLUMN PRESS DURING SYSTEM RESET P.122

NESTED "WRITE ONLY" ROUTINES (NO Z80 STACK AREA UTILIZED)
EXITS WITH JP(IX) OR JP(IY) INSTEAD OF RET

A. WRITE RELATIVE, WRITR P.5
B. WRITE WITH PATTERN SIZE, WRITP P.6
C. WRITE PATTERN WITH COORDINATES CONVERSION, WRIT P.6
D. WRITE PATTERN, WRITM P.7
E. NORMAL WRITE, MWRT P.8-9
F. WRITE EXPANDED, WEXPD P.10-11
G. WRITE WITH FLOP, WFLOP P.12-13
H. WRITE WITH EXPANDED FLOP, WXFLOP P.14-15

ADDITIONAL "WRITE ONLY" ROUTINES

CONVERT X,Y COORDINATES TO MAGIC ADDRESS P.16-18
FILL AREA, FILL P.19
RESTORE AREA, RESTOR P.20-21
MAGIC ENTRY POINT, MENTR P.21

SAMPLE PARAGRAPHS P.35-40

WRITE A CUSTOM PARAGRAPH OF TEXT, WPGPH P.22-24

WRITE ONLY (NO Z80 STACK UTILIZED) FISH TANK +15 MAGIC PATTERN WRITES, PGM P.26-32

7x9 PIXEL CHARACTER TABE, P.33-35

CLEAR SCREEN, CSCRN P.41

WRITE ENLARGED TEXT - RELATED SUBROUTINES

CUSTOM WRITE ONE LINE OF TEXT WITH EXPANDED ENLARGEMENT
WRTLIN, P.51
WRITE CHARACTER, WCHAR P.57
WRITE CHARACTER STRING, WSTR P.58
SLICE X4 TEXT STRING, SLICE P.60

(NORMAL HI-RES) CONVERT COORDINATES TO MAGIC ADDRESS WITH PROVISION FOR CUSTOM MAGIC FLOP REQUEST (BITS 7,6=1 FOR MAGIC REGISTER VALUE)
RELTA1, P.55-56

NORMAL WRITE ROUTINES (UTILIZES Z80 STACK AREA)
A. WRITE RELATIVE FROM VECTOR BLOCK, VWRITR P.64
B. WRITE RELATIVE, WRITR P.64
C. WRITE WITH PATTERN SIZE, WRITP P.65
D. WRITE WITH COORDINATE CONVERSION. P.65
E. WRITE PATTERN, WPATHR P.65-67

PROCESS CUSTOM MR FLOP REQUEST
FLOP PATTERN IN SAME PATTERN AS A NORMAL (UNFLOPPED) WRITE, FREQ P.69
(USED WITH RELTA1, P.55)

"FLIP PAGE" AUDIO SOUND, FSND P.81

RANGED (RANDOMIZER), RANGE P.82-83
CUSTOM UPDATE X AND Y COORDINATES IN VECTOR BLOCK WITH AUTO REVERSE DELTA (NO MANUAL SETTING) P.84-89

FILL SCREEN AREA WITH HORIZONTAL ALTERNATING COLOR STRIPES
LFILLA, LFILLB OR LFILLC PAGE 103

HI-RES HAND CONTROL MASK TO DELTA
(CONVERTS HAND CONTROL PORT INPUT TO VECTOR BLOCK DELTA X (ΔX) OR DELTA Y (ΔY).
MKTD, PAGES 104-105

COPY (MOVE) DATA BLOCK TO RAM IVBLK, PAGE 105

FILL 1 TO 4 PIXEL COLUMNS FROM BOTTOM TO TOP F4COL, PAGE 126, 127

CUSTOM MAGIC WRITE (WITH NO CLEAR SHIFTER BYTE AT THE END OF EACH LINE WRITTEN) CWRT, PAGE 129

USE SCREEN INTERRUPTS TO PROVIDE ADDITIONAL COLORS AND TO UPDATE ELAPSED TIME CLOCK
INITIALIZE SCREEN INTERRUPTS
INTERRUPT VECTORS
INTERRUPT SERVICE ROUTINE
ELAPSED TIME HANDLER ETIMER, PAGE 130

DISPLAY ELAPSED TIME HR:MIN:SEC DTIME, P. 131-132

CUSTOM WRITE WITH EXPAND (WITH NO CLEAR SHIFTER BYTE AT THE END OF EACH LINE WRITTEN) CWXP, PAGE 133

CUSTOM FLOP (FLOP PATTERN IN SAME PATTERN USED FOR A NORMAL WRITE)
USE WITH CONVERT COORDINATES X, Y TO MAGIC ADDRESS RELTA1, PAGE 55
CUFLOP, PAGE 140

FIRST RAM BYTE ADDRESS IN EACH HORIZONTAL SCREEN LINE
(BOTTOM 22 LINES) PAGE 113

OVERVIEW

HI-RES MULTI-PAGER TEST DEMO
DEMO WRITES STATIC GRAPHICS IN 8 PAGES
PROGRAM FLIPS THROUGH EACH PAGE USING 5 CYCLES
FLIP TIME DECREASES WITH EACH CYCLE.
5TH CYCLE FLIPS ABOUT 2 PAGES EVERY SECOND
AN AUDIO OUTPUT (FLIP SOUND) OCCURS WITH EACH FLIP.

ENTRANCE TO MULTI-PAGER DEMO IS AT 2996_H PAGE 0, SEE P. 41

INITIALIZE VARIOUS PARAMETERS FOR PAGE 0 P.41

0 WRITE 3 PARAGRAPHS OF INTRO TEXT IN PAGE 0 P.42, 22-24, 33-40

WHILE PAGE 0 IS BEING DISPLAYED,

1 FILL NARROW VERTICAL STRIPES IN PAGE 1* P.44
2 WRITE AQUARIUM (END VIEW) +15 MAGIC WRITES* IN PAGE 2 P.44, 26-32
3 FILL NARROW HORIZONTAL STRIPS IN PAGE 3* P.44
4 WRITE 10 COLOR TEXTURED TEST PATTERN* IN PAGE 4 P.45-48
5 WRITE NARROW VERTICAL + HORIZONTAL STRIPES* IN PAGE 5 P.48-50
6 WRITE MULTIPAGER TITLE PAGE IN PAGE 6 P.50-54, 56-62, 25
7 WRITE GUNFIGHT SCREEN SHOT IN PAGE 7 P.62A-80

VIEW AND FLIP THE 8 PAGES P.90-93, 81

COPY "MOVE CRITTER" PROGRAM TO PAGE 7 RAM P.93, P.100, P.99-100, P.89,

(COPY PROGRAM; PROGRAM TO COPY; SCREEN INTERRUPT SETUP)

P.94, 95, 96-98, BOTTOM OF PAGE P.84-88, P.101

(INTERRUPT ROUTINE; RANDOMIZE CRITTER XORY COORDINATE, SET UP PAGE TO VIEW; CUSTOM WRITE CRITTER; VECTOR CRITTER; MOVE CRITTER VARIABLES LISTING)

AT END OF MOVE CRITTER IN PAGE 7, JUMP TO HI-RES FISH DEMO

* THIS IS A WRITE ONLY ROUTINE. NO Z80 STACK AREA IS UTILIZED.

MOVE CRITTER WITHIN 3 INTERCONNECTION GRAPHIC SCENES PAGE (vi)
(SCENES HAVE SIMPLIFIED STATIC GRAPHICS)

- PROGRAM BEGINS @ $36DD_H$, PAGE 106
  PROGRAM PAGES 106-111
  - INITIALIZE VECTOR BLOCK TO MOVE AT RAM $7FC0_H$ PAGE 102
  - SUBROUTINE TO MOVE INITIAL VECTOR BLOCK TO RAM, IVBLK PAGE 105
  - CRITTER LIMITS TABLES FOR ALL 3 SCENES PAGES 102-103
  - 3 INTERCONNECTING SCENES DIAGRAM P. 103
  - FILL SCREEN WITH HORIZONTAL LINES COMMON TO ALL 3 SCENES PAGE 103
  - HAND CONTROL MASK TO DELTA SUBROUTINE (CUSTOM HI-RES), MKTD PAGE 104-105

OVERVIEW

HI-RES FISH DEMO

DEMO VARIABLES PAGE 112, 112A

1ST RAM BYTE ADR IN EACH SCREEN LINE (BOTTOM 24_D LINES) PAGE 113

MAIN PROGRAM PAGES 114–117

A. INITIALIZE VARIOUS PARAMETERS PAGE 114

B. INITIALIZE DUAL SCREEN INTERRUPTS TO ADD 3 NEW COLORS TO THE SEA BOTTOM PLUS UPDATE ELAPSED TIME PAGE 114, 148 AND 130

C. INTERRUPT VECTORS PAGE 147

D. DETAIL THE SEABOTTOM PAGES 114, 117–121, 122, 126–128

E. DISPLAY ELAPSED TIME PAGES 114, 116, 131

F. INITIALIZE WITH RANDOMIZATION ALL THE FISH PAGE 115

G. UPDATE FISH VECTOR BLOCK, BLANK AND REWRITE ALL FISH PAGES 115–116

H. CHECK FOR ELAPSED TIME UPDATE PAGE 116

I. CHECK FOR DEMO RUN NONSTOP FLAG SET P. 116

J. WRITE "UP" ARROW NEXT TO TIME IF NONSTOP MODE IS ENABLED P. 116

K. CHECK FOR AUTO RESTART (JMP TO 2000_H) WHEN TIME = 2:00 MINUTES P. 117

FACTORY PROGRAM REVISION (ADJUSTMENT) FOR:

(1) 13" RCA CRT TV

TO REMOVE "WHITE SCAN LINES" AT VERY BOTTOM OF TV SCREEN

P.148 $3FED_H$ 3E D4 ← CHANGE TO DA

INCREASES SCREEN INTERRUPT SCAN LINES TO 218_D.

(2) 19" INSIGNIA LeD TV

MOVE HORIZONTAL BORDER LOCATION 1 BYTE TO THE RIGHT TO ELIMINATE UNDESIRABLE "VERTICAL STRIPE" AT RIGHT SIDE OF SCREEN RAM AREA.

P.41 $29AC_H$ 3E 2A ← 2B

P.90 $340D_H$ 3E 2A ← 2B

P.92 $34A7_H$ 3E AA ← AB

P.94 $351A_H$ 2A ← 2B

P.94 $352C_H$ AA ← AB

P.148 $3FDA_H$ 3E 2A ← 2B

P.148 $3FF5_H$ 3E EA ← EB

MCM DESIGN 8kB HI-RES MULTI-PAGE TEST DEMO PACKAGE FOR USE WITH MODIFIED HI-RES ASTROCADE WITH MULTI-PAGER AND 128kB SCREEN STATIC RAM (SRAM).

INCLUDES A HI-RES FISH DEMO AND MOVE CRITTER WITHIN 3 SCENES.

2000_H C3 5F3A JP A3A5F — JUMP TO CHECK LEFT/RIGHT COLUMN PRESS DURING SYSTEM RESET, PAGE 122

LEFT JMP TO FISH DEMO
RIGHT JMP TO MOVE CRITTER WITHIN 3 INTERCONNECTING SCENES

COLOR TABLE

CLRT1 2003_H

07	WHITE	PIXEL 11	LEFT COLORS
A3	GREEN	10	
FC	LIGHT BLUE	01	
00	BLACK	00	
7D	MAIZE	PIXEL 11	RIGHT COLORS
A3	GREEN	10	
FC	LIGHT BLUE	01	
00	BLACK	00	

$200A_H$

CRITTER PATTERN (16 PIXELS WIDE X 18 PIXEL LINES HIGH)

$200B_H$ 00 RELATIVE X
00 ↓ Y

$200D_H$ 04 X SIZE
12 Y SIZE

$200F_H$ 00 00 00 00

2013_H 00 14 14 00
01 55 55 40
05 41 41 50
05 41 41 50

2023_H 05 55 55 50
01 51 45 40
00 54 15 00
04 05 50 10

2033_H 16 AA AA 94
04 AA AA 10
00 BB EE 00
02 AA AA 80

2043_H 05 55 55 50
05 40 01 50
05 00 00 50
3F 00 00 FC

2053_H 00 00 00 00

WAGON PATTERN

2057_H	00	RELATIVE X
	00	↓ Y
	04	X SIZE
	16	Y SIZE
$205B_H$	00 05 50 00	
$205F_H$	00 55 55 00	
2063_H	01 55 55 40	
	05 55 55 50	
	15 54 15 54	
	15 50 05 54	
2073_H	15 40 01 54	
	15 40 01 54	
	15 50 05 54	
	05 54 15 50	
2083_H	01 55 55 40	
	00 55 55 00	
	00 15 54 00	
	02 AA AA 80	
2093_H	00 AA AA 00	
	12 AA AA 84	
	10 A8 2A 04	
	10 20 08 04	
$20A3_H$	52 AA AA 85	
	10 20 08 04	
	10 00 00 04	
$20AF_H$	10 00 00 04	

TREE PATTERN

$20B3_H$	01	X SIZE
	11	Y SIZE
$20B5_H$	08	
	1C	
	3E	
	6B	
	08	
	08	
	3C	
	7E	
	A9	
$20BE_H$	08	

NOTE:
GRAPHIC PATTERNS
PAGES 2-4
SIMILAR OR SAME AS
LOW-RES PATTERNS
TAKEN FROM
NUTTING MANUAL
ROM CODE LISTING

HI-RES 3
STATIC RAM
WRITE ONLY
GRAPHICS

$20BF_H$ 3C
$20C0_H$ 7E
EB
89
08
1C
AE

CACTUS PATTERN

$20CG_H$ 01 X SIZE
0C Y ↓
20
30
38
30
B2
F2
F6
3C
$20D0_H$ 3C
30
30
30

COWBOY'S ARM PATTERN

$20D4_H$ 0A RELATIVE X
07 ↓ Y
02 X SIZE
04 Y SIZE
10 00
05 40
54 00
$20DE_H$ 50 00

COWBOY PATTERN (INCLUDES LEGS/FEET)

HI-RES STATIC RAM WRITE ONLY GRAPHICS

4

$20EO_H$	03	X SIZE		
	14	Y	↓	
$20E2_H$	00	44	00	
	11	55	10	
	15	55	50	
	02	AA	00	
	02	AZ	00	
$20F1_H$	02	AA	80	
	00	AA	00	
	04	A8	00	
	15	55	00	
	55	55	50	
2100_H	51	55	50	
	41	55	00	
	41	55	00	
	45	55	00	
	01	55	00	
	01	55	00	
2112_H	05	45	40	
	15	01	40	
	50	01	40	
$211B_H$	15	00	54	

WRITE RELATIVE

"WRITE ONLY" VARIATION OF ON-BOARD SUB#32

NO RAM STACK USED

ENTER WITH: DE = X COORDINATE OF MAIN PATTERN

B = Y COORDINATE

A = MAGIC REGISTER VALUE

HL = PATTERN ADDRESS - 4 (POINTING AT RELATIVE X)

IX = PROGRAM "CONTINUE ADDRESS" (SUBSTITUTE FOR RET)

WRITR $211E_H$	08	EX AF, AF'	SAVE MAGIC REG VALUE
	7E	LD A, (HL)	A = RELATIVE X
2120_H	23	INC HL	POINT HL AT RELATIVE Y
	83	ADD A, E	ADD RELATIVE X TO MAIN PATTERN X COORD
	5F	LD E, A	
	7A	LD A, D	
	CE 00	ADC A, 0	
	57	LD D, A	DE = UPDATED X COORDINATE
	7E	LD A, (HL)	A = RELATIVE Y
	23	INC HL	POINT HL AT PATTERN X SIZE
WRITR1 2129_H	80	ADD A, B	ADD RELATIVE Y TO MAIN PATTERN Y COORD
	ED 47	LD I, A	PUT UPDATED Y IN REG Y
$212C_H$	08	EX AF, AF'	A = MAGIC REG VALUE AGAIN

INTERRUPT REG I USED

NOTE: WRITE ROUTINES PAGES 5–18 ARE SIMILAR TO LOW-RES ROUTINES. THEY ARE REVISED FOR A Z80 WRITE ONLY MODE. NO Z80 RAM READS OR STACK AREA ARE UTILIZED. NO PUSH/POP OR RETURN INSTRUCTIONS ARE USED.

WRITE WITH PATTERN SIZE

"WRITE ONLY" VARIATION OF ON-BOARD SUB#34.
NO RAM STACK USED.

ENTER WITH: I = REGY = Y COORDINATE
DE = X COORDINATE
A = MAGIC REGISTER VALUE
HL = PATTERN ADDRESS-2 (POINTING AT X SIZE)
IX = PROGRAM "CONTINUE ADDRESS" (SUBSTITUTE FOR RET)

WRITP $212D_H$	4E	LD C,(HL)	C = X SIZE
	23	INC HL	POINT HL AT Y SIZE
	46	LD B,(HL)	B = Y SIZE
2130_H	23	INC HL	POINT AT PATTERN ADDRESS

WRITE PATTERN WITH COORDINATES CONVERSION

"WRITE ONLY" VARIATION OF ON-BOARD SUB#36
NO RAM STACK USED.

ENTER WITH: I = REGY = Y COORDINATE
DE = X COORDINATE
C = X SIZE
B = Y SIZE
A = MAGIC REGISTER VALUE
HL = PATTERN ADDRESS
IX = PROGRAM "CONTINUE ADDRESS" (SUBSTITUTE FOR RET)

MAGIC ENTRANCE

WRIT 213_H C3 C6 22 JP MENTR

P.21

COLOR TABLE (PAGE 0 TEXT INTRO)

CLRT2 2134_H	63	RED	PIXEL 11	CLR REG 7	
	FC	LT BLUE	10		6
	A5	GREEN	01		5
2137_H	00	BLACK	00		4

WRITE PATTERN (WITH MAGIC ADDRESS ONLY) 7

"WRITE ONLY" VARIATION OF ON-BOARD SUB#38

NO RAM STACK USED

ENTER WITH: DE = MAGIC ADDRESS
C = X SIZE
B = Y SIZE
A = MAGIC REGISTER VALUE
HL = PATTERN ADDRESS
IX = PROGRAM "CONTINUE ADDRESS" (SUBSTINTE FOR RET)

```
WRITM  2138H  CB 77      BIT 6,A        CHECK FOR A FLOP WRITE
              20 6D      JR NZ, WFLOP   JMP IF FLOP   <- DOG
              CB 5F      BIT 3,A        CHECK FOR EXPAND WRITE
       213EH  20 35      JR NZ, WEXP    JMP IF EXPAND
```

NORMAL WRITE

MWRT 2140_H	79	LD A, C	A = XSIZE
	08	EX AF, AF'	A' = XSIZE
	78 ← DOG	LD A, B	A = YSIZE
	D9	EXX	SAVE
	47	LD B, A	B' = YSIZE
	08	EX AF, AF' A = XSIZE	C' = XSIZE
	4F	LD C, A	
	D9	EXX	
	7B ← BOY	LD A, E	SAVE
	08	EX AF, AF' ← NOW E	DE' = MAGIC ADDRESS
	7A	LD A, D	
	D9	EXX	
	57	LD D, A	
	08	EX AF, AF'	
	5F	LD E, A	
	D9	EXX	
2150_H	AF	XOR A	ZERO B
	47	LD B, A	
	ED B0 ← BOY	LDIR (DE) ← (HL)	WRITE PATTERN LINE
		DE ← DE+1	
		HL ← HL+1	
		BC ← BC-1	
	12	LD (DE), A	CLEAR SHIFTER END BYTE
	D9	EXX	DE = MAGIC ADDRESS
	7B ← BOY	LD A, E	
	08	EX AF, AF' ← NOW E	
	7A	LD A, D	
	D9	EXX	
	57	LD D, A	
	08	EX AF, AF'	
	5F ← BOY	LD E, A	
	EB	EX DE, HL	DE = PATTERN ADR
	0E 50	LD C, 80_D	HL = MAGIC ADR
2160_H	09	ADD HL, BC	C = BYTES/LINE B = 0
2161_H	EB	EX DE, HL	POINT HL AT NEXT LINE (MAGIC ADR)
			DE = MAGIC ADR
			HL = PATTERN ADR

NORMAL WRITE (CONT'D

2162H	7B ← BOY	LD A, E	SAVE DE' = MAGIC ADR
	08	EX AF, AF' ← NOW E	
	7A	LD A, D	
	D9 ← DOG	EXX	
	57	LD D, A	
	08	EX AF, AF'	
	5F	LD E, A	
	D9	EXX	
	7D ← DOG	LD A, L ← NOW L	B = YSIZE C = XSIZE DE = MAGIC ADR HL = PATTERN ADR
	08	EX AF, AF'	
	7C	LD A, H	
	D9	EXX	
	67	LD H, A	
	08	EX AF, AF'	
217DH	6F	LD L, A	
	10 CD ← DOG	DJNZ MWRT	
2173H	DDE9 ← DOGS	JP (IX)	

-51

3 2 1 4 8 4 2 1
0 0 1 1 0 0 1 1
1 1 0 1 1 1 0 0
+1
1 1 0 0 1 1 0 1
C D

WRITE EXPANDED

WEXPD	2175_H	EB ← BOY	EX DE, HL	HL = SCREEN ADR DE = PATTERN ADR
WEXPD1		79	LD A, C ← NOW C	B' = Y SIZE C = X SIZE
		08	EX AF, AF'	
		78	LD A, B	
		D9	EXX	
		47	LD B, A	
		08	EX AF, AF'	
		4F	LD C, A	
		D9	EXX	
		7D ← DOG	LD A, L ← NOW L	HL' = SCREEN ADR
		08	EX AF, AF'	
	2180_H	7C	LD A, H	
		D9	EXX	
		67	LD H, A	
		08	EX AF, AF'	
		6F	LD L, A	
		D9	EXX	
		41	LD B, C	B = X SIZE
WEXPD2		1A	LD A, (DE)	WRITE EXPANDED BYTE
		13	INC DE	
		77	LD (HL), A	
		23	INC HL	
		77	LD (HL), A	
		23	INC HL	
-8 CW00 1000 1111 0111 +1		10 F8	DJNZ, WEXPD2	LOOP BACK TO FINISH LINE
		70	LD (HL), B	CLEAR 2 END SHIFTER BYTES
	2190_H	23	INC HL	
		70	LD (HL), B	
		D9	EXX	HL = SCREEN ADR
		7D	LD A, L ← NOW L	
		08	EX AF, AF'	
		7C	LD A, H	
		D9	EXX	
		67	LD H, A	
		08	EX AF, AF'	
		6F	LD L, A	
		0E 50	LD C, 80_D	HI-RES = 80_D BYTES/LINE B STILL = 0
	$219C_H$	09	ADD HL, BC	

$219D_H$	D9	EXX		
	79	LD A, C		
	08	EX AF, AF'	NOW C	
$21A0_H$	78	LD A, B		B = Y SIZE
	D9	EXX		C = X SIZE
	47	LD B, A		
	08	EX AF, AF'		
	4F	LD C, A		
	10 CF	DJNZ WEXPD1	-49	
$21A7_H$	DD E9	JP (IX)		

```
  3216 8421
  0011 0001
  1100 1110
          +1
  1100 1111
    C    F
```

WRITE WITH FLOP

WFLOP 21A9$_H$	CB 5F		BIT 3, A	CHECK FOR EXPANDED FLOP
	20 37		JR NZ, WXFLOP	JMP IF SO
	AF		XOR A	← 55$_D$
	ED 47	INTERRUPT →	LD I, A	
WFLOP1 21B0$_H$	79	REG I	LD A, C	
	08	USED	EX AF, AF'	← NOW C
	78		LD A, B	B' = Y SIZE
	D9		EXX	C' = X SIZE
	47		LD B, A	
	08		EX AF, AF'	
	4F		LD C, A	
	D9		EXX	
	7B		LD A, E	
	08		EX AF, AF'	← NOW E
	7A		LD A, D	
	D9		EXX	DE' = PATTERN ADR
	57		LD D, A	
	08		EX AF, AF'	
	5F		LD E, A	
	D9		EXX	
21C0$_H$	ED 57			
	47	INTERRUPT →	LD A, I	A = 0
	← 00	REG I	LD B, A	
WFLOP2 21C3$_H$	ED A0	USED	LDI	(DE) ← (HL)
				DE ← DE + 1
				HL ← HL + 1
				BC ← BC - 1
				↑ 0
	1B		DEC DE	
	1B		DEC DE	
	EA C3 21		JP PE, WFLOP 2	
	12		LD (DE), A	
	D9		EXX	
	7B		LD A, E	
	08		EX AF, AF'	← NOW E
	7A		LD A, D	DE = PATTERN ADR
	D9		EXX	
21D0$_H$	57		LD D, A	
	08		EX AF, AF'	
	5F		LD E, A	
	EB		EX DE, HL	HL = PATTERN ADR
	0E 50		LD C, 80$_D$	
21D6$_H$	09		ADD HL, BC	

← 06

$21D7_H$	EB	EX DE, HL		
	D9	EXX		
	79	LD A, C		
	08	EX AF, AF'	NOWC	
	78	LD A, B		B'=Y SIZE
	D9	EXX		C'=X SIZE
	47	LD B, A		
	08	EX AF, AF'		
	4F	LD C, A		
	10 CE	DJNZ WFLOP1	-50	
$21E2_H$	DD E9	JP(IX)		

32K 8421
00 11 00 10
11 00 11 01
+1
1100 1110
C E

WRITE WITH EXPANDED FLOP

WXFLOP 21E4	EB	EX DE,HL	HL=SCREEN ADR, DE=PATTERN ADR
WXFLOP1	79	LD A,C	
	08	EX AF,AF'	NOW C
	78	LD A,B	B'=Y SIZE
	D9	EXX	C'=X SIZE
	47	LD B,A	
	08	EX AF,AF'	
	4F	LD C,A	
	D9	EXX	
	7D	LD A,L	DOG
	08	EX AF,AF'	NOW L
	7C	LD A,H	
21F0H	D9	EXX	
	67	LD H,A	HL'=SCREEN ADR
	08	EX AF,AF'	
	6F	LD L,A	
	D9	EXX	
	41	LD B,C	
	1A	LD A,(DE)	
XFLOP2	13	INC DE	
	77	LD (HL),A	
	2B	DEC HL	
	77	LD (HL),A	
	2B	DEC HL	
	10 F8	DJNZ WXFLOP2	-8 0000 1000 → 1111 0111 +1 → 1111 1000 = F 8
	70	LD (HL),B	
	2B	DEC HL	
2200H	70	LD (HL),B	
	D9	EXX	
	7D	LD A,L	DOG
	08	EX AF,AF'	NOW L
	7C	LD A,H	HL=SCREEN ADR
	D9	EXX	
	67	LD H,A	
	08	EX AF,AF'	
	6F	LD L,A	
	0E 50	LD C,80_D	HI-RES 80_D BYTES/LINE
220BH	09	ADD HL,BC	

$220C_H$	D9	EXX	
	79	LD A, C	
	08	EX AF, AF' (NOW C)	
	78	LD A, B	B'=Y SIZE
2210_H	D9	EXX	C'=X SIZE
	47	LD B, A	
	08	EX AF, AF'	
	4F	LD C, A	
	10 CF	DJNZ WXFLOP1	-49
2216_H	DD E9	JP (IX)	

3216 8421
0011 0001
1100 1110
1100 1111
C F

CONVERT X,Y COORDINATES TO MAGIC ADDRESS

NO USE OF RAM STACK IS PERMITTED (WRITE ONLY GRAPHICS)

ENTER WITH: I=REGY = Y COORDINATE

DE = X COORDINATE

C = X SIZE] USED FOR PATTERN WRITE

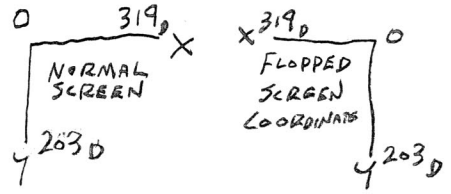
B = Y ↓] DO NOT CLOBBER THESE VALUES

A = MAGIC REGISTER VALUE

HL = PATTERN ADDRESS

IX = PROGRAM "CONTINUE ADDRESS" (SUBSTITUTE FOR RET)

IY = RETURN SUBSTITUTE FOR THIS ROUTINE



MADDR	2218_H	08	EX AF, AF'	A' = MAGIC REGISTER VALUE
		79	LD A, C	A = X SIZE
		08	EX AF, AF'	A' = X SIZE, A = MAGIC REG VALUE
		4F	LD C, A	SAVE MR VALUE IN C
		78	LD A, B	A = Y SIZE
		D9	EXX	
		47	LD B, A	B' = Y SIZE
		08	EX AF, AF'	C' = X SIZE
	2220_H	4F	LD C, A	
		D9	EXX	
		7D ← DOG	LD A, L	NOW L
		08	EX AF, AF'	
		7C	LD A, H	
		D9	EXX	HL' = PATTERN ADDRESS
		67	LD H, A	
		08	EX AF, AF'	
		6F	LD L, A	
		D9	EXX	
		79	LD A, C	A = MAGIC REG VALUE
		E6 78	AND 78_H	
		6F	LD L, A	
		7B	LD A, E	
		E6 03	AND 03	
	2234_H	B5	OR L	

```
2232H  4F        LD C,A        SAVE UPDATED MR VALUE IN C
       6B        LD L,E        ] HL = X COORDINATE
       62        LD H,D        ]
       E6 40     AND 40H
       28 0B     JR Z, MADDR1  <- BOY
       7B        LD A,E        ]
       2F        CPL           ]
       5F        LD E,A        ] 2'S COMPLEMENT X
       7A        LD A,D        ] (NEGATE X)
       2F        CPL           ]
       57        LD D,A        ]
       13        INC DE        ]
2240   21 3F 01  LD HL, 319D   ] XFLOD = 319 - X
       19        ADD HL,DE     ]
MADDR1 7D        LD A,L        <- NOW L
       08        EX AF,AF'     ]
       7C        LD A,H        ]
       D9        EXX           ] DE' = SAVE THIS X
       57        LD D,A        ]
       08        EX AF.AF'     ]
       5F        LD E,A        ]
       D9        EXX           ]
       ED57      LD A,I        <- INTERRUPT REG I
       6F        LD L,A          USED
       26 00     LD H,0
2251H  29        ADD HL,HL
       29        |
       29        |
       29        |
       54        LD D,H
       5D        LD E,L
       29        ADD HL,HL
       29        ADD HL,HL
       19        ADD HL,DE
       D9        EXX           ]
       7B        LD A,E        <- NOW E  ]
       08        EX AF,AF'     ]         ] DE = X SAVED ABOVE
       7A        LD A,D        ]         ]
       D9        EXX           ]         ]
       57        LD D,A        ]         ]
2260H  08        EX AF,AF'     ]         ]
2261H  5F        LD E,A        ]         ]
```

```
2262   CB 1A     RR D
       CB 1B     RR E
       CB 3B     SRL E
       16 00     LD D,0
       19        ADD HL,DE
       EB        EX DE,HL        DE = MAGIC ADR

226C_H D9        EXX
       7D        LD A,L
       08        EX AF,AF'       NOW L
       7C        LD A,H
2270_H D9        EXX             HL = PATTERN ADDRESS
       67        LD H,A
       08        EX AF,AF'
       6F        LD L,A
       D9        EXX
       79        LD A,C
       08        EX AF,AF'       NOW C
       78        LD A,B          B = Y SIZE
       D9        EXX             C = X SIZE
       47        LD B,A
       79        LD A,C          A = UPDATED MR VALUE
       08        EX AF,AF'       MR VALUE
       4F        LD C,A
       08        EX AF,AF'       A = UPDATED MR VALUE
227E_H FD E9     JP (IY)
```

FILL AREA (USE TO FILL AQUARIUM WITH WATER)

ENTER WITH: E = XSIZE (# OF BYTES WIDE TO FILL)
D = YSIZE (# OF LINES HIGH TO FILL)
B = DATA TO FILL WITH
HL = SCREEN ADDRESS TO BEGIN FILLING
IX = PROGRAM "CONTINUE ADDRESS" (SUBSTITUTE FOR RETURN

FILL	2280_H	3E 50	LD A, 50_H ; HI-RES 80_D BYTES/LINE
		93	SUB E
		4F	LD C, A
		78	LD A, B
FILL1		43	LD B, E
FILL2		77	LD (HL), A
		23	INC HL
		10 FC	DJNZ FILL2
		09	ADD HL, BC
		15	DEC D
		20 F7	JR NZ, FILL1
	$228E_H$	DD E9	JP (IX)

RESTORE AREA (USED TO FILL AQUARIUM WITH PEBBLES)

"WRITE ONLY" VARIATION SIMILAR TO ON-BOARD SUB#46

NO RAM STACK USED

ENTER WITH: DE = SAVE AREA ADDRESS
HL = ADDRESS TO RESTORE TO
IX = PROGRAM "CONTINUE ADDRESS" (SUBSTITUTE FOR RET)

```
RESTOR 2290H  EB        EX DE,HL        DE = ADDRESS TO RESTORE TO
                                        HL = SAVE AREA ADDRESS
              4E        LD C,(HL)       C = X SIZE
              23        INC HL
              46        LD B,(HL)       B = Y SIZE
              23        INC HL
RESTOR1 2295H AF        XOR A           ZERO A
              ED 47     LD I,A          INTERRUPT REGI USED
RESTOR2 2298H 78        LD A,B          NOW B
              08        EX AF,AF'
              79        LD A,C          B' = Y SIZE
              D9        EXX             C' = X SIZE
              4F        LD C,A
              08        EX AF,AF'
              47        LD B,A
              D9        EXX
22A0H         7B        LD A,E          BOY
              08        EX AF,AF'       NOW E
              7A        LD A,D          DE' = ADR TO RESTORE TO
              D9        EXX
              57        LD D,A
              08        EX AF,AF'
              5F        LD E,A          DOG
              D9        EXX
              ED 57     LD A,I          INTERRUPT REGI USED
              47        LD B,A          ZERO B
              ED B0     LDIR            WRITE A LINE
22ADH         EB        EX DE,HL
```

```
22AE_H  D9          EXX
        7B  ← BOY   LD A,E   ← NOW E
22B0_H  08          EX AF,AF'
        7A          LD A,D
        D9          EXX
        67          LD H,A
        08          EX AF,AF'
        6F          LD L,A
        0E 50       LD C,50   HI-RES 80 BYTES/LINE
        09          ADD HL,BC
        EB          EX DE,HL
        D9          EXX
        79          LD A,C
        08          EX AF,AF'   ← NOW C
        78          LD A,B
        D9          EXX
        47          LD B,A
22C0_H  08          EX AF,AF'
        4F          LD C,A
        10 D4       DJNZ RESTOR2
22C4    DD E9       JP(IX)
```

HL = ADR TO RESTORE TO

B = Y SIZE

C = X SIZE

WRITE ANOTHER LINE?

-44

```
  32 16 8 4 2 1
  0 0 1 0 1 1 0 0
  1 1 0 1 0 0 1 1
                +1
  1 1 0 1   0 1 0 0
  D         4
```

DETERMINE MAGIC ADDRESS FROM X,Y COORDINATES

MAGIC ENTRY POINT

```
MENTR  22C6_H  FD 21 CD 22   LD IY,MENTR1   ← DOG
               C3 18 22     JP MADDR
MENTR1 22CD_H  D3 0C        OUT(MAGIC),A
       22CF_H  C3 38 21     JP WRITM
```

PAGE 0 INTRO
WRITE WITH FLOP EXPAND XOR TREE (PATTERN 12) 22
PAGE 2

PGM29 $22D2_H$	3E 08	LD A, 08	0000 1000 (BLK) — EXPAND WRITE TREE WITH COLOR GRN (GRN)
	D3 19	OUT (19_H), A	
	3E 8E	LD A, 142_D	I = REGY = Y COORD
	ED 47	LD I, A	
	11 4E00	LD DE, 78_D	DE = X_{COORD} = 78_D
	3E 68 (BOY)	LD A, 0110 1000	MR VALUE = EXPAND WITH OR'D FLOP
	21 B3 20	LD HL, $20B3_H$	HL = PATTERN ADDRESS - 2 (POINTING AT XSIZE)
$22E2_H$	DD 21 32 2A	LD IX, PAG3	IX = PROGRAM "CONTINUE" ADR
$22E6_H$	C3 2D 21	JP WRITP	WRITE TREE

WRITE A PARAGRAPH
(FOR MULTI-PAGER DEMO, PAGE 0 TEXT INTRO)

ENTER WITH: ($7FC0_H$) = NUMBER OF LINES IN PARAGRAPH
I = REGY = Y COORDINATE OF 1ST LINE
HL = ASCII CHAR STRING ADDRESS FOR 1ST LINE
EXPAND REGISTER SET UP

WPGPH $22E9_H$	AF	XOR A	DE = X_{COORD} = 0 FOR CHAR SCREEN FRAME
	5F	LD E, A	
	57	LD D, A	
	3A C07F	LD A, ($7FC0_H$)	B = NUMBER OF LINES IN PARAGRAPH
	47	LD B, A	
WPGPH1 $22F0_H$	C5	PUSH BC	SAVE LINE CTR
	D5	PUSH DE	SAVE INITIAL X_{COORD}

GET CHARACTER CODE FROM STRING

GCHAR	7E	LD A, (HL)	A = ASCII CHAR CODE
	FE 00	CP 00_H	JUMP IF CHAR CODE IS STRING TERMINATOR (A = 00_H) JUMP TO POINT TO NEXT LINE
	28 30	JR Z, PNLIN	
$22F7_H$	E5	PUSH HL	SAVE STRING ADR

PAGE 0 INTRO POINT TO CHR PATTERN IN TABLE

PCHAR 22F8H	D6 30	SUB 30H	SUBTRACT 30H FROM ASCII CODE
	67	LD H, A	IF ASCII CODE IS 30H, THEN HL = 0000H
	6F	LD L, A	
	A7	AND A	IF ASCII CODE IS 30H, INDEX ADJUSTMENT IS NOT REQ'D
	28 0C	JRZ, PCHAR2	
	47	LD B, A	B = CHAR INDEX
	D5	PUSH DE	SAVE X COORD
2300H	21 00 00	LD HL, 0000	ADJUST CHAR INDEX TO POINT AT CHAR PATTERN
	11 09 00	LD DE, 0009	
PCHAR1	19	ADD HL, DE	
-3 0000 0011 1111 1101 F D	10 FD ← DOG	DJNZ PCHAR1	
	D1 DOG	POP DE	DE = X COORD
PCHAR2	01 67 25	LD BC, CHART	BC = CHAR TABLE ADR
	09	ADD HL, BC	POINT HL AT CHAR PATTERN (1 BYTE WIDE × 9 BYTE HIGH PATTERN)

SET UP MAGIC REGISTER VALUE

3E 18	LD A, 0001 1000	A = REQUEST FOR EXPAND WITH OR WRITE (OR WRITE FOR 40TH CHAR WRAP AROUND)

WRITE CHAR ON SCREEN

WRCHR 2311H	01 01 09	LD BC, 0901H	B = YSIZE = LINES HIGH = 09 C = XSIZE = BYTES WIDE = 01
	DD 21 1B 23 ← BOY	LD IX, WRCHR1	IX = PROGRAM "CONTINUE ADDRESS"
	C3 31 21	JP WRIT	WRITE THE CHAR

POINT TO NEXT STRING CHAR CODE

WRCHR1 231BH	E1	POP HL	HL POINTS TO LAST CHAR ASCII CODE
	23	INC HL	POINT TO NEXT CODE

POINT TO NEXT SCREEN CHAR FRAME

PNFRM	D1	POP DE	DE = OLD LINE X COORD
	EB	EX DE, HL	HL = OLD LINE X COORD DE = POINTS TO NEXT CHAR CODE
	01 08 00	LD BC, 0008	POINT HL TO NEXT SCREEN CHAR FRAME
2322H	09	ADD HL, BC	
	EB	EX DE, HL	DE = NEXT FRAME X COORD HL POINTS TO NEXT CHAR CODE
	D5	PUSH DE	SAVE UPDATED FRAME X COORD
-53 32 16 8 4 2 1 0 0 1 0 1 0 0 1 0 1 0 1 1 0 0 1 0 1 0 1 1 E B	18 CB ← BOY 2325H	JR GCHAR	LOOP BACK TO WRITE NEXT CHAR

PAGE 0 INTRO

POINT TO NEXT LINE

PNLIN 2327H	D1 ← DOG	POP DE	TOSS OUT LAST CHAR FRAME X COORD
	AF	XOR A	DE = X COORD = 0
	5F	LD E,A	
	57	LD D,A	
	C1	POP BC	B = LINE COUNTER
	ED 57	LD A,I	I = REG Y = Y COORD FOR NEXT LINE (LEAVE 3 LINES OF SPACE BETWEEN CHAR LINES
	C6 0C	ADD A,12D	
	ED 47	LD I,A	
	05	DEC B	DEC LINE CTR
	C8	RET Z	RETURN IF LAST PARAGRAPH LINE IS WRITTEN
	23	INC HL	POINT HL AT NEXT CHAR STRING
2335H	18 B9	JR WPGH1	(STEP PAST PREVIOUS STRING TERMINATOR) LOOP BACK TO WRITE NEXT LINE

-71
54321 68421
0 0 0 1 1 1
0 1 0 1 1 0 0 0 +1
1011 1001
B 9

PAGE 1 COLOR TABLE

CLRT3 2337H	09	MEDIUM BLUE	PIXEL 11	LEFT COLORS
	86	YELLOW	10	
	4D ← DOG	PINK	01	
	00	BLACK	00	
	FA	LIGHT BLUE	PIXEL 11	RIGHT COLORS
	07	WHITE	10	
	63	ORANGE	01	
233E	F9	BLUE	00	

CLRT4 PAGE 4 COLOR TABLE

233FH	AC	GREEN	PIXEL 11	LEFT COLORS
2340H	86	YELLOW	10	
	07	WHITE	01	
	00	BLACK	00	
	CD ← DOG	CYAN	PIXEL 11	RIGHT COLORS
	5A	RED	10	
	2B	MAGENTA	01	
2346H	F9	BLUE	00	

PAGE 7 HI-RES GUNFIGHT SCREENSHOT

COLOR TABLE

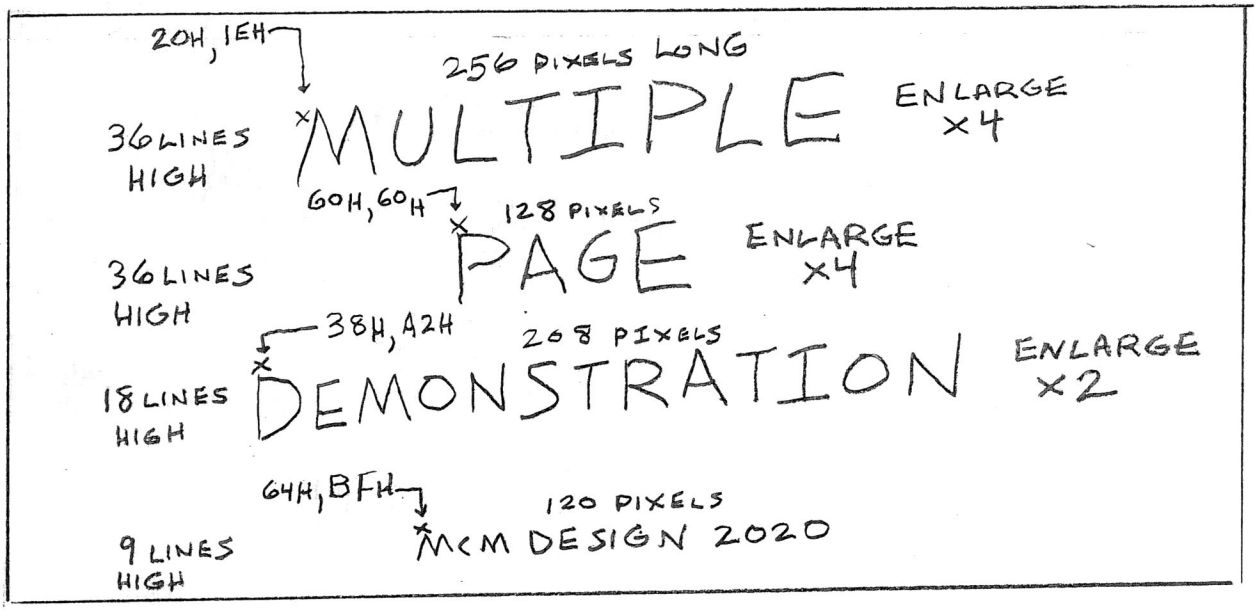
CLRT6 2347_H

A2	GREEN	PIXEL 11	LEFT COLORS
7D	YELLOW	10	
0B	BLUE	01	
00	BLACK	00	
A2	GREEN	PIXEL 11	RIGHT COLORS
7D	YELLOW	10	
6C	RED	01	
00	BLACK	00	

