

# SetScreen 2 - An Upgraded Bally Arcade/Astrocade Troubleshooting Utility

By MCM Design/Michael Matte, October 2017

WRITE ROUTINES AND A READ ROUTINE HAVE BEEN ADDED TO SetScreen 2 WITH THE INTENTION OF PROVIDING A TROUBLESHOOTER MORE INFO TO DIAGNOSE AND ISOLATE A PROBLEM AREA FOR A MOTHERBOARD THAT PRODUCES A BLANK TV SCREEN DURING POWER UP.

IN ADDITION TO THE 3 PARAMETER SCREEN SET UP, SetScreen 2 WRITES TO SCREEN RAM AS FOLLOWS:

- ① THE TOP 1/3 OF SCREEN RAM (34 LINES) IS FILLED WITH THE COLOR 1010 1010 (AA<sub>H</sub>), YELLOW ON THE LEFT AND BROWN ON THE RIGHT. SEE SetScreen 2 CODE LISTING FOR COLOR TABLE USED AT ADDRESS LOCATION 2003<sub>H</sub>.
- ② THE CENTER AREA OF SCREEN RAM IS ZEROED (CLEARED) IN 17 LINES WITH BLACK ON THE LEFT AND BLUE ON THE RIGHT.
- ③ THE 4 CORNER BYTES, 4000, 4027, 4FC8 AND 4FEF<sub>H</sub> ARE SET TO COLOR 0101 0101 (55<sub>H</sub>) WITH THE LEFT CORNER BYTES WHITE AND THE RIGHT CORNER BYTES MAGENTA.

SEE PAGE 2 FOR DETAILS ON SetScreen's DISPLAY ATTEMPT AT POWER UP.

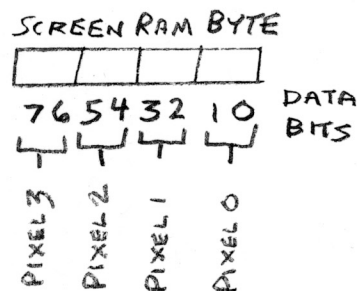
IN ADDITION TO SetScreen 2 ATTEMPTING TO WRITE DATA INDICATED ABOVE, SetScreen 2 SCANS KEYPAD COLUMNS 3 AND 2 TO EXECUTE ADDITIONAL SCREEN RAM WRITE ROUTINES AND A READ ROUTINE.

## COLUMN 3 PIXEL WRITE ROUTINES

BUT HOLDING DOWN THE APPROPRIATE KEY IN COLUMN 3 (LEFT MOST COLUMN) AND PRESSING THE "RESET" BUTTON, THE USER CAN CHOOSE FROM 5 DIFFERENT WRITE ROUTINES.

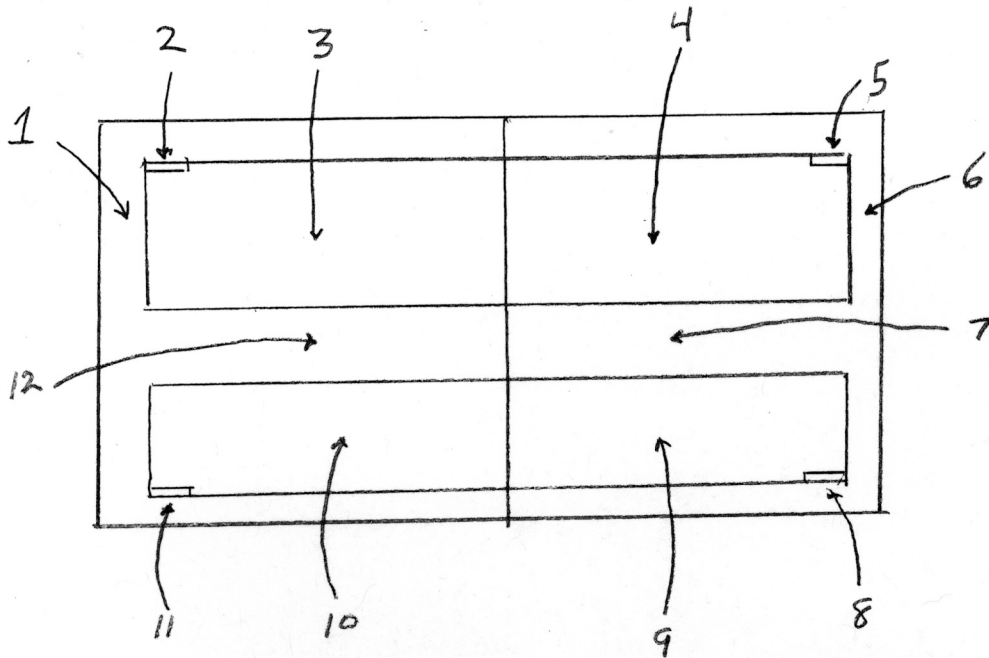
- ① FOUR PIXEL COLOR VARIATION WRITE ROUTINES ARE AVAILABLE, EACH ROUTINE EMPHATICIZES ONE PARTICULAR PIXEL IN EVERY SCREEN RAM BYTE. IT TAKES 2 DATA BITS TO DEFINE A PIXEL (PICTURE ELEMENT). SO, A PIXEL CAN BE ONE OF FOUR POSSIBLE COLORS (00, 01, 10 OR 11). THE EMPHATICIZED PIXEL IS WRITTEN WITH THE 4 VARIATIONS USING 4 WRITE PASSES. THE 4 WRITE PASSES ARE:

PASS 1, PIXEL VARIATION 00  
↓ 2 01  
↓ 3 10  
↓ 4 11



THE VISUAL DISPLAY ON THE TV SCREEN IS 40 VERTICAL STRIPES SIMULTANEOUSLY WRITTEN TOP TO BOTTOM DOWN THE ENTIRE HEIGHT OF THE SCREEN.

# INITIAL NORMAL DISPLAY AT POWER ON



LEFT SIDE

RIGHT SIDE

- 1 BLACK BACKGROUND AREA
- 2 WHITE BYTE WRITE AT 4000<sub>H</sub>.  
0101 0101
- 3 YELLOW WRITE, TOP AREA  
FILLED WITH 1010 1010
- 12 BLACK WRITE, CENTER AREA  
FILLED WITH 0000 0000
- 11 WHITE BYTE WRITE AT 4FC8