UNUSED BYTE

$234F_H$ FF

PAGE 6 MULTIPAGER TITLE PAGE



TEXT STRING PLACEMENTS
(FOR REFERENCE ONLY)

"HI-RES "WRITE ONLY" GRAPHICS PROGRAM (WITH FILL + MAGIC WRITES) 26

← 15 MAGIC PATTERN WRITES + FISH TANK

```
PGM 2350   F3          DI
           3E 01       LDA,1
           D3 08       OUT(08),A
           AF          THIS PROGRAM COPIED FROM
           01 18 08    BALCHECKHR 8KB PACKAGE
           ED 79       2350 - 236B_H NOT USED WITH
           10 FC       MULTI-PAGER TEST DEMO
           3E CC
           D3 0A
  2361_H   3E 9C
           D3 09
           21 03 20
           01 0B 08
  236B_H   ED B3
```

SET CUSTOM CHIPS IN HI-RES MODE

STOP ALL SOUND

SET VERT BLNK REG TO 204_D

SET HOR CLR BNDRY
10 01 1100
W BCK GRD 28_D

SET COLORS

JUMP → MULTI-PAGER PAGE 2 WRITE JUMPS TO THIS ADDRESS

```
PGMF 236D_H   AF          XORA    A=0
             01 C0 3F     LD BC, 3FC0_H    BC = # OF BYTES TO CLEAR
  2371_H     11 00 40     LD DE, 4000_H    DE = START ADR
             EB           EX DE,HL         HL = SCREEN ADR
PGM1 2375_H  77           LD (HL),A        CLEAR BYTE
             ED A1        CPI              HL ← HL+1, BC ← BC-1
  2378_H     EA 75 23     JP PE, PGM1      LOOP BACK IF NOT DONE
```

CLEAR SCREEN
16,320 BYTES

CONTINUE NEXT PAGE 27

PROGRAM CNT'D

PAINT LEFT SIDE OF FISH TANK

$237B_H$	3E 20	LD A, 32_D	} I=REGY=32_D
	ED 47	LD I, A	
	11 6B00 ←BOY	LD DE, 107_D	X=107_D
			MR=0
2382_H	AF	XOR A	
	FD21 8A 23	LD IY, PGM2	
	C3 18 22	JP MADDR	DETERMINE MAGIC ADR
PGM2 $238A_H$	CB F2	SET 6, D	NON MAGIC, NOW SCREEN ADR
	EB	EX DE, HL	HL=SCREEN ADR
	11 01 89	LD DE, 8901_H	E=XSIZE=1
			D=YSIZE=137_D
2390_H	06 0F	LD B, $0F_H$	FILL DATA = 0000 1111
	DD 21 99 23	LD IX, PGM3	
	C3 80 22	JP FILL	

PAINT RIGHT SIDE OF FISH TANK

PGM3 2399_H	11 D4 00	LD DE, 212_D	
	AF	XOR A	
	FD 21 A4 23	LD IY, PGM4	
$23A1_H$	C3 18 22	JP MADDR	
PGM4 $23A4_H$	CB F2	SET 6, D	
	EB	EX DE, HL	
	11 01 89	LD DE, 8901_H	
	06 F0	LD B, $F0_H$	FILL DATA = 1111 0000
	DD 21 B3 23	LD IX, PGM5	
$23B0_H$	C3 80 22	JP FILL	

PAINT BOTTOM OF FISH TANK

PGM5 $23B3_H$	3E A6	LD A, 166_D	
	ED 47	LD I, A	
	11 6C 00	LD DE, 108_D	
	AF	XOR A	
	FD 21 C2 23	LD IY, PGM6	
	C3 18 22	JP MADDR	
PGM6 $23C2_H$	CB F2	SET 6, D	
	EB	EX DE, HL	
	11 1A 01	LD DE, $011A_H$	
	06 FF	LD B, FF_H	FILL DATA = 1111 1111
	DD 21 D1 23	LD IX PGM7	
$23CE_H$	C3 80 22	JP FILL	

PLACE PEBBLES IN BOTTOM OF TANK

PGM 7 $23D1_H$	3E9C	LD A, 156_D	
	ED 47	LD I, A	
	116C00	LD DE, 108_D	$X=108_D$
	AF	XOR A	MR=0
	FD 21 E0 23	LD IY, PGM8	
	C3 1822	JP MADDR	
PGM 8 $23E0_H$	CB F2	SET 6, D	
	218504	LD HL, 0485_H	HL= SAVE AREA ADDRESS
	011A0A	LD BC, $0A1A_H$	C= XSIZE = 26_D B= YSIZE = 10_D
	DD 21 EF 23	LD IX, PGM9	
	C3 9522	JR RESTOR1	

WRITE WITH "OR" COMBOY (PATTERN 1)

PGM 9 23EF	3E0A	LD A, 10_D	
$23F1_H$	ED 47	LD I, A	$Y=10_D$
	111200	LD DE, 18_D	$X=18_D$
	3E10	LD A, 10_H	OR IT
	21 E0 20	LD HL, $20E0_H$	
	DD 21 0224	LD IX, PGM10	
	C3 2D 21	JP WRITP	

WRITE WITH "OR" COWBOY'S ARM (PATTERN 1)

PGM10 2402_H	111200	LD DE, 18_D	$X=18_D$
	060A	LD B, 10_D	$B=Y=10_D$
	3E10	LD A, 10_H	OR IT
	21 D4 20	LD HL, $20D4_H$	
	DD 21 1324	LD IX, PGM11	
2410_H	C3 1E 21	JP WRITR	

WRITE WITH PLOP WAGON (PATTERN 2)

PGM11 2413_H	112E00	LD DE, 46_D	$X=46_D$
	0609	LD B, 9_D	$Y=9_D$
	AF	XOR A	PLOP IT
	21 5720	LD HL, 2057_H	
	DD 21 2324	LD IX, PGM12	
2420_H	C3 1E 21	JP WRITR	

FILL FISH TANK WITH WATER

```
PGM12 2423H  3E 2A        LD A, 42D
             ED 47        LD I, A
             11 6C 00     LD DE, 108D
             AF           XOR A
             FD 21 32 24  LD IY, PGM13
             C3 18 22     JP MADDR
PGM13 2432H  CB F2        SET 6, D
             EB           EX DE, HL
             11 1A 72     LD DE, 721AH   E=XSIZE=26D, D=YSIZE=114D
             06 55        LD B, 55H
             DD 21 41 24  LD IX, PGM14
             C3 80 22     JP FILL
```

WRITE WITH FLOP PLOP COWBOY (PATTERN 3)

```
PGM14 2441H  3E 0A        LA, 10D
             ED 47        LD I, A        Y=10D
             11 E6 00     LD DE, 230D    X=230D
             3E 40        LD A, 40H      FLOP PLOP IT
             21 E0 20     LD HL, 20E0H
             DD 21 54 24  LD IX, PGM15
      2451H  C3 2D 21     JP WRITP
```

WRITE WITH "OR" COWBOYS ARM (PATTERN 3)

```
PGM15 2454H  11 E6 00     LD DE, 230D    X=230D
             06 0A        LD B, 10D      Y=10D
             3E 50        LD A, 50H      FLOP OR IT
             21 D4 20     LD HL, 20D4H
             DD 21 65 24  LD IX, PGM16
      2462H  C3 1E 21     JP WRITR
```

WRITE WITH EXPAND PLOP TREE (PATTERN 4)

```
PGM16 2465   3E 08        LD A, 08H
             D3 19        OUT (19H), A   COLOR IT YELLOW
             3E 86        LD A, 134D
             ED 47        LD I, A        Y=134D
             11 32 00     LD DE, 50D     X=50D
      2470H  3E 08        LD A, 08H      EXPAND PLOP IT
             21 B3 20     LD HL, 20B3H
             DD 21 7C 24  LD IX, PGM17
      2479H  C3 2D 21     JP WRITP
```

WRITE WITH OR CRITTER (PATTERN 5)

```
PGM17  247C_H  11 1C 00      LD DE,28_D
               0699          LD B,153_D
       2481_H  3E 10         LD A,10_H
               210B20        LD HL,200B_H
               DD218D24      LD IX,(PGM18)
               C31E21        JP WRITR
```

WRITE WITH EXPAND OR CACTUS (PATTERN 6)

```
PGM18  248D_H  3E0C          LD A,0C
               D319          OUT(19_H),A   COLOR IT YELLOW
       2491_H  3E9C          LD A,156_D
               ED47          LD I,A
               114400        LD DE,68_D
               3E18          LD A,18_H     OR EXPAND IT
               21C620        LD HL,20C6_H
               DD21A424      LD IX,PGM19
       24A1_H  C32D21        JP WRITP
```

WRITE WITH EXPAND XOR CACTUS (PATTERN 7)

```
PGM19  24A4_H  3E0C          LD A,0C_H
               D319          OUT(19_H),A
               3E13          LD A,19_D
               ED47          LD I,A
               11EB00        LD DE,235_D   X=235_D
               3E28          LD A,28_H     XOR EXPAND IT
       24B1_H  21C620        LD HL,20C6_H
               DD21BB24      LD IX,PGM20
               C32D21        JP WRITP
```

WRITE WITH FLOP OR COWBOY (PATTERN 8)

```
PGM20  24BB_H  3E0F          LD A,15_D
               ED47          LD I,A        Y=15_D
               111400        LD DE,20_D    X=20_D
       24C2_H  3E50          LD A,50_H     FLOP OR IT
               21E020        LD HL,20E0_H
               DD21CE24      LD IX,PGM21
       24CB_H  C32D21        JP WRITP
```

WRITE WITH FLOP OR COWBOY'S ARM (PATTERN 8)

PGM21	$24CE_H$	111400	LD DE, 20_D	$X=20_D$
	$24DI_H$	060F	LD B, $0F_H$	$Y=15_D$
		3E50	LD A, 50_H	FLOP OR IT
		21D420	LD HL, $20D4_H$	
		DD21DF24	LD IX, PGM22	
		C31E21	JP WRITR	

WRITE WITH XOR CRITTER (PATTERN 9)

PGM22	$24DF_H$	110101	LD DE, 257_D	$X=257_D$
	$24E2_H$	0628	LD B, 40_D	$Y=40_D$
		3E20	LD A, 20_H	XOR IT
		210B20	LD HL, $200B_H$	
		DD21F024	LD IX, PGM23	
		C31E21	JP WRITR	

WRITE WITH FLOP EXPAND PLOP TREE (PATTERN 10)

PGM23	$24F0_H$	3E04	LD A, 04	
		D319	OUT (19_H), A	
		3E4E	LD A, 78_D	
		ED47	LD I, A	$Y=78_D$
		114E00	LD DE, 78_D	$X=78_D$
		3E48	LD A, 48_H	FLOP EXPAND PLOP IT
		21B320	LD HL, $20B3_H$	
	2500_H	DD21072 5	LD IX, PGM24	
		C32D21	JP WRITP	

WRITE WITH FLOP EXPAND OR CACTUS (PATTERN 11)

PGM24	2507_H	3E0C	LD A, 0C	
		D319	OUT (19_H), A	
		3E70	LD A, 112_D	
		ED47	LD I, A	$Y=112_D$
		114F00	LD DE, 79_D	$X=79_D$
	2512_H	3E58	LD A, 58_H	FLOP EXPAND OR IT
		21C620	LD HL, $20C6_H$	
		DD211E25	LD IX, PGM25	
	$251B_H$	C32D21	JP WRITP	

PATTERN 12, WRITE WITH FLOP EXPAND OR TREE WAS ADDED AT END OF PROGRAM ← P.32

WRITE WITH EXPAND PLOP CACTUS (PATTERN 13)

```
PGM25  251E_H  3E 08        LD A,08
       2520_H  D3 19        OUT (19_H),A
               3E 50        LD A,80_D
               ED 47        LD I,A        Y=80_D
               11 2201      LD DE,290_D   X=290_D
               3E 08        LD A,08  EXPAND PLOP IT
               21 C620      LD HL,20CG
               DD21 3525    LD IX,PGM26
       2532_H  C3 2D 21     JP WRITP
```

WRITE WITH EXPAND OR TREE (PATTERN 14)

```
PGM26  2535_H  3E 04        LD A,04
               D3 19        OUT (19_H),A
               3E 6E        LD A,110_D
               ED 47        LD I,A   Y=110_D
               11 2101      LD DE,289_D  X=289_D
       2540_H  3E 18        LD A,18_H
               21 B320      LD HL,20B3_H
               DD21 4C25    LD IX,PGM27
               C3 2D 21     JP WRITP
```

WRITE WITH FLOP EXPAND PLOP CACTUS (PATTERN 15)

```
PGM27  254C_H  3E 0C        LD A,0C
               D3 19        OUT (19_H),A
       2550_H  3E 90        LD A,144_D
               ED 47        LD I,A        Y=144_D
               11 1900      LD DE,25_D    X=25_D
               3E 48        LD A,48_H  FLOP EXPAND PLOP IT
               21 C620      LD HL,20C6_H
               DD21 6325    LD IX,PGM28
       2560_H  C3 2D 21     JP WRITP
```

← P.22

```
PGM28  2563_H  C3 D2 22     JP PGM29   JMP TO WRITE PATTERN 12
                  DOG
PGM30  2566_H  76           HALT
```

15 MAGIC PATTERN WRITES + FISH TANK ←

END OF PROGRAM

CHARACTER TABLE

CHAR SIZE = 7 PIXELS WIDE X 9 PIXEL LINES HIGH
CHAR FRAME WIDTH = 2 BYTES
40 CHAR MAX PER SCREEN LINE
THERE WILL BE NO CARRIAGE RETURN
1 PIXEL HEIGHT BETWEEN CHAR LINES (REVISED. SPACE INCREASED TO ?)
20 CHAR LINES MAX PER PAGE
20 x 10 = 200 PIXEL LINES DEVOTED TO CHAR TEXT
THERE IS NO FONT DESCRIPTOR TABLE.
NEED: CAPITAL LETTERS A THRU Z, ASCII CODES 41 THRU 5A
· 0 THRU 9, ASCII CODES 30 THRU 39

USE NONSTANDARD ASCII CODES FOR:

ASCII	
3A	:
3B	- HYPHEN
3C	, COMMA
3D	=
3E	. PERIOD
3F	?
40	SPACE

CHARACTER STRING WRITE ROUTINE WILL DISPLAY ONLY
ASCII CODES 30 THRU 5A

CHART

											ASCII
2567_H	7C	82	86	8A	92	A2	C2	82	7C	0	30
2570_H	10	30	10	10	10	10	10	10	7C	1	31
2579_H	7C	82	02	02	7C	80	80	80	FE	2	32
2582_H	7C	82	02	02	1E	02	02	82	7C	3	33
$258B_H$	84	84	84	84	FE	04	04	04	04	4	34
2594_H	FE	80	80	FC	02	02	02	82	7C	5	35
$259D_H$	7C	82	80	80	FC	82	82	82	7C	6	36
$25A6_H$	FE	02	04	08	10	20	40	40	40	7	37
$25AF_H$	7C	82	82	82	7C	82	82	82	7C	8	38
$25B8_H$	7C	82	82	82	7E	02	02	82	7C	9	39
$25C1_H$	00	00	00	10	00	10	00	00	00	:	3A
$25CA_H$	00	00	00	00	7C	00	00	00	00	-	3B
$25D3_H$	00	00	00	00	00	00	10	10	20	,	3C
$25DC_H$	00	00	00	7C	00	7C	00	00	00	=	3D
$25E5_H$	00	00	00	00	00	00	00	00	10	.	3E
$25EE_H$	08	1C	2A	49	08	08	08	08	08	↑	3F
$25F7_H$	00	00	00	00	00	00	00	00	00	SPACE	40
2600_H	7C	82	82	82	FE	82	82	82	82	A	41
2609_H	FC	82	82	82	FC	82	82	82	FC	B	42
2612_H	7C	82	80	80	80	80	80	82	7C	C	43
$261B_H$	F8	84	82	82	82	82	82	84	F8	D	44
2624_H	FE	80	80	80	F0	80	80	80	FE	E	45
$262D_H$	FE	80	80	80	F0	80	80	80	80	F	46
2636_H	7E	80	80	80	8E	82	82	82	7E	G	47
$263F_H$	82	82	82	82	FE	82	82	82	82	H	48
2648_H	FE	10	10	10	10	10	10	10	FE	I	49
2651_H	02	02	02	02	02	02	02	82	7C	J	4A
$265A_H$	82	84	88	D0	A0	90	88	84	82	K	4B
2663_H	80	80	80	80	80	80	80	80	FE	L	4C
$266C_H$	82	C6	AA	92	82	82	82	82	82	M	4D
2675_H	82	82	C2	A2	92	8A	86	82	82	N	4E
$267E_H$	7C	82	82	82	82	82	82	82	7C	O	4F
2687_H	FC	82	82	82	FC	80	80	80	80	P	50

			ASCII
2690_H	7C 82 82 82 82 82 8A 84 7A	Q	51
2699_H	FC 82 82 82 FC 90 88 84 82	R	52
$26A2_H$	7C 82 80 80 7C 02 02 82 7C	S	53
$26AB_H$	FE 10 10 10 10 10 10 10 10	T	54
$26B4_H$	82 82 82 82 82 82 82 82 7C	U	55
$26BD_H$	82 82 82 82 82 82 44 28 10	V	56
$26C6_H$	82 82 82 82 82 92 AA C6 82	W	57
$26CF_H$	82 82 44 28 10 28 44 82 82	X	58
$26D8_H$	82 82 44 28 10 10 10 10 10	Y	59
$26E1_H$	FE 02 04 08 10 20 40 80 FE	Z	5A

COLOR TABLE (PAGE 0 INTRO)

THE COLOR TABLE CLRT2 IS LOCATED AT 2132_H

CHARACTER STRINGS (PAGE 0 INTRO)

PARAGRAPH 1

		M C M D E S I G N I S P R
LINE 0	$26EA_H$	4D 43 4D 40 44 45 53 49 47 4E 40 49 53 40 50 52
		O U D T O A N N O U N C E
	$26FA_H$	4F 55 44 40 54 4F 40 41 4E 4E 4F 55 4E 43 45
		I T S .
	2709_H	40 49 54 53 00

44

		C R E A T I O N O F A M O D
LINE 1	$270E_H$	43 52 45 41 54 49 4F 4E 40 4F 46 40 41 40 4D 4F
		I F I E D H I - R E S A S T
	$271F_H$	49 46 49 45 44 40 48 49 3B 52 45 53 40 41 53 54
		R O C A D E
	$272F_H$	52 4F 43 41 44 45 00

```
                W H I C H   U T I L I Z E S
LINE2 2736H 57 48 49 43 48 40 55 54 49 4C 49 5A 45 53 40
            O N L Y   4   S T A T I C   S C
      2745H 4F 4E 4C 59 40 34 40 53 54 41 54 49 43 40 53 43
            R E E N   R A M
      2755H 52 45 45 4E 40 52 41 4D 00

            C H I P S .   I N   A D D I T I
LINE3 275EH 43 48 49 50 53 3E 49 4E 40 41 44 44 49 54 49
            O N   T O   T H E   N E W   R A
      276DH 4F 4E 40 54 4F 40 54 48 45 40 4E 45 57 40 52 41
            M   S C H E M E ,
      277DH 4D 40 53 43 48 45 4D 45 3C 00

            A   S C R E E N   M U L T I - P
LINE4 2787H 41 40 53 43 52 45 45 4E 40 4D 55 4C 54 49 3B 50
            A G E R   W A S   A D D E D .
      2797H 41 47 45 52 40 57 41 53 40 41 44 44 45 44 3E
              E A C H   R A M
      27A6H 40 45 41 43 48 40 52 41 4D 00

            C H I P   S T O R E S   3 2 K B
LINE5 27B0H 43 48 49 50 40 53 54 4F 52 45 53 40 33 32 4B 42
              O F   D A T A .   T H E   H I -
      27C0H 40 4F 46 40 44 41 54 41 3E 40 54 48 45 40 48 49 3B
            R E S   R A M
      27D1H 52 45 53 40 52 41 4D 00

            M O D E   U T I L I Z E S   1 6
LINE6 27D9H 4D 4F 44 45 40 55 54 49 4C 49 5A 45 53 40 31 36
            K B .   S O ,   T H E   M U L T
      27E9H 4B 42 3E 40 53 4F 3C 54 48 45 40 4D 55 4C 54
            I - P A G E R
      27F8H 49 3B 50 41 47 45 52 00
```

```
                CAN DISPLAY 8 PAG
LINE 7  2800H 43 41 4E 40 44 49 53 50 4C 41 59 40 38 40 50 41 47
                ES OF SCREEN RAM,
        2811H 45 53 40 4F 46 40 53 43 52 45 45 4E 40 52 41 4D 3C
                  WHICH
        2822H 40 57 48 49 43 48 00

                MAPS 320 X 204 P
LINE 8  2829H 4D 41 50 53 40 33 32 30 40 58 40 32 30 34 40 50
                IXELS TO THE SCR
        2839H 49 58 45 4C 53 40 54 4F 40 54 48 45 40 53 43 52
                EEN.
        2849H 45 45 4E 3E 00
```

PARAGRAPH 2

```
LINE 9  284EH THIS DEMO TESTS
              54 48 49 53 40 44 45 4D 4F 40 54 45 53 54 53 40
                AND DEMONSTRATES
        285EH 41 4E 44 40 44 45 4D 4F 4E 53 54 52 41 54 45 53 4C
                THE
        286FH 54 48 45 00

                HI-RES MULTI-PA
LINE 10 2873H 48 49 3B 52 45 53 40 4D 55 4C 54 49 3B 50 41
                GER, A HI-RES FI
        2882H 47 45 52 3E 40 41 40 48 49 3B 52 45 53 40 46 49
                SH DEMO
        2892H 53 48 40 44 45 4D 4F 00
```

```
                F O L L O W S   T H E   M
LINE11 289A_H 46 4F 4C 4C 4F 57 53 40 54 48 45 40 4D
                U L T I - P A G E R   D E M O
       28A7_H 55 4C 54 49 3B 50 41 47 45 52 40 44 45 4D 4F
                .   T O   R U N   T H E
       28B6_H 3E 40 54 4F 40 52 55 4E 40 54 48 45 00

                F I S H   D E M O   N O N S T O P
LINE12 28C3_H 46 49 53 48 40 44 45 4D 4F 40 4E 4F 4E 53 54 4F 50
                ,   P R E S S   A N Y   K E Y
       28D4_H 3C 40 50 52 45 53 53 40 41 4E 59 40 4B 45 59 40
                O N   T H E
       28E4_H 4F 4E 40 54 48 45 00

                K E Y P A D   W H I L E   T
LINE13 28EB_H 4B 45 59 50 41 44 40 57 48 49 4C 45 40 54
                H E   F I S H   D E M O   I S
       28F9_H 48 45 40 46 49 53 48 40 44 45 4D 4F 40 49 53
                R U N N I N G .
       2908_H 40 52 55 4E 4E 49 4E 47 3E 00
```

PARAGRAPH 3

```
                Y O U   C A N   R E D U C E
LINE14 2912_H 59 4F 55 40 43 41 4E 40 52 45 44 55 43 45 40
                T H E   I N T R O   P A G E   40
       2921_H 54 48 45 40 49 4E 54 52 4F 40 50 41 47 45
                R E A D   D E L A Y
       2930_H 52 45 41 44 40 44 45 4C 41 59 00
```

B Y H O L D I N G D O W N

LINE15 293B$_H$ 42 59 40 48 4F 4C 44 49 4E 47 40 44 4F 57 4E 40

A K E Y F O R A

294B$_H$ 41 40 4B 45 59 40 46 4F 52 40 41 40

W H I L E . T H E

2957$_H$ 57 48 49 4C 45 3E 40 54 48 45 00

T E S T D E M O W I L L

LINE16 2962$_H$ 54 45 53 54 40 44 45 4D 4F 40 57 49 4C 4C 40

E X E C U T E A U T O M A T I

2971$_H$ 45 58 45 43 55 54 45 40 41 55 54 4F 4D 41 54 49

C A L L Y .

2981$_H$ 43 41 4C 4C 59 3E 00

PAGE 0 INTRO

MCM DESIGN IS PROUD TO ANNOUNCE ITS CREATION OF A MODIFIED HI-RES ASTROCADE WHICH UTILIZES ONLY 4 STATIC SCREEN RAM CHIPS. IN ADDITION TO THE NEW RAM SCHEME, A SCREEN MULTI-PAGER WAS ADDED. EACH RAM CHIP STORES 32kB OF DATA. THE HI-RES RAM MODE UTILIZES 16kB. SO, THE MULTI-PAGER CAN DISPLAY 8 PAGES OF SCREEN RAM, WHICH MAPS 320 x 204 PIXELS TO THE SCREEN.

← COLOR 01 GREEN

NO SPACE

THIS DEMO TESTS AND DEMONSTRATES THE HI-RES MULTI-PAGER. A HI-RES FISH DEMO FOLLOWS THE MULTI-PAGER DEMO. TO RUN THE FISH DEMO NONSTOP, PRESS ANY KEY ON THE KEYPAD WHILE THE FISH DEMO IS RUNNING.

← COLOR 02 LT BLUE

YOU CAN REDUCE THE INTRO PAGE READ DELAY BY PRESSING ANY KEY ON THE KEYPAD. THE TEST DEMO WILL EXECUTE AUTOMATICALLY.

← COLOR 03 RED

CLEAR SCREEN (SUBROUTINE)

REVISED

CLEARS ALL ~~204~~ 203 LINES IN SCREEN.
TO CLEAR FEWER LINES, SET UP BC WITH BYTE COUNT TO CLEAR, THEN CALL CSCRN 1

```
CSCRN   2988H  01 70 3F      LD BC, ~~3FC0~~ 3F70H
CSCRN1  298BH  AF            XOR A
CSCRN2  298CH  21 00 40      LD HL, 4000H
CSCRN3  298FH  77            LD (HL), A
        2990H  ED A1         CPI
               EA 8F 29      JP PE CSCRN3
        2995H  C9            RET
```

CLEAR SCREEN (CLEAR ALL ~~204~~ 203 LINES)
203 × 80 = 16,240 BYTES = 3F70H

ENTRANCE TO MULTI-PAGER DEMO

PAGE 0 DEMO TEXT INTRO PROGRAM

```
PAG0    2996H  F3            DI                  DISABLE INTERRUPTS
               AF            XOR A
               01 18 08      LD BC, 0818H        } STOP ALL SOUND
PAG0A          ED 79         OUT (C), A          }
               10 FC         DJNZ PAG0A          }
               3E 81         LD A, 81H           } ENABLE HI-RES MODE
        29A1H  D3 08         OUT (08H), A        } USING MULTI-PAGER
               AF            XOR A               A=0
               D3 74         OUT (74), A         DISPLAY PAGE 0
               D3 75         OUT (75H), A        R/W PAGE 0
               3E CB         LD A, 203D          } SET VERT BLANK REG = 203D
               D3 0A         OUT (0AH), A        }
        29ACH  3E 2B         LD A, 00 10 10 11   } SET HORIZ COLOR BNDRY
               D3 09         OUT (09), A         } 00 10 1011
                                                 } BLK ↑ BORDER COLOR  43D
        29B0H  21 34 21      LD HL, 2134H        } SET SCREEN COLORS
               01 0B 08      LD BC, 080BH        }
               ED B3         OTIR                }
        29B8H  31 C0 7F      LD SP, 7FC0H        SET STACK POINTER
               01 70 3F      LD BC, 3F70H        CLEAR 203 LINES
        29BEH  CD 8B 29      CALL CSCR1          203 × 80 = 16,240 BYTES
                                                 = 3F70H
```

29C1$_H$	AF	XOR A	] INITIALIZE I=REG Y=Y COORD=0
	ED 47	LD I, A	] FOR FIRST TEXT LINE

WRITE PARAGRAPH 1 (GREEN TEXT)

	3E 09	LD A, 9	] (7FC0$_H$) = LINES IN PARAGRAPH
	32 C0 7F	LD (7FC0$_H$), A	] PARAGRAPH 1 HAS 9 LINES
	21 EA 26	LD HL, LINE0	POINT HL AT 1ST PARAGRAPH 1 LINE
	3E 04	LD A, 0000 0100	] EXPAND WRITE WITH
	D3 19	OUT (XPAND), A	] GREEN TEXT ON BLK BACKGROUND
29D0$_H$	CD E9 22	CALL WPGPH	WRITE 1ST PARAGRAPH

WRITE PARAGRAPH 2 (LT BLUE TEXT)

	3E 05	LD A, 5	] (7FC0$_H$) = LINES IN PARAGRAPH = 5
	32 C0 7F	LD (7FC0$_H$), A	
	21 4E 28	LD HL, LINE9	POINT HL AT 1ST PARAGRAPH 2 LINE
	3E 08	LD A, 0000 1000	] EXPAND WRITE WITH
	D3 19	OUT (XPAND), A	] LT BLUE TEXT ON BLK BACKGROUND
	CD E9 22	CALL WPGPH	WRITE 2ND PARAGRAPH

WRITE PARAGRAPH 3 (RED TEXT)

29E2$_H$	3E 03	LD A, 3	] (7FC0$_H$) = LINES IN PARAGRAPH = 3
	32 C0 7F	LD (7FC0$_H$), A	
	21 12 29	LD HL, LINE 14	POINT HL AT 1ST PARAGRAPH 3 LINE
	3E 0C	LD A, 0000 1100	] EXPAND WRITE WITH
	D3 19	OUT (XPAND), A	] RED TEXT ON BLK BACKGROUND
	CD E9 22	CALL WPGPH	
29F1$_H$	C3 19 2A	JP PAG1	JUMP TO WRITE PAGE 1

PAGE 44

TIME DELAY (A VARIATION FROM BALCHECKHR)

ENTER WITH : REG I = VARY DELAY COUNTER 1
↑ Z80 INTERRUPT REGISTER
EXAMPLES I=2 NEARLY 1 SEC
I=10 NEARLY 4 SEC (INITIAL SETTING)

NOTE:
SINCE REG I IS USED AS THE TIME DELAY VARIABLE, SCREEN INTERRUPTS CAN NOT BE UTILIZED DURING THIS DEMO.

```
DELAY   29F4  ED 57      LD A, I          ] D = CTR 1
              57         LD D, A          ]
              1E FF      LD E, 255D       E = CTR 2 = 255D
DELAY 1       06 FF      LD B, 255D       B = CTR 3 = 255D
DELAY 2       10 FE      DJNZ DELAY 3     LOOP BACK TO DELAY 3  255 TIMES
DELAY 3       1D  <- DOG DEC E            ] LOOP BACK TO DELAY 2  255 TIMES
              20 F9      JR NZ, DELAY 2   ]
                                          A = ABOVE 11 CTR 1 TIME
        2A00H 3E 0B      LD A, 11D
                 <- BOY
              BA         CP D             ] IF D IS AT OR BELOW INITIAL PAGE SKIP DELAY (10D),
              30 10      JR NC, DELAY 6   ] THEN SKIP KEYPAD SCAN.
              06 04      LD B, 4          B = PORT LOOP CTR
              0E 14      LD C, 14H        C = INPUT PORT 14H        4 KEYPAD COLUMN INPUT PORTS 14-17H
                 <- DOG
DELAY 4       ED 78      IN A, (C)        ] IF KEY IN COLUMN IS PRESSED, JMP TO DELAY 5
              A7         AND A            ]
              20 05      JR NZ, DELAY 5   ]
              0C         INC C            ] CHECK NEXT KEYPAD COLUMN
              10 F8      DJNZ DELAY 4     ]
        2A11H 18 02      JR DELAY 6       NO KEY PRESSED
              16 04      LD D, 4          KEY PRESSED, SET CTR 1 TO INITIAL PAGE SKIP DELAY
              15         DEC D            DECREMENT CTR 1 LOOP
              20 E1      JR NZ, DELAY 1
        2A18H C9         JP (HL)          RETURN TO MAIN PROGRAM
```

-8
0000 1000
1111 0111 +1
F 8

DELAY 5
DELAY 6

-31
32 16 8 4 2 1
0001 1111
1110 0000 +1
E 1

PAGE 1 COLOR TABLE

LOCATED AT 2337H (SEE PAGE 24)

WRITE PAGE 1 WRITE NARROW VERTICAL STRIPES

```
PAG1   2A19H  3E 10      LD A, 0001 0000
               D3 75      OUT (75H), A
       2A1DH   3E 22      LD A, 0010 0010
               21 00 40   LD HL, 4000H
       2A22H   01 C0 3F   LD BC, 3FC0H
PAG1A  2A25H   77         LD (HL), A
               ED A1      CPI
               EA 25 2A   JP PE, PAG1A
```

Z80 TO WRITE TO PAGE 1

FILL PAGE 1 WITH NARROW VERTICAL STRIPES (WRITE EVERY BYTE WITH 0001 0001)

80 × 204 = 16,3200 BYTES = 3FC0H

WRITE PAGE 2 AQUARIUM +15 MAGIC WRITES

```
PAG2   2A2BH  3E 22      LD A, 0010 0010
               D3 75      OUT (75H), A
               C3 6D 23   JP PGMF   ~P.26
```

Z80 TO W/R TO PAGE 2 READ NEEDED FOR MAGIC XOR/OR WRITES

JUMP TO WRITE PAGE 2

WRITE PAGE 3 NARROW HORIZONTAL STRIPES

```
PAG3   2A32H  3E 30      LD A, 0011 0000
               D3 75      OUT (75H), A
               21 00 40   LD HL, 4000H     POINT HL AT 1ST LINE
               06 CC      LD B, 204D       B= SCRN LINE LOOP CTR
                                           FILL 204 LINES TOTAL
               50         LD D, B          SAVE LOOP CTR IN D
PAG3A          3E AA      LD A, AA         A= PIXELS CLR 10101010
               CB 40      BIT 0, B         IF LOOP CTR IS ODD (BIT 0=1),
               20 01      JR NZ, PAG3B     USE CLR AA AND JUMP FORWARD 1 BYTE  (BIT 1=0)
               AF         XOR A            OTHERWISE, CTR IS EVEN
                                           USE PIXELS CLR ZERO
       2A42H   06 50      LD B, 80D        B= BYTES/LINE = 80D
PAG3B  PAG3C   77         LD (HL), A       FILL RAM BYTE WITH THE COLOR
               23         INC HL           POINT TO NEXT RAM BYTE
               10 FC      DJNZ PAG3C       LOOP BACK TO WRITE NEXT BYTE
                                           IN THE LINE? IF SO, JUMP
                                           BACK 4 BYTES.
               42         LD B, D          PUT LOOP CTR BACK IN B
       2A4AH   10 EF      DJNZ PAG3A
       2A4CH   C3 61 2A   JP PAG4          JMP TO WRITE PAGE 4
```

FILL PAGE 3 WITH NARROW HORIZONTAL STRIPES

P.45

WRITE (FILL) VERTICAL BAR - COMMON PORTION (FOR PAGE 4) 45

VBAR	$2A4F_H$	62	LD H,D	] HL = INITIAL FILL ADR
	$2A50_H$	6B ← BOY	LD L,E	
MFILL1		0609	LD B,9	B = LOOP CTR = XSIZE = 9 BYTES WIDE
MFILL2		77	LD (HL),A	FILL A BYTE
		23	INC HL	POINT TO NEXT BYTE ADR
		10 FC	DJNZ MFILL2	FINISHED WITH A LINE?
		09	ADD HL,BC	POINT TO NEXT LINE ADR
		08	EX AF,AF'	A = YSIZE, A' = FILL DATA
		47	LD B,A	PUT YSIZE IN B
		3D ← DOG	DEC A	] DECREMENT YSIZE
		08	EX AF,AF'	] AND SAVE IT
		05	DEC B	DECREMENT YSIZE AGAIN FOR LOOP
		20 F2	JR NZ, MFILL1	
		DD E9	JP (IX)	JMP TO NEXT VERTICAL BAR INITIALIZATION

WRITE PAGE 4 10 COLOR TEXTURED TEST PATTERN

PAG4	$2A61_H$	3E 40	LD A, 0100 0000	] 280 TO WRITE TO PAGE 4
		D375	OUT (75_H),A	
		AF	XOR A	] CLEAR SCREEN
		21 0040	LD HL, 4000	204 LINES x 80 BYTES/LINE
	$2A69_H$	01 C0 3F	LD BC, $3ED0_H$	TOTAL 16,320 BYTES = $3FC0_H$
PAG4A	$2A6C_H$	77	LD (HL),A	
		ED A1	CPI	
		EA 6C 2A	JP PE, PAG4A	

WRITE BORDERS

	$2A72_H$	11 03 40	LD DE, 4003	DE = START ADR
		3E C9	LD A, CP_H	] A' = YSIZE
		08	EX AF,AF'	
		3E 50	LD A, 50_H	A = 80 BYTES/LINE
		D6 4A	SUB $4A_H$	XSIZE = $74_D = 4A_H$
		4F	LD C, A	C = # OF BYTES TO SKIP FORWARD
	$2A7D_H$	3E 11	LD A,	FILL WITH 0001 0001 (TEXTURED)
		62 ← BOY	LD H,D	
	$2A80_H$	6B	LD L,E	
MBLAN1		06 4A	LD B, 4A	B = LOOP CTR = XSIZE
MBLAN2		77	LD (HL),A	
		23	INC HL	
	$2A85_H$	10 FC	DJNZ MBLAN2	

WRITE UNDERLYING BORDERS AROUND 8 BARS WITH TEXTURED BORDER

```
2A87H 09        ADD HL,BC
      08        EX AF,AF'   A=YSIZE, A'=FILL DATA
      47 ~DOG   LD B,A
      3D        DEC A
      08        EX AF,AF'   A=FILL DATA, A'=YSIZE DEC
      05        DEC B
      20 F2     JR NZ,MBLAN1
      DD 21 A3 2A  LD IX,PAG4B   SETUP "CONTINUE" JUMP FOR VERTICAL BAR 2
                                 NO SUBROUTINES ALLOWED.
                                 NO STACK AREA OR Z80 READS FROM RAM
```

WRITE 8 VERTICAL BARS (ON TOP OF BORDER AREA)

WRITE ONLY, HI-RES TEST PATTERN

```
2A93H 11 44 41  LD DE,START ADR
      3E C1     LD A,C1H      A'=YSIZE=193D LINES
      08        EX AF,AF'
      3E 50     LD A 50H      X SIZE=9 BYTES/LINE
      D6 09     SUB 9
      4F        LD C,A
      3E 00     LD A,00H      FILL WITH BLK
2AA0H C3 4F 2A  JP WVBAR      JMP TO WRITE(FILL) VERTICAL BAR
```

WRITE BAR 1 (LEFT-MOST BAR) FILL WITH

```
PAG4B 2AA3H 7B  LD A,E
      C6 09     ADD A,9
      5F        LD E,A
      DD 21 B8 2A  LDIX,PAG4C   IX=CONTINUE PGM ADR
      3E C1
      08
      3E 50
2AB0H D6 09
      4F
      3E 33     LD A,33       FILL BAR WITH 00 11 BLK GRN
      C3 4F 2A  JP WVBAR
```

POINT DE AT NEXT BAR'S START ADR

WRITE BAR 2 FILL WITH 00 11 BLK GRN

```
PAG4C 2AB8H 7B
      C6 09
      5F
      DD 21 CD 2A  LD IX,PAG4D
2AC0H 3E C1
      08
      3E 50
      D6 09
      4F
      3E AA     LD A,AAH      FILL BAR WITH YELLOW
2AC4H C3 4F 2A  JP WVBAR
```

WRITE BAR 3 FILL WITH SOLID YEL

```
PAG4D   2ACD_H  7B
                C6 09
        2AD0_H  5F
                DD 21 E2 2A     LD IX, PAG4E
                3E C1
                08
                3E 50
                D6 09
                4F
                3E EE           LD A, EE_H  FILL BAR WITH GRN YEL
                C3 4F 2A        JP WVBAR
PAG4E   2AE2_H  7B
                C6 09
                5F
                DD 21 F7 2A     LD IX, PAG4F
                3E C1
                08
                3E 50
                D6 09
        2AF1_H  4F
                3E 33           LD A, 33_H  FILL BAR WITH BLU GRN
                C3 4F 2A        JP WVBAR
PAG4F   2AF7_H  7B
                C6 09
                5F
                DD 21 0C 2B     LD IX, PAG4H
                3E C1
        2B01_H  08
                3E 50
                D6 09
                4F
                3E 00           LD A, 00_H  FILL BAR WITH BLUE
                C3 4F 2A        JP WVBAR
PAG4H   2B0C_H  7B
                C6 09
                5F
        2B10_H  DD 21 21 2B     LD IX, PAG4I
                3E C1
                08
                3E 50
                D6 09
                4F
                3E AA           LD A, 00_H  FILL BAR WITH RED
                C3 4F 2A        JP WVBAR
        2B1E_H
```

WRITE BAR4 FILL WITH 11 10 GRN YEL

WRITE BAR5 FILL WITH 00 11 BLU GRN

WRITE BAR6 FILL WITH SOLID BLUE

WRITE BAR7 FILL WITH SOLID RED

```
PAG4I  2B21H  7B
              C609
              5F
              DD21362B   LDIX, PAG5   IX = CONTINUE PROGRAM ADR
              3EC1
              08
              3E50
              D609
       2B30H  4F
       2B31H  3EEE       LD A, EEH   FILL BAR WITH 11 10 (GRN YEL)
       2B33H  C34F2A     JP WVBAR
```

WRITE BAR & FILL WITH 11 10 GRN YEL

WRITE PAGE 5 NARROW VERTICAL + HORIZONTAL STRIPES

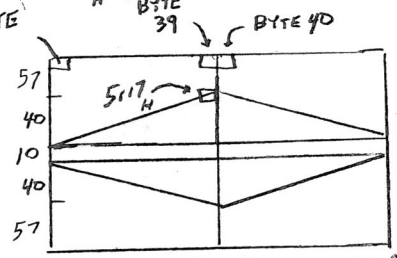
```
PAG5   2B36H  3E50       LD A, 01010000   } Z80 TO WRITE TO PAGE 5
              D375       OUT (75H), A     }
       2B3AH  3E11
              210040
              01603F
PAG5A  2B42H  77
              ED A1      BC
       2B45H  EA422B     JP PE, PAG5A
```

FILL PAGE 5 WITH NARROW VERTICAL STRIPES (SAME AS PAGE 1 SEE P.44)

OVERLAY PAGE 5 WITH NARROW HORIZONTAL STRIPES

BEGIN FILL ADR 5117_H
FILL ADR NEXT LINE $-1+80=79$
BYTES/LINE INCREASE 2, 4, 6, 8...
FILL 40 LINES = 28_H

BYTE 39 + 80(57)
$= 4599 = 11F7_H$
SCREEN ADR = 51F7



FILL TOP TRIANGE

```
PAG4   2B48H  3E02       LD A, BPLINE   A = BYTES PER LINE TO FILL INITIALLY
              08         EX AF, AF'     SAVE BPLINE IN A'
              21F751     LD HL, 51F7H   POINT HL AT 1ST LINE
              0628       LD B, 40D      B = "SCREEN LINES TO FILL" LOOP CTR
PAG4A  2B50H  48         LD C, B        SAVE LOOP CTR IN C
              3EAA       LD A, AAH      } LOOP CTR ODD, FILL WITH AAH
              CB40       BIT 0, B       }   EVEN ↓ 00H
              2001       JR NZ, PAG4B   }
              AF         XOR A          }
PAG4B         08         EX AF, AF'     A = BPLINE  A' = FILL DATA
              5D  DOG    LD E, L        } SAVE "1ST ADR IN LINE TO FILL" IN DE
       2B5AH  54         LD D, H        }
```

```
2B5B_H  47        LD B,A      B="BPLINE TO FILL" LOOP CTR
        08        EX AF,AF'   A=FILL DATA, A'=BPLINE
PAG4C   77        LD (HL),A   FILL SCREEN ADR WITH COLOR BYTE
        23        INC HL      POINT TO NEXT ADR TO FILL
        10 FC     DJNZ PAG4C  LOOP BACK TO FILL NEXT SCREEN BYTE
2B61_H  79        LD A,C      SAVE LINE LOOPCTR IN A
        0E 4F     LD C,79_D   C=79_D, B=0. C=INCREMENT FOR NEXT LINE
        EB        EX DE,HL    HL=1ST ADR ON LAST LINE
                              DE=LAST
        09        ADD HL,BC   POINT HL AT 1ST ADR ON NEXT LINE
        47        LD B,A      PUT LINE LOOP CTR BACK IN B
        08        EX AF,AF'   A=BPLINE AGAIN
        C6 02     ADD A,2     A=NEW (WIDDENED) BPLINE TO FILL
        08        EX AF,AF'   A'=BPLINE
        10 E3     DJNZ PAG4A

ADD SPACE BETWEEN TRIANGLE FILLS

2B6D_H  01 21 03  LD BC,801_D   800_D = 0321_H
2B70_H  09        ADD HL,BC

FILL BOTTOM TRIANGLE

2B71_H  3E 50     LD A,BPLINE
        08        EX AF,AF'
        06 28     LD B,40_D
PAG4D   48        LD C,B                    SIMILAR
        3E AA     LD A,AA                     TO
        CB 40     BIT 0,B                   FILL
        20 01     JRNZ,PAG4E                 T.P
        AF        XOR A                   TRIANGLE
PAG4E   08        EX AF,AF'
        5D        LD E,L
2B80_H  54        LD D,H
        47        LD B,A
        08        EX AF,AF'
PAG4F   77        LD (HL),A
        23        INC HL
        10 FC     DJNZ PAG4F
        79        LD A,C
        0E 51     LDC,80+1
        EB        EX DE,HL
2B8B_H  09        ADD HL,BC
```

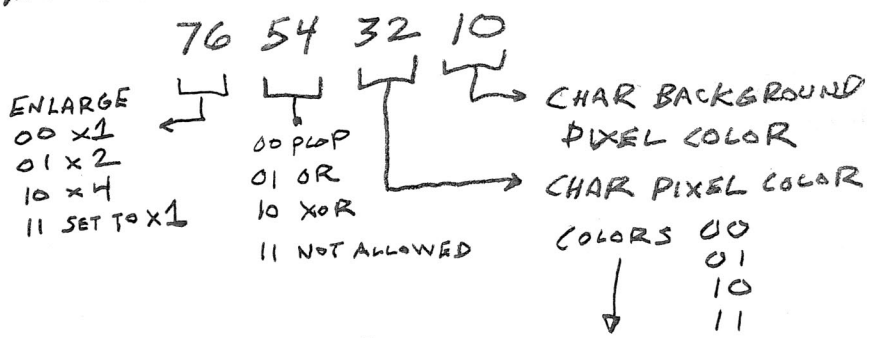
2B8C	47	LD B,A
	08	EX AF,AF'
	D602	SUB 2
2B90H	08	EX AF,AF'
	10E3	DJNZ PAG4D
2B93H	C3012D	JP PAG6

JUMP TO WRITE PAGE 6
(ROUTINE IS ON P. 61)

PAGE 6 MULTIPAGER TITLE PAGE

PAGE 6 DATA BASES

7FC0H CHAR DISPLAY OPTIONS FOR EXPANDED WRITE



7FC1, 7FC2H — INITIAL MAGIC ADDRESS OF CHAR LINE TO WRITE

7FC3, 7FC4H — CHAR X COORDINATE (UPPER LEFT PIXEL)

7FC7H — REGY = CHAR Y COORDINATE (UPPER LEFT PIXEL)

SEE DIAGRAM ON P. 25

WRITE ONE LINE OF TEXT WITH EXPANDED ENLARGEMENT

ENTER WITH: HL = CHAR PATTERN ADDRESS (TO WRITE)
(7FC6$_H$) = CHAR PATTERN OPTIONS
(7FC1 / 7FC2$_H$) = INITIAL MAGIC ADDRESS OF CHR STRING

x1, x2 OR x4

(7FF7$_H$) = REGY = Y COORDINATE

```
WRTLIN 2B96H 0E 01      LD C,1          CHAR PATTERN WIDTH = 1 BYTE
             06 00      LD B,0          B=0
             DD 21 00 00 LD IX,0        ]
             DD 39      ADD IX,SP       ] POINT DE AT
             DD E5      PUSH IX         ] STACK TOP
2BA0H        D1         POP DE          ]
             3E 0C      LD A,0CH        ] SET UP EXPAND REG FOR PATTERN
             D3 19      OUT (XPAND),A   ] EXPANSION IN STACK (FOR ENLARGEMENT)
                                        ] ON CHAR = 11  OFF BKGND = 00
             3E 08      LD A,08H        ] EXPAND WITH PLOP IN STACK
             D3 0C      OUT(MAGIC),A    ]
2BABH        3A C0 7F   LD A,(7FC0H)    ] PUT CHAR DISPLAY OPTIONS IN A.
             E6 C0      AND C0H         ] ISOLATE ENLARGEMENT BITS 7,6
```

```
             28 21      JR Z,WRTL3      SKIP ENLARGEMENT PROCESSING IF
                                        BITS 7,6 = ZERO, MULT FACTOR x1.
```

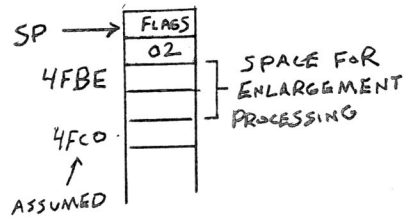
BEGIN PROCESSING PATTERN BITS FOR ENLARGEMENT.
EXPAND PATTERN BYTES FIRST WITHIN STACK AREA.
THEN, USE STACK PATTERN BYTES TO EXPAND WITH ENLARGEMENT
THE ENTIRE HORIZONTAL LINE.

```
2BBZH 07        RLCA    ]
      07        RLCA    ] SHIFT ENLARGEMENT BITS 7,6 INTO
                        ] BITS 1,0 RESPECTIVELY
                        ] A = 0000 0010
                                  └→ ENLARGEMENT FACTOR x4
WRTL1 2BB4H EB  EX DE,HL        DE = ADDRESS OF CHAR PATTERN
                                HL = POINTS TO TOP OF STACK.
```

```
2BB5H  A7       AND A          CLEAR CARRY
       ED 42    SBC HL,BC
       ED 42    SBC HL,BC
       F9       LD SP,HL
```

COMPUTE HOW MANY BYTES MUST BE SAVED IN STACK FOR THE PATTERN LINE EXPANSION WITH ENLARGEMENT PROCESSING. POINT SP AT THE FIRST OF THOSE SAVED BYTES. SP=HL
IN THIS CASE, BC=01 ← 1 CHAR PATTERN BYTE WIDE

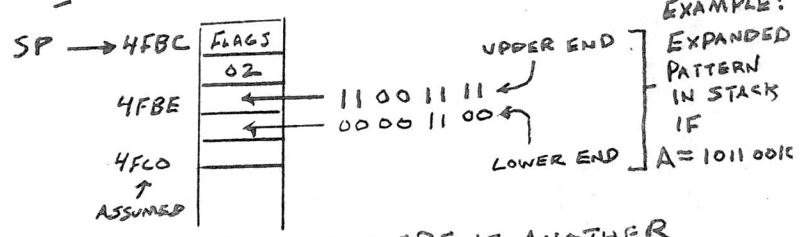
```
       CB B4    RES 6,H        HL=MAGIC ADR
                               =4FBE (ASSUMED)
       F5       PUSH AF        SAVE CTR1, ENLARGEMT
                               FACTOR
```




```
       41       LD B,C         B= COUNTER 2= X CHAR PATTERN WIDTH (SIZE)
                               =
```

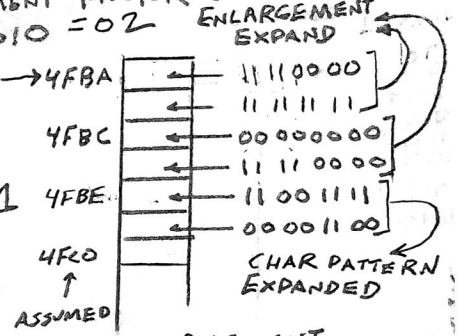
EXPAND CHAR PATTERN LINE WITHIN STACK
(IN THIS CASE, CHAR LINE PATTERN IS ONLY 1 BYTE)

```
WRTL2  2BBFH 1A       LD A,(DE)      DE POINTS TO GRAPHICS PATTERN TO EXPAND
                                     PUT IT IN A   EXAMPLE=10110010
       2BC0H 13       INC DE         POINT DE AT NEXT PATTERN
              77       LD (HL),A      HL POINTS AT MAGIC ADR TO EXPAND WITHIN
              23       INC HL         STACK.
              77       LD (HL),A      EXPAND REG (19H) WAS PRESET TO
              23       INC HL           0000 1100
                                        ↑ OFF EXPAND CLR 00
                                        ON EXPAND CLR 11
```



-8
000 1000
1111 0111 +1
1111 1000

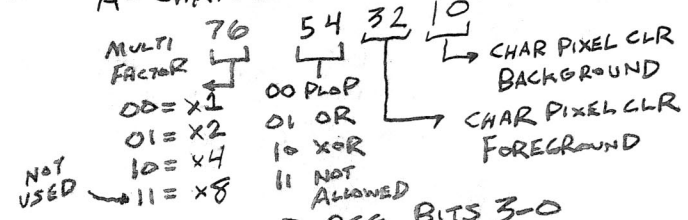
```
       10 F8    DJNZ WRTL2     LOOP BACK IF THERE IS ANOTHER
                               PATTERN IN LINE TO EXPAND
       CB 21    SLA C          SHIFT LEFT X PATTERN SIZE C=0000 0010
                               (=ENLARGED) PATTERN CTR
       F1       POP AF         A= COUNTER 1
                               = ENLARGEMENT FACTOR BITS REQUEST ENLARGEMENT EXPAND
                               = 0000 0010 =02
       21 00 00 LD HL,0        POINT DE AT TOP OF STACK AGAIN
       39       ADD HL,SP
       54       LD D,H
       5D ← DOG LD E,L
       3D       DEC A          DECREMENT CTR1
2BD0H  20 E1    JRNZ,WRTL1     LOOP BACK TO CONTINUE ENLARGEMENT PROCESSING?
2BD1H                          DE=HL=SP
                               DE POINTS TO NEW CHAR PATTERN (IN STACK) TO BE ENLARGED WITHIN STACK
```



ENLARGEMENT EXAMPLE SHOWN ABOVE FOR X4 ENLARGEMENT USING INITIAL CHAR PATTERN EXAMPLE A=1011 0010

EXPAND ENLARGED LINE WITHIN STACK

WRTL3 2BD3H	CD 2E 2C	CALL AMULF	B = MULT FACTOR 1, 2 OR 4 ONLY (X 8 IS NOT ALLOWED)
	ED5B C1 7F	LD DE, (7FC1H)	DE = MAGIC ADDRESS
	3A C0 7F	LD A, (7FC0H)	A = CHAR DISPLAY OPTIONS



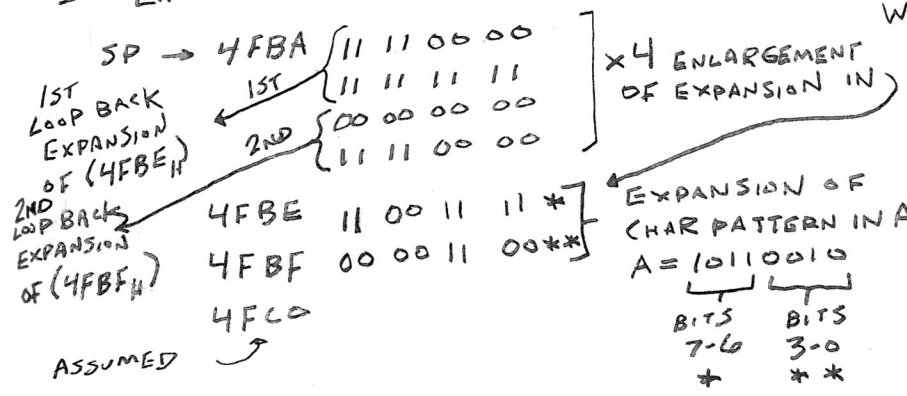
	D3 19	OUT (XPAND), A	SET EXPAND REG BITS 3-0
	E6 30	AND 30H	ISOLATE BITS 5, 4 FOR MAG REG VALUE
	F6 08	OR 08H	TURN ON EXPAND BIT FOR MAGIC REG VALUE
2BE1H	D3 0C	OUT A, (MAGIC)	
	EB	EX DE, HL	HL = MAGIC ADDRESS DE = ENLARGED CHAR PATTERN ADR IN STACK (MULT FACTOR 2, 4 OR 8 OR = CHAR LINE PATTERN ADR (MULT FACTOR 1)
WRTL4	F5	PUSH AF	SAVE A = MAGIC REG VALUE
	C5	PUSH BC	SAVE C = CTR FOR ENLARGED PATTERN IN STACK B = MULT FACTOR 1, 2, 4 OR 8
	D5	PUSH DE	SAVE DE = ADDRESS OF 1ST EXPANDED PATTERN IN STACK (MULT FACTOR 2, 4 OR 8 OR CHAR PATTERN (MULT x 1)
	E5	PUSH HL	SAVE HL = WRITE TO MAGIC ADDRESS
	41	LD B, C	B = ENLARGED PATTERN CTR
WRTL5	1A	LD A, (DE)	A = CHAR PATTERN TO EXPAND IF ENLARGED, DE POINTS TO ENLARGED CHR PATTERN IN STACK.
	13	INC DE	POINT DE AT NEXT PATTERN TO EXPAND
	77	LD (HL), A	WRITE TO MAGIC RAM EXPANDING BITS 7-4 IN CHAR PATTERN
	23	INC HL	POINT HL AT NEXT MAGIC ADDRESS
	77	LD (HL), A	WRITE TO MAGIC RAM EXPANDED BITS 3-0 IN CHAR PATTERN
2BF0H	23	INC HL	HL = NEXT MAGIC ADDRESS
2BF1H	10 F8	DJNZ WRTL5	LOOP BACK TO FINISH THE WRITING OF THE ENLARGED EXPANDED LINE

DELETE WRITE OF CLEAR SHIFTE BYTE
INITIAL X COORDINATE WILL HAVE BITS 1 AND 0 = ZERO

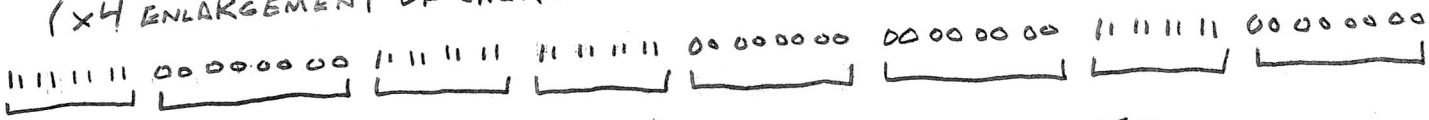
WRTL6	2BF3H	E1	POP HL	HL = INITIAL MAGIC ADR FOR LINE
		0E 50	LD C, 50H	HL = MAGIC ADR OF NEXT LINE (B=0 FROM LAST DJNZ INSTR)
		09	ADD HL, BC	
		D1	POP DE	DE = ADDRESS OF 1ST EXPANDED PATTERN IN STACK (MULT FACTOR 2 OR 4 OR CHAR PATTERN (MULT x1)
		C1	POP BC	C = CTR FOR ENLARGED PATTERN IN STACK B = MULT FACTOR 1, 2 OR 4
		F1	POP AF	A = MAGIC REGISTER
	2BFAH	10 EA	DJNZ WRTL4	LOOP BACK TO DUPLICATE (ENLARGE) LINES (MULT FACTOR x2, 4 OR 8)

-22
168421
00101110
00101001
1110 1010 +1

LINE ENLARGEMENT FOR PATTERN EXAMPLE A = 10110010 (SEE PAGE B) WITH x4 ENLARGEMENT.



FULL PIXEL LINE ENLARGEMENT OF 4FBA – 4FBDH
(x4 ENLARGEMENT OF CHAR PATTERN A = 1011 0010



DUPLICATE WRITES FOR 3 MORE LINES
TO DISPLAY x4 ENLARGEMENT OF CHAR PATTERN LINE A = 1011 0010

2BFCH	DD F9	LD SP, IX	RESTORE STACK POINTER SAME AS ENTRY OF WRTLN (SEE BEGINNING OF WIRTLN)
2BFE	C9	RET	
	UNUSED	BYTE	
2BFFH	FF		

CONVERT COORDINATES TO MAGIC ADDRESS

ENTER WITH: HL = PATTERN ADDRESS (SAVED IN THIS SUB)
DE = X COORDINATE
A = MR VALUE TO OUTPUT TO MAGIC REGISTER (PORT 08_H)
($7FF7_H$) = REGY = Y COORD

NORMALLY, BC = YSIZE XSIZE
HL AND BC ARE NOT CLOBBERED

EXIT WITH: DE = MAGIC ADDRESS CONVERSION
ADJUSTED MR VALUE OUTPUTED TO MAGIC REGISTER IN THIS SUBROUTINE.

```
RELTA1  2C00H  E5          PUSH HL          SAVE PATTERN ADDRESS
                           PASS BITS 7 CUSTOM FLOP, 6 FLOP, 5-4 XOR OR FLOP, 3 EXPAND
        2C01H  E6 F8       AND F8H          CLEAR SHIFT AMOUNT IN MR VALUE (BITS 1 AND 0)
               6F          LD L,A           SAVE MR IN L
               7B          LD A,E           PUT X LOW ORDER IN A
               E6 03       AND 03H          ISOLATE SHIFT BITS 1 AND 0
               B5          OR L             OR A WITH MAGIC FUNCTIONS
                                            ISOLATE SHIFT AMOUT
                                            COMBINE IT WITH MR VALUE AT ENTRY.
        2C08H  F5          PUSH AF          SAVE MR VALUE
               CB 77       BIT 6,A          IF FLOP BIT IS SET,
               C2 4D 2F    JP NZ,FREQ       JUMP TO PROCESS IT
                                            (DOG)
                                            (p.69)

RELTAC  2C0EH  D5          PUSH DE          SAVE X COORD
               3A F7 7F    LD A,(REGY)      HL=Y
        2C12H  6F          LD L,A
               26 00       LD H,0
               29          ADD HL,HL        HL=COMPUTED MAGIC ADR
               29
               29
        2C18H  29
               54          LD D,H
               5D          LD E,L
               29          ADD HL,HL
               29          ADD HL,HL
               19          ADD HL,DE
               D1          POP DE
               CB 1A       RR D
        2C21H  CB 1B       RR E
               CB 3B       SRL H
               16 00       LD E,A
               19          ADD HL,DE
        2C28H  EB          EX DE,HL         DE=MAGIC ADR
               F1          POP AF           A=MR VALUE
               E1          POP HL           HL=PATTERN ADR
               D3 0C       OUT (MAGIC),A    OUT MAGIC REQUEST
```

MAGIC OUTPUT HERE

ADJUST THE ENLARGEMENT BITS

```
AMULF 2C2E_H 3A C0 7F    LD A,(7FC0_H)   A= CHAR DISPLAY OPTIONS
      2C31_H 07          RLCA  ]
             07          RLCA  ] SHIFT THE ENLARGEMENT BITS IN 7-6
             E6 03       AND 03          INTO BITS 1-0
                                         ISOLATE THE ENLARGEMENT BITS
                                         00 x1
                                         01 x2
                                         10 x4
                                         11 x8 (WILL NOT USE, EXIT WITH x1
```

```
             FE 03       CP 3
             20 04       JR NZ, AMULF1   ] IF x8 ENLARGEMENT,
             3E 01       LD A,1          ]   SET IT TO x1
             18 06       JR AMULF2       ] ADJUST, NOW
AMULF1       3C          INC A           ]   01 x1
             FE 03       CP 3            ]   10 x2
      2C40_H 20 01       JR NZ, AMULF2   ] IF A= 0000 0011,
             3E          INC A           ]   SET A= 04
AMULF2       47          LD B,A          B= MULT FACTOR x1, x2 OR x4
      2C44_H C9          RET
```

PAGE 6 CHARACTER STRINGS

MULTIPLE

SLN0 2C45_H 4D 55 4C 54 49 50 4C 45 00

PAGE

SLN1 2C4E_H 50 41 47 45 00

DEMONSTRATION

SLN2 2C53_H 44 45 4D 4F 4E 53 54 52 41 54 49 4F 4E 00

MCM DESIGN

SLN3 2C61_H 4D 43 4D 40 44 45 53 49 47 4E 40

2020

2C6C_H 32 30 32 30 00

↳ 2C70_H

WRITE CHARACTER

ENTER WITH: ($7FF7_H$) = REGY = CHAR Y COORDINATE (UPPER LEFT PIXEL)

($7FC0_H$) = CHAR DISPLAY OPTIONS

76 54 32 10

- Bits 7 6: ENLARGE
  - 00 ×1
  - 01 ×2
  - 10 ×4
  - 11 ×1 (SET TO)
- Bits 5 4:
  - 00 PLOP
  - 01 OR
  - 10 XOR
  - 11 NOT ALLOWED
- Bits 3 2: CHR COLOR
- Bits 1 0: CHAR BACKGND COLOR

DE = CHAR X COORDINATE (UPPER LEFT PIXEL)

HL = CHAR PATTERN ADDRESS

Label	Address	Code	Instruction	Comment
WCHAR	$2C71_H$	3E 08	LD A, 08	A = PRESET MAG REG VALUE (FOR USE WITH SUB RELTA1
		CD 00 2C	CALL RELTA1	CONVERT X, Y COORDINATES TO CORRESPONDING MAGIC ADDRESS IN DE.
		ED 53 C1 7F	LD ($7FC1_H$), DE	SAVE THIS MAGIC ADDRESS AT $7FC1_H$
		06 09	LD B, 9	B = CHAR LINE CTR (CHAR HAS 9 LINES)
WCHAR1		C5	PUSH BC	SAVE LINE CTR
		E5	PUSH HL	SAVE CHAR PATTERN ADR
		CD 96 2B	CALL WRTLIN	WRITE CHAR LINE THIS SUB EXITS WITH: HL = POINTS TO MAGIC ADR FOR NEXT LINE TO WRITE
	$2C81_H$	EB	EX DE, HL	DE = MAGIC ADR OF NEXT PATTERN LINE TO WRITE
		ED 53 C1 7F	LD ($7FC1_H$), DE	SAVE THAT MAGIC ADR FOR NEXT LINE WRITE
		E1	POP HL	HL = CHAR PATTERN ADR
		23	INC HL	POINT HL AT NEXT CHAR PATTERN LINE
		C1	POP BC	B = CHAR PAT LINE CTR
-15		10 F1	DJNZ WCHAR1	
	$2C8B_H$	C9	RET	

WRITE CHAR STRING

ENTER WITH: $(7FC0_H)$ = CHAR DISPLAY OPTIONS FOR EXPAND WRITE

76 54 32 10

- 76: ENLARGE — 00 ×1, 01 ×2, 10 ×4, 11 —
- 54: 00 PLOP, 01 OR, 10 XOR, 11 —
- 32: CHR COLOR
- 10: CHR BACKGROUND COLOR

$(7FC1, 7FC2_H)$ = CHAR X COORDINATE (UPPER LEFT PIXEL)

$(7FC7_H)$ = REGY = CHAR Y COORD (UPPER LEFT PIXEL)

HL = CHAR STRING ADDRESS

FETCH CHAR CODE FROM STRING

```
WSTR  2C8CH  7E        LD A,(HL)      A = ASCII CHAR CODE
             FE 00     CP 00H         } RETURN IF CODE IS
             C8        RET Z          } STRING TERMINATOR 00H
      2C90H  E5        PUSH HL        SAVE HL = CHAR CODE ADDRESS
```

POINT TO CHAR PATTERN IN CHAR TABLE

```
PTCHR        D6 30     SUB 30H        SUBTRACT 30H FROM ASCII CODE
             67        LD H,A         } IF ASCII CODE IS 30H,
             6F        LD L,A         } THEN HL = 0000H
             A7        AND A          } IF ASCII CODE IS 30H,
             28 0A     JR Z, PTCHR2   } INDEX ADJUSTMENT IS NOT REQ'D
             47        LD B,A         B = CHAR INDEX
             21 00 00  LD HL,0000     } ADJUST CHAR INDEX TO
             11 09 00  LD DE,0009     } POINT AT CHAR PATTERN
PTCHR1       19        ADD HL,DE      }
      2CA0H  10 FD     DJNZ PTCHR1    }
PTCHR2       01 6725   LD BC,CHART    BC = CHAR TABLE ADR
             09        ADD HL,BC      POINT HL AT CHAR PATTERN
                                      (1 BYTE WIDE x 9 BYTES HIGH CHR PATTERN)
```

HL POINTS TO CHAR PATTERN NOW

WRITE CHAR ON SCREEN

```
        (DOG)  (BOY)
             ED5B C37F  LD DE,(7FC1H)  DE = CHAR X COORDINATE
      2CAAH  CD 71 2C   CALL WCHR      WRITE THE CHAR
```

PAGE 6 MULTIPAGER TITLE PAGE

POINT TO NEXT STRING CHAR CODE

```
2CAD_H  E1      POP HL          POINT HL AT CHAR CODE
        23      INC HL          POINT HL TO NEXT CHAR CODE
        E5      PUSH HL         SAVE CHAR CODE POINTER
```

POINT TO NEXT CHAR FRAME IN LINE

```
PFRM 2CB0_H  3A C0 7F   LD A,(7FC0_H)   A= CHAR DISPLAY OPTIONS
             07         RLCA            SHIFT ENLARGEMENT BITS 7-6
             07         RLCA            INTO BITS 1-0.
             E6 03      AND 03          ISOLATE BITS 1-0
             3C         INC A           NOW A= 01
                                               10
                                               11
                                               100
             CB 97      RES 2,A         KILL x8 BIT
             A7         AND A           x8 REQUEST
             20 02      JR NZ, PFRM1    NOT ALLOWED.
             3E 01      LD A,1          DEFAULT TO x1.
PFRM1        FE 03      CP 3            ADJUST x4
        2CC1_H  20 01   JR NZ, PFRM2    TO A=4
             3C         INC A           (11 BECOME 100)
PFRM2        47         LD B,A          B= CTR 1,2 OR 4
             AF         XOR AF
PFRM3        C6 08      ADD A,8
             10 FC      DJNZ PFRM3
```

ENLARGE		ADD
01	x1	8
10	x2	16
100	x4	32

A= X COORD INCREMENT

-4
0000 0100
1111 1011 +1
1111 1100

```
             ED 5B C3 7F   LD DE,(7FC1_H)   DE= LAST CHAR X COORD
             EB            EX DE,HL         HL=  (NEXT)
             4F            LD C,A           C= X COORD INCREMENT FOR FRAME
                                            B= 0 FROM DJNZ
        2CD0_H  09         ADD HL,BC        HL= NEXT FRAME X COORD
             EB            EX DE,HL         DE=
             ED 53 C3 7F   LD (7FC1_H),DE   SAVE DE= UPDATED FRAME ADR
             E1            POP HL           HL= NEXT CHAR CODE POINTER
-11     2CD7_H  18 B3      JR WSTR          (BOT)
```

PAGE 6 COLOR TABLE

```
CLRTS 2CD9_H  86   YELLOW        PIXEL 11
              73   LT ORANGE          10
              FC   LT BLUE           01
        2CDC_H  00   BLACK           00
```

SLICE x4 TEXT STRING

ENTER WITH: DE = STRING X COORDINATE
A = STRING Y COORDINATE + 1 (Y COORD FOR 1ST SLICE, 1 LINE BELOW INITIAL LINE)
D = BYTES WIDE IN STRING LINE

SLICE 2	32 F7 7F	LD (7FF7$_H$), A	REG Y = STRING Y COORD + 1
2CE0$_H$	AF	XOR A	A = MR VALUE = SLICE CLEAR BYTE = 0
	D5	PUSH DE	SAVE BYTES WIDE
	CD 002C	CALL RELTA1	CONVERT X, Y COORDINATES TO DE = MAGIC ADDRESS
	EB	EX DE, HL	NOW, HL = MAGIC ADR
	D1	POP DE	D = BYTES WIDE AGAIN
	1E 09	LD E, 9$_D$	E = SLICE CTR = 9 DOUBLE SLICES TOTAL
SLICE 1	E5	PUSH HL	SAVE INITIAL LINE MAGIC ADR
	42	LD B, D	B = BYTES/LINE IN STRING LINE
SLICE 2	77	LD (HL), A	SLICE (CLEAR) BYTE IN LINE
	23	INC HL	POINT HL AT NEXT MAGIC ADR
	10 FC	DJNZ, SLICE 2	LOOP BACK TO FINISH SLICING LINE
	E1	POP HL	HL = INITIAL LINE MAGIC ADR
2CF0$_H$	0E 50	LD C, 80$_D$	C = BYTES/SCREEN LINE
			B = 0 FROM DJNZ
	09	ADD HL, BC	POINT HL AT LINE IMMEDIATELY BELOW SLICED LINE
	E5	PUSH HL	SAVE NEW LINE INITIAL ADR
	42	LD B, D	B = BYTES/LINE IN STRING LINE
SLICE 3	77	LD (HL), A	SLICE LINE IMMEDIATELY BELOW 1ST SLICE (DOUBLE SLICE ENLARGED PIXEL)
	23	INC HL	
	1D FC	DJNZ SLICE 3	
	E1	POP HL	HL = INITIAL MAGIC ADR OF LINE SLICED
	0E F0	LD C, 240$_D$	C = NEXT PIXEL SLICE Y INCREMENT
	09	ADD HL, BC	HL = MAGIC ADR OF NEXT LINE TO SLICE
	1D ← DOG	DEC E	DECREMENT SLICE CTR
	20 E9	JR NZ, 1	LOOP BACK TO SLICE STRING AGAIN
2D00$_H$	C9	RET	

-23
168421
0001 0111
1110 1000 +1
1110 1001 ↑ DOG

WRITE PAGE 6
DEMO TITLE SCREEN

(PAGE WRITES CONTINUATION, SEE PAGE 50)

DOG ↓

PAG 6 $2D01_H$	3E66	LD A, 0110 0110	Z80 WRITE TO AND READ FROM PAGE 6
	D375	OUT (75H, A)	
	31C07F	LD SP, $7FC0_H$	INITIALIZE STACK POINTER
	01703F	LD BC, $3F70_H$	CLEAR SCREEN 203 LINES 203×80 = 16,240 BYTES = $3F70_H$
	CD8B29	CALL CSCRN1	
	3E1E	LD A, YCOORD	INITIALIZE Y COORDINATE FOR 1ST TEXT STRING "MULTIPLE"
$2D10_H$	32F77F	LD ($7FF7_H$), A	
	112000	LD DE, XCOORD	INITIALIZE X COORDINATE FOR 1ST TEXT STRING "MULTIPLE"
	ED53C37F	LD ($7FC3_H$), DE	
	3E84	LD A, 1000 0109	SET UP CHAR DISPLAY OPTIONS: ENLARGE ×4, PLOP, BACKGROUND COLOR 00, CHAR COL 01
	32C07F	LD ($7FC0_H$), A	
	21452C	LD HL, SLNO	POINT HL AT STRING
$2D22_H$	CD8C2C	CALL WSTR	WRITE THE TEXT STRING

PAGE 6 MULTIPAGER TITLE PAGE

```
2D25H 3E 60      LD A,YCOORD          ] SET YCOORD
      32 F7 7F   LD (7FF7H),A         ]
      11 60 00   LD DE,XCOORD         ] SET XCOORD
      ED 53 C3 7F LD (7FC3H),DE       ]
2D31H 3E 84      LD A,1000 0100       ] SET CHAR DISPLAY OPTIONS
      32 C0 7F   LD (7FC0H),A         ]
      21 4E 2C   LD HL,SLN1           POINT HL AT STRING
      CD 8C 2C   CALL WSTR            WRITE STRING
                                                          ] WRITE "PAGE"

      3E A2      LD A,YCOORD          ] SET YCOORD
      32 F7 7F   LD (7FF7H),A         ]
2D41H 11 38 00   LD DE,XCOORD         ] SET XCOORD
      ED 53 C3 7F LD (7FC3H),DE       ]
                    ↑x2   ↑LT BLUE
      3E 48      LD A,0100 1000       ] SET CHAR DISPLAY OPTIONS
      32 C0 7F   LD (7FC0H),A         ]
      21 53 2C   LD HL,SLN2           POINT HL AT STRING
      CD 8C 2C   CALL WSTR            WRITE STRING
                                                          ] WRITE "DEMONSTRATION"

2D53H 3E BF      LD A,YCOORD          ] SET YCOORD
      32 F7 7F   LD (7FF7H),A         ]
      11 64 00   LD DE,XCOORD         ] SET XCOORD
      ED 53 C3 7F LD (7FC3H),DE       ]
      3E 0C      LD A,0000 1100       ] SET CHAR DISPLAY OPTIONS
2D61H 32 C0 7F   LD (7FC0H),A         ]
      21 61 2C   LD HL,SLN3           POINT HL AT STRING
      CD 8C 2C   CALL WSTR            WRITE STRING
                                                          ] WRITE "M&M DESIGN 2020"

      11 20 00   LD DE,0020H          DE=STRING X COORDINATE
      3E 1F      LD A,1FH             A=STRING Y COORD+1
      16 40      LD D,64D             D=BYTES WIDE
2D71H CD DD 2C   CALL SLICE           SLICE THE STRING
                                                          ] SLICE "MULTIPLE STRING"

      11 60 00   LD DE,0060
      3E 61      LD A,61H
      16 20      LD D,32D
2D7BH CD DD 2C   CALL SLICE
                                                          ] SLICE "PAGE STRING"
```

ERROR CORRECTION

SET UP PAGE 7 FOR Z80 WRITE AND READ

$2D7E_H$	3E 77	LD A, 0111 0111	Z80 WRITE TO AND READ FROM PAGE 6
	D3 75	OUT (75_H, A)	
$2D82_H$	C3 D3 30	JP TO PAG 7	JUMP TO WRITE PAGE 7

WAVE 1 @ $3E92_H$ IN FISH DEMO SEA BOTTOM

WAVE 1	$2D85_H$	42	0100 0010
	$2D86_H$	E7	1110 0111
	$2D87_H$	FF	UNUSED BYTE

(This page intentionally left blank.)

PAGE 7 HI-RES GUNFIGHT SCREEN SHOT

BULLET PATTERN

BULT $2D88_H$ 18 3C 3C 3C 3C 3C 3C 7E 00

WRITE PLAYER SCORE (ZERO)

ENTER WITH: A = CHAR DISPLAY OPTIONS

= 76 54 32 10

00 00 ×1 PLOP

10 → BACKGROUND COLOR

32 → CHAR COLOR

DE = X COORDINATE OF SCORE CHAR

WPLS $2D91_H$	32 C0 7F	LD ($7FC0_H$), A	($7FC0_H$) = CHAR DISPLAY OPTIONS
	3E 04	LD A, 4	REGY = Y COORD OF SCORE CHAR
	32 F7 7F	LD ($7FF7_H$), A	
	21 67 25	LD HL, 2567_H	HL = CHAR PATTERN (ZERO) ADR
$2D9C_H$	CD 71 2C	CALL WCHAR	WRITE THE "ZERO"
$2D9F_H$	C9	RET	

WRITE 6 BULLETS

ENTER WITH: DE = X COORDINATE OF 1ST BULLET

WBUL $2DA0_H$	06 06	LD B, 6	B = BULLET CTR
	21 88 2D	LD HL, $2D88_H$	HL = BULLET PATTERN ADR
WBUL1	E5	PUSH HL	SAVE PAT ADR
	C5	PUSH BC	SAVE BULLET CTR
	D5	PUSH DE	SAVE CHAR FRAME ADR
	CD 71 2C	CALL WCHAR	WRITE A BULLET
	D1	POP DE	DE = CHAR FRAME ADR
	EB	EX DE, HL	NOW HL = ↓
	01 08 00	LD BC, 8	POINT HL AT NEXT CHAR FRAME
$2DB0_H$	09	ADD HL, BC	
	EB	EX DE, HL	DE = NEXT BULLET FRAME X COORD
	C1	POP BC	B = BULLET CTR
	E1	POP HL	HL = BULLET PAT ADR
-17	10 EF	DJNZ WBUL1	LOOP BACK TO WRITE 5 MORE BULLETS
$2DB6_H$	C9	RET	

WRITE RELATIVE FROM VECTOR BLOCK (SIMILAR TO SUB#30) 64

ENTER WITH: IX = VECTOR BLOCK (PACKET) ADDRESS

HL = PATTERN ADDRESS -4 (POINTING AT RELATIVE X)

VECTOR PACKET IS 15 BYTES LONG. (X_H AND ΔX_H ARE 2 BYTES LONG)

VWRITR $2DB7_H$	DD 7E 00	LD A,(IX)	A = MAGIC REGISTER VALUE
	DD 46 0D	LD B,(IX+0D$_H$)	B = Y_H
	DD 5E 07	LD E,(IX+7)	DE = X_H
$2DC0_H$	DD 56 08	LD D,(IX+8)	
	DD CB 01 F6	SET 6,(IX+1)	SET BLANK BIT

WRITE RELATIVE (SIMILAR TO SUB#32)

ENTER WITH: HL = PATTERN ADDRESS -4 (POINTING AT RELATIVE X)

DE = X COORDINATE (X_H IN VECTOR PACKET)

B = Y COORDINATE (Y_H TO OUTPUT TO MAGIC REGISTER PORT 05$_H$)

A = MR VALUE

WRITR $2DC7_H$	F5	PUSH AF	SAVE MR VALUE
	7E	LD A,(HL)	A = RELATIVE X
	23	INC HL	POINT HL AT RELATIVE Y
	83	ADD A,E	
	5F	LD E,A	DE = X_{COORD} + RELATIVE X OR X_H
	7A	LD A,D	
	CE 00	ADC A,0	
	57	LD D,A	
$2DD0_H$	7E	LD A,(HL)	A = RELATIVE Y
	23	INC HL	POINT HL AT X SIZE
	80	ADD A,B	A = Y_{COORD} + RELATIVE Y OR Y_H
	32 F7 7F	LD (REGY),A	REGY = Y + RELATIVE Y
$2DD6_H$	F1	POP AF	A = MR VALUE

WRITE WITH PATTERN SIZE (SIMILAR TO SUB #34) 65

ENTER WITH: HL = PATTERN ADDRESS - 2
(POINTING AT XSIZE)

DE = X_{COORD} + RELATIVE X

($7FF7_H$) = REGY = Y_{COORD} + RELATIVE Y

A = MR VALUE TO OUTPUT TO MAGIC REG ($PORT08_H$)

WRITP $2DD7_H$	4E	LD C, (HL)	C = XSIZE
	23	INC HL	POINT HL AT Y SIZE
	46	LD B, (HL)	B = YSIZE
	23	INC HL	POINT HL AT PATTERN

WRITE WITH COORDINATE CONVERSION (SIMILAR TO SUB #36)

WRIT $2DDB_H$ CD 002C CALL RELTA1

WRITE PATTERN (SIMILAR TO SUB #38)

ENTER WITH: HL = PATTERN ADDRESS
DE = MAGIC ADDRESS TO WRITE TO
C = XSIZE OF PATTERN (# OF BYTES WIDE)
B = YSIZE ↓ (# OF LINES HIGH)

ALSO MR VALUE MUST BE IN Z80 REGISTER A AT ENTRY. TO FLAG (APPROPRIATE WRITE ROUTINE →

NOTE: MAGIC REGISTER VALUE MUST BE OUTPUT TO THE MAGIC REGISTER (PORT 08H) PRIOR TO CALLING THIS SUB.

VPATHR	$2DDE_H$ CB 77	
	$2DE0_H$ 20 2C	
	CB 5F	
	20 11	
WPLOP →	AF	
(NORMAL PLOP, XOR, OR)	C5	
	D5	
	47	
DOG ←	ED B0	
	12	
DOG →	D1	
	EB	
	0E 50	LD C, BYTEPL HI-RES 80D BYTES/LINE
$2DF1_H$	09	
	EB	
	C1	
	10 F1	
$2DF6_H$	C9	

REFERENCE NUTTING MANUAL
Z80/ROM CODE BREAKDOWN P. 49-50

```
WEXPD  2DF7H  EB
(WRITE        C5
EXPANDED)     E5
              41
              1A
              13
              77
              23
              77
       2E00H  23
              10 F8
              70
              23
              70
              E1
              0E 50
              09
              C1
              10 EB
              C9
              CB 5F
              20 16
       2E10H  AF
WFLOP         C5
(WRITE        D5
FLOPPED)      47          ← DOG
WFLOP1 2E16H  ED A0
              1B
              1B
              EA 16 2E
              12
              D1
              EB
       2E20H  0E 50
              09
              EB          ← BOT
              C1
              10 EC
       2E27H  C9
```

REFERENCE NUTTING MANUAL
Z80/ROM CODE BREAKDOWN P. 50

LD C, BYTEPL HI-RES 80_D BYTES/LINE

REFERENCE NUTTING MANUAL
Z80/ROM CODE BREAKDOWN P. 50-51

JP PE, WFLOP1

WXFLOP (WRITE WITH EXPANDED FLOP)

REFERENCE NUTTING MANUAL
Z80/ROM CODE BREAKDOWN P.51

```
2E28H EB
      C5
      E5
      41
      1A
      13
      77
      2B
2E30H 77
      2B  <- BOY
      10 F8
      70
      2B
      70
      E1
      0E 50        LD C, BYTEPL   HI-RES 80D BYTES/LINE
      09
      C1
      10 EB
2E3EH C9
```

COWBOY (LEFT SIDE)

```
2E3FH 06 XSIZE
2E40H 28 YSIZE
      00 01 50 15 00 00   ]
      01 05 55 55 50 00   ] - HAT, BLUE
      01 55 55 55 50 00   ]
2E53H 01 55 55 55 54 00   ]
      00 55 55 55 55 00   ]
      00 0A AA AA A1 40   ]
2E65H 00 0A AA AA A0 00   ]
      00 0A AA A0 A0 00   ]
2E71H 00 0A AA AA A8 00   ] - HEAD,
      00 0A AA AA A8 00   ]
      00 0A AA AA A0 00   ]
2E83H 00 02 AA AA 00 00   ]
      00 00 AA AA A0 00   ]
2E8FH 00 00 AA AA 80 00   ]
```

```
2E95H 00 00 55 55 00 00
      00 01 55 55 40 00
2EA1H 01 55 55 55 50 00
      05 55 55 55 54 00
      15 55 55 55 55 50
2EB3H 55 55 55 55 55 50
      55 05 55 55 55 50
      54 05 55 55 50 00
2EC5H 54 05 55 55 50 00
      54 05 55 55 50 00
2ED1H 54 05 55 55 50 00
      54 05 55 55 50 00
      54 3F FF FF FC 00
2EE3H 54 3F FF FF FC 00
      AA 05 55 55 50 00
      AA 05 55 55 50 00
2EF5H AA 15 55 55 54 00
      06 15 54 55 54 00
2F01H 00 55 54 55 54 00
      00 55 50 15 54 00
      01 55 40 15 54 00
2F13H 01 55 00 05 54 00
      05 54 00 05 54 00
      0F FF C0 0F FF C0
2F25H 3F FF F0 3F FF F0
2F2BH 3F FF F0 3F FF F0
```

UPPER BODY, BLUE (PIXEL 01)

BELT, GREEN (PIXEL 11)

LEGS, BLUE

BOOT BOTTOMS, GREEN

COWBOY FRONT ARM + GUN

```
2F31H 16 RELATIVE X    +22
      12 ↓       Y     +18
      04 X SIZE
      06 Y SIZE
2F35H 00 40 00 10
      00 15 55 54
      00 05 55 54
2F41H 55 59 90 00
      55 5A 40 00
2F49H 55 5A 00 00
```

↙ 2F4CH

PROCESS MR FLOP REQUEST FOR "CONVERT COORDINATES TO MAGIC ADDRESS", P.55

FREQ $2F4D_H$	CB7F	BIT 7,A (P.140)	IF BIT 7 IN MR VALUE = 1,
	C28B3E	JP NZ, CUFLOP	JUMP TO EXECUTE CUSTOM FLOP ROUTINE
$2F52_H$	7B	LD A,E	2'S COMPLEMENT X COORD (NEGATE IT)
	2F	CPL	
	5F	LD E,A	
	7A	LD A,D	
	2F	CPL	
	57	LD D,A	
	13	INC DE	
	213F01	LD HL, 319D	DE = XFLOP = 319 - X
	19	ADD HL, DE	
	EB	EX DE, HL	
$2F5E_H$	C30E2C	JP RELTAC	

319 − X = FLOPPED COORDINATE SYSTEM

2'S COMPLEMENT YOU COULD DELETE THIS INSTRUCTION AND USE LD HL, 320D

TREE

$2F61_H$	02	X SIZE	
	22	Y SIZE	
	01 80		0000 0001 1000 0000
	01 C0		0000 0001 1100 0000
	03 A0		0000 0011 1010 0000
DOG	07 90		0000 0111 1001 0000
	8D 88		1000 1101 1000 1000
	79 E4		0111 1001 1110 0100
	31 A3		0011 0001 1010 0011
	03 B0		0000 0011 1011 0000
$2F71_H$	07 9C		0000 0111 1001 1100
DOG	0D 80		0000 1101 1000 0000
	F9 80		1111 1001 1000 0000
	31 9F		0011 0001 1001 1111
	01 BC		0000 0001 1011 1100
	01 B0		0000 0001 1011 0000
	F1 E0		1111 0001 1110 0000
$2F81_H$	79 E0		0111 1001 1110 0000
	0D 00		0000 1101 1100 0000
	07 80		0000 0111 1000 0000
$2F87_H$	03 81		0000 0011 1000 0001

$2F89_H$	E1 C2	1110 0001 1100 0010
	71 E4	0111 0001 1110 0100
	19 BC	0001 1001 1011 1100
DOG	0D 98	0000 1101 1001 1000
$2F91_H$	07 80	0000 0111 1000 0000
	03 80	0000 0011 1000 0000
	79 87	0111 1001 1000 0111
	05 98	0000 0101 1001 1000
	03 A0	0000 0011 1010 0000
	01 C0	0000 0001 1100 0000
	01 80	0000 0001 1000 0000
	01 80	0000 0001 1000 0000
$2FA1_H$	03 C0	0000 0011 1100 0000
	0F F0	0000 1111 1111 0000
$2FA5_H$	3F FC	0011 1111 1111 1100

CACTUS

$2FA7_H$	02 XSIZE	
	18 YSIZE	
$2FA9_H$	04 00	0000 0100 0000 0000
	0E 00	0000 1110 0000 0000
	0E 40	0000 1110 0100 0000
	0E E0	0000 1110 1110 0000
$2FB1_H$	0E E0	0000 1110 1110 0000
	0F E0	0000 1111 1110 0000
	0F C0	0000 1111 1100 0000
	4F 84	0100 1111 1000 0100
	EE 0E	1110 1110 0000 1110
	EE 0E	1110 1110 0000 1110
	EE 0E	1110 1110 0000 1110
	EE 0E	1110 1110 0000 1110
$2FC1_H$	FE 1E	1111 1110 0001 1110
	7E 3C	0111 1110 0011 1100
	3F FC	0011 1111 1111 1100
	0F F8	0000 1111 1111 1000
	0F F0	0000 1111 1111 0000
	0F C0	0000 1111 1100 0000
	0F 00	0000 1111 0000 0000
$2FCF_H$	0F 00	0000 1111 0000 0000

$2FD1_H$	OF 00	0000 1111 0000 0000
	OF 00	0000 1111 0000 0000
	OF 00	0000 1111 0000 0000
$2FD7_H$	OF 00	0000 1111 0000 0000

WAGON CANOPY
(1 BYTE WIDE x 7 BYTES HIGH)
Block 1

$2FD9_H$	10	RELATIVE X (16_D)
	00	RELATIVE Y
$2FDB_H$	01	X SIZE
	07	Y SIZE
$2FDD_H$	FF FF FF FF FF FF FF	EXPAND BLUE (01) CANOPY ON BLACK (00) BACKGROUND

(1 BYTE WIDE x 16 BYTES HIGH)
Block 2

$2FE4_H$	08	RELATIVE X
	00	RELATIVE Y
$2FE6_H$	01	X SIZE
	10	Y SIZE
	07	0000 0111
	0F	0000 1111
	1F	0001 1111
	3F	0011 1111
	7F	0111 1111
	FF	1111 1111
	FF	1111 1111
	FE	1111 1110
$2FF0_H$	FC	1111 1100
	F8	1111 1000
	F0	1111 0000
	E0	1110 0000
	C0	1100 0000
	C0	1100 0000
	80	1000 0000
$2FF7_H$	80	1000 0000

Block 3 (1 byte wide x 10 lines high)

$2FF8_H$	00	Relative X
	06	Relative Y
$2FFA_H$	01	X size
	0A	Y size
	01	0000 0001
	03	0000 0011
	07	0000 0111
	0F	0000 1111
3000_H	0F	0000 1111
	1F	0001 1111
	3F	0011 1111
	3F	0011 1111
	7F	0111 1111
	7F	0111 1111

Block 4 (1 byte wide x 17 lines high)

3006_H	00	Relative X
	10	Relative Y (16_D)
3008_H	01	X size
	11	Y size (17_D)
	7F	0111 1111
	7F	0111 1111
	7F	0111 1111
	7F	0111 1111
	7F	0111 1111
	7E	0111 1110
3010_H	7E	0111 1110
	7E	0111 1110
	7E	0111 1110
	7E	0111 1110
	7C	0111 1100
	7C	0111 1100
	7C	0111 1100
	7C	0111 1100
	7C	0111 1100
	7C	0111 1100
$301A_H$	7E	0111 1110

WAGON CARGO AREA (LEFT HALF)

$301B_H$ 06 X SIZE
18 Y SIZE (24_D)

3F FF FF FF FF FF
3023_H 3F FF FF FF FF FF
3F FF FF FF FF FF
0F C0 00 00 00 00
3035_H 0F CF FF FF FF FF
0F CF FF FF FF FF
3041_H 0F CF FF FF FF FF
0F CF FF FF FF FF
0F CF FF FF FF FF
3053_H 0F C0 00 00 00 00
0F CF FF FF FF FF
0F CF FF FF FF FF
3065_H 0F CF FF FF FF FF
0F CF FF FF FF FF
3071_H 0F CF FF FF FF FF
0F CF FF FF FF FF
0F C0 00 00 00 00
3083_H 0F FF FF FF FF FF
FF FF FF FF FF FF
$308F_H$ 55 55 55 55 55 55
3095_H 55 55 55 55 55 55
55 55 55 55 55 55
$30A1_H$ 55 55 55 55 55 55
$30A7_H$ 55 55 55 55 55 55

WHEEL HUB (LEFT SIDE)

$30AD_H$ 01 X SIZE
06 Y SIZE ← $30B4_H$
$30AF_H$ 03 0F 3F 3F 0F 03

WAGON WHEEL

$30B5_H$ 01 X SIZE
1C Y SIZE (28_D)

$30B7_H$ FF FF FF FF FF FF FF
FF FF FF FF FF FF FF
$30C5_H$ FF FF FF FF FF FF FF
$30CC_H$ FF FF FF FF FF FF FF ↘ $30D2_H$

WRITE PAGE 7 HI-RES GUNFIGHT SCREEN SHOT

(NOTE: ERROR CORRECTION ON PAGE 62A, $207E_H$)

```
PAG7 30D3H  31 C0 7F     LD SP, 7FC0H      INITIALIZE STACK POINTER
            CD 88 29     CALL CSCRN        CLEAR THE SCREEN
            3E FF        LD A, FFH         FILL TOP 18 LINES
            01 A0 05     LD BC, 05A0H      WITH GREEN (COLOR 11)
            CD 8C 29     CALL CSCRN2       80 x 18 = 1440 BYTES
                                             = 05A0H
```

WRITE PLAYER 1 SCORE (SINGLE DIGIT) X=16_D, Y=4

```
30E1H  3E 0B     LD A, 00 00 1011      BCK GND CLR 11
                                       CHAR CLR 10
                                       x1 PLOP
       11 10 00  LD DE, 0010           DE = XCOORD = 16D
       CD 91 2D  CALL WPLS             WRITE PLAYER SCORE
```

WRITE PLAYER 2 SCORE X=296_D=0128_H

```
       3E 0B     LD A, 0000 1011
       11 28 01  LD DE, 0128H
       CD 91 2D  CALL WPLS             WRITE PLAYER SCORE
```

WRITE PLAYER 1 BULLETS

```
30F1H  11 40 00  LD DE, 0040H          DE = 1ST BULLET XCOORD = 64D
       CD A0 2D  CALL WBUL             WRITE THE 6 BULLETS
```

WRITE PLAYER 2 BULLETS

```
       11 D0 00  LD DE, 00D0H          DE = 1ST BULLET XCOORD = 208D
                 (DOG)
30FAH  CD A0 2D  CALL WBUL             WRITE THE 6 BULLETS
```

WRITE LEFT COWBOY

30FD$_H$	21 3F 2E	LD HL, PAT-2	HL = PATTERN ADR-4
3100$_H$	11 08 00	LD DE, XCOORD	DE = XCOORD = 8
	3E 50	LD A, 80$_D$	(7FF7$_H$) = REGY = YCOORD
	32 F7 7F	LD (7FF7$_H$), A	
	AF	XOR A	A = 0000 0000 = MR VALUE
			→ FLOP WRITE
	CD D7 2D	CALL WRITP	WRITE COWBOY (LESS SHOOTING ARM)
	21 31 2F	LD HL, PAT-4	WRITE RELATIVE LEFT ARM + GUN
	11 08 00	LD DE, XCOORD	
3112$_H$	06 4D ← DOG	LD B, YCOORD	
	3E 10	LD A, 0001 0000	
		→ OR WRITE	
	CD C7 2D	CALL WRITR	

FLOP WRITE RIGHT COWBOY

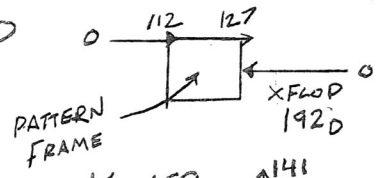
	21 3F 2E		
	11 08 00		XCOORD
	3E 50		YCOORD
3121$_H$	32 F7 7F		FLOP WRITE COWBOY (LESS SHOOTING ARM)
	3E 40	LD A, 0100 0000	
		→ FLOP WRITE	
		→ FLOP WRITE	
	CD D7 2D		
	21 31 2F		XCOORD
	11 08 00		Y ↓
	06 4D ← DOG		FLOP WRITE RELATIVE SHOOTING ARM + GUN
3131$_H$	3E 50	LD A, 0101 0000	
		→ OR WRITE	
		→ FLOP WRITE	
	CD C7 2D		

EXPAND WRITE TOP LEFT TREE

	3E 0C	LD A, 0000 1100	SETUP EXPAND REG GREEN TREE ON BLACK BACKGROUND
	D3 19	OUT (XPAND), A	
	21 61 2F	LD HL, PAT-2	POINT HL PATTERN ADR-2
	11 70 00	LD DE, XCOORD	DE = XCOORD = 112$_D$ = 70
3140$_H$	3E 1C	LD A, YCOORD	(7FF7$_H$) = REGY = YCOORD = 28$_D$ = 1C$_H$
	32 F7 7F	LD (7FF7$_H$), A	
	3E 08	LD A, 0000 1000	A = MR VALUE = EXPAND WRITE
		→ EXPAND WRITE	
3147$_H$	CD D7 2D	CALL WRITP	WRITE TREE

FLOP EXPAND WRITE BOTTOM LEFT TREE

$314A_H$	21 61 2F	LD HL, PAT-2	POINT HL AT PATTERN ADR-2
	11 C0 00	LD DE, XCOORD	XFLOP 192_D



XFLOP = 319 − X
= 319 − 127
= 192 = $C0_H$

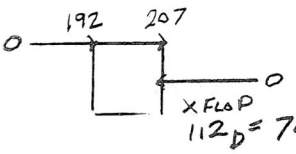
3150_H	3E 9F	LDA, YCOORD	$Y = 159_D$

141 TREE 2
175 } $1C_H = 28_D$ SPACING
203
ADJUSTMENT 141 + 18 GREEN TOP LINES
$= 159_D$
$(7FF7_H)$ = REGY = YCOORD

	32 F7 7F	LD ($7FF7_H$), A	
	3E 48	LDA, 01 00 1 00	A = MR VALUE
		FLOP, PLOP, EXPAND WRITE	
	CD D7 2D	CALL WRITP	WRITE TREE

FLOP EXPAND WRITE TOP RIGHT TREE

	21 61 2F	LD HL, PAT-2	POINT HL AT PATTERN ADR-2
	11 70 00	LD DE, XCOORD	XFLOP = $112_D = 70_H$



XFLOP = 319 − X
X = 319 − XFLOP
= 319 − 112
= 207

3160_H	3E 1A	LDA, YCOORD	RAISE UP 2 PIXELS $1C_H - 2 = 1A_H$
	32 F7 7F	LD ($7FF7_H$), A	$(7FF7_H)$ = REGY = YCOORD = $1A_H$
	3E 48	LD A, 01 00 1 000	A = MR VALUE
		FLOP, PLOP, EXPAND WRITE	
	CD D7 2D	CALL WRITP	WRITE TREE

EXPAND WRITE BOTTOM RIGHT TREE

	21 61 2F	LD HL, PAT-2	POINT HL AT PATTERN ADR-2
	11 C0 00	LD DE, XCOORD	XCOORD = $192_D = C0_H$
$317O_H$	3E 9D	LD A, YCOORD	$(7FF7_H)$ = REGY = YCOORD = $1A_H$
	32 F7 7F	LD ($7FF7_H$), A	
	3E 08	LD A, 0000 1000	A = MR VALUE
		PLOP, EXPAND WRITE	
3177_H	CD D7 2D	CALL WRITP	WRITE TREE

EXPAND WRITE LEFT CACTUS

317A$_H$	3E 08	LD A, 0000 1000	] SET UP EXPAND REG COLOR 10 ON BLACK BACKGROUND
	D3 19	OUT (XPAND), A	
	21 A7 2F	LD HL, PAT-2	POINT HL AT CACTUS PATTERN -2
3181$_H$	11 70 00	LD DE, XCOORD	DE = XCOORD = 112$_D$
	3E 62	LD A, YCOORD	] (7FF7$_H$) = REGY = YCOORD = 98$_D$ = 62$_H$
	32 F7 7F	LD (7FF7$_H$), A	
	3E 08	LDA, 0000 1000	A = MR VALUE
	CD D7 2D	CALL WRITP ↳	WRITE CACTUS

TREE 28$_D$ → 62$_D$ → CACTUS 24 LINES 98$_D$ → 122$_D$ → TREE 159$_D$

FLOP EXPAND WRITE RIGHT CACTUS

	21 A7 2F	LD HL, PAT-2	POINT HL AT CACTUS PAT-2
3191$_H$	11 71 00	LD DE, XCOORD	
	3E 60	LD A, YCOORD	] YCOORD = 62-2 RAISE CACTUS 2 PIXELS ↑
	32 F7 7F	LD (7FF7$_H$), A	
	3E 48	LD A, 01 00 1000	A = MR VALUE
		FLOP ↲ PLOP ↳ EXPAND	
	CD D7 2D	CALL WRITP	WRITE CACTUS

* * * * * * * WRITE LEFT SIDE OF WAGON CANOPY * * * * * * * *

	3E 04	LD A, 0000 0100	] EXPAND PATTERN WRITE WITH BLUE ON BLACK BACKGROUND
	D3 19	OUT (XPAND), A	
31A0$_H$	EXPAND WRITE	BLOCK 1	
	21 D9 2F	LD HL, PAT-4	POINT HL AT BLOCK 1 PATTERN-4
	11 87 00	LD DE, XCOORD	DE = XCOORD, WAGON CANOPY WRITE FRAME = 135$_D$
	06 4A	LD B, YCOORD	B = YCOORD, = 74$_D$
	3E 08	LD A, 0000 1000	A = MR VALUE
		PLOP ↳ EXPAND	
	CD C7 2D	CALL WRITR	WRITE RELATIVE BLOCK 1
	EXPAND WRITE	BLOCK 2	
	21 E4 2F	LD HL, PAT-4	POINT HL AT BLOCK 2 PATTERN-4
31B2$_H$	11 87 00	LD DE, XCOORD	XCOORD = 135$_D$
	06 4A	LD B, YCOORD	YCOORD = 74$_D$
	3E 18	LD A, 0001 1000	A = MR VALUE
		OR ↲ ↳ EXPAND	EXPAND OR WRITE
31B9$_H$	CD C7 2D	CALL WRITR	WRITE RELATIVE BLOCK 2

EXPAND WRITE BLOCK 3

31BC$_H$	21 F8 2F	LD HL, PAT-4	POINT HL AT BLOCK3 PATTERN-4
	11 87 00	LD DE, XCOORD	XCOORD = 135$_D$
	06 4A	LD B, YCOORD	YCOORD = 74$_D$
31C2$_H$	3E 18	LD A, 0001 1000	A = MR VALUE
		OR ↵ ↳ EXPAND	EXPAND OR WRITE
	CD C7 2D	CALL WRITR	WRITE RELATIVE BLOCK 3

EXPAND WRITE BLOCK 4

	21 06 30	LD HL, PAT-4	POINT HL AT BLOCK4 PATTERN-4
	11 87 00	LD DE, XCOORD	XCOORD = 135$_D$
	06 4A	LD B, YCOORD	YCOORD = 74$_D$
	3E 18	LD A, 0001 1000	A = MR VALUE, EXPAND OR WRITE
		OR ↵ ↳ EXPAND	
31D1$_H$	CD C7 2D	CALL WRITR	WRITE RELATIVE BLOCK 4

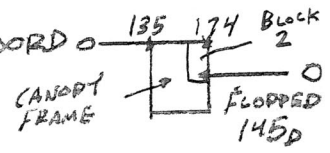
********** WRITE RIGHT SIDE OF WAGON CANOPY ******

OR WITH EXPAND WRITE BLOCK 1

21 DB 2F	LD HL, PAT-2	POINT HL AT BLOCK 1 XSIZE
11 9F 00	LD DE, X	X = XCOORD + RELATIVE X = 135$_D$ + 24$_D$ = 159$_D$ = 9F$_H$
3E 4A	LD A, 74$_D$	(7FF7$_H$) = REGY = YCOORD + RELATIVE Y
32 F7 7F	LD (7FF7$_H$), A	= 74$_D$ + 0 = 74$_D$ = 4A$_H$
31E1$_H$ 3E 18	LD A, 0001 1000	A = MR VALUE
	OR ↵ ↳ EXPAND	
CD D7 2D	CALL WRITP	WRITE BLOCK 1

OR WITH FLOPPED EXPAND BLOCK 2

21 E6 2F	LD HL, PAT-2	POINT HL AT BLOCK 2 XSIZE
11 91 00	LD DE, XCOORD	XFLOP = 145$_D$

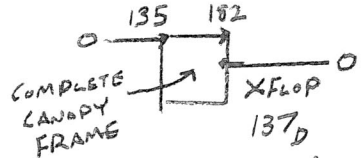


XFLOP = 319 − X = 319 − 174 = 145$_D$ = 91$_H$

3E 4A	LD A, 74$_D$	Y = YCOORD + RELATIVE Y = 74$_D$ + 0 = 74$_D$
32 F7 7F	LD (7FF7$_H$), A	(7FF7$_H$) = REG = Y = 4A$_H$
31F1$_H$ 3E 58	LD A, 0101 1000	A = MR VALUE
	FLOP ↵ ↳ OR ↳ EXPAND	
31F3$_H$ CD D7 2D	CALL WRITP	FLOP WRITE BLOCK 2

OR WITH FLOPPED EXPAND BLOCK 3

$31F6_H$	21 FA 2F	LD HL, PAT-2	POINT HL AT BLOCK 3 XSIZE
	11 89 00	LD DE, XCOORD	



XFLOP = 319 − X
= 319 − 182
$= 137 = 89_H$

	3E 50	LD A, 80_D	Y = YCOORD + RELATIVE Y = $74_D + 6$ = $80 = 50_H$ → $(7FF7_H)$ = REG Y = Y
	32 F7 7F	LD $(7FF7_H)$, A	
$320I_H$	3E 58	LD A, 0101 1000	A = MR VALUE
	CD D7 2D	CALL WRITP	FLOP WRITE BLOCK 2

(LD A: FLOP ↵, EXPAND OR WRITE)

OR WITH FLOPPED EXPAND BLOCK 4

	21 08 30	LD HL, PAT-2	POINT HL AT BLOCK 4 XSIZE
	11 89 00	LD DE, XCOORD	X = COORD = $137 = 89_H$ SEE ABOVE ILLUSTRATION
	3E 5A	LD A, 90_D	Y = YCOORD + RELATIVE Y = $74_D + 16_D$ = $90_D = 5A_H$ → $(7FF7_H)$ = REG Y = 90_D
	32 F7 7F	LD $(7FF7_H)$, A	
3211_H	3E 58	LD A, 0101 1000	A = MR VALUE
	CD D7 2D	CALL WRITP	FLOP WRITE BLOCK 3

(LD A: FLOP ↵, EXPAND OR WRITE)

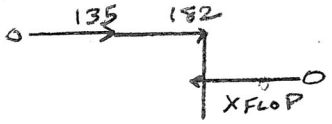
WRITE CARGO AREA (LEFT HALF)

	21 1B 30	LD HL, PAT-2	POINT HL AT CARGO WAGON XSIZE
	11 87 00	LD DE, XCOORD	XCOORD = $135_D = 87_H$
	3E 6B	LD A, 107_D	Y = YCOORD + RELATIVE Y = $74_D + 33_D = 107_D = 6B_H$ → $(7FF7_H)$ = REG Y = 107_D
	32 F7 7F	LD $(7FF7_H)$, A	
3221_H	3E 10	LD A, 0001 0000	A = MR VALUE
	CD D7 2D	CALL WRITP	WRITE THE CARGO AREA

(LD A: OR ↵)

FLOP WRITE CARGO AREA (RIGHT HALF)

	21 1B 30	LD HL, PAT-2	POINT HL AT CARGO WAGON XSIZE
	11 89 00	LD DE, XFLOP	



XFLOP = 319 − 182
$= 137_D = 89_H$

	3E 6B	LD A, 107_D	A = Y, SAME AS LEFT CARGO AREA → $(7FF7_H)$ = REG Y = 137_D
	32 F7 7F	LD $(7FF7_H)$, A	
3231_H	3E 50	LD A, 0101 0000	A = MR VALUE (FLOP OR WRITE)
3233_H	CD D7 2D	CALL WRITP	FLOP WRITE CARGO AREA

WRITE LEFT WHEEL

3236_H	21 B5 30	LD HL, PAT-2	POINT HL AT WAGON WHEEL XSIZE
	11 83 00	LD DE, XCOORD	XCOORD = $131_D = 83_H$
	3E 72	LD A, YCOORD	($7FF7_H$) = REGY = YCOORD
	32 F7 7F	LD ($7FF7_H$), A	$= 114_D = 72_H$
3241_H	3E 10	LD A, 0001 0000	A = MR VALVE
	CD D72D	CALL WRITP (OR)	OR WRITE LEFT WHEEL

WRITE RIGHT WHEEL (WRITE SIMILAR TO LEFT WHEEL)

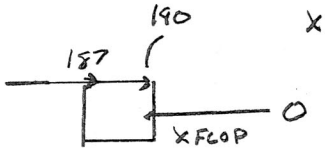
	21 B5 30	LD HL, PAT-2	
	11 B7 00	LD DE, XCOORD	XCOORD = $183_D = B7_H$
	3E 72	LD A, YCOORD	
	32 F7 7F	LD ($7FF7_H$), A	
3251_H	3E 10	LD A, 0001 0000	
	CD D72D	CALL WRITP	OR WRITE RIGHT WHEEL

WRITE LEFT WHEEL HUB

	21 AD 30	LD HL, PAT-2	POINT HL AT LEFT WHEEL HUB XSIZE
	11 7F 00	LD DE, XCOORD	XCOORD = $127_D = 7F_H$
	3E 7D	LD A, YCOORD	($7FF7_H$ = REGY = YCOORD
	32 F7 7F	LD ($7FF7_H$), A	$= 125_D = 7D_H$
3261_H	3E 10	LD A, 0001 0000	
	CD D72D	CALL WRITP (OR)	OR WRITE

FLOP WRITE RIGHT WHEEL HUB

	21 AD 30	LD HL, PAT-2	
	11 81 00	LD DE, XCOORD	XFLOP = 129_D
	3E 7D	LD A, YCOORD	Y=COORD = SAME AS LEFT WHEEL
	32 F7 7F	LD ($7FF7_H$), A	
3271_H	3E 50	LD A, 0101 0000	A = MR VALVE
		FLOP ↵ ↳ OR WRITE	
	CD D72D	CALL WRITP	OR WRITE RIGHT WHEEL HUB



XFLOP = 319 − X
= 319 − 190
$= 129_D = 81_H$

JUMP TO VIEW/FLIP PAGES

3276_H	C3 0034	JP FPAG	

↵ P.90

PAGE 3 COLOR TABLE

CLRT7 3279_H

09	DK BLUE	PIXEL 11	LEFT COLORS
FB	BLUE	10	
A3	NOT USED	01	
00	BLK	00	
85	RED	PIXEL 11	RIGHT COLORS
87	NOT USED	10	
FC	LT BLUE	01	
00	BLK	00	

FLIP PAGE SOUND

```
FSND  3281H  3E 9F     LD A, 1001 1111      SET VOLUME FOR WHITE NOISE GENERATION
                                  └→ VOLUME WHT NOISE
              D3 17     OUT (17H), A
              3E A1     LD A, A1H            SET UP MASTER OSC 5
              D3 10     OUT (10H), A
              3E 3F     LD A, 00 11 1111     VOICE C MAX VOLUME
                                  └→ VOICE C MAX VOL
                        BIT 5=1              MIX NOISE WITH VOICE C
                        MIX NOISE WITH VOICE C OUTPUT
                        BIT 4=1              TURN ON NOISE
                        TURN ON NOISE
              D3 15     OUT 15H
              3E 70     LD A, 70H            PLAY NOTE VOICE C
              D3 13     OUT (13H), A
              3E 40     LD A, 40H            EXECUTE SOUND DURATION DELAY
      3291H
FSND1         06 FF     LD B, 255D
FSND2         10 FE     DJNZ, FSND2
              3D        DEC A
              20 F9     JR NZ, FSND1
              AF        XOR A                TURN OFF MIX NOISE BIT 5 VOICE C
              D3 15     OUT (15H), A
      329DH   C9        RET
```

SELECT TIME DELAY

```
SLTD  329EH   ED 57     LD A, I              IF I=0 EXECUTE NEAR 1/4 SEC DELAY
      32 A0H  A7        AND A
              28 03     JR Z, SLTD1
              C3 F4 29  JP DELAY             ← USE THIS DELAY FOR I=10, 7, 4 AND 1
SLTD1         3E 80     LD A, 128D           CTR A=128D
SLTD2         06 FF     LD B, 255D           CTR B=255D
SLTD3         10 FE     DJNZ SLTD3           LOOP BACK 255 TIMES
              3D        DEC A                DEC CTR A
              20 F9     JR NZ, SLTD2         LOOP BACK 128 TIMES
      32AFH   C9        RET
```

MOVE WITH PLOP CRITTER THROUGH 8 PAGES

CRITTER PATTERN (16 PIXELS WIDE X 18 PIXEL LINES HIGH)
IS AT
$200B_H$ POINTING AT RELATIVE X (SEE PAGE 1)

INITIAL CRITTER VECTOR PACKET

$32B0_H$ 00 MR (PLOP WRITE)
00 VECTOR STATUS BIT 5 1 = NO MOTION OCCURED
0 = MOTION OCCURED
← USE FOR 1ST WRITE

$32B2_H$ 03 TIME BASE
50 ΔX_L
00] ΔX_H (2 BYTES)
00]
00 X_L
00] X_H (2 BYTES)
00]

$32B9_H$ 03 X CHECKS MASK — BIT 0 1 = CHECK FOR X LIMIT ATTAINED
1 1 = REVERSE ΔX AT X LIMIT
3 1 = LIMIT ATTAINED
0 = LIMIT NOT ATTAINED
40 ΔY_L
00 ΔY_H
00 Y_L
$32BD_H$ 00 Y_H
$32BE_H$ 03 Y CHECKS MASK — BIT 0 1 = CHECK FOR Y LIMIT ATTAINED
1 1 = REVERSE ΔY AT Y LIMIT
3 1 = LIMIT ATTAINED
0 = LIMIT NOT ATTAINED

CRITTER LIMITS TABLE

CLT $32BF_H$ 0000 X LOWER LIMIT
$32C1_H$ 2F01 X UPPER ← $319-16=303_D=012F_H$
00 Y LOWER
B8 Y UPPER

RANGED (SIMILAR TO LOW-RES SUB #118) +1 (A-1 = MAX # TO GENERATE)
ENTER WITH: A = MAXIMUM NUMBER TO GENERATE

RANGE	$32C5_H$	F5	PUSH AF
		2A EF 7F	LD HL, $7FEF_H$
		CD F0 32	CALL SHIFTR
		01 17 00	LD BC 23_D
		09	ADD HL, BC
	$32D0_H$	8A	ADC A, D

$32D1_H$	22 EF 7F	LD ($7FEF_H$), HL
	2A F1 7F	LD HL, $7FF1_H$
	5F	LD E, A
	CD F0 32	CALL SHIFTR
	19	ADD HL, DE
	22 F1 7F	LD ($7FF1_H$), HL
	5A	LD E, D
	EB	EX DE, HL
$32E1_H$	F1	POP AF
	A7	AND A
	4F	LD C, A
	7A	LD A, D
	28 08	JR, R3
	AF	XOR A
	19	ADD HL, DE
R1	30 01	JR NC, R2
	3C	INC A
R2	0D ← DOG	DEC C
	20 F9	JR NZ, R1
R3	C9	RET
		LD B, H
SHIFTR $32F0_H$	44	LD C, L
	4D	XOR A
	AF	LD D, 7
SH1	16 07	ADD HL, HL
	29	RLA
	17	DEC D
	15	JR NZ, SH1
	20 FB	ADD HL, BC
	09	ADC A, D
	8A	RET
$32FC_H$	C9	

CONVERT COORDINATES TO MAGIC ADDRESS

USE RELTA1 @ $2C00_H$ (SEE P. 55)

UPDATE X AND Y COORDINATES IN VECTOR PACKET

ENTER WITH: IX = VECTOR PACKET ADDRESS

HL = LIMITS TABLE ADDRESS
(POINTING TO LOWER X LIMIT)

INITIALIZE VECTOR PACKET IN RAM TO SUIT PROGRAM APPLICATION

NOTES

(1) THIS SUB IS SIMILAR TO ON-BOARD LOW-RES SUB #62 (REVISED FOR HI-RES)

(2) VECTOR STATUS (IX+1)
BIT 7, ACTIVE BIT IS NOT SET
BIT 5 1 = NO MOTION OCCURED
0 = MOTION OCCURED (X_H OR Y_H CHANGED)

(3) TIME BASE (IX+2) THE TIME BASE IS NOT ZEROED (INITIALIZE MOVE SPEED)

(4) X OR Y CHECKS MASK (IX+9) OR (IX+0E)
BIT 0 1 = DO LIMITS CHECK
0 = NO
BIT 1 1 = REVERSE DELTA AT LIMIT
0 = NO REVERSE DELTA AT LIMIT
BIT 3 1 = X (OR Y) LIMIT WAS ATTAINED
0 = LIMIT NOT ATTAINED

NOTE
THIS CUSTOM HI-RES SUBROUTINE AUTOMATICALLY CHECKS FOR A LIMIT AND REVERSES DELTA WHEN A LIMIT IS REACHED

(5) AN UPI IS NOT UTILIZED WITH THIS SUBROUTINE. THERE IS NO PASSING OF DATA OR SETTING BITS WITHIN A CONTEXT BLOCK.

```
MVECT 32FD_H  DD 4E 02      LD C, (IX+2)        C = TIME BASE
              DD CB 01 EE   SET 5, (IX+01)      USE BIT 5 IN VECTOR STATUS AS MOTION BIT
                                                1 = NO MOTION
                                                0 = MOTION OCCURED (X_H OR Y_H CHANGED)
      DOGS -> 11 03 00      LD DE, 3            ] POINT IX AT ΔX_L
              DD 19         ADD IX, DE          ]
```

UPDATE X COORDINATE

```
VECTX         E5            PUSH HL             SAVE LIMITS TABLE POINTER
              DD 56 02      LD D, (IX+2)        ] DE = ΔX_H
              DD 5E 01      LD E, (IX+1)        ]
3310_H        DD 66 05      LD H, (IX+5)        ] HL = OLD X_H
              DD 6E 04      LD L, (IX+4)        ]
3316_H        E5            PUSH HL             SAVE OLD X_H
```

```
3317H  41          LD B,C      B = TIME BASE = LOOP CTR
       DD 4E 00    LD C,(IX)   IX → ΔXL, C = ΔXL
       DD 7E 03    LD A,(IX+3) A = XL
VCTX1  81          ADD A,C     A = XL + ΔXL   (RESULT)
       ED 5A       ADC HL,DE   HL = OLD XH + ΔXH + CARRY
                                    H        DE
3321H  10 FB       DJNZ VCTX1
       DD 77 03    LD (IX+3),A LOAD UPDATED (NEW) XL INTO PACKET NOW
```

ADD DELTA TO COORDINATE "TIME BASE TIMES"

OOPS! IF NO MOTION OCCURED, THERE IS NO NEED TO JUMP TO VTX1A. COULD HAVE JUMPED RIGHT TO VECTY TO UPDATE Y COORDINATE

```
       C1          POP BC      BC = OLD XH
       5D ← DOG    LD E,L
       54          LD D,H      DE = SAVE UPDATED (NEW) XH
       A7 ← DOG    AND A       CARRY = 0
       ED 42       SBC HL,BC   HL = NEW XH − OLD XH − CARRY
                                    HL      BC
       28 04 →     JRZ,VTX1A
       DD CB FE AE RES 5,(IX-2) RESET BIT 5 IN VECTOR STATUS
                                0 = MOTION OCCURED
VTX1A  3332H  E1   POP HL      POINT HL AT X LOWER LIMIT
```

IF MOTION OCCURED (XH HAS CHANGED), RESET BIT 5 IN VECTOR STATUS

GET X LOWER LIMIT

```
       4E          LD C,(HL)
       23          INC HL
       46          LD B,(HL)   BC = X LOWER LIMIT
       23          INC HL      POINT HL AT X UPPER LIMIT
       C5          PUSH BC     SAVE LOWER LIMIT
```

CHECK IF X REACHED LOWER LIMIT

```
       01 6F 01    LD BC,367D  ← -48 0→319 +48 (SIMILAR TO LOW-RES VECTOR ROUTINE)
       CD DE 33    CALL LCHK
       C1          POP BC      BC = LOWER LIMIT
       30 0E       JR NC,VCTX2
3341H  CD DE 33    CALL LCHK
3344H  38 09       JRC,VCTX2
```

HANDLE < 0 CASE. ANY X > 367D IS CONSIDERED NEGATIVE

IF NEW XH < LOWER LIMIT JMP TO VCTX2

```
                GET X UPPER LIMIT
3346H 4E        LD C,(HL)
      23        INC HL
      46        LD B,(HL)
                CHECK X UPPER LIMIT                  ] IF NEW XH < UPPER LIMIT
      CD DE33   CALL LCHK                            ] JMP TO VCTX3
      38 2E     JR C,VCTX3
      2B ← BOY  DEC HL
VCTX2 23        INC HL
3350H 23        INC HL                               HL POINTS AT Y LOWER LIMIT
      DD 71 04  LD (IX+4),C                          ] XH IN VECTOR PACKET
      DD 70 05  LD (IX+5),B                          ] = X LIMIT
      DD 36 03 00  LD (IX+3),0                       XL IN VECTOR PACKET = 0
      DD CB ← BOY 06 DE  SET 3,(IX+6)                SET X LIMIT ATTAINED BIT
                                                     (IN VP'S X CHECKS MASK)
                PROCEED TO REVERSE THE DELTA (2'S COMPLEMENT)
335FH E5        PUSH HL                              SAVE LIMIT POINTER
                                                     (POINTS AT Y LOWER LIMIT)
3360H DDE5      PUSH IX                              ] POINT DE AT ΔXL
DOG →  D1       POP DE                               ]
      1A        LD A,(DE)   A=ΔXL                    ] 2'S COMPLEMENT ΔXL
      2F        CPL                                  ]
      C6 01     ADD A,1                              ]
      12        LD (DE),A                            ]
      13        INC DE                               ] HL = COMPLEMENT ΔXH
      1A        LD A,(DE)                            ]
      2F        CPL                                  ]
      6F        LD L,A                               ]
      13        INC DE                               ]
      1A        LD A,(DE)                            ]
      2F        CPL                                  ]
336FH 67        LD H,A                               ]
```

```
3370H 01 00 00      LD BC,0
      ED 4A         ADC HL,BC     HL = HL+BC+CARRY
      EB  ← BOY     EX DE,HL      DE = ΔXH + CARRY
                                  HL = POINTS TO HIGH ORDER XH
UPDATE
X ONLY,
INSERT
OPTIONAL
RETURN
HERE  72            LD (HL),D     ┐
      2B            DEC HL        ├ LOAD 2'S COMPLEMENT OF XH INTO VECTOR PACKET
      73            LD (HL),E     ┘
      E1            POP HL        HL POINTS AT Y LOWER LIMIT
  └→ 337AH 18 0B    JR VECTY      JUMP TO VECTY
```

NEW X_H WHERE: LOWER LIMIT < NEW X_H < UPPER LIMIT

LOAD NEW X_H IN VECTOR PACKET

```
                    INC HL        POINT HL NOW AT Y LOWER LIMIT
VCTX3  23  ← 2 DOGS
       DD 73 04     LD (IX+4),E   ┐ SET XH IN VECTOR PACKET
                    LD (IX+5),D   ┘ TO UPDATED (NEW) XH
UPDATE
X ONLY,
INSERT
OPTIONAL
RETURN  3380H DD 72 05
       DD CB 06 9E  RES 3,(IX+6)  CLEAR X LIMIT ATTAINED BIT
ALSO HERE →                       (IN VP X CHECKS MASK)
```

UPDATE Y COORDINATE

ENTER WITH: IX POINTING TO ΔX_L IN VECTOR PACKET
HL = LOWER Y LIMIT (IN LIMITS TABLE)

```
VECTY 3387H DD 4E FF    LD C,(IX-1)   C = TIME BASE
      11 07 00          LD DE,7       ┐ POINT IX AT ΔYL
      DD 19             ADD IX,DE     ┘
2 DOGS → E5             PUSH HL       SAVE Y LOWER LIMIT POINTER
      3390H DD 56 01    LD D,(IX+1)   ┐ DE = ΔY (ΔYH, ΔYL)
      DD 5E 00          LD E,(IX+0)   ┘
      DD 66 03          LD H,(IX+3)   ┐ HL = YH YL
      DD 6E 02          LD L,(IX+2)   ┘
      7C                LD A,H        A = YH, SAVE YH
      41                LD B,C        B = TIME BASE CTR
VECT1 19                ADD HL,DE     ┐ ADD DELTA TO COORDINATE
      10 FD ← DOG       DJNZ VECT1    ┘ "TIME BASE TIMES"
      33A1H ← BOY  BC   CP H          ┐ IF NO CHANGE IN Y,
      28 04             JR Z,VCT1A    ┘ JMP TO VCT1A
      33A4H DD CB F7 AE RES 5,(IX-9)  MOTION OCCURED
                                      RESET MOTION BITS IN VECTOR STATUS
```

```
                GET Y LOWER LIMIT
VCT1A 33A8H 7C      LD A,H          A=YH
            E3      EX (SP),HL      TOP OF STACK = Y COORDINATE
                                    HL = Y LIMITS POINTER
            46      LD B,(HL)         B = Y LOWER LIMIT
            23      INC HL          POINT HL AT Y UPPER LIMIT
        CHECK IF Y REACHED LOWER LIMIT
            FE FA   CP 250D         -48 0 -> 203D + 48 ] HANDLE <0 CASE.
            30 07   JR NC,VECT2     ANY X>250 IS CONSIDERED NEG
                                    ] IF YH < LOWER LIMIT,
            B8      CP B            JMP TO VECT2 (LIMIT WAS
33B0H       38 04   JR C,VECT2      REACHED)
        GET Y UPPER LIMIT, CHECK IF IT REACHED UPPER LIMIT
            46      LD B,(HL)       B = Y UPPER LIMIT ] IF YH < UPPER LIMIT,
            B8      CP B            JMP VECT3
            38 1A   JR C,VECT3
        LOAD UPPER LIMIT INTO VECTOR PACKET
VECT2       DD 70 03    LD (IX+3),B     YH IN VP = NEW YH
            DD 36 02 00 LD (IX+2),0     YL ↓ = 0
            DD CB 04 DE SET 3,(IX+4)    SET Y LIMIT ATTAINED BIT 3 IN
                                        Y CHECKS MASK.
33C2H       F1          POP AF          CLEAN UP STACK
    LIMIT ATTAINED, REVERSE THE DELTA (ΔYH ΔYL), DE = ΔYH ΔYL
            7A      LD A,D  ] D = ΔȲH   ] REVERSE DELTA
            2F      CPL                 (2'S COMPLEMENT)
            57      LD D,A  ]
            7B      LD A,E  ] E = ΔȲL
            2F      CPL
            5F      LD E,A  ]
            13      INC DE              ]
            DD 73 00    LD (IX+0),E     ] LOAD REVERSE DELTA
            DD 72 01    LD (IX+1),D     IN VECTOR PACKET
33D0H       C9      RET
VECT3       E3      EX (SP),HL      HL = YH YL = NEW COORDINATE
            DD 75 02    LD (IX+2),L ] LOAD COORDINATE
            DD 74 03    LD (IX+3),H ] IN VECTOR PACKET
            E1          POP HL
            DD CB 04 9E RES 3,(IX+4)    CLEAN UP STACK
                                        RESET LIMIT ATTAINED BIT 3
                                        IN Y CHECKS MASK
33DDH       C9          RET
```

LIMIT CHECK (X DIRECTION)

ENTER WITH: DE = UPDATED (NEW) X_H
BC = X LIMIT (LOWER OR UPPER LIMIT)

```
LCHK  33DE_H  E5        PUSH HL      SAVE LIMIT POINTER
              6B  ← BOY  LD L,E       } HL = UPDATED (NEW) X_H
      33E0_H  62        LD H,D
              A7        AND A        CARRY = 0                    } COMPARE
              ED 42     SBC HL,BC    HL = NEW X_H - LIMIT - CARRY  } NEW X_H
                                     (HL)  (BC)                   } WITH LIMIT
              E1        POP HL       HL = LIMIT POINTER
      33E5_H  C9        RET
```

MOVE CRITTER VECTOR

```
IVECT  33E6_H  0135   MCINT   INTERRUPT SERVICE ROUTINE
```

MOVE CRITTER IM2 VECTOR SET UP

```
                                DON'T NEED ← DI   EXECUTED WHEN INTERRUPT IS ACKNOWLEDGED BY Z80
CINTS  33E8_H  F3
               3E 33    LD A,?      } OUTPUT
               ED 47    LD I,A      } PAGE OF VECTOR
               3E E6    LD A,?      } OUTPUT
               D3 0D    OUT (0D_H),A} LINE OF VECTOR
       33F1_H  3E 08    LD A,8      } CONTINUAL INT REQUESTS
               D3 0E    OUT (0E_H),A} FOR Z80 ACKNOWLEDGEMENT
               3E C8    LD A,C8_H   } GENERATE INTERRUPT
               D3 0F    OUT (0F_H),A} AT 200 TV SCAN LINES
               ED 5E    IM2         USE INTERRUPT MODE 2
               FB       EI
       33FC_H  C9       RET
```

WAVE2 @ 3E97$_H$ IN FISH DEMO SEA BOTTOM

```
WAVE2  33FD_H  62   0110 0010
               F7   1111 0111
       33FF_H  FF_H  UNUSED BYTE
```

VIEW AND FLIP PAGES

```
                                    Z80 INTERRUPT REG I IS USED.
                                    OK, IF NO SCREEN INTERRUPT IS EXECUTING
FPAGE
FPAGE  3400H  3E 70     LD A,112D   ]
              ED 47     LD I,A      ] SET VIEW TIME CTR FOR ≈ 57 SEC
                                    FOR PAGE 0 INTRO VIEWER READ
              18 16     JR SKPAG    SKIP PAGE 0 SET UP

SET UP PAGE 0 FOR FLIP "LOOP BACK" (TEXT INTRO)

FPAG0 3406H   AF        XOR A       ]
              D3 74     OUT (74H),A ] SET TV DISPLAY TO PAGE 0
              3E CB     LD A,203D   ]
              D3 0A     OUT (0AH),A ] SET VERT BLNK REG = 203D
              3E 2B     LD A,0010 1011 ]
              D3 09     OUT (09),A  ] HORIZ SET CLR BOUNDARY
                                    0010 1011
                                      43D BORDER CLR BLK
      3411H   21 34 21  LD HL,2134H ]
              01 0B 08  LD BC,080BH ] SET PAGE 0 COLORS
      3417H   ED B3     OTIR        ]
              CD 81 32  CALL FSND ← P.81   PLAY FLIP SOUND
SKPAG         CD 9E 32  SLTD ← P.81        SELECT VIEW TIME DELAY

CHECK FOR FIRST PAGE FLIP

              ED 57     LD A,I      ]
                                    ] IF THIS IS FIRST FLIP,
      3421H   FE 70     CP 70H      ] I REG WILL = 70H.
              20 04     JR NZ,FPAG1 ] IF SO, SET I = 0AH
              3E 0A     LD A,21D    ] TO INITIAL FLIP VIEW
              ED 47     LD I,A      ] TIME (4 SEC)

SET UP PAGE 1 FOR VIEWING (NARROW VERTICAL STRIPES)

FPAG1         3E 01     LD A,01     ]
              D3 74     OUT (74H),A ] FLIP TV DISPLAY TO PAGE 1
              3E D4     LD A,1101 0100 ]
      342FH   D3 09     OUT (09),A  ] HORIZ SET CLR BOUNDARY
                                    11 01 0100
                                      20 SPLIT SCREEN
                                    BORDER CLR 11
      3431H   21 37 23  LD HL,CLRT3 ]
              01 0B 08  LD BC,080BH ] SET COLORS
      3437H   ED B3     OTIR        ]
              CD 81 32  CALL FSND   PLAY FLIP SOUND
      343CH   CD 9E 32  CALL SLTD   SELECT VIEW TIME DELAY
```

NEW FLIP B

91

SET UP PAGE 2 FOR VIEWING (FISH AQUARIUM + 15 MAGIC WRITES)

FPAG2	$343F_H$	3E 02	LD A,02	FLIP TV DISPLAY TO PAGE 2
	3441_H	D3 74	OUT (74_H),A	
		3E 9C	LD A,1001 1100	SET HORIZ CLR BOUNDARY 10 01 1100 (28D); BORDER CLR 10
	3445_H	D3 09	OUT (09),A	
		21 03 20	LD HL,CLRT1	SET COLORS
		01 0B 08	LD BC,$080B_H$	
	$344D_H$	ED B3	OTIR	
		CD 81 32	CALL FSND	PLAY FLIP SOUND
	3452_H	CD 9E 32	CALL SLTD	SELECT VIEW TIME DELAY

SET UP PAGE 3 FOR VIEWING (NARROW HORIZONTAL STRIPES)

FPAG3		3E 03	LD A,03	FLIP TV DISPLAY TO PAGE 3
		D3 74	OUT (74_H),A	
		3E D4	LD A,1101 0100	SET HORIZ CLR BOUNDARY 11 01 0100; 20 SPLIT SCREEN BORDER CLR 11
	$345B_H$	D3 09	OUT (09),A	
		21 79 32	LD HL,CLRT7	SET COLORS
	3460_H	01 0B 08	LD BC,$080B_H$	
	3463_H	ED B3	OTIR	
		CD 81 32	CALL FSND	PLAY FLIP SOUND
		CD 9E 32	CALL SLTD	SELECT VIEW TIME DELAY

SET UP PAGE 4 FOR VIEWING (TEXTURED 10 COLOR TEST PATTERN)

FPAG4		3E 04	LD A,04	FLIP TV DISPLAY TO PAGE 4
		D3 74	OUT (74_H),A	
		3E C9	LD A,201D	SET VERICAL BLANK REG TO 201D
	3471_H	D3 0A	OUT ($0A_H$),A	
		3E 14	LD A,00 01 0100	SET HORIZ CLR BOUNDARY 00 01 0100; 20 SPLIT SCREEN BORDER CLR 00
	3475_H	D3 09	OUT (09),A	
		21 3F 23	LD HL,CLRT4	SET COLORS
		01 0B 08	LD BC,$080B_H$	
	$347D_H$	ED B3	OTIR	
		CD 81 32	CALL FSND	PLAY FLIP SOUND
	3482_H	CD 9E 32	CALL SLTD	SELECT VIEW TIME DELAY

SET UP PAGE 5 FOR VIEWING (NARROW VERTICAL + HORIZONTAL STRIPES)

```
FPAG5   3485H  3E 05      LD A,05            ] FLIP TV DISPLAY TO PAGE 5
               D3 74      OUT (74H),A        ]
               3E CB      LD A,203D          ] SET VERTICAL BLANK REG TO 203D
               D3 0A      OUT (0AH),A        ]
               3E D4      LDA,11010100       ] SET HORIZ CLR BNDRY
                          OUT (09),A         ]   11 01 0100
                                                  |  |  |
                                                  |  |  20 SPLIT SCREEN BORDER CLR 11
                                                  |  -> 01
                                                  -> 11
       348FH   D3 09
       3491H   21 3723    LD HL,CLRT3        ] SET COLORS
               01 0B 08   LD BC,080BH        ]
       3497H   ED B3      OTIR               ]
               3E 09      LD A,09            ] SET RIGHT BOUNDARY COLOR 03
               D3 03      OUT (03),A         ]   SAME AS LEFT
               CD 8132    CALL FSND          PLAY FLIP SOUND
       34A0H   CD 9E 32   CALL SLTD          SELECT VIEW TIME DELAY
```

SET UP PAGE 6 FOR VIEWING (DEMO TITLE PAGE)

```
FPAG6           3E 06      LD A,06            ] FLIP TV DISPLAY TO PAGE 6
                D3 74      OUT (74H),A        ]
                3E AB      LD A,1010 1011     ] SET HORIZ CLR BNDRY
       34A9H    D3 09      OUT (09),A         ]   10 10 1011
                                                  |  |  |
                                                  |  |  43D
                                                  |  -> 10
                                                  -> 10
                21 D92C    LD HL,CLRT5        ] SET COLORS
                01 0B 08   LD BC,080BH        ]
       34B1H    ED B3      OTIR               ]
                CD 8132    CALL FSND          PLAY FLIP SOUND
                CD 9E 32   CALL SLTD          SELECT VIEW TIME DELAY
```

PIXEL 11 YEL
10 ORG
01 BLU
00 BLK

SET UP PAGE 7 FOR VIEWING (HI-RES GUNFIGHT SCREEN SHOT)

```
FPAG7           3E 07      LD A,07            ] FLIP TV DISPLAY TO PAGE 7
                D3 74      OUT (74H),A        ]
                3E DE      LD A,11 01 1110    ] SET HORIZ CLR BNDRY
                                                  11 01 1110
                                                  |  |  |
                                                  |  |  30D RIGHT OF CENTER
                                                  |  -> 01
                                                  -> 11
                                                  BORDER CLR GREEN
       34BFH    D3 09      OUT (09),A         ]
       34C1H    21 4723    LD HL,CLRT6        ] SET COLORS
                01 0B 08   LD BC,080BH        ]
       34C7H    ED B3      OTIR               ]
                CD 8132    CALL FSND          PLAY FLIP SOUND
       34CCH    CD 9E 32   CALL SLTD          SELECT VIEW TIME DELAY
```

VIEWING TIME FLIP LOOP

NEW FLIP D

93

Z80 I REG = VIEW TIME COUNTER, INITIALIZED TO 10_D (5 SEC)

I-3 DECREMENTATIONS 10, 7, 4, 1, 0 (5 PASSES)

4 SEC — 3 SEC — 2 SEC — ≈ ½ SEC — ≈ 1/4 SEC

WITH 6 PASS "RUN ON" EACH

```
RVTM  34CF_H  ED 57      LD A,I
      34D1_H  A7         AND A
              28 1C      JR Z, RVTM4        } IF I=0, JUMP TO DECREMENT "RUN ON" PASS COUNTER
              FE 01      CP 1
              20 14      JR NZ, RVTM3       } IF I=1, DECREMENT "RUN ON" CTR IN 7FCO_H
              3A C0 7F   LD A, (7FCO_H)
              3D         DEC A
              32 C0 7F   LD (7FCO_H), A
              20 08      JR NZ, RVTM2       LOOP BACK ANOTHER PASS IF CTR≠0
RVTM1 34E1_H  AF         XOR AF             } CTR=0, SET I=0
              ED 47      LD I, A
              3E 06      LD A, 6            } INITIALIZE "RUN ON" CTR IN 7FCO TO 6 WHEN I=4 AND 1. (OK IF INITIALIZED WHEN I=7
              32 C0 7F   LD (7FCO_H), A
RVTM2         C3 06 34   JP FPAGO
DOG ->
RVTM3 +20     D6 03      SUB 3              } DECREMENT I=10, 7 AND 4 BY 3
-14           18 F2      JR, RVTM1
```

-14

```
  1421
0000 1110
1111 0001
        +1
1111 0010
```

```
+28 RVTM4 34F0_H  3A C0 7F   LD A, (7FCO_H)     } DECREMENT I=0 "RUN ON" CTR IN 7FCO_H
                  3D         DEC A
                  32 C0 7F   LD (7FCO_H), A
                  C2 06 34   JP NZ, FPAGO      IF "RUN ON" CTR≠0, LOOP BACK FOR ANOTHER PASS
                                               IF CTR=0, END "RUN ON"
                  3E 0A      LD A, 10_D        } INITIALIZE I=10 FOR OPTIONAL FLIP PAGE NONSTOP
                  ED 47      LD I, A
          34FE_H  C3 47 36   JP COPYP          JMP TO COPY MVCRT TO PAGE 7 FOR EXECUTION
                                               ← P.100
OPTIONS           06 34      JP FPAGO          OPTIONAL FLIP PAGE NONSTOP SUBSTITUTION
                                               ← PAGE 90
                  E5 35      JP MVCRT          OPTIONAL MOVE CRITTER (PAGE 99)
```

MOVE CRITTER INTERRUPT ROUTINE

VECTOR @ $33E6_H$

DECREMENT SECONDS COUNTER (NO BCD FOR SECS REQUIRED)

SAVING REG A IS NECESSARY TO RUN GRAPHICS PERFECTLY.

HL= CLOBBERED IN THIS ROUTINE

ACINT	3501_H	F5	PUSH AF	
		E5	PUSH HL	
		21 EB 7F	LD HL, TMRGO	DECREMENT TMRGO *
		35	DEC (HL)	
		20 0D ← DOG	JR NZ, MCINT1	EXIT IF TMRGO ≠ 0
		36 3B	LD (HL), 59_D	SET TMRGO = 59_D
		21 ED 7F	LD HL, SECS	DECREMENT SECS COUNT DOWN TIMER *
		35	DEC (HL)	
		20 05	JR NZ, MCINT1	EXIT IF SECS ≠ 0
	3511_H	21 F8 7F	LD HL, NPAGE	SET BIT 7 IN NEW PAGE (TO INDICATE TIME TO MOVE CRITTER TO NEXT PAGE)
		CB FE	SET 7, (NPAGE)	
MCINT1		E1	POP HL	
		F1	POP AF	
		FB	EI	
		C9	RET	
	3519_H			

* PARAMETER IS INITIALIZED IN MOVE CRITTER PROGRAM

VIEW PAGE TABLE

VPAGO	$351A_H$	2B	HOR CLR BOUNDARY	PAGE 0 SCRN PARAMETERS
		34 21	COLOR TABLE ADR	
	$351D_H$	D4		PAGE 1
		37 23		
	3520_H	9C		2
		03 20		
		D4		3
		79 32		
		14		4
		3F 23		
		D4		5
		37 23		
		AB		6
		D9 2C		
		DE		7
		47 23		
	3530_H			

WAVE3	3532_H	22	0010 0010	WAVE3 @ 3E9C IN FISH DEMO, SEA BOTTOM
	3533_H	77	0111 0111	

SET UP PAGE TO VIEW

```
                                          DISPLAY THIS PAGE
SP2V   3534H  D3 74      OUT (74H),A
              11 1A 35    LD DE,VPAGO     DE = VIEW PAGE TABLE
              A7          AND A           IF PAGE NUMBER = 0,
              2808        JR Z,SP2VB      DE IS ALREADY POINTING AT
                                          PAGE'S SCRN SET UP PARAMETERS
              47          LD B,A
              AF          XOR A
SP2VA                     ADD A,3         POINT DE AT PAGE'S
              C603        DJNZ,SP2VA      SCRN SET UP PARAMETERS
       3540H  10FC        ADD A,E
              83          LD E,A
              5F          LD A,(DE)       SET UP PAGE'S
SP2VB         1A          OUT (09),A      HORIZ CLR BOUNDARY
              D309        INC DE
-4            13          LD A,(DE)
              1A          LD L,A          HL = PAGE'S COLOR TABLE
              6F          INC DE
              13          LD A,(DE)
              1A          LD H,A
              67          LD BC,080BH     SET UP PAGE'S COLORS
              010B08      OTIR
       3550H  EDB3        LD A,(7FF9H)    IS THIS PAGE 5?
              3AF97F      CP 5            IF NO, JUMP AHEAD
              FE05        JR NZ,SP2VC     IF YES, CONTINUE
              2004        LD A,09         SET RIGHT BORDER CLR 03
              3E09        OUT (03),A        SAME AS LEFT
              D303
                                          PLAY FLIP SOUND
SP2VC         CD8132      CALL FSND
       3560H  C9          RET

                                          CRITTER
                                          16 PIXELS WIDE
                                          18 LINES HIGH
RANDOMIZE CRITTER X OR Y INITIAL COORDINATE

RXORY  3561H  3E0B        LD A,0BH        RANDOMIZE REG A
              CDC532      CALL RANGE        FROM 0-10D
              21B07F      LD HL,7FB0H     POINT HL AT STACK AREA AT TOP OF
                                          10 BYTE INDEX FOR INITIAL RANDOMIZATION
              85          ADD A,L         ADD TO RANDOMIZATION A BYTE
              6F          LD L,A            INDEXED IN THE STACK.
              7E          LD A,(HL)
              CB47        BIT 0,A         IF BIT 0 IN RANDOMIZED A
              280A        JR Z,VARX         = 0, VARYX
                                            = 1, VARYY
VARY   3570H  3EB4        LD A,180D       INITIALIZE
              CDC532      CALL RANGE      RANDOMIZED Y COORD
              32CC7F      LD (7FCCH),A    IN VECTOR BLOCK
              1808        JR EXIT           OR
VARX          3EFF        LD A,FFH        INITIALIZE
       3582H  CDC532      CALL RANGE      RANDOMIZED X COORD
              32C67F      LD (7FC6H),A
EXIT   3582H  C9          RET
```

C9 RET

CUSTOM MULTI-PAGER WRITE ROUTINE 96

USED TO PLOP WRITE A CRITTER IN ANY OF 8 PAGES WHILE THE Z80 R/Ws ARE WORKING THE STACK AND VARIABLE(S) WITHIN SCREEN RAM PAGE 7 FOR THE MAIN PROGRAM.

WRITE THE CRITTER USING VECTOR BLOCK (SIMILAR TO LOW-RES SUB#30)
VECTOR BLOCK IS 15 BYTES USING 2 BYTES EACH FOR X_H AND ΔX_H.

ENTER WITH: IX = VECTOR BLOCK (PACKET) ADDRESS
HL = PATTERN ADDRESS - 4 (POINTING AT RELATIVE X)
$(7FF9_H)$ = PAGE NUMBER 0-7 TO WRITE (VIEW) CRITTER.

NOTE: $(7FF7_H)$ = REGY = Y COORD SAVED FOR WRITING CRITTER.

CVWRIT	3583_H	DD 7E 00	LD A,(IX)	A = MAGIC REGISTER VALUE
		DD 46 0D ← D.G	LD B,(IX+$0D_H$)	$B = Y_H$
		DD 5E 07	LD E,(IX+7)	$DE = X_H$
		DD 56 08	LD D,(IX+8)	
		DD CB 01 F6	SET 6,(IX+1)	SET BLANK BIT

WRITE RELATIVE (SIMILAR TO LOW-RES SUB#32)

CWRITR	3593_H	F5	PUSH AF	SAVE MR VALUE
		7E	LD A,(HL)	A = RELATIVE X
		23	INC HL	POINT HL AT RELATIVE Y
		83	ADD A,E	
		5F	LD E,A	$DE = X_H$ OR = XCOORD + RELATIVE X
		7A	LD A,D	
		CE 00	ADC A,0	
		57	LD D,A	
		7E	LD A,(HL)	A = RELATIVE Y
		23	INC HL	POINT HL AT XSIZE
		80	ADD A,B	$A = Y_H$ OR YCOORD + RELATIVE Y
		32 F7 7F	LD (REGY),A	SAVE Y = REGY
	$35A2_H$	F1	POP AF	A = MR VALUE

WRITE WITH PATTERN SIZE (SIMILAR TO LOW-RES SUB#34)

CWRITP		4E	LD C,(HL)	C = XSIZE
	$35A4_H$	23	INC HL	POINT HL AT Y SIZE

$35A5_H$ 46 LD B,(HL) B=YSIZE
23 INC HL POINT HL AT PATTERN TO WRITE

WRITE WITH COORDINATES CONVERSION (SIMILAR TO LOW-RES SUB#36)

CWRIT $35A7_H$ CD 002C CALL RELTA1

WRITE THE PATTERN

THIS SUBROUTINE IS AN EXAMPLE OF HOW TO UTILIZE McM DESIGN'S MULTI-PAGER HARDWARE TO MAGIC WRITE A PATTERN TO ONE PAGE WHILE THE Z80 IS (NORMALLY WORKING) THE STACK AND SCRATCHPAD VARIABLE(S) IN ANOTHER PAGE SPECIFIED BY THE MULTI-PAGER OUTPUT PORT 75_H. THIS SUBROUTINE IS SIMILAR TO THE LOW-RES NORMAL MAGIC WRITE SUB MWRT. ENTER THIS SUBROUTINE WITH THE Z80 R/W'S POINTING TO PAGE 7 AND WITH ($7FF9_H$) = THE PAGE NUMBER (0-7) TO WRITE CRITTER PATTERN INTO.

POINT THE Z80 TO R/W A PAGE USING OUTPUT PORT 75_H = 0XXX 0XXX
(Z80 WRITE; Z80 READ, READ MAGIC HARDWARE FOR XOR, OR WRITE)

WHERE XXX = THE PAGE NUMBER (0-7).

POINTING THE Z80 IN THIS MANNER ALLOWS THE Z80 TO WORK IN THIS PAGE, THE STACK AND ANY SCRATCHPAD VARIABLES, PLUS ALLOWS THE MAGIC HARDWARE TO READ RAM BYTES IN THE PAGE FOR MAGIC XOR, OR LOGICAL WRITES.

POINT Z80 R/W AT PAGE CRITTER IS BEING WRITTEN INTO

CMWRIT $35AA_H$ C5 PUSH BC SAVE YSIZE, XSIZE
3A F9 7F LD A,($7FF9_H$) A=PAGE NUMBER (0-7) TO WRITE PATTERN INTO
47 LD B,A
CB00 RLC B
$35B1_H$ CB00 RLC B
CB00 RLC B
CB00 RLC B
(SHIFT LEFT PAGE NUMBER INTO BITS 4-7 IN B)
80 ADD A,B A=0XXX 0XXX = PAGE R/W
C1 POP BC BC=YSIZE, XSIZE
FE 77 CP 77
28 09 JR Z, CWRT
(IS THIS R/W PAGE 7? IF SO, JUMP AHEAD. MAIN PROGRAM STACK IS ALREADY SETUP.)
ED 73 FA 7F LD ($7FFA_H$), SP SAVE PAGE 7 STACK POINTER IN PAGE 7 SCRATCHPAD.
$35C1_H$ D3 75 OUT (75_H), A POINT Z80 R/W AT THIS PAGE (0-6)
$35C3_H$ 31 C0 7F LD SP, $7FC0_H$ INITIALIZE THE STACK POINTER IN THIS PAGE (FOR CRITTER WRITE, STACK PUSH/POP).

(PLOP) WRITE CRITTER INTO PAGE (SIMILAR TO LOW-RES NORMAL PLOP CRIT*)

CWRT	35C6H	AF	XOR A,F	SIMILAR TO LOW-RES MWRT. SEE NUTTING MANUAL Z80/ROM CODE BREAKDOWN, PAGE 50.
		C5	PUSH BC	
		D5	PUSH DE	
		47	LD B,A	
		ED B0	LDIR	
		12	LD (DE),A	
		D1	POP DE	
		EB	EX DE,HL	
		0E 50	LD C,80D	HI-RES 80 BYTES/LINE
		09	ADD HL,BC	
	35D1H	EB	EX DE,HL	
		C1	POP BC	
		10 F1	DJNZ CWRT	
		3A F9 7F	LD A,(7FF9) A=PAGE #	IS CRITTER BEING WRITTEN IN PAGE 7? IF SO, RETURN NOW. NO STACK RESTORATION REQ'D.
		FE 07	CP 7	
		C8	RET Z	
		3E 77	LD A,77H	RESTORE Z80 R/W TO POINT BACK TO PAGE 7
		D3 75	OUT (75H),A	
	35E0H	ED 7B FA 7F	LD SP,(7FFAH)	RESTORE PAGE 7 STACK POINTER
	35E4H	C9	RET	

* SEE NUTTING MANUAL Z80/ROM CODE LISTING, PAGE 50

MOVE CRITTER PROGRAM (JUMP FROM $34FE_H$, P.93)

(COPIED INTO PAGE 7 SCRATCH PAD AREA FOR EXECUTION)

```
MVCRT 35E5H  3E 77      LD A, 0111 0111     ] POINT Z80 TO R/W PAGE 7
              D3 75      OUT (75H), A       ] FOR PROGRAM EXECUTION
              31 BF 7F   LD SP, 7FBFH       INITIALIZE STACK POINTER
              3E CC      LD A, 204D         ] SET VERT BLANK REG = 204D
              D3 0A      OUT (0AH), A       ] (REVEAL ALL SCREEN RAM)
35F0H         3E 3B      LD A, 59D          ] INITIALIZE TMRGO
              32 EB 7F   LD (TMRGO), A      ] AT 7FEBH TO 59D
                                            ] (FOR SCREEN INT TIMER)
              AF         XOR A              ] ZERO NPAGE @ 7FF8H
              32 F8 7F   LD (7FF8H), A      ] PAGE NUMBER @ 7FF9H
              32 F9 7F   LD (7FF9H), A
              CD E8 33   CALL CINTS  (P.89)  START UP PAGE VIEW COUNTDOWN TIMER

8 PAGE LOOP BACK ENTRY HERE

MVCRT1 35FFH  21 B0 32   LD HL, 32B0H       HL = "COPY FROM" ADR   ] INITIALIZE
3602H         11 BF 7F   LD DE, 7FBFH       DE = "COPY TO" ADR     ] CRITTER
              01 0F 00   LD BC, 15D         BC = # OF BYTES TO COPY] VECTOR
              ED B0      LDIR                                      ] BLOCK
              3E 1F      LD A, SECS         ] SET SECS
              32 ED 7F   LD (7FEDH), A      ] AT 7FED = 31D *
              CD 61 35   CALL RXORY         RANDOMIZE XH OR YH
3612H         3A F9 7F   LD A, (7FF9H)      A = PAGE NUMBER TO VIEW
              CD 34 35   CALL SP2V          SET UP PAGE TO VIEW
              F3         DI                 DISABLE ANY SCREEN INTERRUPT DURING CRITTER WRITE
MVCRT2        DD 21 BF 7F LD IX, 7FBFH      ] PLOP WRITE CRITTER
              21 0B 20   LD HL, 200BH       ] FROM VECTOR BLOCK
3620H         CD 83 35   CALL CVWRITE
              FB         EI                 START UP SCREEN INTERRUPT AGAIN
              DD 21 BF 7F LD IX, 7FBFH      ] UPDATE CRITTER
              21 BF 32   LD HL, CLT         ] VECTOR BLOCK
              CD FD 32   CALL MVECT
              3A F8 7F   LD A, (NPAGE)      ] IF SECS COUNTDOWN ≠ 0
3631H         CB 7F      BIT 7, A           ] (THAT IS, 7FF8 BIT7 = 0),
3633H         28 E3      JR Z, MVCRT2       ] LOOP BACK TO CONTINUE
                                            ] MOVE/WRITE CRITTER
```

* SCREEN INTERRUPTS ARE DISABLED DURING THE CRITTER WRITE SUBROUTINE.

3635_H	AF	XOR A	RESET SEC=0 FLAG AT 7FF8
	32 F8 7F	LD ($7FF8_H$), A	
	3A F9 7F	LD A, (PAGE NUMBER)	INCREMENT PAGE NUMBER @ $7FF9_H$
	3C	INC A	
	32 F9 7F	LD (PAGE NUMBER), A	
3640_H	FE 08	CP 8	
	20 BB ← 2 BYTS	JR NZ, MVCRT1	

-69
64 32 16 8 4 2 1
0 1 0 0 0 1 0 1
0 1 0 1 1 0 1 0
1011

3644_H C3 42 38 JP IFDEMO JUMP TO HI-RES FISH DEMO

↑ OPTIONAL SUBSTITUTION 96 29 JP PAGO MULTI-PAGE DEMO ENTRANCE (PAGE FLIPS + CRITTER ERASE NONSTOP)

MOVE CRITTER MAIN PROGRAM = 98 BYTES
CRITTER VIEW TIME ON PAGE = 30 SECONDS ← P.99

COPY "MOVE CRITTER" PROGRAM MVCRT @ $35E5_H$
TO PAGE 7 SCRATCH PAD AREA FOR EXECUTION

COPYP 3647_H	21 E5 35	LD HL, $35E5_H$	HL = "COPY FROM" ADR	COPY MOVE CRITTER PROGRAM TO PAGE 7 SCRATCH PAD
	11 20 7F	LD DE, $7F20_H$	DE = "COPY TO" ADR	
	01 62 00	LD BC, 98_D	BC = # OF BYTES TO COPY	
3650_H	ED B0	LDIR		
3652_H	C3 20 7F	JP $7F20_H$	JP TO $7F20_H$ IN PAGE 7 TO EXECUTE MOVE CRITTER PROGRAM THERE	

NOTES: MULTI-PAGE DEMO = 1654_H = 5716 BYTES WITH 2476 BYTES REMAINING

IN THE MOVE CRITTER PROGRAM, THE VERTICAL BLANK REGISTER WAS DROPPED ALL THE WAY DOWN, SO YOU CAN SEE:

A. THE Z80 WORKING THE STACK AREA WITH PUSH/POP INSTRUCTIONS AT THE BOTTOM RIGHT CORNER OF THE SCREEN RAM AREA FOR THE CRITTER WRITE SUBROUTINE EXECUTING IN PAGES 0 THRU 6.

B. THE CRITTER MOVE PROGRAM LOADED INTO THE BOTTOM OF PAGE 7 BEGINNING AT $7F20_H$, WHICH IS THE LEFT MOST BYTE 2 LINES UP FROM THE BOTTOM OF SCREEN RAM. THE TOP LINE CONTAINS 80 BYTES OF THE 98 BYTE MAIN PROGRAM. THE MAIN PROGRAM CALLS 5 SUBROUTINES IN CARTRIDGE ROM. A Z80 JUMP TO $7F20_H$ IS MADE TO EXECUTE THE MAIN PROGRAM.

MOVE CRITTER PROGRAM
SCREEN RAM DATA BASES IN PAGE 7

↑

$7FBE_H$ STACK BEGIN HERE

$7FBF_H$ MR (PLOP WRITE)

$7FC0$ VECTOR STATUS — BIT 5 1=NO MOTION, 0 MOTION OCCURED; BIT 7 1= VECTOR IS ACTIVE; BIT 6 BLANK BIT SET BY VECTOR WRITE ROUTINE

$7FC1$ 02 TIME BASE

$7FC2_H$ ΔX_L ← USED IN SCREEN INTERRUPT ROUTINE

ΔX_H E
2 BYTES D

X_L

$7FC6_H$ 00 X_H
00 2 BYTES

$7FC8_H$ 03 X CHECKS MASK — BIT 0 1= DO LIMITS CHECK; BIT 1 1= REVERSE Δ; BIT 3 1=X LIMIT ATTAINED, 0=LIMIT NOT ATTAINED

$7FC9_H$ ΔY_L

ΔY_H

Y_L

$7FCC_H$ 00 Y_H

$7FCD_H$ 03 Y CHECKS MASK — BIT 0 1= DO LIMITS CHECK; BIT 1 1= REVERSE Δ; BIT 3 1=Y LIMIT ATTAINED, 0=LIMIT NOT ATTAINED

$7FEB_H$ TMR60 (FOR SCREEN INTERRUPTS) INITIALIZED TO 59_D

$7FED_H$ SECS

7FEE

7FEF
7FF0
7FF1
7FF2 — USED FOR HI-RES RANGED SUBROUTINE

7FF7 REG Y = Y COORDINATE

7FF8 NPAGE BIT 7 1 = SECS HAS REACHED ZERO (MOVE CRITTER TO NEXT PAGE)
0 CURRENT PAGE IN PROGRESS

7FF9 PAGE NUMBER (BEING VIEWED, 0-7)

7FFA
7FFB — SAVE PAGE 7 STACK POINTER FOR USE WITH CUSTOM WRITE ROUTINE @ ?

MULTI-PAGER TEST DEMO OPTION

(RIGHT MOST COLUMN KEY DOWN AT SYSTEM RESET)

MOVE CRITTER USING HAND CONTROLLER (PLAYER 1) WITHIN 3 PAGES (SCENES)

H PAD

RAM

MOVE TO RAM VECTOR BLOCK AT $7FC0_H$

INITIAL CRITTER VECTOR BLOCK (PACKET), 15 BYTES

FOR RAM			
7FC0 →	3655_H	00	MR VALUE (PLOP WRITE)
7FC1		00	VECTOR STATUS
7FC2		02	TIME BASE
7FC3		00	ΔX_L
7FC4		00	ΔX_H (2 BYTES)
7FC5		00	
7FC6		00	X_L
7FC7		97	X_H (2 BYTES)
7FC8		00	
7FC9		00	X CHECKS MASK
7FCA		00	ΔY_L
7FCB	3660_H	00	ΔY_H
7FCC		00	Y_L
CD		5C	Y_H
7FCE →	3663_H	00	Y CHECKS MASK

INITIALIZE CRITTER IN PAGE 0, NEAR CENTER

$X, Y = 151_D, 92_D$

CRITTER WILL BE INITIALIZED WITH NO MOTION.

X_H = X COORDINATE $0-319_D$ RANGE

BIT 3 = LIMIT ATTAINED IF 1

INITIALIZE IN PAGE 0 AT $X = 151_D = 97_H$

Y_H = Y COORDINATE = $0-203_D$ RANGE

INITIALIZE IN PAGE 0 AT $92_D = 5C_H$

CRITTER LIMITS (PAGE 0) ← CENTER SCENE

LIMT0	3664_H	00	X LOWER LIMIT
		00	
		30	X UPPER LIMIT (304_D)
		01	
		1B	Y LOWER LIMIT (27_D)
	3669_H	9F	Y UPPER LIMIT *

* A0 LIMIT EATS 1 BLUE BYTE AT BOTTOM RIGHT CORNER

CRITTER LIMITS (PAGE 1) ← LEFT SCENE

LIMT1	$366A_H$	50	X LOWER LIMIT (80_D)
		00	
		30	X UPPER LIMIT
		01	
		1B	Y LOWER LIMIT
	$366F_H$	9F	Y UPPER ↓

CRITTER LIMITS (PAGE 2) ← RIGHT SCENE

USE WITH DEMO MOVE CRITTER WITHIN 3 INTERCONNECTING SCENES

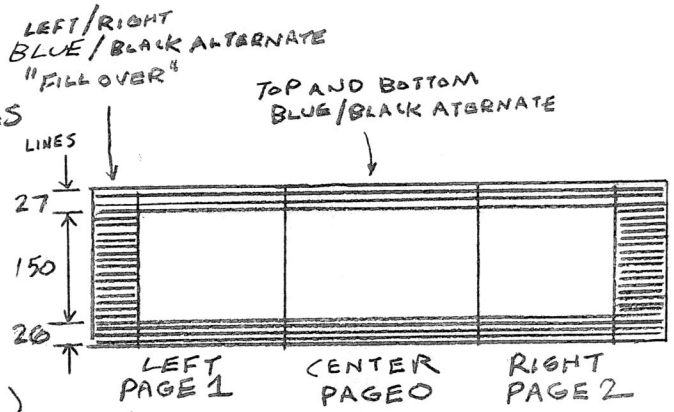
LIMT2 3670_H 00] X LOWER LIMIT
00]

DF] X UPPER LIMIT (224_D)
00]

1B Y LOWER LIMIT

3675_H 9F Y UPPER LIMIT

AREA → COMMON → ALL 3 PAGES



FILL SCREEN PAGE WITH LINES
(SIMPLISTIC STATIC GRAPHICS)

ENTRY Z80 PARAMETERS ARE:

HL = LINE "START ADDRESS" TO BEGIN FILL

C = FILL "ON" COLOR (CLR 11, 10 OR 01)
FILL "OFF" COLOR SET TO CLR 00 (BLACK) BY PROGRAM

B = NUMBER OF VERTICAL LINES TO FILL

E = HORIZONTAL BYTES/LINE TO FILL

SUBROUTINE HAS 3 ENTRY POINTS LFILLA, LFILLB AND LFILLC DEPENDING ON FILL APPLICATION.

```
LFILLA  3676H  21 00 40     LD HL,4000H   HL = LINE START ADR = 4000H, ENTRY A
LFILLB  3679H  1E 50        LD E,80D      E = 80 BYTES/LINE, FULL SCREEN, ENTRY B
LFILLC  367BH  79           LD A,C        A = "ON" COLOR, ENTRY C
               E5           PUSH HL       SAVE LINE FILL START ADR,
               50           LD D,B        SAVE # OF LINES (LOOP CTR) IN D
               CB40         BIT 0,B       ] IF LOOP CTR IS ODD (BIT 0=1), USE "ON" COLOR
               3680H 2001   JR NZ,LFILL1  ] AND JMP FORWARD 1 BYTE.
                            ]             ] OTHERWISE, CTR IS EVEN (BIT 1=0).
                            ]             ] USE PIXEL CLR 00 (BLK).
               AF           XOR A
               43           LD B,E        B = BYTES/LINE TO FILL
LFILL1         77           LD (HL),A     FILL RAM BYTE WITH THE COLOR
LFILL2         23           INC HL        POINT TO NEXT RAM BYTE *
               10 FC        DJNZ LFILL2   LOOP BACK TO WRITE NEXT BYTE IN THE LINE
               E1           POP HL        HL = PREVIOUS LINE FILL START ADR
               C5           PUSH BC       SAVE FILL "ON" COLOR. B=0 FROM DJNZ
               0E 50        LD C,80D      ] POINT HL AT NEXT
               09           ADD HL,BC     ] LINE START ADR
               C1           POP BC        C = "ON" COLOR AGAIN
               42           LD B,D        B = # OF LINES (LOOP CTR) AGAIN
               10 EA        DJNZ LFILLC
               3691H C9     RET
```

-22
16 8 4 2 1
0 0 0 1 0 1 1 0
1 1 1 0 1 0 0 1
+1
1 0 1 0

* WHEN 80 BYTES/LINE ARE WRITTEN, HL WILL EXIT POINTING TO THE SCREEN ADDRESS STARTING THE NEXT LINE.

HAND CONTROL MASK TO DELTA (REVISED)

SIMILAR TO LOW-RES SUB#126, REVISED FOR HI-RES**

NO FLOP STATUS PROCESSED

ENTER WITH: B= JOYSTICK MASK

```
          7 6 5 4 3 2 1 0
B = X X X X R L UP
                  └→ DOWN
└→ COULD USE AS A FLOP STATUS FLAG
```

C = POSITIVE ΔX_L *

DE = ↓ ΔX_H (2 BYTES) *

HL = ↓ $\Delta Y_H \Delta Y_L$ *

* EXIT WITH UPDATED (ADJUSTMENT PER HAND CONTROL INPUT) DELTA FOR LOADING INTO CRITTER VECTOR BLOCK (PACKET).

** REFERENCE NUTTING MANUAL, Z80/ROM CODE LISTING, PAGE 27.

```
MKTD 3692H  CD 9936   CALL CONCPL     PROCESS Y MOTION FIRST
            CD AE36   CALL CONC2      PROCESS X MOTION SECOND
            C9        RET
```

SUBROUTINE TO CONDITIONALLY 2'S COMPLEMENT DELTA

```
CONCPL 3699H CB08     RRC B           ] CONTINUE IF "UP"
             3002     JR NC,CONC1     ] OTHERWISE, JUMP TO CHECK "DOWN"
             7D ← DOG LD A,L          ]
             2F       CPL             ] 2'S COMPLEMENT HL FOR "UP" ↑
             6F       LD L,A          ] (REVERSE DELTA)
             7C       LD A,H          ]
       36A0H 2F       CPL             ]
             67       LD HL,A         ]
             23       INC HL          ]
             CB08     RRC B           PUT R AND L IN BITS 1 AND 0.
             C9       RET             NO MORE PROCESSING REQ'D

CONC1        CB08     RRC B           ] CHECK "DOWN" ↓
             D8       RETC            ] IF SET, USE YH YL POSITIVE Δ
DOG →        210000   LD HL,0         IF NOT SET, NO Y MOTION, ZERO HL.
             C9       RET

CONC2 36AEH  CB08     RRC B           ] CONTINUE IF "LEFT"
      36B0H  3002     JR NC,CONC3     ] OTHERWISE, JUMP TO CHECK "RIGHT"
```

(REVISED)

ENTER DE = ΔX_H, C = ΔX_L

$36B2_H$	79	LD A,C	A = ΔX_L	2's COMPLEMENT $\Delta X_H \Delta X_L$ (REVERSE DELTA)
	2F	CPL		
	C6 01	ADD A,1	SET CF IF CARRY	
	4F	LD C,A		
	7B	LD A,E		
	2F	CPL		
	5F	LD E,A		
	7A	LD A,D		
	2F	CPL		
	57	LD D,A		
	D0	RET NC		
	13	INC DE		
	C9	RET		
CONC3 $36C0_H$	CB 08	RRR C		CHECK IF "RIGHT" — USE POSITIVE ΔX, IF "RIGHT"
	D8	RET C		
	AF	XOR A		IF NO "RIGHT", NO MOTION — ZERO $\Delta X = \Delta X_H X_L$
	4F	LD C,A		
	11 0000	LD DE,00		
	C9	RET		

$36C8_H$ COPY DATA BLOCK TO RAM

INITIALIZE THE CRITTER VECTOR BLOCK

IVBLK $36C9_H$	21 5536	LD HL, 3655_H	HL = BLOCK ADR
	11 C0 7F	LD DE, $C07F_H$	DG = "COPY TO" ADR
	01 0F 00	LD BC, $000F_H$	BC = # OF BYTE TO COPY
$36D2_H$	ED B0	LDIR	
$36D4_H$	C9	RET	

PROGRAM INITIALIZATION

```
36D5H  0E14   LD C,14H      ] IF RIGHT MOST COLUMN KEY
       ED78   IN A,(C)      ] IS NOT HELD DOWN AT SYSTEM RESET,
       A7     AND A         ] JUMP TO MAIN PAGE FLIP DEMO
       CA9629 JP Z,PAGO     ] ENTRANCE AT 2996H
```

DISABLE INTERRUPTS

```
IPGM 36DDH  F3       DI
            AF       XOR A
            011808   LD BC,0818H   ] KILL ANY SOUND
IPGM1 36E2H ED79     OUT (C),A     ]
            10FC     DJNZ PGM1     ]
            3E81     LD A,81H      ] ENABLE HI-RES MODE
            D308     OUT (08H),A   ] WITH MULTI-PAGER
            3ECC     LD A,204      ] SET VERTICAL BLANK REG = 204D
            D30A     OUT (0AH),A   ] (HIDE ONE LINE FOR Z80 STACK)
            3EEA     LD A,1110 1010 ] SET HORIZ COLOR BNDRY
                     OUT (09),A    ] 1110 1010
36F0H       D309                   ]   42D BORDER CLR = YELLOW
            21D92C   LD HL,2CD9H   ] SET SCREEN COLORS
            010B08   LD BC,080BH   ] USE EXISTING COLOR TABLE
            EDB3     OTIR          ]
            31BE7F   LD SP,7FBEH   SET STACK POINTER
                                   POINT TO BOTTOM SCREEN LINE
```

MOVE CRITTER WITHIN 3 INTER-CONNECTING SCENES

PAGES 1, 0, 2

CLRT5 2CD9H

		PIXEL
86	YEL	11
73	BRN	10
FC	BLUE	01
00	BLK	00

WRITE (FILL) SIMPLISTIC STATIC GRAPHICS IN 3 PAGES

(SEE DIAGRAM OF P. 103)

```
PGM         0603     LD B,3        B = LOOP CTR = 3 PAGE LOOP
                                   START WITH PAGE 0, END WITH PAGE 2
                                   (SAVE THIS CTR IN THE STACK IN THE PAGE
                                    THAT IS BEING FILLED.)
            AF       XOR A         A = PAGE NUMBER FOR Z80 R/W = 0
                                   START WITH PAGE.
3700H       D374     OUT (74H),A   TV DISPLAY PAGE 0
PGM1        D375     OUT (75H),A   POINT Z80 R/W TO FILL GRAPHICS TO THIS PAGE
                                   Z80 SP = 7FBEH FOR THIS PAGE.
            08       EX AF,AF'     SAVE PAGE NUMBER FOR Z80 R/W IN A'
            C5       PUSH BC       SAVE LOOP CTR IN THIS PAGE'S STACK
```

FILL GRAPHICS COMMON TO ALL 3 PAGES
START WITH PAGE 0, END WITH PAGE 2

PAGE 0 CENTER SCENE
1 LEFT
2 RIGHT

```
            0E55     LD C,BLUE     C = 1010 1010 "ON" COLOR = BLUE
            061B     LD B,27D      B = FILL 27D LINES
370AH       CD7636   CALL LFILLA   FILL SCREEN
```

FILL UPPER SCREEN BLU/BLK ALTERNATE

PAGE 103

```
370D_H  0E 00      LD C, BLK       "ON" COLOR IS BLK      ┐  FILL
        06 96      LD B, 150_D     FILL 150_D LINES       ├  CENTER OF SCREEN
3711_H  CD 7936    CALL LFILLB     FILL WITH ALL BLK      ┘  ALL BLACK
        0E 55      LD C, BLUE      "ON" COLOR IS BLUE     ┐  FILL
        06 1A      LD B, 26_D      FILL 26_D LINES        ├  LOWER SCREEN
        CD 7936    CALL LFILLB     ← PAGE 103             ┘  BLU/BLK ALTERNATE
        C1         POP BC          B=LOOP CTR AGAIN
                                   SP IS BACK AT 7FBE_H
        08         EX AF, AF'      PUT PAGE NUMBER FOR Z80/RW FROM A'
                                   BACK IN A
        C6 11      ADD A, 11_H     INCREMENT PAGE NUMBER TO NEXT PAGE
        10 E1      DJNZ PGM1       LOOP BACK TO FILL NEXT PAGE
```

-31 = 16 8 4 2 1 : 0 0 0 1 1 1 1 1 : 1 1 1 0 0 0 0 0 → F1 → 0001

FILL PAGE 1 WITH "FILL OVER" ON LEFT SIDE
SP IS BACK TO 7FBE_H AFTER ABOVE FILLS

```
3721_H  21 0040    LD HL, 4000_H   HL= LINE "START ADR" TO BEGIN FILL
        0E 55      LD C, BLUE      "ON" COLOR = BLUE
        06 CB      LD B, 203_D     FILL 203_D LINES
        1E 14      LD E, 20_D      FILL 20_D BYTES/LINE
        3E 11      LD A, 11        ┐  POINT Z80 R/W AT PAGE 1
        D3 75      OUT (75_H), A   ┘
        CD 7B36    CALL LFILLC     ← PAGE 103   FILL THE LEFT SIDE OF SCREEN
```

FILL PAGE 2 WITH "FILL OVER" ON RIGHT SIDE
SP = 7FBE_H AGAIN

```
3731_H  21 3C40    LD HL, 403C_H   HL= LINE "START ADR" TO BEGIN FILL
        0E 55      LD C, BLUE      SAME
        06 CB      LD B, 203_D     AS
        1E 14      LD E, 20_D      ABOVE
        3E 22      LD A, 22_H      ┐  POINT Z80 R/W AT PAGE 2
        D3 75      OUT (75_H), A   ┘
373E_H  CD 7B36    CALL LFILLC
                                   ← PAGE 103
```

NOTE: Z80 R/W IS POINTED NOW AT PAGE 2

MOVE CRITTER IN PAGE 0

```
3741H  31 BE 7F     LD SP,7FBEH          SET STACK POINTER
       3E 22        LD A,22H             START WITH Z80 R/W → PAGE2
3746H  D3 75        OUT (75H),A          ┐
       CD C9 36     CALL IVBLK           INITIALIZE VB IN THIS PAGE
       D6 11        SUB 11H              A → PAGE 1
       20 F7        JR NZ, A3746         LOOP BACK FOR PAGE 1
       D3 75        OUT (75H),A          A=0  A → PAGE 0
       CD C9 36     CALL IVBLK           INITIALIZE PAGE 0 VB
3751H  32 D0 7F     LD (TVPAGE),A        (7FD0H)=0=TVPAGE     ┘ INITIALIZE VECTOR BLOCK IN ALL PAGES 0-2

       LOOPED MOVE
               IN A,(10H)               GET JOYSTICK MOVEMENT FROM HAND CONTROL #1
LOOP 3757H  DB 10
       47           LD B,A               B= JOYSTICK MASK
       0E 20        LD C,20              C= POSITIVE ΔXL = 20
       11 00 00     LD DE,0000           DE= POSITIVE ΔXH = 0
       21 20 00     LD HL,0020           HL= POSITIVE ΔYH YL
       CD 92 36     CALL MKTD            PROCESS ANY MOTION REQUEST
3762H  79           LD A,C               ┐ PUT UPDATED ΔXL
       32 C3 7F     LD (7FC3H),A         ┘ IN VECTOR BLOCK
       ED 53 C4 7F  LD (7FC4H),DE        PUT UPDATED ΔXH IN VB
       22 C6 7F     LD (7FC6H),HL        PUT UPDATED ΔYH ΔYL = VB
       DD 21 C0 7F  LD IX,7FC0H          ┐ WRITE
3770H  21 0B 20     LD HL,PAT-4          │ CRITTER
       CD B7 2D     CALL VWRITR          ┘ (PAGE 64)

       3A D0 7F     LD A,(TVPAGE)        ┐
       E6 03        AND 0000 0011        │ POINT HL AT
       21 64 36     LD HL,LIMTO          │ TV PAGE'S
3782H  A7           AND A                │ IF PAGE=0,
       28 07        JR Z,A3786           │ SKIP LOOP CTR.
       47           LD B,A               │ USE LIMTO
3786H  7D           LD A,L               │ B= LOOP CTR
       C6 06        ADD A,6              │ LIMITS TABLE
       6F           LD L,A               │
       10 FA        DJNZ, A3786          │ LOOP BACK
                                         ┘
3789H  DD 21 C0 7F  LD IX,7FC0H
3790H  CD FD 32     CALL MVECT           VECTOR (MOVE) CRITTER, PAGE 84
```

-9 = 0000 1001 → 1111 0110 +1 → 0111

-6 = 0000 0110 → 1111 1001 +1 → 1010

CHECK TO MOVE INTO ANOTHER PAGE

```
3793  3A C9 7F    LD A,(7FC9H)   ] DID CRITTER REACH A X LIMIT?
      CB 5F       BIT 3,A        ] BIT 3 = 1 IN X CHECKS MASK IF SO.
      28 BD       JR Z,LOOP      ] LOOP BACK IF NOT.
                  (-67)
      21 C8 7F    LD HL,7FC8H    ] IF XH (HIGH ORDER BYTE) = 01H,
      7E          LD A,(HL)      ] THIS X LIMIT MUST BE 304D.
      A7          AND A          ] JUMP TO PROCESS THIS LIMIT.
      20 5B       JR NZ,SWTR     ]

X LIMIT POSSIBILITIES NOW ARE 00, 50 OR DFH

37A1H 2B          DEC HL         POINT HL AT XH (L.O. BYTE)  ] IF XH (L.O. BYTE)
      7E          LD A,(HL)                                  ] IS NOT 00H,
      A7          AND A                                      ] THEN LOOP BACK
      20 B1       JR NZ,LOOP     (-79)                       ]

THIS LOWER LIMIT MUST BE X=0
      3A D0 7F    LD A,(TVPAGE)  ] IS THIS TVPAGE POINTING TO PAGE0?
      A7          AND A          ] IF NOT, JUMP TO SWITCH
      20 20       JR NZ,SW2TO    ] PAGE 0 ← PAGE 2

SWITCH PAGE 1 ← PAGE 0 (X=0)
(Z80 R/W IS POINTING TO PAGE 0)
      3A F7 7F    LD A,(7FF7H)   GET Y COORD FROM REG Y IN PAGE 0
SWOT1 47          LD B,A         SAVE Y COORD IN B
37B0H 3E 11       LD A,11H
      21 B7 37    LD HL,SWOT1A   POINT HL AT PROGRAM CONTINUE ADR
      18 2C       JR CONT1       JUMP TO BYTE SAVER ROUTINE
SWOT1A 37B7H D3 75  OUT (75H),A  ] POINT Z80 R/W AT PAGE 0 ] BLANK LAST CRITTER WRITE IN PAGE 0
      CD C3 37    CALL BLKCTR    ] BLANK CRITTER           ]
      3E 11       LD A,11H       ] POINT Z80 R/W AGAIN AT PAGE 1
      D3 75       OUT 75H,A      ]
37C0H C3 57 37    JR LOOP        LOOP BACK TO MOVE CRITTER

BLANK CRITTER (USING XOR WRITE) WITH PATTERN SIZE
ENTER WITH: DE = X COORD (0 OR 304D) OF CRITTER TO BLANK
            (7FF7H) = REG Y = Y COORDINATE
            IN BLANKING PAGE
            POINTING Z80 R/W TO THE BLANKING PAGE
BLKCTR 37C3H 21 0D 20   LD HL,PAT-2   HL = PATTERN ADDRESS - 2 (POINTING AT X SIZE)
      3E 20       LD A,00100000      A = MAGIC REGISTER VALUE
      CD D7 2D    CALL WRITP (XOR)   BLANK (WRITE WITH XOR) THE LAST
                                     CRITTER WRITE WITHIN OLD PAGE
37CBH C9          RET
```

SWITCH PAGE 0 ← PAGE 2 (X=0)

```
SW2TO   37CC_H  3A F7 7F    LD A,(7FF7_H)   GET Y COORD FROM REGY IN PAGE 2
                47          LD B,A          SAVE Y COORD IN B
                AF          XOR A           A=0
        37D1_H  21 D6 37    LD HL,SW2TOA    POINT HL AT PROGRAM CONTINUE ADR
                18 0D       JR CONT1        JUMP TO BYTE SAVER ROUTINE
SW2TOA  37D6_H  3E 22       LD A,22         ] POINT Z80 R/W  ] BLANK LAST
SW2TOB          D3 75       OUT (75_H),A    ] AT PAGE 2      ] CRITTER WRITE
                CD C3 37    CALL BLKCTR     BLANK THE CRITTER ] IN PAGE 2
                AF          XOR A           ] POINT Z80 R/W BACK AT PAGE 0
                D3 75       OUT (75_H),A    ]
        37E0_H  C3 57 37    JP LOOP         LOOP BACK TO MOVE CRITTER
```

CONTINUE 1 PAGE SWITCH (BYTE SAVER TRICK)

```
CONT1           D3 74       OUT (74_H),A    SET TV DISPLAY TO NEW PAGE
                D3 75       OUT (75_H),A    POINT Z/80 R/W (SP IS ALREADY SET UP)
                32 D0 7F    LD (TVPAGE),A   LOAD NEW PAGE NUMBER IN TVPAGE
                11 2F 01    LD DE,304-1     ] SET X_H IN NEW PAGE TO
                ED 53 C7 7F LD (7FC7_H),DE  ] "RIGHT LIMIT-1" (RIGHT SIDE OF THE PAGE)
        37F1_H  78          LD A,B          A=Y COORD
                32 CD 7F    LD (7FCD_H),A   SET Y_H IN NEW PAGE VECTOR BLOCK (SAME AS OLD PAGE EXIT)
                                            NOTE: CRITTER WRITE VWRITR LOADS Y COORD INTO REGY=(7FF7_H).
                AF          XOR A           ] MAKE SURE NEW PAGE
                32 C9 7F    LD (7FC9_H),A   ] X LIMIT ATTAINED BIT IS RESET
                57          LD D,A          ] DE=X COORD 0 FOR CRITTER BLANK IN OLD PAGE
                5F          LD E,A          ]
                E9          JP (HL)         JUMP BACK TO FINISH PAGE SWITCH
                                            ↑ PAGE 1 OR PAGE 0
```

CRITTER AT X UPPER LIMIT (304_D) SWITCH TO PAGE ON THE RIGHT

```
SWTR            3A D0 7F    LD A,(TVPAGE)   A=TVPAGE
                A7          AND A           ] IS CRITTER IN PAGE 0?
        3800_H  28 0E       JR Z,SWOT2      ] JUMP AHEAD IF SO.
```

SWITCH PAGE 1 (X=304_D) → PAGE 0

```
SW1TO           3A F7 7F    LD A,(7FF7_H)   GET Y COORD FROM REGY IN PAGE 1
                47          LD B,A          SAVE Y COORD IN B
                AF          XOR A           A=0
                21 0C 38    LD HL,SW1TOA    POINT HL AT PROGRAM CONTINUE ADR
                18 1C       JR CONT2        JUMP TO BYTE SAVE ROUTINE
SW1TOA  380C_H  3E 11       LD A,11_H
        380E_H  18 C8       JR SW2TOB       JUMP TO FINISH THIS ROUTINE
                                            (ANOTHER BYTE SAVER)
```

-56
32 16 8 4 2 1
0 0 1 1 1 0 0 0
1 1 0 0 0 1 1 1
+1
1 0 0 0

SWITCH PAGE 0 ($X=304_D$) → PAGE 2

```
SWOT2 3810H 3A F7 7F     LD A,(7FF7H)   GET Y COORD FROM REGY IN PAGE 1
            47           LD B,A         SAVE Y COORD IN B
            3E 22        LD A,22H
            21 1B 38     LD HL,SWOT2    POINT HL AT PROGRAM CONTINUE ADDRESS
            18 0D        JR CONT2
WOT2  381B  AF           XOR A          A=0   ] POINT Z80 R/W AT PAGE 2
            D3 75        OUT (75H),A          ]
            CD C3 37     CALL BLKCTR    BLANK THE CRITTER  ] BLANK LAST CRITTER WRITE IN PAGE 0

      3821H 3E 22        LD A,22H       ] POINT Z80 R/W BACK AT PAGE 2
            D3 75        OUT (75H),A    ]
            C3 57 37     JP LOOP        LOOP BACK TO MOVE CRITTER

      CONTINUE 2 IS PAGE SWITCH (BYTE SAVER TRICK)
CONT2       D3 74        OUT (74H),A    SET TV DISPLAY TO NEW PAGE
            D3 75        OUT (75H),A    POINT Z80 R/W TO ↓
            32 D0 7F     LD (TVPAGE),A  LOAD NEW PAGE NUMBER IN TVPAGE
            11 01 00     LD DE,0+1      ] SET XH IN NEW PAGE TO "LEFT LIMIT+1"
      3832H ED 53 C7 7F  LD (7FC7H),A   ] (LEFT SIDE OF PAGE
            78           LD A,B         A=Y COORD
            32 CD 7F     LD (7FCDH),A   SET YH IN NEW PAGE VECTOR BLOCK (SAME AS OLD PAGE EXIT)
            AF           XOR A          ] MAKE SURE NEW PAGE
            32 C9 7F     LD (7FC9H),A   ] X LIMIT ATTAINED BIT IS RESET
            11 2F 01     LD DE,303D     DE=X COORD = 303D
      3841H E9           JP (HL)        JUMP BACK TO FINISH PAGE SWITCH
```

1982 BYTES LEFT FOR HI-RES FISH DEMO

HI-RES FISH DEMO VARIABLES

↑
STACK BEGINS HERE *

* STACK AREA IS ON 2ND SCREEN RAM FROM BOTTOM. FIRST BYTE ON THIS LINE IS $7F20_H$

7F63	GLDF 1	X_H (2 BYTES), Y_H, ΔX_L, ΔY_L, Δ TIMER
7F69	2	
7F6F	3	
7F75	4	
7F7B	5	
7F81	TROPA 1	
7F87	2	
7F8D	3	
7F93	TROPB 1	
7F99	2	
7F9F	3	
7FA5	TROPC 1	
7FAB	2	
7FB1	3	
7FB7	TROPD 1	
7FBD	2	
7FC3	BOTF 1	
7FC9	2	
7FCF 7FD0	"FISH PATTERN ADDRESS-4" TO WRITE (POINTING AT PATTERN'S RELATIVE X)	
7FD1	MR (XORIT)	
7FD2	VECTOR STATUS	
7FD3	TIME BASE	
7FD4	ΔX_L	
7FD5 7FD6	ΔX_H (2 BYTES)	
7FD7	X_L	
7FD8 7FD9	X_H (2 BYTES)	

7FDA	X CHECKS MASK
7FDB	ΔY_L
7FDC	ΔY_H
7FDD	Y_L
7FDE	Y_H
7FDF	Y CHECKS MASK
7FE0	SECS
7FE1	MINS
7FE2	HOURS
7FE3	BIT 7 1PPS, BIT 0 1 KEY PRESSED, RUN NONSTOP / 0 KEY NOT PRESSED
7FE4	VECTOR BLOCK VARIABLES — INDEX TABLE ADDRESS-1 (FOR TYPE OF FISH) ← POINT TO 1 BYTE PRIOR TO THE INDEX TABLE
7FE5	
7FE6	FISH TYPE LIMITS TABLE
7FE7	

7FEB	TMR60
7FEC	
7FED	
7FEE	
7FEF	RANDOMIZE NUMBER
7FF0	
7FF1	
7FF2	

7FF7 REG Y = Y COORDINATE, USED FOR WRITE ROUTINES

(This page intentionally left blank.)

SET HI-RES MLM TEXT LINES TO 12_H (18_D)

7840$_H$ 3E CC → DROP VERT BLNK LINE.
3 BYTES FOR TEXT LINE 19_D D3 0A
C9

INITIAL DEVELOPMENT. MLM SETUP COMMENT.

1ST RAM BYTE IN EACH SCREEN LINE

24_D LINES

7840 ← LINE 180
7890
78E0
7930
7980
79D0
7A20
7A70
7AC0
7B10 ← LINE 189
7B60
7BB0
7C00 ← LINE 192 TOP PIXEL IN TIME DIGITS
7C50
7CA0
7CF0
7D40
7D90
7DE0
7E30
7E80 LINE 201
7ED0 ←
7F20
7F70 ← BOTTOM LEFT SCREEN RAM BYTE

BOTTOM 2 LINES USED FOR VARIABLES, DATA BLOCKS, STACK AREA, ETC.

LINE 203

HI-RES FISH DEMO (STRAIGHT JUMP TEST 2000H C34238)

```
IFDEMO 3842H F3          DI          DISABLE INTERRUPTS
       AF                XORA
       011808            LD BC,0818H  ┐ STOP ALL SOUND
3847H  ED79              OUT (C),A    │
       10FC              DJNZ A3847   ┘
       3E01              LD A,01H     ┐ ENABLE HI-RES MODE
       D308              OUT (08H),A  ┘ WITH NO MULTI-PAGER
       3E CA             LD A,202D    ┐ SET
3851H  D30A              OUT (0AH),A  ┘ VERT BLNK REG = 202D
       31637F            LD SP,7F63H  INITIALIZE STACK POINTER
       21637F            LD HL,7F63H  ┐ CLEAR DATA BASES
       AF                XORA         │ 7F63-7FEEH = 140 BYTES = 8CH
       018C00            LD BC,8CH    │ LEAVE RANDOMIZING 7FEF-7FF2H AS IS
       CD8F29            CALL CSCRN3  ┘
3860H  3E3B              LD A,59D     ┐ TMR60 = 59D
       32EB7F            LD (TMR60),A ┘ (FOR ELAPSED TIMER)
       3E3F              LD A,IPAGE   ┐ OUTPUT PAGE
       ED47              LD I,A       ┘ OF IM2 VECTOR
       3EBA              LD A,LINE1   ┐ OUTPUT LINE 1
       D30D              OUT (0DH),A  ┘ OF IM2 VECTOR
       ED5E              IM2          ┐ GENERATE INITIAL INTERRUPT
       3EC8              LD A,C8H     │ AT 200 SCREEN SCAN LINES
3871H  D30F              OUT (0FH),A  ┘
       FB                EI          ENABLE SCREEN INTERRUPTS
3874H  01203F            LD BC,3F20H  ┐ CLEAR 202 LINES
       CD8B29            CALL CSCRN1  ┘ 202x80 = 16,160 = 3F20H
       CD3E39            CALL CBOT   DETAIL THE SEA BOTTOM
       3EFF              LD A,FFH     ┐ RANDOMIZE
       CDC532            CALL RANGE   │ THE RANDOMIZER
3882H  32F17F            LD (7FF1H),A ┘
3885H  CD0B3C            CALL DTIME  DISPLAY HR:MIN:SEC CLOCK
       3E20              LD A,20H     ┐ SET MR VALUE IN VECTOR BLOCK
       32D17F            LD (7FD1H),A ┘ FOR A XOR WRITE
       3E03  ← SPEED ADJUSTMENT
       32D37F            LD (7FD3H),A ┐ SET TIME BASE
                                      ┘ IN VECTOR BLOCK = 03
3892H  1809              JR A389DH   SKIP THE NOPS
       000000
       000000
389AH  000000            NOP'S
                         ↓
```

INITIALIZE GOLDFISH PARAMETERS

$389D_H$	CD 60 3E	CALL SUGLDF	SET UP GOLDFISH PARAMETERS
$38A0_H$	06 03	LD B,03	B=3=GOLDFISH TO PROCESS
	CD 0A 3F	CALL IFISH	INITIALIZE AND WRITE THE GOLDFISH SAVE THE VECTOR BLOCK VARIABLES

INITIALIZE TROPICAL FISH A

$38A5_H$	CD E9 3D	CALL SUTRPA	
	06 02	LD B,02	
	CD 0A 3F	CALL IFISH	

INITIALIZE TROPICAL FISH B PARAMETERS

	CD 68 3D	CALL SUTRPB	
$38B0_H$	06 02	LD B,02	
	CD 0A 3F	CALL IFISH	

INITIALIZE TROPICAL FISH C PARAMETERS

	CD F0 3C	CALL SUTRPC	
	06 02	LD B,02	
	CD 0A 3F	CALL IFISH	

INITIALIZE TROPICAL FISH D PARAMETERS

	CD 85 3C	CALL SUTRPD	
$38C0_H$	06 02	LD B,02	
	CD 0A 3F	CALL IFISH	

INITIALIZE SEA BOTTOM FISH

	CD F9 3A	CALL SUBOTF	
	06 01	LD B,01	} REVISION NECESSARY FOR B=2 FISH OPTION
	CD 0C 3B	CALL IBFISH	

VECTOR/WRITE 3 GOLDFISH

LOOPB $38CD_H$	CD 60 3E	CALL SUGLDF	SET UP AGAIN SOME GOLDFISH PARAMETERS
$38D0_H$	06 03	LD B,03	B=3=GOLDFISH TO PROCESS (LOOP CTR)
	CD D8 3E	CALL SETVB	SET UP VECTOR BLOCK VECTOR (MOVE A FISH) BLANK LAST FISH WRITE

LOOP BACK HERE

VECTOR/WRITE 2 TROPICAL FISH A

	CD E9 3D	CALL SUTRPA	
	06 02	LD B,02	
	CD D8 3E	CALL SETVB	

VECTOR/WRITE 2 TROPICAL FISH B

	CD 68 3D	CALL SUTRPB	
$38E0_H$	06 02	LD B,02	
$38E2_H$	CD D8 3E	CALL SETVB	

VECTOR/WRITE 2 TROPICAL FISH C

```
38E5H CD F0 3C     CALL SUTRPC
      06 02        LD B,02
      CD D8 3E     CALL SETVB
```

VECTOR/WRITE 2 TROPICAL FISH D

```
      CD 85 3C     CALL SUTRPD
38F0H 06 02        LD B,02
      CD D8 3E     CALL SETVB
```

VECTOR/WRITE SEA BOTTOM FISH

```
38F5H CD F9 3A     CALL SUBOTF
      06 01        LD B,01
      CD 863A      CALL SVBF
```

UPDATE ELAPSED TIME? CHECK FOR 1PPS=1

```
38FDH 21 E3 7F     LD HL,7FE3H   POINT HL AT 1PPS
3900H CB 7E        BIT 7,(HL)    ] IF 1PPS=0,
      28 05        JR Z,A3909    ] YOUR DONE
      CB BE        RES 7,(HL)    ] IF 1PPS=1,
      CD 0B 3C     CALL DTIME    ] UPDATE TIME DISPLAY AND RESET 1PPS
```

CHECK FOR DEMO RUN NONSTOP FLAG SET

```
3909H 21 E3 7F     LD HL,7FE3H   POINT HL AT 1PPS BYTE
      CB 46        BIT 0,(HL)    ] IF BIT0 AT 7FE3H=1,
      20 BD        JR NZ,LOOPB   ] RUN DEMO NONSTOP
```

-67
64 32 16 8 4 2 1
0 1 0 0 0 0 1 1
1 0 1 1 1 1 0 0
+1
1 1 0 1

CHECK FOR A PRESSED KEY

HL IS STILL POINTING AT 7FE3H

```
3910H 06 04        LD B,4        B=INPUT PORT LOOP CTR
      0E 14        LD C,14H      START WITH INPUT PORT 14H
3914H ED 78        IN A,(C)      ] IF A KEY IN THIS COLUMN
      A7           AND A         ] IS NOT PRESSED,
3917H 28 18        JR Z,A3931    ] JMP TO TEST NEXT COLUMN
      CB C6        SET 0,(HL)    ] IF A KEY WAS PRESSED, SET BIT0 AT 7FE3H.
```

WRITE UP ARROW FOR DEMO NONSTOP CONFIRMATION

```
      3E 08        LD A,00001000 ] OUTPUT MAGIC REG
      D3 0C        OUT (MAGIC),A ] EXPAND WITH PROP
      3E 06        LD A,00000110 ] EXPAND WITH
3921H D3 19        OUT (19H),A   ] 0 → 10 LT BRN
                                 ] 1 → 01 YEL
      11 303C      LD DE,3C30H   DE=MAGIC ADR TO WRITE
      21 EE 25     LD HL,25EEH   HL=↑ UP PATTERN
      01 01 09     LD BC,YSIZE XSIZE
      CD 6B 3C     CALL CWXP     WRITE UP ARROW
      18 9C        JR LOOPB      LOOP BACK TO FISH DEMO
```

-100
64 32 16 8 4 2 1
0 1 0 0 1 0 0
1 0 0 1 1 0 1 1
+1
1 1 0 0

```
3931H 0C           INC C
3932H 10 E0        DJNZ A3914
```

-32
32 16 8 4 2 1
0 0 1 0 1 1 1 1
1 1 0 1 0 0 0 0
+1
0 0 0 0

IF A KEY WAS PRESSED, SET BIT0 AT 7FE3H. WRITE "UP" ARROW. LOOP BACK TO FISH DEMO

CHECK FOR AUTO RESTART AT TIME 0:02:00
AT FIRST 02 MINUTES, TIME TO RESTART

3934_H	3A E1 7F	LD A,($7FE1_H$)	A=MINUTES
	FE 02	CP 02	IF MINUTES=02, RESTART DEMOS
	20 92	JR NZ, LOOPB	OTHERWISE, CONTINUE
$393B_H$	C3 00 20	JP 2000_H	WITH FISH DEMO

-110
6432168421
11011110
001 0001
+1
0010

DETAIL SEA BOTTOM

FILL BLOCK 1

7B1E

52 BYTES WIDE

13 LINES HIGH

FILL WITH LIGHT BROWN COLOR 10

DOG

CBOT $393E_H$	11340D	LD DE, YSIZE XSIZE	
3941_H	06 AA	LD B, 10101010	B= FILL DATA, LT BRN, COLOR 10
	21 1E 7B	LD HL, $7B1E_H$	SCREEN ADR TO BEGIN FILL
	DD 21 4D 39	LD IX, CBOT1	
	C3 8022	JP FILL	PAGE 19

FILL BLOCK 2

CBOT1 $394D_H$	AF	XOR A	
	D3 0C	OUT (MAGIC), A	
3950_H	11 6C 3B	LD DE, $3B6C_H$	
	21 A6 3B	LD HL, CBLK2	
	01 02 0C	LD BC, $0C02_H$	
	CD D6 3B	CALL CWRT	

FILL BLOCK 3

	AF	XOR A	
	D3 0C	OUT (MAGIC), A	
	11 A2 3B	LD DE, $3BA2_H$	
3962_H	21 BE 3B	LD HL, CBLK3	
	01 02 0C	LD BC, $0C02_H$	
3968_H	CD D6 3B	CALL CWRT	

FILL BLOCK 4

FILL COLUMNS OF PIXELS FROM BOTTOM TO TOP STARTING WITH THE BOTTOM MAGIC ADDRESS. INDEX A TABLE THAT LISTS THE NUMBER OF PIXELS IN A COLUMN THAT SHOULD BE WRITTEN, USING A MAGIC "OR" WRITE. FIRST WRITE A COLUMN OF PIXELS 3, THEN FOLLOWING WITH COLUMNS OF PIXELS 2, 1 AND 0. REPEAT FOR EVERY BOTTOM MAGIC ADDRESS. THAT WILL BE WRITTEN ONE COLUMN AT A TIME.

THIS TECHNIQUE REQUIRES LESS BYTES COMPARED TO WRITING STATIC GRAPHICS USING MULTIPLE WIDTH X HEIGHT GRAPHIC PATTERNS.

NO MAGIC SHIFTING

FBLK4	$396B_H$	3E10	LD A, MR	A = MAGIC REG VALUE = 0001 0000 (OR WRITE)
		D30C	OUT (MAGIC), A	OUTPUT MR VALUE TO MAGIC REG
		11D03E	LD DE, $3ED0_H$	DE = INITIAL COLUMN MAGIC ADR (FOR SCREEN RAM ADR $7ED0_H$)
	3972_H	21463B	LD HL, B4CPC	POINT HL AT BLOCK4 TABLE OF COLUMN PIXEL COUNTERS
		060C	LD B, BCOL	B = "BYTE COLUMNS TO FILL" COUNTER = $12_D = 0C_H$, SEE HAND DWG
FBK4A		C5	PUSH BC	SAVE THIS COUNTER
		3E03	LD A, 0000 0011	A = INITIAL PIXEL BYTE SET UP. NEXT SUB SHIFTS RIGHT. FOR A PIXEL 3 WRITE 11 00 00 00 (PIXEL3, PIXEL2, PIXEL1, PIXEL0)
		CD2A3B	CALL F4COL	WRITE 4 COLUMNS OF PIXELS WITH COLOR 11. WRITE BOTTOM TO TOP OF SCREEN. USE MAGIC OR WRITE. BEGIN WITH PIXEL 3, THEN WRITE THE COLUMNS OF PIXEL 2, 1 AND 0.
		13	INC DE	POINT DE AT NEXT COL MAGIC ADR
		C1	POP BC	B = "BYTE COLUMNS TO FILL" COUNTER
	$397F_H$	10F6	DJNZ FBK4A	LOOP BACK TO WRITE 4 MORE COLUMNS

-10
8421
2001010
1111 0101
+1
0110

FILL BLOCK 5 (SIMILAR TO ABOVE BLOCK 4)

FBLK5	3981_H	11143F	LD DE, $3F14_H$
		21763B	LD HL, B5CPC
		060C	LD B, BCOL
FBK5A		C5	PUSH BC
		3E03	LD A, 0000 0011
		CD2A3B	CALL F4COL
		13	INC DE
	3990_H	C1	POP BC
		10F6	DJNZ FBK5A

WRITE INDENTATIONS AT TOP CENTER OF SEA BOTTOM

```
3993H 3E 08      LD A, 0000 1000   (EXPAND, PLOP)   USE MAGIC EXPAND WITH A PLOP
      D3 0C      OUT (MAGIC), A
      3E 08      LD A, 0000 10 00  (EXPAND 0 WITH CLR 00 BLUE; EXPAND 1 WITH CLR 10 LT BRN)
      D3 19      OUT (19H), A      OUTPUT A TO EXPAND REG

                                   WRITE INDENTATION 1
      21 6D 3A   LD HL, IDENT1     HL = PATTERN ADR
      11 21 3B   LD DE, 3B21H      DE = MAGIC ADR
      01 02 02   LD BC, YSIZE XSIZE   B=03, C=02
39A1H CD 6B 3C   CALL CWXP

                                   WRITE INDENTATION 2
      21 71 3A   LD HL, IDENT2
      11 29 3B   LD DE, 3B29H
      01 02 02   LD BC, 0202H
39B0H CD 6B 3C   CALL CWXP

                                   WRITE INDENTATION 3
      21 75 3A   LD HL, IDENT3
      11 31 3B   LD DE, 3B31H
      01 02 01   LD BC, C202H
      CD 6B 3C   CALL CWXP

                                   WRITE INDENTATION 4
      21 77 3A   LD HL, IDENT4
39C2H 11 39 3B   LD DE, 3B39H
      01 01 01   LD BC, 0101H
      CD 6B 3C   CALL CWXP

                                   WRITE INDENTATION 5
      21 78 3A   LD HL, IDENT5
      11 3F 3B   LD DE, 3B3FH
      01 02 02   LD BC, 0202
39D1H CD 6B 3C   CALL CWXP

                                   WRITE INDENTATION 6
      21 7C 3A   LD HL, IDENT6
      11 47 3B   LD DE, 3B47H
      01 02 02   LD BC, 0202
39E0H CD 6B 3C   CALL CWXP

      21 80 3A   LD HL, IDENT7
      11 4C 3B   LD DE, 3B46H
      01 02 03   LD BC, 0302H
39ECH CD 6B 3C   CALL CWXP
```

WRITE LEFT STARFISH

```
39EF_H  3E 03      LD A, 0000 0011
                                  └─→ EXPAND 0 WITH CLR 11 DK BRN
                                  └─→ EXPAND 1 WITH CLR 00 BLUE
39F1_H  D3 19      OUT (19_H), A   OUTPUT A TO EXPAND REG
        21 683A    LD HL, STARF
        11 01 3C   LD DE, 3C01_H
        01 01 05   LD BC, 0501
        CD 6B 3C   CALL CWXP
```

WRITE RIGHT STARFISH

```
        3E 07      LD A, 0000 0111   ┐ SET UP
3A01_H  D3 19      OUT (19_H), A     ┘ EXPAND REG
        21 683A
        11 AD 3B   LD DE, 3BAD_H
        01 01 05
        CD 6B 3C
```

ADD WAVES BETWEEN LIGHT AND DARK BROWN SEA BOTTOM

```
3A0F_H  3E 0E      LD A, 0000 1110
                                  └─→ EXPAND 0 WITH CLR 10 LT BRN
                                  └─→ EXPAND 1 WITH CLR 11 DK BRN
3A11_H  D3 19      OUT (19_H), A   OUTPUT A TO EXPAND REG
        21 852D    LD HL, WAVE1    HL = WAVE PATTERN ADR   ┐ WRITE
        11 923E    LD DE, 3E92_H   DE = MAGIC ADR          ├ WAVE 1
        01 01 02   LD BC, 0201     BC = YSIZE XSIZE        ├ @3E92_H
        CD 6B 3C   CALL CWXP       EXPAND WITH PLOP PATTERN┘

        21 FD33    LD HL, WAVE2                        ┐ WRITE
3A22_H  11 973E    LD DE, 3E97_H                       ├ WAVE 2
        01 01 02   LD BC, 0201                         ├ @3E97_H
        CD 6B 3C   CALL CWXP                           ┘

        21 3235    LD HL, WAVE3                        ┐ WRITE
        11 9C3E    LD DE, 3E9C_H                       ├ WAVE 3
3A31_H  01 01 02   LD BC, 0201                         ├ @3E9C_H
3A34_H  CD 6B 3C   CALL CWXP                           ┘
```

2:01
CORK 1/4

```
3A37H  21 5C 3A     LD HL, WAVE4
       11 B2 3E     LD DE, 3EB2H
       01 01 02     LD BC, 0210H
3A40   CD 6B 3C     CALL CWXP
```

WRITE WAVE 4 @ 3EB2H

```
3A43H  21 32 35     LD HL, WAVE3      USE WAVE3 PATTERN
       11 B7 3E     LD DE, 3EB7H
       01 01 02     LD BC, 0201
3A4CH  CD 6B 3C     CALL CWXP
```

WRITE WAVE 5 @ 3EB7

```
3A4FH  21 85 2D     LD HL, WAVE1      USE WAVE1 PATTERN
3A52H  11 BC 3E     LD DE, 3EBCH
       01 01 02     LD BC, 0201
3A58H  CD 6B 3C     CALL CWXP
```

WRITE WAVE 6 @ 3EBCH

```
3A5BH  C9           RET
```

WAVE 4 @ 3EB2H

```
WAVE 4  3A5C  3C    00111100
        3A5D  7E    01111110
        3A5E  FF    UNUSED BYTE
```

WAVE PATTERN BYTES

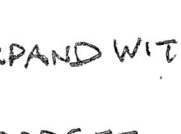
			CODE
PAGE 62A	2D85	01000010 11100111	42 E7
PAGE 89	33FD	01000010 11110111	42 F7
PAGE 94	3532	00100010 01110111	22 77
← THIS PAGE	3A5C	00111100 01111110	

DIRECT JUMP TO HI-RES FISH DEMO FROM RESET 122

```
3A5F_H  DB17      IN A,(COL3)     A = INPUT FROM KEYPAD COLUMN 3
        A7        AND A           ✓RIGHT COL
        CAD536    JP Z,A36D5      IF NO KEY IN LEFT MOST COLUMN IS HELD DOWN AT RESET, JUMP TO CHECK COLUMN 0 AT RESET
                                  PAGE 106
3A65_H  C34238    JP IFDEMO       ← JUMP TO INITIALIZE FISH DEMO IF KEY IN COLUMN 3 IS PRESSED AT RESET
```

STARFISH PATTERN (MAGIC EXPAND WITH PLOP)

```
STARF   3A68_H  10
                54
                38
                28
                44
```



SEA BOTTOM INDENTATIONS (MAGIC EXPAND WITH PLOP)

```
        INDENTATION @ 3B21_H
IDENT1  3A6D_H  01 C0     ← MAGIC RAM ADDRESS
                C3 E9

        INDENTATION @ 3B29_H
IDENT2  3A71_H  00 00
                81 89

        INDENTATION @ 3B31_H
IDENT3  3A75_H  0C 4C

        INDENTATION @ 3B39_H
IDENT4  3A77_H  08

        INDENTATION @ 3B3F_H
IDENT5  3A78_H  00 00
                A5 99

        INDENTATION @ 3B47_H
IDENT6  3A7C_H  00 00
                86 19

        INDENTATION @ 3B4C_H
IDENT7  3A80_H  04 60
                8E 71
        3A84_H  DE 73
```

ROMAN
I

VARIATION OF SUBROUTINE SETVB AT $3ED8_H$

COMMENT: BECAUSE THE SEA BOTTOM FISH MOVES ONLY ALONG A NARROW RANGE OF Y, FROM 180 TO 182_D, AN ADJUSTMENT WAS NECESSARY TO KEEP THE FISH FROM FREQUENTLY MOVING UP AND DOWN. REFER TO SETVB FOR ADDITIONAL Z80 INSTRUCTION COMMENTS NOT INCLUDED BELOW.

SET UP THE VECTOR BLOCK FOR SEA BOTTOM FISH

SVBF $3A86_H$	C5 (DOG)	PUSH BC	
	CD 4C 3F	CALL INDEXB	POINT HL AT THE VB VARIABLES FOR THIS FISH. (POINTS TO IN THIS CASE HL = $7FC3_H$)
Y_H IS LOADED INTO VB →	CD AE 3E	CALL SUVB	LOAD THE 3 VARIABLES X_H (2 BYTES), Y_H AND ΔX_L INTO THE VB. Y_H IS ALSO LOAD INTO ($7FF7_H$) = REGY
	E5	PUSH HL	
	CD A0 3F	CALL FLPCK	CHECK ΔX_L IN VECTOR BLOCK. ADJUST MR VALUE AND ΔX_H IN VB FOR A POSITIVE OR REVERSE DELTA IN ΔX_L
$3A91_H$	E1	POP HL	
Y_L IS LOADED INTO VB. ΔY_H IS ADJUSTED FOR POSITIVE OR REVERSE DELTA →	CD C7 3E	CALL SUVB1	LOAD VARIABLES ΔY_L AND ΔY_H IN VB. ADJUST ΔY_H FOR POSITIVE OR REVERSE Δ.
	E5	PUSH HL	
	D5	PUSH DE	
	VECTOR THE FISH		
	DD 21 D1 7F	LD IX, $7FD1_H$	
HL POINTS AT LIMITS TABLE →	2A E6 7F	LD HL, ($7FE6_H$)	
	CD FD 32	CALL MVECT	
	BLANK LAST FISH WRITE		
$3AA1_H$	76	HALT	HALT Z80 TEMPORARILY TO KEEP "BLANK TIME" AT A MINIMUM
	D1	POP DE	← FISH PATTERN ADR POINTER
	2A CF 7F	LD HL, ($7FCF_H$)	
	3A F7 7F	LD A, ($7FF7_H$)	
	47	LD B, A	
	3A D1 7F	LD A, ($7FD1_H$)	
	CD C7 2D	CALL WRITR	
	WRITE NEW FISH		
$3AB0_H$	E1	POP HL	
	C1	POP BC	SIMILAR TO WFISH @ $3F1E_H$ SEE NEXT PAGE FOR DESCRIPTION BEGIN THE VARIATION
	CD B8 3A	CALL WBOTF	←
	10 CF	DJNZ SVBF	
$3AB7_H$	C9		

CONTINUE AT $3AB8_H$ ON NEXT PAGE

A

SUBROUTINE WFISH (PARTIAL) DESCRIPTION

USES SUB TIMERCK @ $3F75_H$ UPDATE ΔTIMER → DECREMENT ΔTIMER VARIABLE. IF ≠0, YOUR DONE. IF=0, RANDOMIZE A NEW ΔTIMER (0–79_D). LOAD NEW ΔTIMER IN FISH VARIABLES DATA BLOCK.

TIMERCK FALLS INTO DELTXY @ $3F7F_H$ → RANDOMIZE A NEW ±ΔX_L AND A NEW ±ΔY. LOAD THESE NEW VALUES INTO VB

★ MAKE ADJUSTMENT HERE. LOWER ΔY INCREMENT/DECREMENT ie, REDUCE SPEED AT WHICH FISH MOVES UP AND DOWN

WRITE SEA BOTTOM FISH (SIMILAR TO WFISH @ $3F1E_H$)

WBOTF	$3AB8_H$	C5	PUSH BC	
		CD CA 3A	CALL TCKBF	HERE'S THE ADJUSTMENT
	$3ABC_H$	C3 22 3F	JP WFISH1	FINISH THE WFISH SUBROUTINE

SEA BOTTOM FISH ΔY (SPEED ADJUSTMENT, SLOW DOWN Y SPEED)

RANDOMIZE A ΔY_L (FOR SEA BOTTOM FISH, A VARIATION OF RANDELT @ $3FGA_H$)

RDELTY	$3ABF_H$	3E10	LD A, 10_H	← LOWER FROM $7F_H$ TO 10_H
	$3AC1_H$	CD C5 32	CALL RANGE	
		4F	LD C, A	
		1F	RRA	
		79	LD A, C	
		D0	RET NC	
		2F	CPL	
		C9	RET	

SPEED ADJUSTMENT
LOWER Y SPEED

CHECK THE DELTA TIMER (FOR SEA BOTTOM FISH, A VARIATION OF TIMERCK @ $3F75_H$)

TCKBF	$3AC4_H$	35	DEC (HL)	
		C0	RETNZ	
		E5	PUSH HL	
		3E 50	LD A, 80_D	
		CD C5 32	CALL RANGE	
	$3AD2_H$	E1	POP HL	
		77	LD (HL), A	

NOW LOAD RANDOM DELTAS (ΔX_L, ΔY_L) INTO VECTOR BLOCK (FOR SEA BOTTOM FISH)

DXYBF	$3AD4_H$	CD 6A 3F	CALL RANDELT	
		32 D4 7F	LD ($7FD4_H$), A	
		CD BF 3A	CALL RDELTY	← THIS LOWERS Y SPEED
		32 DB 7F	LD ($7FDB_H$), A	
	$3AE0_H$	C9	RET	

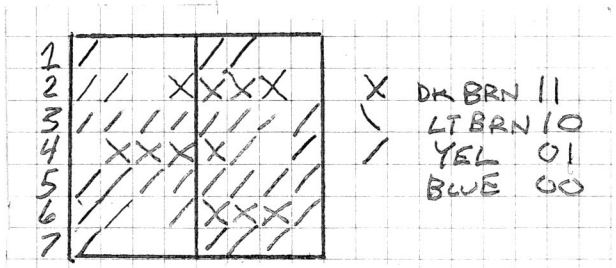
B

SEA BOTTOM FISH PATTERN

PCBOT-4 $3AE1_H$

```
00 RELATIVE X
00      ↓    Y
02 X SIZE (BYTES WIDE)
07 Y ↓    (LINES HIGH
40 50
53 FC
55 55
3F D1
55 55
51 FD   <-DOG
40 54
```

YELLOW WITH DK BRN STRIPES



$3AF1_H$

SEA BOTTOM FISH LIMITS TABLE

CBTFL $3AF3_H$

```
10 00   X LOWER LIMIT = 16D
2A 01   X UPPER LIMIT = 298D
B4      Y LOWER LIMIT = 180D
B6      Y UPPER LIMIT = 182D
```

SET UP SOME SEA BOTTOM FISH PARAMETERS

```
SUBOTF  3AF9  21 E1 3A    LD HL, PCBOT-4      ] SET UP SEA BOT FISH PATTERN ADR-4
              22 CF 7F    LD (7FCF_H), HL     ] (POINTING AT PATTERN REL X)

              21 67 3F    LD HL, INDEX6-1     ] SET UP VECTOR BLOCK VARIABLES
3B02_H        22 E4 7F    LD (7FE4_H), HL     ] INDEX TABLE FOR SEA BOT FISH
                                              ] (POINT HL TO 1 BYTE AHEAD OF TABLE)
              21 F3 3A    LD HL, CBTFL        ] SET UP
              22 E6 7F    LD (7FE6_H), HL     ] SEA BOT FISH LIMITS TABLE
              C9          RET
```

INITIALIZE 4 VECTOR BLOCK VARIABLES X_H (LOW ORDER), Y_H, ΔX_L, ΔY_L FOR THIS FISH TYPE.
SIMILAR TO IFISH @ $3F0A_H$ ON PAGE ?

```
IBFISH  3B0C_H  C5          PUSH BC          SAVE THE FISH NUMBER (LOOP COUNTER)
THIS                3E A0   LD A, 160D     ] INITIALIZE
INITIALIZATION      32 D8 7F LD (7FD8_H), A ] X_H (LOW ORDER) IN VECTOR BLOCK
IS FOR      3B12_H  3E B4   LD A, 180D      ] INITIALIZE Y_H IN VECTOR BLOCK
ONLY                32 DE 7F LD (7FDE_H), A ]
1 FISH              CD 7F 3F CALL DELTXY     RANDOMIZE ΔX_L AND ΔY_L -
                                             PUT THEM IN VB
                    C1       POP BC          B = FISH NUMBER AGAIN.
                    CD 4C 3F CALL INDEXB     INDEX ADR OF VB VARIABLES FOR FISH
                    11 05 00 LD DE, 5        ] POINT HL AT ΔTIMER FOR THIS FISH
              3B21_H 19      ADD HL, DE      ]
                    36 80    LD (HL), 80     SET ΔTIMER = 148D FOR THIS FISH
                    CD 1E 3F CALL WFISH      WRITE THE FISH
NO LOOP BACK FOR JUST 1 FISH → 00 00 DJNZ IBFISH  NOT REQ'D (YOU ARE ONLY MOVING 1 FISH)
              3B29_H C9      RET
```

C

FILL 4 PIXEL COLUMNS

THIS SUBROUTINE FILLS 4 COLUMNS OF PIXELS FROM THE BOTTOM OF SCREEN RAM TO TOP BEGINNING AT A SPECIFIC RAM ADDRESS. ONE COLUMN IS WRITTEN AT A TIME EMPHACIZING FIRST PIXEL 3 LOCATIONS, THEN PIXEL 2, 1 AND 0 LOCATIONS.

ENTER WITH: A = "PIXEL BYTE TO WRITE" SETUP
DE = INITIAL COLUMN MAGIC ADDRESS
HL = "COLUMN PIXEL COUNTER" TABLE ADDRESS (SEE PAGE 127)

F4COL	$3B2A_H$	06 04	LD B, 4	B = COLUMN CTR — SAVE COLUMN PIXEL CTR FOR PIXELS 3, 2, 1 AND 0
F4CL1		C5	PUSH BC	
		CB 0F	RRCA	7 → 0 → CF
	$3B2F_H$	CB 0F	RRCA	A = PIXEL BYTE TO WRITE IN COLUMN SHIFT RIGHT FOR EMPACIZED PIXEL FOR NEXT COLUMN FILL
	$3B31_H$	D5	PUSH DE	SAVE THE COLUMN MAGIC ADR
		E5	PUSH HL	SAVE THE COL PIXEL CTR TABLE ADR
		46	LD B, (HL)	GET COL PIXEL CTR, PUT IT IN B
F4CL2		C5	PUSH BC	SAVE THIS CTR
		12	LD (DE), A	WRITE THE PIXEL USING A MAGIC "OR"
		EB	EX DE, HL	HL = COLUMN MAGIC ADR
		01 B0 FF	LD BC, $FFB0_H$	DE = COL PIXEL CTR TABLE ADR BC = -80_D HI-RES = 80 BYTES/LINE
		09	ADD HL, BC	POINT HL TO NEXT ABOVE MAGIC ADR IN THE COLUMN
		EB	EX DE, HL	DE = NEXT MAGIC ADR IN COLUMN HL = COL PIXEL CTR TABLE ADR
		C1	POP BC	B = COLUMN PIXEL CTR
		10 F5	DJNZ F4CL2	LOOP BACK TO WRITE NEXT PIXEL IN COLUMN (WRITES ARE BOT TO TOP)
		E1	POP HL	HL = COL PIXEL COUNTER TABLE ADR
	$3B40_H$	23	INC HL	HL POINTS AT PIXEL CTR TABLE ADR FOR THE NEXT COLUMN
		D1	POP DE	DE = INITIAL COLUMN MAGIC ADR
		C1	POP BC	B = THE COLUMN PIXEL CTR FOR PIXELS 3, 2, 1, 0
		10 E7	DJNZ F4CL1	LOOP BACK TO FILL NEXT PIXEL COLUMN
	$3B45_H$	C9	RET	

-11
128 64 32 16 8 4 2 1
0 0 0 1 0 1 1
1 1 1 1 0 1 0 0
+1
0 1 0 1

-25
128 64 32 16 8 4 2 1
0 0 0 1 1 0 0 1
1 1 1 0 0 1 1 0
+1
0 1 1 1

EXIT WITH: HL = TABLE ADR OF NEXT COLUMN PIXEL CTR
DE = INITIAL (BOTTOM) COLUMN MAGIC ADR FOR LAST 4 COLUMN FILL.

D

BLOCK 4 COLUMN PIXEL COUNTERS

	COL CNTRS	FOR MAGIC RAM
34CPC $3B46_H$	16 15 14 13	3EDO
	13 13 13 13	1
$3B4E_H$	12 11 10 0F	2
$3B52_H$	0F 0F 0F 0E	3
	0D 0C 0B 0A	4
	0A 0A 0A 09	5
$3B5E_H$	08 07 06 05	6
$3B62_H$	05 05 05 06	7
	07 08 09 09	8
	08 07 06 05	9
$3B6E_H$	04 03 03 03	A
$3B72_H$	03 03 03 03	B

BLOCK 5 COLUMN PIXEL COUNTERS

	COL CNTRS	FOR MAGIC RAM
35CPC $3B76_H$	04 04 03 02	3F14
	02 03 03 04	5
$3B7E_H$	04 05 06 06	6
$3B82_H$	06 07 08 09	7
	0A 0D 0E 0F	8
	0F 0F 0E 0D	9
$3B8E_H$	0C 0B 0A 0A	A
$3B92_H$	0A 0B 0C 0D	B
	0D 0D 0E 0F	C
	10 11 12 13	D
$3B9E_H$	12 11 11 11	E
$3BA2_H$	11 14 15 16	F

SEA BOTTOM BLOCK 2 PATTERN (INITIAL MAGIC ADR $3B6C_H$) /128

CBLK2 $3BA6_H$	00	0A	LINE 1
	00	AA	2
	08	2A	3
	28	2A	4
	A8	2A	5
$3BB0_H$	2A	AA	6
	0A	AA	7
	02	AA	8
	02	AA	9
	FA	AA	10
	EA	AA	11
	AA	AA	12

SEA BOTTOM BLOCK 3 PATTERN (INITIAL MAGIC ADR $3BA2_H$)

CBLK3 $3BBE_H$	80	28	LINE 1
$3BC0_H$	AA	AA	2
	AA	AA	3
	AA	A8	4
	AA	A0	5
	AA	80	6
	AA	A0	7
	AA	A8	8
	AA	AB	9
$3BD0_H$	AA	BF	10
	AF	FF	11
$3BD4_H$	FF	FF	12

F

CUSTOM APPLICATION MAGIC WRITE APPLICATION

SIMILAR TO LOW-RES SUB #38, NUTING MANUAL ROM LISTING PAGE 50.

NO CLEAR SHIFTER BYTE AT END OF EACH PATTERN LINE WRITTEN.

NORMAL WRITE WITH MAGIC PLOP, OR, XOR

ENTER WITH: DE = MAGIC ADDRESS TO BEGIN WRITE
HL = PATTERN ADDRESS
BC = YSIZE XSIZE YSIZE = PATTERN LINE HIGH
XSIZE = PATTERN BYTE WIDE

OUT TO MAGIC REGISTER (PORT α_H) 00xx 0000
↳ 00 PLOP
01 OR
10 XOR

CWRT	$3BD6_H$	AF	XOR A	
CWRT1		C5	PUSH BC	SAVE YSIZE XSIZE
		D5	PUSH DE	SAVE MAGIC ADDRESS FOR LINE TO WRITE
		47	LD B,A	B=0
		EDB0	LDIR	WRITE A LINE
		D1	POP DE	
		EB	EX DE,HL	DE = PATTERN ADR HL = MAGIC ADR
		0E 50	LD C, 50_H	HI-RES = 80 BYTES/LINE
	$3BE0_H$	09	ADD HL,BC	HL = MAGIC ADR OF NEXT LINE
		EB	EX DE,HL	DE = MAGIC ADR OF NEXT LINE HL = PATTERN LINE TO WRITE
		C1	POP BC	BC = YSIZE XSIZE
		10 F2	DJNZ CWRT1	
	$3BE5$	C9	RET	

-14
64 32 16 8 4 2 1
0 0 0 0 1 1 1 0
1 1 1 1 0 0 0 1
+1
0 0 1 0

G

ELAPSED TIME HANDLER

```
FTIMER  3BE6H  21 EB 7F    LD HL,TMRG0          ] DECREMENT
               35           DEC (HL)             ] TMRG0
               C0           RET NZ               YOU'RE DONE IF TMRG0 ≠ 0
               36 3B        LD (HL),59D          TMRG0=59D
               21 E3 7F     LD HL,7FE3H          ] SET 1PPS
       3BF0H   CB FE        SET 7,(HL)           ] (BIT 7 AT 7FE3H)
               06 02        LD B,2               B=LOOP CTR
               21 E0 7F     LD HL,SECS           POINT HL AT SECS
ETIMER1        7E           LD A,(HL)            GET SECS (OR MINS)
               C6 01        ADD A,1              INCREMENT TIME
               27           DAA                  ADJUST BCD FOR
                                                 CARRY OVER TO MSD
               FE 60        CP 60H               ] IF <60, YOUR DONE.
               20 0A        JR NZ,ETIMER2        ] NO FURTHER UPDATE REQ'D
               3E 00        LD A,0               ]
       3C01H   77           LD (HL),A            ]
               23           INC HL               POINT HL AT MINS (OR HRS)
               10 F2        DJNZ ETIMER1         LOOP BACK TO UPDATE MINS
               7E           LD A,(HL)            A=HOURS   ] UPDATE HOURS
               C6 01        ADD A,1                        ]
               27           DAA                            ]
ETIMER2        77           LD (HL),A            SAVE THE UPDATE
       3C0AH   C9           RET
```

DISPLAY THE TIME

7FE0 SECS
7FE1 MINS
7FE2 HRS 0-9 AS LSD IN BITS 3-0
7FE3 BIT 7 ONLY, 1PPS SET BY SCREEN INTERRUPT INTR2 $3FE2_H$ ↓
7FEB TMR60 DECREMENTED

CALCULATE MAGIC ADDRESS FOR HOURS DIGIT SCREEN FRAME
COORDINATES OF THIS FRAME ARE $X=132_D(84_H)$, $Y=192_D(C0_H)$
BITS 1,0=00; NO PIXEL SHIFT REQ'D →
ENTER RELTA1 WITH: HL = IRRELEVANT WITH THIS CUSTOM WRITE

```
DTIME  3C0B_H  11 84 00     LD DE,132_D      DE = X COORD
       3C0E_H  3E C0        LD A,192_D       } (7FF7_H) = REGY = Y COORD
               32 F7 7F     LD (7FF7_H),A    }
               3E 08        LD A,MR          A = MAGIC REG VALUE = 0000 1000
                                                  PLOP ← 2 → EXPAND
               CD 002C      CALL RELTA1      EXIT DE = MAGIC ADR FOR WRITE FRAME
                                                  MR VALUE OUTPUT TO MAGIC REG
                            PAGE 55 →
```

SET UP EXPAND REGISTER

```
3C18_H  3E 06        LD A,0000 01 10
                                  → EXPAND 0 TO 10, LIGHT BROWN
                                  → EXPAND 1 TO 10, YELLOW
        D319         OUT (19_H),A
```

DISPLAY HOURS

```
        3A E2 7F     LD A,(7FE2_H)    A = HRS DIGIT, LSD = 0 TO 9
        CD 4E 3C     CALL DRDGT       DISPLAY HOURS
```

DISPLAY COLON (:)

```
3C22_H  3E 0A        LD A,0A_H        A = COLON INDEX = 0A_H
        CD 4E 3C     CALL DRDGT       DISPLAY RIGHT DIGIT, LSD ONLY
```

DISPLAY MINUTES

```
        3A E1 7F     LD A,(7FE1_H)
        CD 39 3C     CALL DDGTS
```

DISPLAY COLON (:)

```
        3E 0A        LD A,0A_H
        CD 4E 3C     CALL DRDGT
```

DISPLAY SECONDS

```
3C32_H  3A E0 7F     LD A,(7FE0_H)
        CD 39 3C     CALL DDGTS

3C38_H  C9           RET
```

I

DISPLAY DIGIT(S) IN REG A 7654 (MSD) 3210 (LSD) DIGIT = 0 TO 9

DDGTS DISPLAY BOTH DIGITS, MSD THEN LSD
DRDGT DISPLAY RIGHT DIGIT (LSD) ONLY
ENTER WITH: A = DIGIT(S) TO DISPLAY
DE = MAGIC ADDRESS TO WRITE TO FOR DIGIT FRAME ON SCREEN

```
DDGTS  3C39H  D5          PUSH DE      SAVE THIS MAGIC ADR FOR NEXT FRAME
              F5          PUSH AF      SAVE LSD PRESENT IN BITS 3-0
              E6 F0       AND 1111 0000   ISOLATE MSD IN REG A
              0F          RRCA  ↳7→0→CF
              0F          RRCA          ROTATE MSD RIGHT
              0F          RRCA          INTO BITS 3-0
       3C40H  0F          RRCA
              CD 5E 3C    CALL PDCHR   POINT HL AT DIGIT PATTERN TO DISPLAY
              01 01 09    LD BC,0901H  B=YSIZE=9, C=XSIZE=1
              CD 6B 3C    CALL CWXP    WRITE THE MSD
                                       A=LSD IN BITS 3-0
              F1          POP AF
              D1          POP DE       DE = MAGIC ADDRESS OF NEXT
              13          INC DE           DIGIT FRAME TO DISPLAY
              13          INC DE
DRDGT  3C4EH  D5          PUSH DE      SAVE THIS MAGIC ADDRESS FOR
                                       NEXT DIGIT DISPLAY FRAME
              E6 0F       AND 0000 1111   A=LSD ONLY
       3C51H  CD 5E 3C    CALL PDCHR   POINT HL AT DIGIT PATTERN TO DISPLAY
              01 01 09    LD BC,0901H  B=YSIZE=9, C=XSIZE=1
              CD 6B 3C    CALL CWXP    WRITE THE LSD
              D1          POP DE       DE = MAGIC ADDRESS OF NEXT
              13          INC DE           DIGIT FRAME TO DISPLAY
              13          INC DE
              C9          RET
```

POINT TO DIGIT CHARACTER PATTERN IN TABLE
(TO SELECT 0 THRU 9 OR A FOR THE COLON :)
ENTER WITH: A = CHAR INDEX = 0 THRU 9 OR A

```
PDCHR  3C5EH  21 67 25    LD HL,CHART   POINT HL AT CHAR TABLE (PAGE 33)
       3C61H  A7          AND A         IF INDEX=0
              C8          RET Z             POINTER ADJUSTMENT IS NOT REQ'D
              47          LD B,A        B=CHAR INDEX=LOOP CTR
              7D          LD A,L
PDCHR1        C6 09       ADD A,9       ADJUST LOW ORDER BYTE IN HL.
              10 FC       DJNZ PDCHR1   POINT HL TO INDEXED CHAR PATTERN
              6F          LD L,A
       3C6AH  C9          RET
```

-4
0000 0100
1111 1011
FB
1100

CUSTOM WRITE WITH EXPAND (SIMILAR TO NUTTING MANUAL Z80 LISTING, PAGE 50)

WRITE WITH NO PIXEL SHIFT AND NO CLEAR SHIFTER BYTE AT THE END OF EACH LINE WRITTEN

ENTER WITH: HL = PATTERN TO EXPAND
DE = MAGIC ADDRESS TO WRITE TO
BC = YSIZE, XSIZE

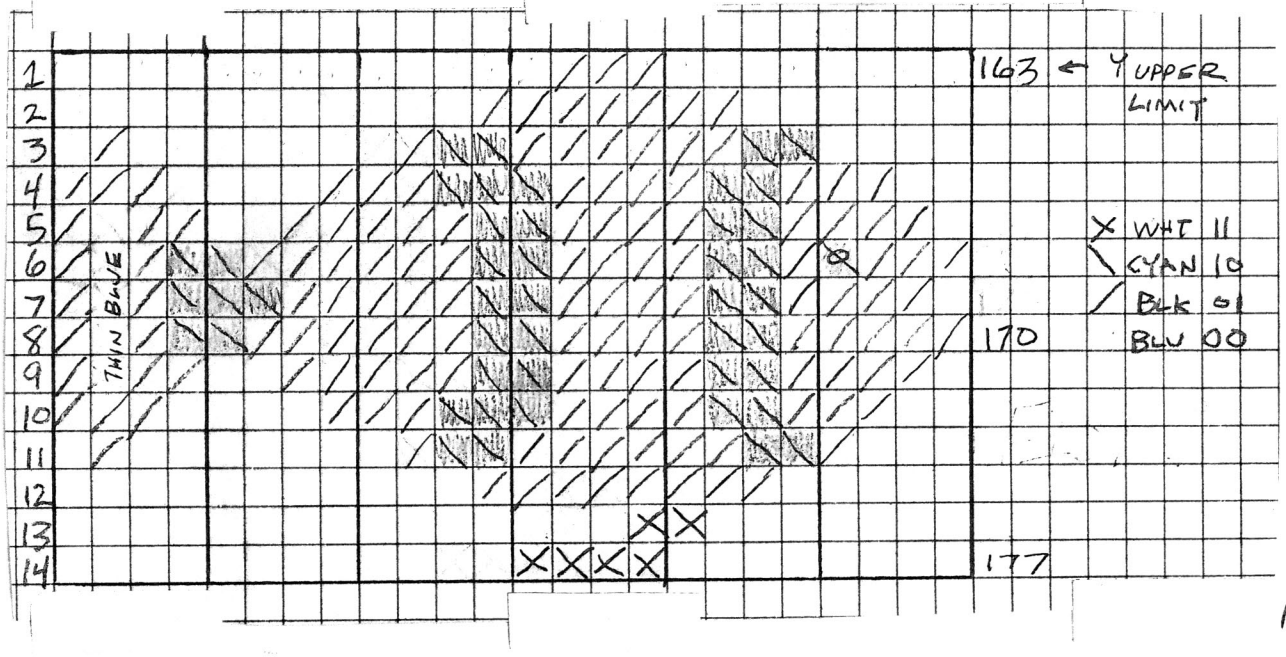
CWXP	$3C6B_H$	EB	EX DE,HL	DE=PATTERN ADR, HL=MAGIC ADR
CWXP1		C5	PUSH BC	SAVE YSIZE, XSIZE
		E5	PUSH HL	SAVE MAGIC ADR
		41	LD B,C	B=XSIZE
CWXP2		1A	LD A,(DE)	A=PATTERN BYTE
	$3C70_H$	13	INC DE	POINT DE AT NEXT PATTERN BYTE
		77	LD (HL),A	WRITE 1ST EXPAND BYTE
		23	INC HL	POINT TO NEXT MAGIC ADR
		77	LD (HL),A	WRITE THE OTHER HALF OF EXPANDED PATTERN
		23	INC HL	POINT TO NEXT MAGIC ADR
		10F8	DJNZ CWXP2	LOOP BACK TO FINISH LINE?
		E1	POP HL	HL=MAGIC ADR OF PREVIOUS LINE
		0E 50	LD C,50_H	HI-RES = 80 BYTES/LINE
		09	ADD HL,BC	B=0 VIA DJNZ
		C1	POP BC	DE=MAGIC ADR FOR NEXT LINE
		10 EE	DJNZ CWXP1	BC=ORIGINAL YSIZE, XSIZE
	$3C7E_H$	C9	RET	

-8
0000 1000
1111 0111
+1
1000

-18
0001 0010
1110 1101
+1
1110

TROPICAL FISH D LIMITS TABLE

TRPDL	$3C7F_H$	0000	X LOWER LIMIT
		2801	X UPPER LIMIT = 296_D
		00	Y LOWER LIMIT
	$3C84_H$	A3	Y UPPER LIMIT = 163_D



FISH D

K

REVISED

SET UP SOME TROPICAL FISH D PARAMETERS

```
SUTRPD 3C85H 21983C    LD HL,PTRPD-4     ] SET UP TROPICAL FISH D
       22CF7F          LD (7FCFH),HL     ] PATTERN ADDRESS-4
                                           (POINTING AT PATTERN RELATIVE X

       21653F          LD HL,INDEX5-1    ] SET UP VECTOR BLOCK VARIABLES
       22E47F          LD (7FE4H),HL     ] INDEX TABLE FOR TROP FISH D
3C91H  217F3C          LD HL,TRPDL       ] SET UP
       22E67F          LD (7FE6H),HL     ] TROPICAL FISH D LIMITS TABLE
       C9              RET
```

TROPICAL FISH D PATTERN

```
PTRPD-4 3C98H 00   RELATIVE X
              00      ↓   Y
              06   XSIZE (BYTES WIDE)
              0E   YSIZE (14 LINES HIGH

PTRPD  3C9CH  00 00 00 15 00 00   LINE 1
       3CA2H  00 00 01 55 50 00        2
              10 00 1A 55 5A 00        3
              54 01 5A 95 69 50        4
       3CB4H  45 05 56 95 69 54        5
              46 95 56 95 69 95        6
       3CC0H  46 A5 56 95 69 54        7
              46 95 56 95 69 55        8
       3CCCH  45 05 56 95 69 54        9
       3CD2H  54 01 5A 95 69 50        10
              10 00 1A 55 5A 40        11
       3CDEH  00 00 01 55 54 00        12
       3CE4H  00 00 00 03 C0 00        13
              00 00 00 FF 00 00        14
```

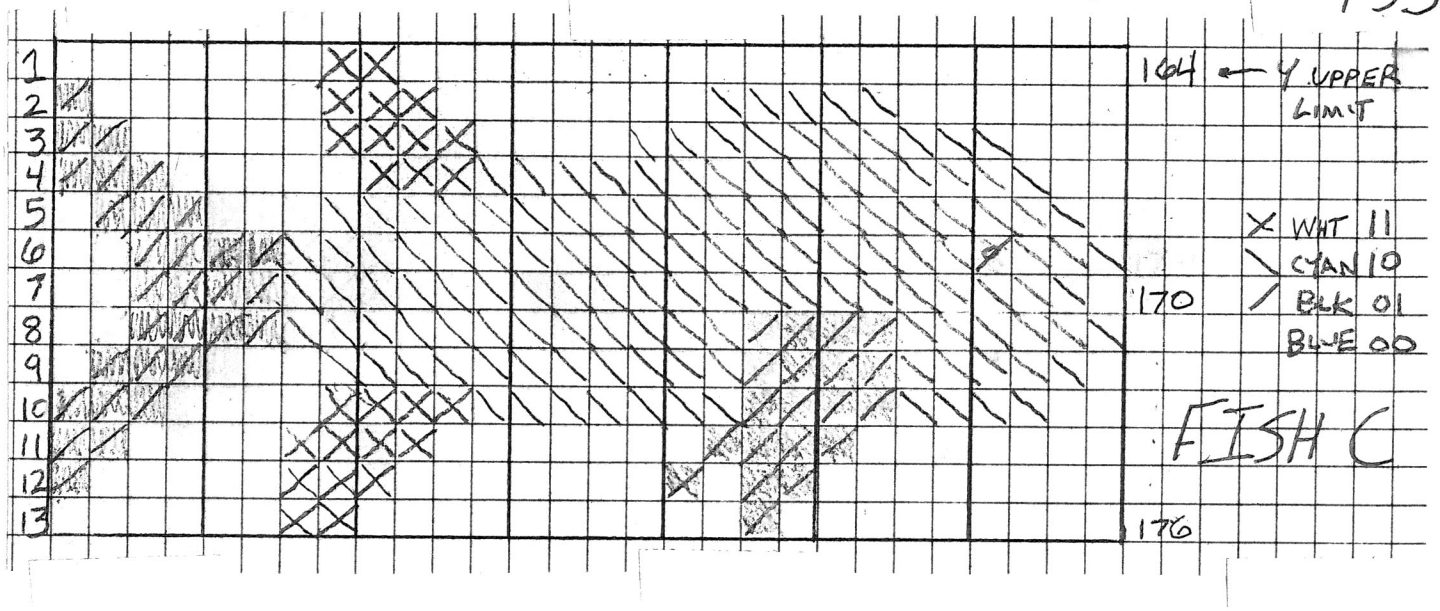
SET UP SOME TROPICAL FISH C PARAMETERS

```
SUTRPC 3CF0H 21033D    LD HL,PTRPC-4     ] SET UP TROPICAL FISH C
       22CF7F          LD (7FCFH),HL     ] PATTERN ADDRESS-4
                                           (POINTING AT PATTERN RELATIVE X)

       21623F          LD HL,INDEX4-1    ] SET UP VECTOR BLOCK VARIABLES
       22E47F          LD (7FE4H),HL     ] INDEX TABLE FOR TROP FISH C

       21623D          LD HL,TRPCL       ] SET UP
3CFF   22E67F          LD (7FE6H),HL     ] TROPICAL FISH C
                                           LIMITS TABLE
3D02H  C9              RET
```

L REVISED



TROPICAL FISH C PATTERN

PTRPC-4	$3D03_H$	00	RELATIVE X							
		00	↓ Y							
		07	X SIZE (BYTES WIDE)							
		0D	Y SIZE (13 LINES HIGH)							
PTRPC	$3D07_H$	00	03	C0	00	00	00	00	LINE	1
	$3D0E_H$	40	03	F0	00	2A	A0	00		2
	$3D15_H$	50	03	FC	02	AA	AA	80		3
	$3D1C_H$	54	00	FE	AA	AA	AA	A0		4
	$3D23_H$	15	02	AA	AA	AA	AA	A8		5
	$3D2A_H$	05	5A	AA	AA	AA	AA	6A		6
	$3D31_H$	05	5A	AA	AA	AA	AA	A8		7
		05	5A	AA	AA	A5	5A	AA		8
	$3D3F_H$	15	02	AA	AA	A5	5A	A8		9
	$3D46_H$	54	03	FE	AA	A5	5A	A0		10
		50	0F	F0	00	15	40	00		11
	$3D54_H$	40	0F	C0	00	45	00	00		12
	$3D5B_H$	00	0F	00	00	04	00	00		13

TROPICAL FISH C LIMITS TABLE

TRPCL	$3D62_H$	00 00	X LOWER LIMIT
		24 01	X UPPER LIMIT = 292_D
		00	Y LOWER LIMIT
	$3D67_H$	A4	Y UPPER LIMIT = 164_D

M
REVISED

SET UP SOME TROPICAL FISH B PARAMETERS

```
SUTRPB 3D68H 217B3D    LD HL,PTRPB-4   ] SET UP TROPICAL FISH B
       22CF7F          LD(7FCFH),HL    ]  PATTERN ADDRESS-4
                                        (POINTING AT PATTERN RELATIVE X)
       215F3F          LD HL,INDEX3-1  ] SET UP VECTOR BLOCK VARIABLES
3D71H  22E47F          LD(7FE4H),HL    ]  INDEX TABLE FOR TROP FISH B
       21E33D          LD HL,TRPBL     ] SET UP
       22E67F          LD(7FE6H),HL    ]  TROPICAL FISH B LIMITS TABLE
3D7AH  C9              RET
```

TROPICAL FISH B PATTERN

```
PTRPB-4 3D7BH 00   RELATIVE X
              00      ↓   Y
              05   X SIZE (BYTES WIDE)
              14   Y   ↓   (LINES HIGH)
PTRPB 3D7FH 00 00 40 00 00   LINE 1
      3D84H 00 00 14 00 00        2
            00 00 15 00 00        3
            80 00 05 40 00        4
      3D93H A8 00 07 50 00        5
            2A 00 05 D4 00        6
            0A 80 05 D5 00        7
      3DA2H 02 90 15 D5 40        8
            02 95 55 D5 D0        9
            00 95 55 D7 54       10
      3DB1H 00 95 57 57 65       11
            00 95 55 D7 54       12
            02 95 55 D7 55       13
            02 90 15 D5 D4       14
      3DC0H 0A 80 05 D5 50       15
            2A 00 07 50 80       16
      3DCFH A8 00 05 40 80       17
      3DD4H 80 00 15 00 80       18
            00 00 14 02 00       19
      3DDEH 00 00 40 08 00       20
```

N
REVISED

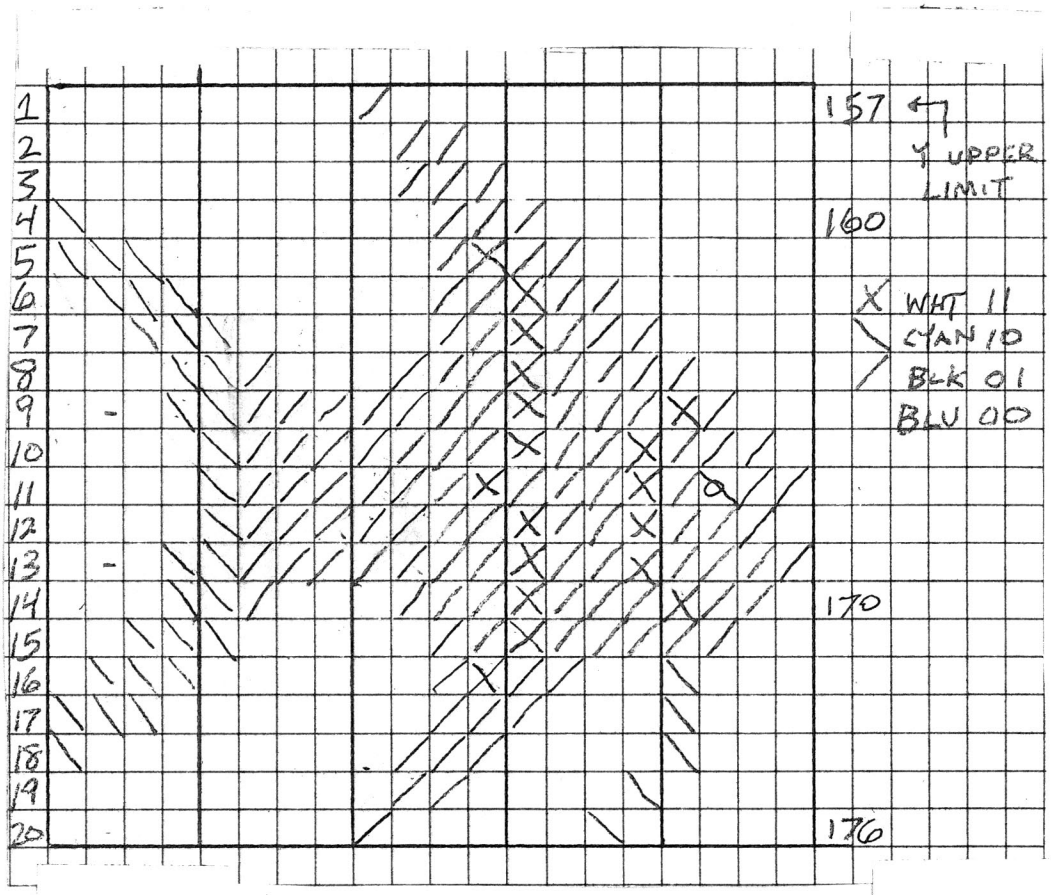
TROPICAL FISH B LIMITS TABLE

TROPBL $3DE3_H$ 00 00 X LOWER LIMIT

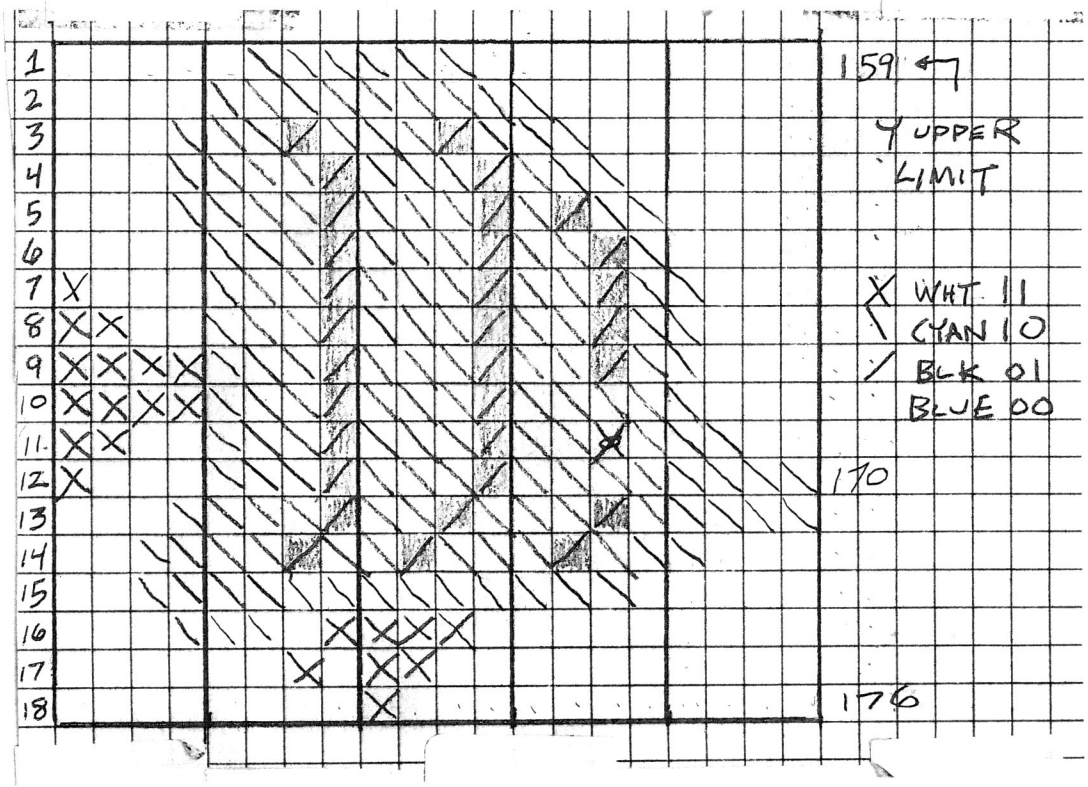
2C 01 X UPPER LIMIT = 300_D

00 Y LOWER LIMIT

$3DE8_H$ 9D Y UPPER LIMIT = 157_D



TROPICAL FISH B



TROPICAL FISH A

REVISED

SET UP SOME TROPICAL FISH A PARAMETERS

SUTRPA	$3DE9_H$	21 FC 3D	LD HL, PTRPA-4	SET UP TROPICAL FISH A PATTERN ADDRESS -4 (POINTING AT PATTERN RELATIVE X)
		22 CF 7F	LD ($7FCF_H$), HL	
		21 5C 3F	LD HL, INDEX2-1	SET UP VECTOR BLOCK VARIABLES INDEX TABLE FOR TROP FISH A POINT HL TO 1 BYTE AHEAD OF TABLE
	$3DF2_H$	22 E4 7F	LD ($7FE4_H$), HL	
		21 5A 3E	LD HL, TRPAL	SET UP TROPICAL FISH A LIMITS TABLE
		22 E6 7F	LD ($7FE6_H$), HL	
		C9	RET	

TROPICAL FISH A PATTERN

TRPA-4	$3DFC_H$	00	RELATIVE X
		00	↓ Y
		05	X SIZE (BYTES WIDE)
		12	Y ↓ (LINES HIGH)
TRPA	$3E00_H$	00 2A AA 00 00	LINE 1
		00 AA AA 80 00	2
		02 A6 A6 A0 00	3
		02 A9 A9 A8 00	4
	$3E14_H$	02 A9 A9 9A 00	5
		00 A9 A9 A6 00	6
		C0 A9 A9 A6 80	7
	$3E23_H$	F0 A9 A9 A6 80	8
		FF A9 A9 A6 80	9
		FF A9 A9 AA 80	10
	$3E32_H$	F0 A9 A9 AE A0	11
		C0 A9 A9 AA AA	12
		02 A9 A6 A6 AA	13
	$3E41_H$	0A A6 9A 9A 80	14
		0A AA AA A8 00	15
		02 A3 FC 00 00	16
	$3E50_H$	00 0C F0 00 00	17
	$3E55_H$	00 00 C0 00 00	18

P

REVISED

TROPICAL FISH A LIMITS TABLE

```
RPAL 3E5A_H  00 00   X LOWER LIMIT
             2C 01   X UPPER LIMIT = 300_D
             00      Y LOWER LIMIT
             9F      Y UPPER LIMIT = 159_D   ← REVISE
```

SET UP SOME GOLDFISH PARAMETERS

```
UGLDF 3E60_H  21 79 3E   LD HL, PGLDF-4      ] SET UP MINNOW PATTERN ADDRESS-4
              22 CF 7F   LD (7FCF_H), HL     ] (POINTING AT PATTERN RELATIVE X)

              21 57 3F   LD HL, INDEX1-1     ] SET UP VECTOR BLOCK VARIABLES
              22 E4 7F   LD (7FE4_H), HL     ] INDEX TABLE FOR GOLDFISH
                                             ] (POINT HL TO 1 BYTE AHEAD OF TABLE)

              21 73 3E   LD HL, GLDFL        ] SET UP
              22 E6 7F   LD (7FE6_H), HL     ] GOLDFISH LIMITS TABLE

       3E72_H C9         RET
```

GOLDFISH LIMITS TABLE

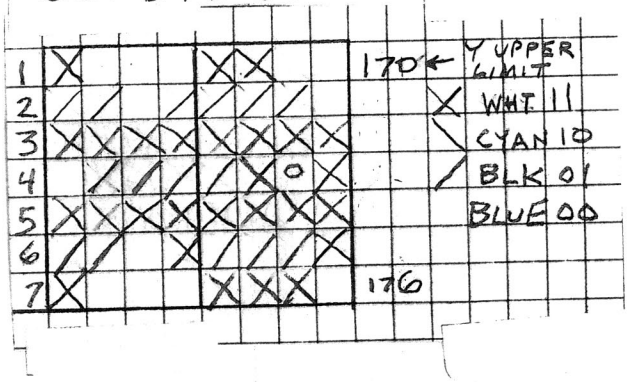
```
GLDFL 3E73_H  0000   X LOWER LIMIT
              3801   X UPPER LIMIT = 312
              00     Y LOWER LIMIT
              AA     Y UPPER LIMIT = 170_D
```

GOLDFISH PATTERN

```
PGLDF-4 3E79_H  00   RELATIVE X
                00      ↓    Y
                02   X SIZE (BYTES WIDE)
                07   Y    ↓  (LINES HIGH)

PGLDF 3E7D_H  C0 F0
              51 54
       3E81_H FF FF
              15 73
              FF FF
              53 57
       3E89_H C0 FC
```

GOLDFISH



Q
REVISED

CUSTOM FLOP

FLOP PATTERN FRAME IN SAME SCREEN LOCATION AS THE NORMAL PATTERN FRAME WRITE

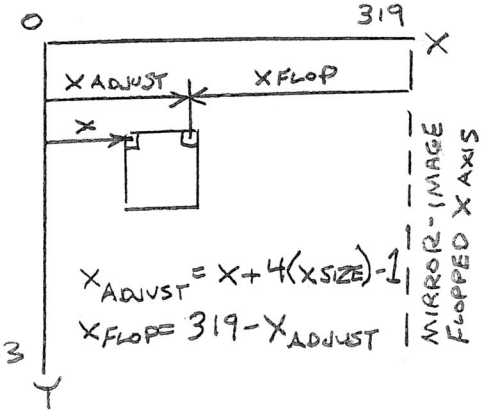
ENTER WITH: A = MR VALUE TO BE OUTPUT TO MAGIC REGISTER PORT $0C_H$

7 6 5 4 3 2 1 0 — SHIFT BITS ADJUSTED PER X COORD

1 1 1 — CUSTOM FLOP, XOR, FLOP

DE = X COORDINATE

BC = YSIZE XSIZE (FOR USE WITH PATTERN WRITE ROUTINE)



NOTES: NORMAL WRITE SHIFTS PIXELS RIGHT.
FLOPPED ↓ LEFT.

THIS CUSTOM FLOP DOES NOT PASS ON A MAGIC OR, EXPAND FUNCTION. CUFLOP IS SPECIFIC FOR THE HI-RES FISH DEMO.

BOY

CUFLOP $3E8B_H$ EB	EX DE,HL	HL = X COORD	
F1	POP AF	DISCARD ENTRY MR VALUE SHIFT BITS 1 AND 0	
ADJUST X COORDINATE FOR FLOP			
79	LD A,C	A = XSIZE	A = 4(XSIZE)-1
CB 27	SLA A		
$3E90_H$ CB 27	SLA A		
C6 FF	ADD A,-1		
5F	LD E,A		HL = X_{ADJUST} = X+4(XSIZE)-1
1600	LD D,0	HL = X COORD	
19	ADD HL,DE	DE = 4(XSIZE)-1	
E5	PUSH HL	SAVE X ADJUST	
ADJUST MR VALUE SHIFT BITS FOR THIS FLOP			
EB	EX DE,HL	DE = X_{ADJUST}	
7B	LD A,E		COMPLEMENT X_{ADJUST}
2F	CPL		
5F	LD E,A		
7A	LD A,D		
2F	CPL		
57	LD D,A		
$3EA0_H$ 21 40 01	LD HL, 319+1	2's COMPLEMENT	HL = X_{FLOP} = 319 - X_{ADJUST}
19	ADD HL,DE		
7D ← 006	LD A,L	GET X_{FLOP} (LOW ORDER)	SET UP NEW ADJUSTED MR VALUE
E6 03	AND 0000 0011	ISOLATE ITS SHIFT BITS	
F6 60	OR 0110 0000	ADD FLOP, XOR REQUESTS	
D1	POP DE	DE = ADJUSTED X COORDINATE FOR THIS FLOP	
F5	PUSH AF	SAVE NEW ADJUSTED MR VALUE FOR RELTAC ROUTINE	
$3EAB_H$ C3 0E 2C	JP RELTAC		

FLOPPED MIRROR IMAGE X COORDINATE

↑ PAGE 55

R

SET UP X_H, Y_H AND ΔX_L IN VECTOR BLOCK FOR FISH VECTOR ROUTINE
ENTER WITH: HL POINTING TO VECTOR BLOCK VARIABLES FOR THIS FISH

```
SUVB  3EAE_H  7E          LD A,(HL)
              32 D8 7F    LD (7FD8_H),A
3EB2_H        5F          LD E,A
              23          INC HL
              7E          LD A,(HL)
              32 D9 7F    LD (7FD9_H),A
              57          LD D,A
              23          INC HL
              7E          LD A,(HL)
              32 DE 7F    LD (7FDE_H),A
              32 F7 7F    LD (7FF7_H),A
3EC1_H        23          INC HL
              7E          LD A,(HL)
              32 D4 7F    LD (7FD4_H),A
              C9          RET
```

LOAD X_H (2 BYTES) INTO VB
DE = OLD FISH X COORDINATE FOR FISH BLANK WRITE

POINT HL AT Y_H
LOAD Y_H INTO VB AND ($7FF7_H$)
($7FF7_H$) = REG Y = Y COORDINATE FOR OLD FISH TO BLANK OLD FISH

POINT HL AT ΔX_L
LOAD ΔX_L INTO VB

EXIT WITH: DE = OLD X COORDINATE ↓ — SAVE TO BLANK OLD FISH WRITE USING XOR WRITE
($7FF7_H$) = OLD Y

SET UP ΔY_L AND ΔY_H IN VECTOR BLOCK FOR FISH VECTOR ROUTINE
ENTER WITH: HL POINTING TO ΔX_L IN FISH VARIABLES DATA BLOCK FOR THIS FISH

```
SUVB1 3EC7_H  23          INC HL
              7E          LD A,(HL)
              32 DB 7F    LD (7FDB_H),A
              CB 7F       BIT 7,A
              3E FF       LD A,FF_H
3ED0_H        20 01       JR NZ, A ?
              AF          XOR A
              32 DC 7F    LD (7FDC_H),A
              23          INC HL
3ED7_H        C9          RET
```

POINT HL AT ΔY_L VARIABLE
LOAD ΔY_L INTO VB
IF ΔY_L IS POSITIVE, SET ΔY_H IN VB = 00_H
IF ΔY_L IS NEGATIVE, SET ΔY_H IN VB = FF_H
POINT HL AT Δ TIMER

EXIT NOTE: DE IS NOT CLOBBERED

SET UP VECTOR BLOCK FOR VECTOR ROUTINE

ENTER WITH: B = NUMBER OF FISH TO PROCESS FOR THIS FISH TYPE

($7FCF_H$) = FISH PATTERN ADDRESS-4 TO WRITE THIS FISH TYPE
(POINTING AT PATTERN'S RELATIVE X)

($7FE4_H$) = VECTOR BLOCK VARIABLES INDEX TABLE ADDRESS-1
POINT TO 1 BYTE BEFORE THE INDEX TABLE

($7FE6_H$) = LIMITS TABLE ADDRESS FOR THIS FISH TYPE

```
SETVB 3ED8H C5          PUSH BC      SAVE FISH NUMBER FOR LOOP BACK
      CD 4C 3F          CALL INDEXB  POINT(INDEX) HL AT VB VARIABLES FOR
                                     THIS FISH
      CD AE 3E          CALL SUVB    SET UP IN VB XHI YH AND ΔXL
      E5                PUSH HL      SAVE HL POINTER TO POINT LATER AT ΔYL
                                     (HL IS STILL POINTING NOW AT ΔXL)
   3EEC CD A0 3F        CALL FLPCK   CHECK TO SET UP VB MR VALUE
                                     FOR A NORMAL OR FLOPPED WRITE
      E1                POP HL       RESTORE HL POINTING TO VARIABLE ΔXL
      CD C7 3E          CALL SUVB1   SET UP ΔYL AND ΔYH IN VB
      E5                PUSH HL      SAVE HL NOW POINTING AT VARIABLE ΔTIMER
      D5                PUSH DE      DE STILL HAS OLD X COORD TO BLANK LAST FISH
                                     WRITE.
                                     SAVE THIS OLD X COORD.
```

VECTOR (MOVE) THIS FISH

ENTER WITH: IX = VECTOR BLOCK ADDRESS
HL = LIMITS TABLE ADDRESS FOR THIS FISH TYPE
(POINTING TO LOWER X LIMIT)

```
      DD21 D1 7F        LD IX, 7FD1
      2A E6 7F          LD HL, (7FE6H)
   3EF0 CD FD 32        CALL MVECT  PAGE 84   CUSTOM MVECT DOES NOT
                                              ZERO TIME BASE IN VB.
```

BLANK (ERASE) OLD (LAST) FISH WRITE

```
      76                HALT         WAIT UNTIL IM'2 ROUTINE IS FINISHED
                                     TO MINIMIZE FISH "OFF" (BLANK) TIME
      D1                POP DE       DE = OLD FISH X COORD
      2A CF 7F          LD HL, (7FCFH)  HL = FISH PATTERN ADR-4
      3A F7 7F          LD A, (7FF7H)  ] B = OLD FISH Y COORD
      47                LD B, A        ]
      3A D1 7F          LD A, (7FD1H)  A = MAGIC REG VALUE (FROM VB)
      CD C7 2D          CALL WRITR   P.64 BLANK LAST FISH WRITE
   3F02 E1              POP HL       RETRIEVE ΔTIMER POINTER (ADDRESS)
                                     FOR THIS FISH
      C1                POP BC       B = FISH NUMBER (LOOPCTR)
      CD 1E 3F          CALL WFISH   WRITE THE NEW VECTORED FISH  P. 64
      10 CF             DJNZ SETVB
   3F09 C9              RET
```

-49
32 16 8 4 2 1
0 0 1 1 0 0 0 1
1 1 0 0 1 1 1 0
1 1 1 1

INITIALIZE 4 VECTOR BLOCK VARIABLES X_H (LOW ORDER), Y_H, ΔX_L, ΔY_L FOR THIS FISH TYPE

ENTER WITH: ($7FCF_H$) = FISH PATTERN ADDRESS − 4 FOR THIS FISH TYPE (POINTING TO PATTERN'S RELATIVE X)
2 BYTES

($7FE4_H$) = VECTOR BLOCK VARIABLES INDEX TABLE ADDRESS − 1 FOR THIS FISH TYPE
2 BYTES — POINT TO 1 BYTE PRIOR TO TABLE

B = NUMBER OF FISH TO BE PROCESSED FOR THIS FISH TYPE (LOOP COUNTER)

IFISH $3F0A_H$	C5	PUSH BC	SAVE THE FISH NUMBER (LOOP COUNTER)
	CD 8C 3F	CALL IVBLK	INITIALIZE FOR THIS FISH IN THE VECTOR BLOCK, X_H (LOW ORDER), Y_H, ΔX_L AND ΔY_L

INITIALIZE THE Δ TIMER FOR THIS FISH

C1	POP BC	B = FISH NUMBER AGAIN
CD 4C 3F	CALL INDEXB	INDEX THE ADDRESS OF THE VECTOR BLOCK VARIABLES FOR THIS FISH HL = ADDRESS OF THE VARIABLES BLOCK
	NOTE →	B IS NOT CLOBBERED IN THIS SUB
$3F12_H$ 11 0500	LD DE, 5	] POINT HL AT ΔTIMER FOR THIS FISH
19	ADD HL, DE	]
36 80	LD (HL), 80	SET ΔTIMER = 148_D FOR THIS FISH

WRITE THE FISH, THEN SAVE THE 4 VECTOR BLOCK VARIABLES

CD 1E 3F	CALL WFISH	WRITE A FISH B = FISH NUMBER (LOOP COUNTER) IS SAVED IN THIS SUB
10 ED	DJNZ IFISH	
$3F1D_H$ C9	RET	

−11 = 1101

128 64 32 16 8 4 2 1
0 0 0 1 0 0 1 1
1 1 1 0 1 1 0 0
+1
1 1 0 1

WRITE A FISH AND SAVE 4 VECTOR BLOCK VARIABLES X_H (2 BYTES), Y_H, ΔX_L, ΔY_L FOR THIS FISH

ENTER WITH: HL POINTING TO THE ΔTIMER FOR THIS FISH
B = FISH NUMBER FOR THIS FISH TYPE TO WRITE
($7FCF_H$) = FISH TYPE'S PATTERN ADDRESS-4 (POINTING TO PATTERN'S RELATIVE X)
($7FE4_H$) = INDEX VARIABLES TABLE FOR THIS FISH TYPE

		C5	PUSH BC	SAVE THE FISH NUMBER (LOOP COUNTER)
WFISH	$3F1E_H$	CD 75 3F	CALL TIMERCK	CHECK IF ΔTIMER = 0. RANDOMIZE ΔX_L, ΔX_L. RANDOMIZE NEW ΔTIMER IF SO.
WFISH1	$3F22_H$	CD A0 3F	CALL FLPCK	CHECK FOR A FLOPPED FISH, THEN SET ΔX_H ACCORDINGLY
		2A CF 7F	LD HL, ($7FCF_H$)	HL = FISH TYPE'S PATTERN ADR-4
		DD 21 D1 7F	LD IX, $7FD1_H$	IX = VECTOR BLK ADR
		CD B7 2D	CALL VWRITR	WRITE THE FISH, P.64
		C1	POP BC	B = FISH NUMBER AGAIN
	$3F30_H$	CD 4C 3F	CALL INDEXB	INDEX VECTOR BLK VARIABLES ADR HL = INDEXED VARIABLES ADR B IS NOT CLOBBERED
		3A D8 7F	LD A, ($7FD8_H$)	GET X_H FROM VB AND SAVE IT
		77	LD (HL), A	
		23	INC HL	2 BYTES
		3A D9 7F	LD A, ($7FD9_H$)	
		77	LD (HL), A	
		23	INC HL	
		3A DE 7F	LD A, ($7FDE_H$)	GET Y_H
	$3F40_H$	77	LD (HL), A	
		23	INC HL	
		3A D4 7F	LD A, ($7FD4_H$)	GET ΔX_L
		77	LD (HL), A	
		23	INC HL	
		3A DB 7F	LD A, ($7FDB_H$)	GET ΔY_L
		77	LD (HL), A	
	$3F4B_H$	C9	RET	

EXIT THIS SUB WITH B = FISH NUMBER

INDEX BYTE IN VECTOR BLOCK VARIABLES INDEX TABLE 145

SET UP HL TO POINT AT THE VARIABLES (DATA BLOCK)

ENTER WITH: $(7FE4_H)$ = VECTOR BLOCK VARIABLES INDEX TABLE ADDRESS - 1 ↑ POINT 1 BYTE PRIOR TO THE INDEX TABLE

B = FISH NUMBER FOR THIS FISH TYPE

EXIT WITH: HL = ADDRESS OF THE VARIABLES (DATA BLOCK)

INDEXB	$3F4C_H$	2A E4 7F	LD HL, $(7FE4_H)$	
		58	LD E, B	DE = INDEX (FISH) NUMBER.
	$3F50_H$	16 00	LD D, 0	
		19	ADD HL, DE	GET LOW ORDER BYTE OF VARIABLES (DATA BLOCK) PUT THIS BYTE IN A
		7E	LD A, (HL)	
		6F	LD L, A	HL = ADR OF VARIABLES
		26 7F	LD H, 7F	
		C9	RET	

VECTOR BLOCK VARIABLES INDEX TABLES (FOR FISH TYPES)

EACH BYTE = LOW ORDER BYTE OF VARIABLES ADDRESS
(HIGH ORDER BYTE OF VARIABLES IS SET TO $7F_H$ BY INDEXB ROUTINE ABOVE

INDEX 1	$3F58_H$	63	GOLDFISH	1
		69		2
		6F		3
		75		4
		7B		5
INDEX 2	$3F5D_H$	81	TROPICAL FISH A	1
		87		2
		8D		3
INDEX 3	$3F60_H$	93	TROPICAL FISH B	1
		99		2
		9F		3
INDEX 4	$3F63_H$	A5	TROPICAL FISH C	1
		AB		2
INDEX 6	$3F65_H$	B1		3
INDEX 5	$3F66_H$	B7	TROPICAL FISH D	1
		BD		2
INDEX 6	$3F68_H$	C3	SEA BOTTOM FISH	1
	$3F69_H$	C9		2

RANDOMIZE A DELTA (ΔX_L OR ΔY_L)

```
RANDELT 3F6A_H  3E 7F      LD A,127_D      ] RANDOMIZE A POSITIVE VALUE
                CD C532    CALL RANGE      ] FROM 0-126_D (SIGN BIT 7=0)
                4F         LD C,A          ] IF RANDOM VALUE IS EVEN,
        3F70_H  1F         RRA  [7 6 5 4 3 2 1 0 → CF]  ] USE THIS VALUE AS A
                79         LD A,C          ] POSITIVE Δ (+Δ=0,2,4...126)
                D0         RET NC          ] WAS PREVIOUS CF SET BY SUBROUTINE RANGE?
                2F         CPL             RANDOM VALUE IS ODD, SO NEGATE (COMPLEMENT)
                                           THIS RANDOM NUMBER. USE AS -Δ.
                                           NOTE: THIS IS NOT A 2'S COMPLEMENT.
                C9         RET
```

ENTER WITH: HL POINTING AT FISH Δ TIMER

CHECK THE DELTA TIMER

```
TIMERCK 3F75_H  35         DEC (HL)        ] (HL) POINTS TO Δ TIMER IN FISH
                C0         RET NZ          ] VARIABLES DATA BLOCK.
                                           ] DEC THE Δ TIMER.
                                           ] IF ΔTIMER=0, CONTINUE (RANDOMIZE NEW Δ TIMER)
                E5         PUSH HL         SAVE THE Δ TIMER POINTER
                3E 50      LD A,80_D       ] RANDOMIZE A NEW Δ TIMER
                CD C532    CALL RANGE      ] (0-79_D)
                E1         POP HL          POINT HL AT Δ TIMER AGAIN
                77         LD (HL),A       LOAD NEW Δ TIMER IN FISH VARIABLES DATA BLOCK
```

NOW LOAD RANDOM DELTAS ($\Delta X_L, \Delta Y_L$) INTO VECTOR BLOCK

```
DELTXY  3F7F_H  CD 6A 3F   CALL RANDELT     ] GET A RANDOM ±ΔX_L.
        3F82_H  32 D4 7F   LD (7FD4_H),A    ] PUT IT IN VECTOR BLOCK
                CD 6A 3F   CALL RANDELT     ] GET A RANDOM ±ΔY_L.
                32 DB 7F   LD (7FDB_H),A    ] PUT IT IN VECTOR BLOCK
                C9         RET
```

INITIALIZE VECTOR BLOCK VARIABLES X_H (LOW ORDER), Y_H, ΔX_L AND ΔY_L

LOAD THESE 4 VARIABLES INTO VECTOR BLOCK

```
IVBLK   3F8C_H  3E FF      LD A,FF         ] A=RANDOMIZED VALUE
                CD C532    CALL RANGE      ] FROM 0-254_D
        3F91_H  32 D8 7F   LD (7FD8_H),A   PUT RANDOMIZED X_H (LOW ORDER)
                                           IN VECTOR BLOCK
                3E AB      LD A,AB_H       ] A=RANDOMIZED VALUE
                CD C532    CALL RANGE      ] FROM 0-170_D
                32 DE 7F   LD (7FDE_H),A   PUT RANDOMIZED Y_H IN VECTOR BLOCK
                CD 7F 3F   CALL DELTXY
        3F9F_H  C9         RET
```

CHECK FOR A FISH FLOP

NOTE: DE IS NOT CLOBBERED

```
FLPCK  3FA0H 3A D4 7F     LDA,(7FD4H)     A=ΔXL
       3FA3H CB 7F        BIT 7,A         ┐ IF ΔXL IS POSITIVE
             2E 00        LDL,0           │ ΔXH=0
             3E 20        LDA,20          ┘ MR VALUE 20H, XOR WRITE
             28 04        JRZ,FLPCK1      ┐ IF ΔXL IS NEG,
             2E FF        LDL,FFH         │ ΔXH=FF        ↱ XOR
       3FADH 3E E0        LDA,E0          ┘ MR VALUE=E0H  1110 0000
                                            USE CUSTOM FLOP ↑ ← FLOP REQUEST
FLPCK1       32 D1 7F     LD(7FD1H),A     UPDATE MR VALUE IN VECTOR BLOCK
       3FB2H 7D ← DOG     LDA,L           ┐
             32 D5 7F     LD(7FD5H),A     │ UPDATE ΔXH IN VECTOR BLOCK
             32 D6 7F     LD(7FD6H),A     ┘   ↖ 2 BYTES
             C9           RET
```

SCREEN INTERRUPT VECTORS

INTERRUPT VECTOR 1/2 ↓

```
VECT1  3FBAH C7 3F
VECT2  3FBCH E2 3F
```

UNUSED BYTE

```
3FBEH FF
```

TOP COLOR TABLE

```
TCLR   3FBFH 07 WHT      PIXEL 11
       3FC0H CB CYAN       ↓  10
             00 BLK           01
             F9 BLUE          00
```

BOTTOM COLOR TABLE

```
BCLR   3FC3H 7A DARK BROWN   PIXEL 11
             7B LIGHT BROWN     ↓  10
             85 YEL                01
       3FC6H F9 BLUE               00
```

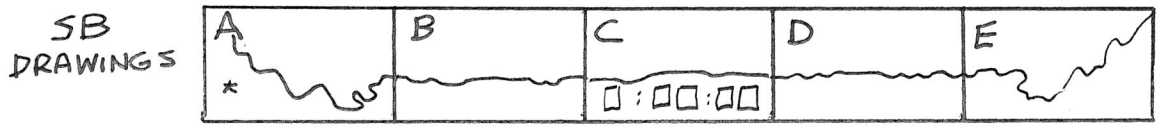
INTERRUPT ROUTINE 1 (FOR TOP COLOR SCAN)

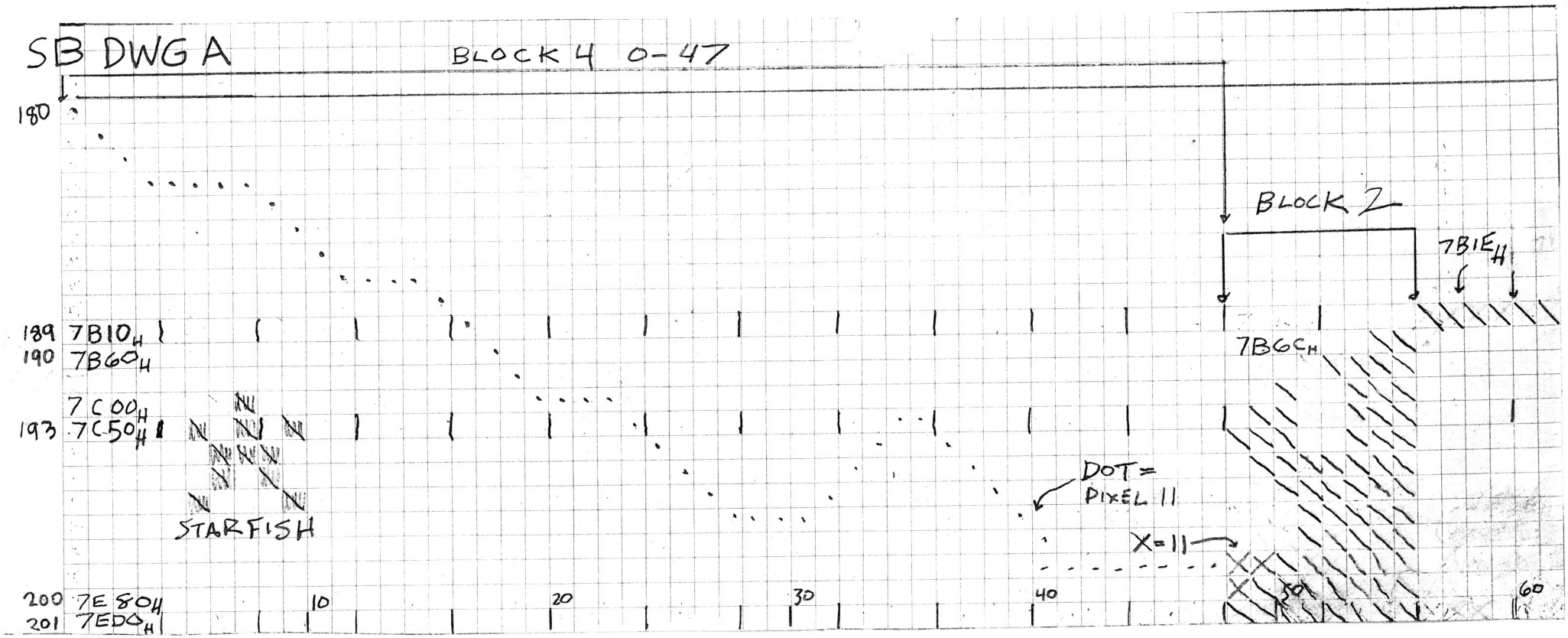
```
INTR1 3FC7H F3          DI
          F5            PUSH AF   ] SET Z80 ENVIRONMENT
          D9            EXX       ]
          21 BF3F       LD HL, TCLR   ] SET COLORS FOR
          01 0B 08      LD BC, 080BH  ]  TOP CLR SCAN
    3FD0H ED B3         OTIR          ]
          3E B0         LD A, B0H     ] INTERRUPT AFTER 176D
          D3 0F         OUT (0FH), A  ]  SCREEN SCAN LINES
          3E BC         LD A, LINE 2  ] SET NEXT INTERRUPT VECTOR
          D3 0D         OUT (0DH), A  ]  TO LINE 2 (@ 3FBCH)
          3E 2B         LD A, 0010 1011  ] SET HORIZ CLR BNDRY 00 10 1011
          D3 09         OUT (09H), A     ]   BORDER COLOR 00 ↑   43D
          D9            EXX       ]
          F1            POP AF    ] RESTORE Z80 ENVIRONMENT
    3FE0H FB            EI        ]
          C9            C9
```

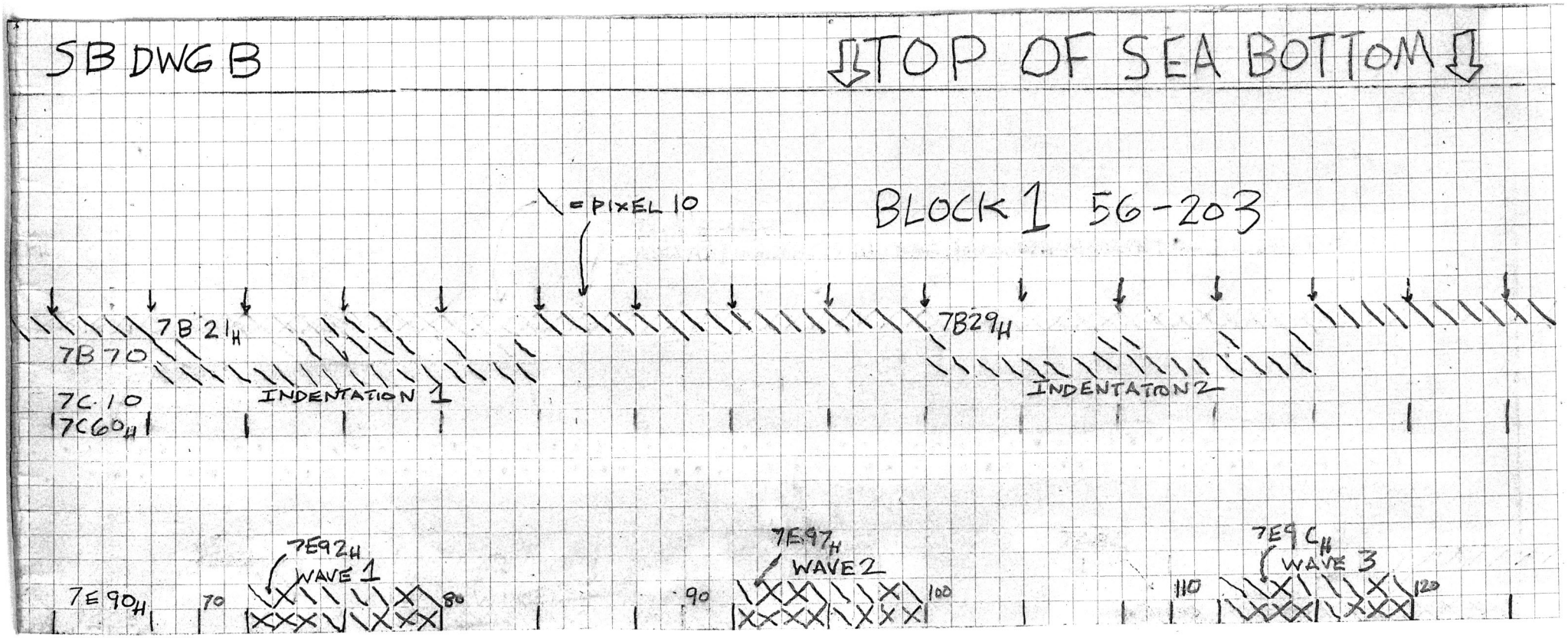
INTERRUPT ROUTINE 2 (FOR BOTTOM COLOR SCAN ELAPSED TIMER)

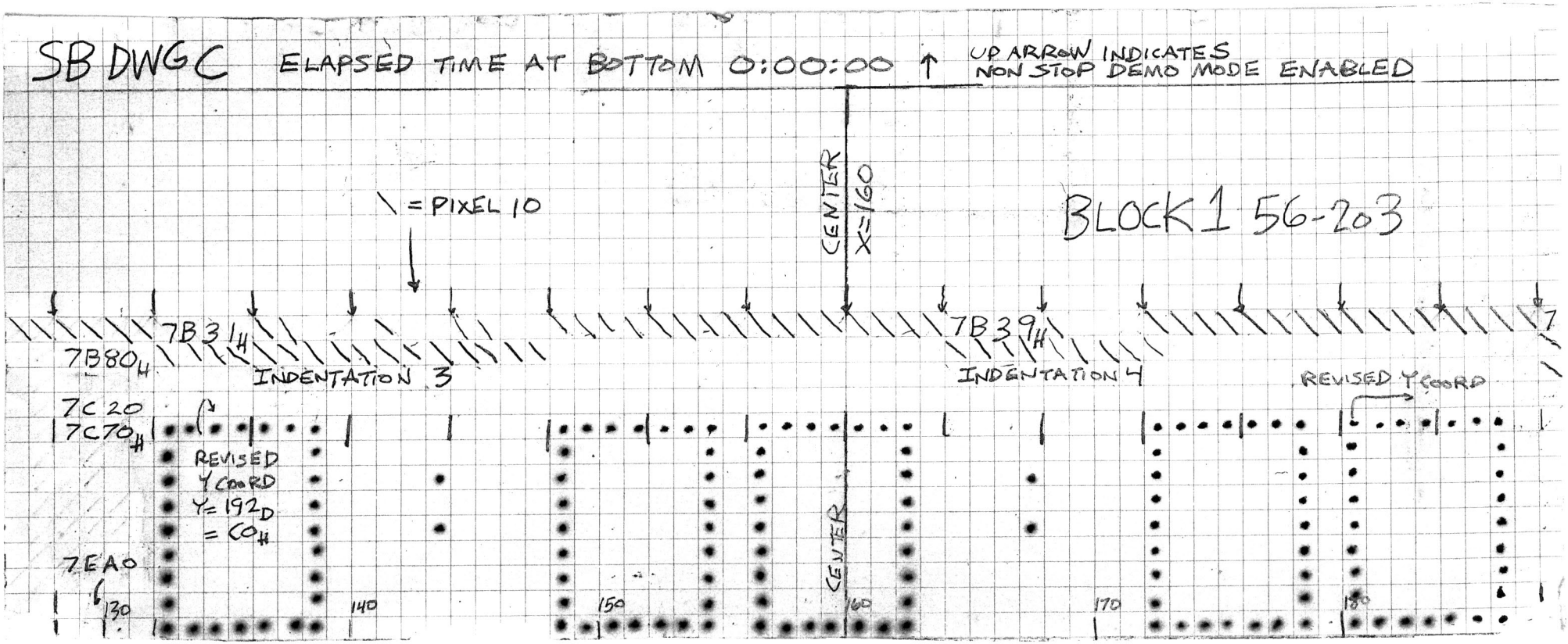
```
INTR2 3FE2H F3          DI
          F5            PUSH AF   ] SAVE Z80 ENVIRONMENT
          D9            EXX       ]
          21 C3 3F      LD HL, BCLR   ] SET COLORS FOR
          01 0B 08      LD BC, 080BH  ]  BOTTOM CLR SCAN
          ED B3         OTIR          ]
    3FED0H 3E DA ← WAS D5   LD A, DAH     ] INTERRUPT AFTER 218D
          D3 0F         OUT (0FH), A  ]  SCREEN LINES SCANNED
    3FF1H 3E BA         LD A, BA      ] SET NEXT INTERRUPT VECTOR
          D3 0D         OUT (0DH), A  ]  TO LINE 1 (@ 3FBAH)
          3E EB         LD A, 1110 1011  ] SET HORIZ CLR BNDRY 11 10 1011
          D3 09         OUT (09H), A     ]   BORDER COLOR 11 ↑   43D
          CD E6 3B      CALL ETIMER   UPDATE ELAPSED TIME, P. 130
          D9            EXX       ]
          F1            POP AF    ] RESTORE Z80 ENVIRONMENT
          FB            EI        ]
    3FFFH C9            RET
```

SEABOTTOM LAYOUT
(21 × 320 PIXELS)

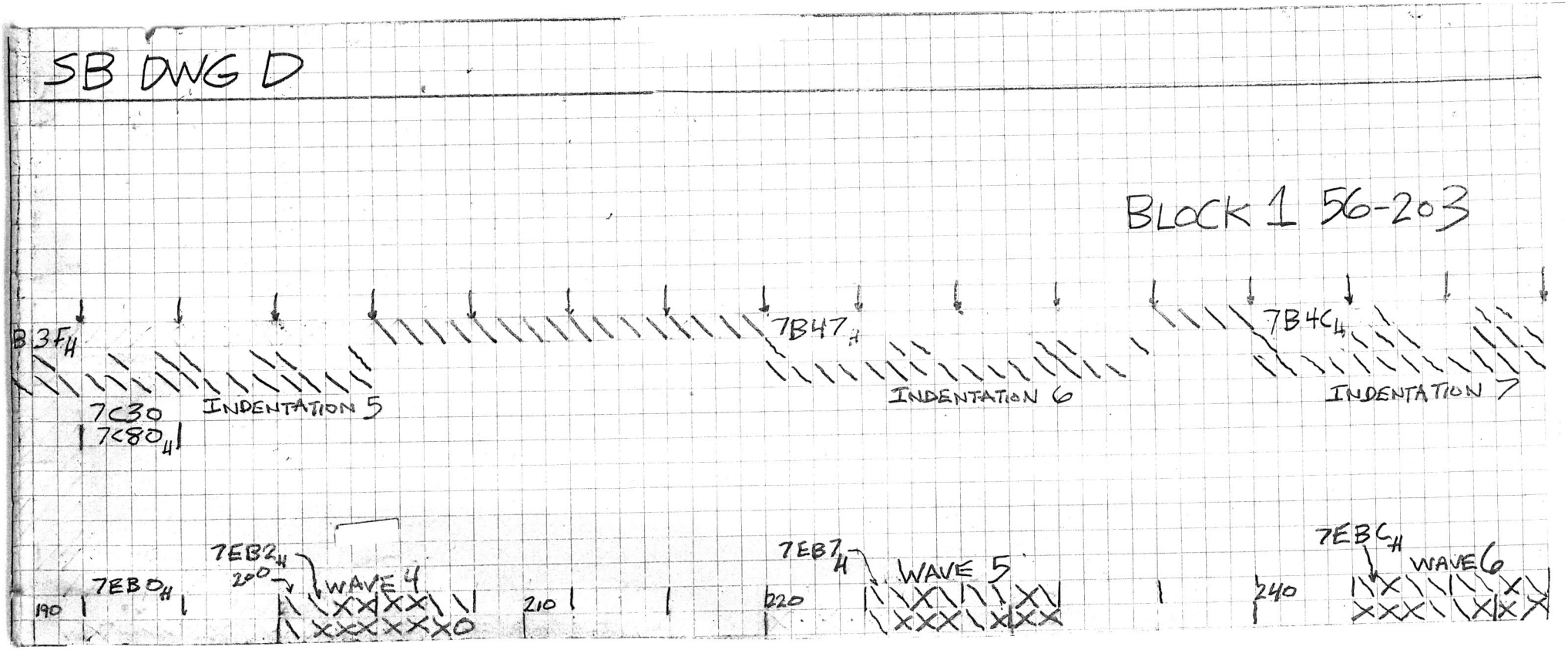




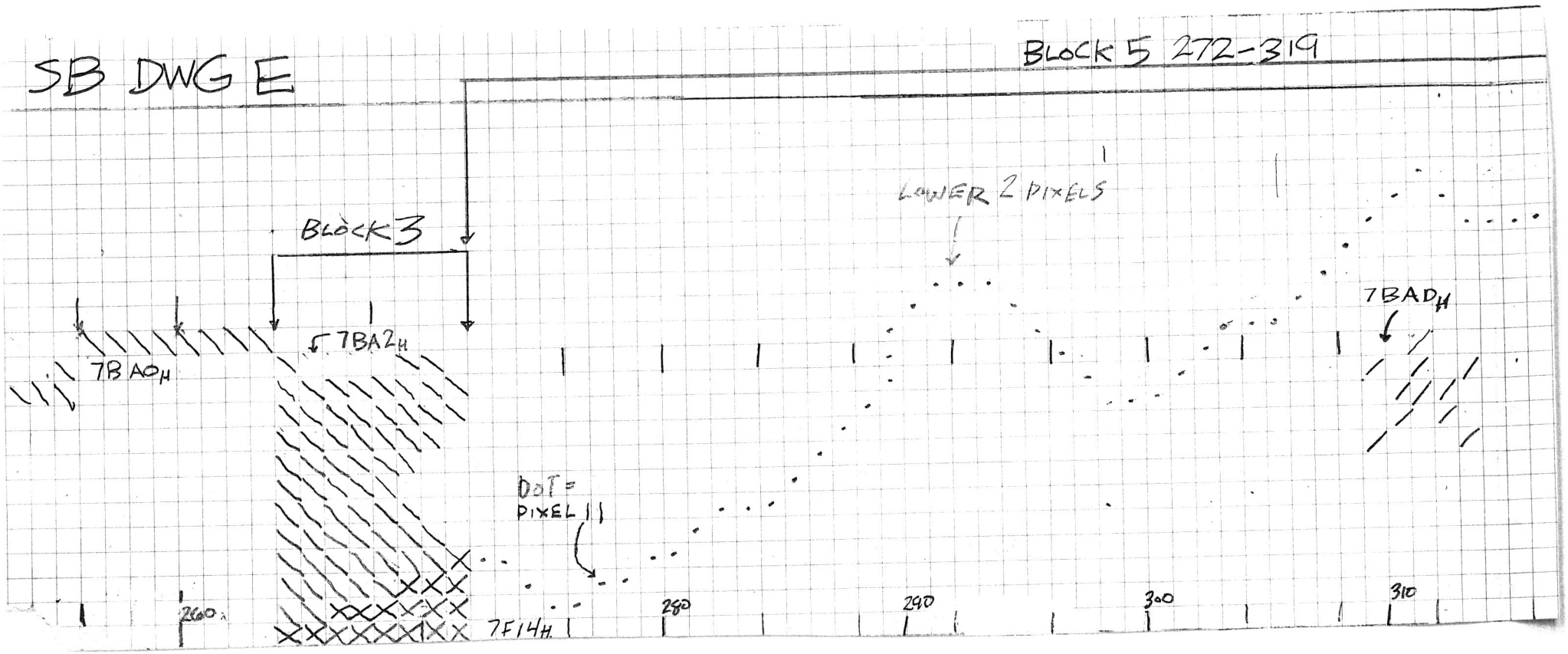

SB DWG B
TOP OF SEA BOTTOM
= PIXEL 10
BLOCK 1 56-203
7B21H
7B70
7C10
7C60H
INDENTATION 1
7B29H
INDENTATION 2
7E92H
WAVE 1
7E90H
7E97H
WAVE 2
7E9CH
WAVE 3
70
80
90
100
110
120

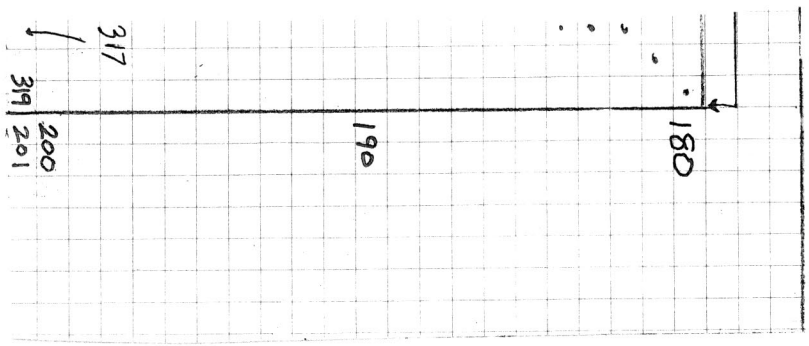

SB DWGC
ELAPSED TIME AT BOTTOM 0:00:00
UP ARROW INDICATES NON STOP DEMO MODE ENABLED
| = PIXEL 10
CENTER X=160
BLOCK1 56-203
7B31H
7B80H
INDENTATION 3
7B39H
INDENTATION 4
REVISED Y COORD
7C20
7C70H
REVISED Y COORD
Y=192D
=C0H
7EA0
CENTER
130
140
150
160
170
180

SB DWG D


BLOCK 1 56-203
B3F$_H$
7B47$_H$
7B4C$_H$
INDENTATION 5
INDENTATION 6
INDENTATION 7
7C30
7C80$_H$
7EB0$_H$
7EB2$_H$
200
WAVE 4
7EB7$_H$
WAVE 5
7EBC$_H$
WAVE 6
190
210
220
240

SB DWG E


BLOCK 5 272-319
LOWER 2 PIXELS
BLOCK 3
7BAD$_H$
7BA2$_H$
7BA0$_H$
DOT = PIXEL 111
260
280
290
300
310
7F14$_H$


317
319
200
201
190
180

(This page intentionally left blank.)

BULLET

X = 135

X = 143

BLOCK 2
2 BYTES WIDE
16 LINES HIGH

X = 151

BLOCK 1
2 BYTES WIDE
7 LINES HIGH

X = 159

X = 174

X = 182

Y = 74

BLOCK 3
2 BYTES WIDE
10 LINES HIGH

Y = 80

BLOCK 4
2 BYTES WIDE
17 LINES HIGH

Y = 90

ALL NUMBERS
ARE DECIMAL

Y = 107

X = 131

Y = 114

X = 183

X = 127

Y = 125

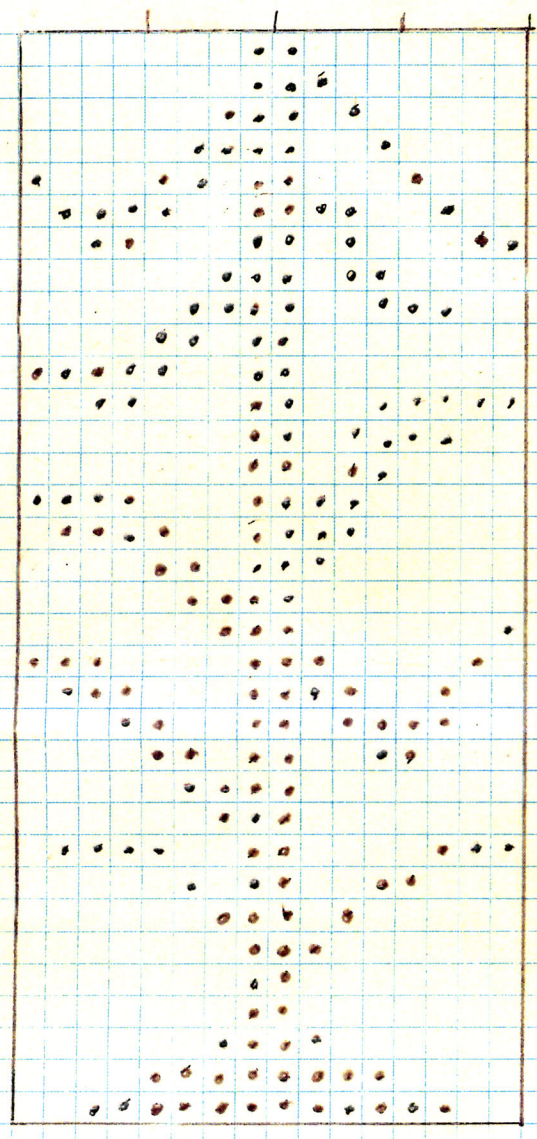
X = 187

GREEN

BLUE

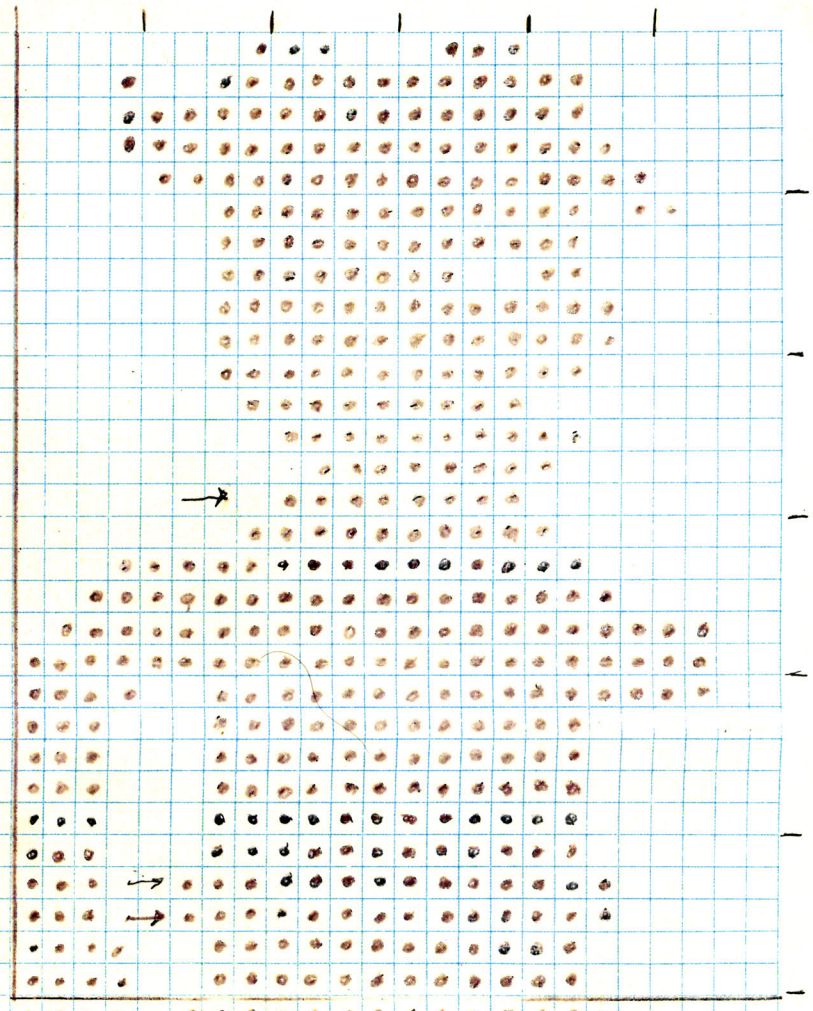
GREEN

(This page intentionally left blank.)



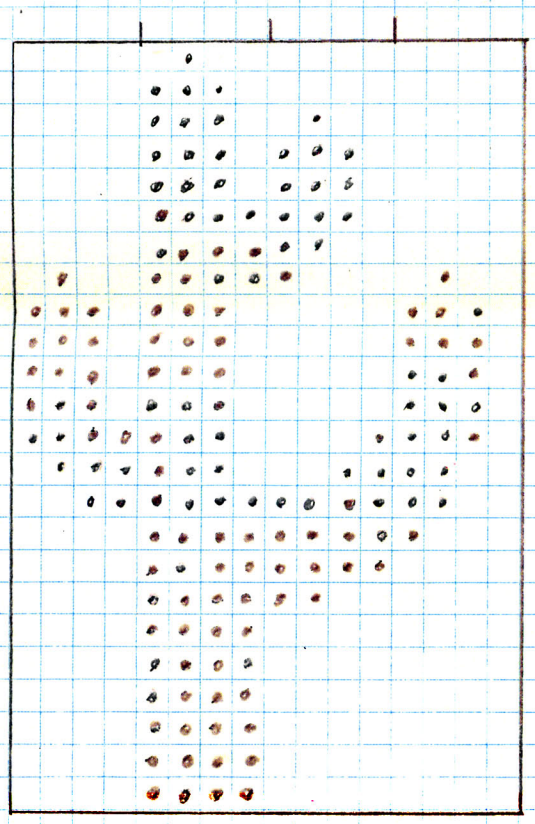
BLUE HAT

YELLOW FACE



GREEN BELT

YELLOW HAND



GREEN BOOTS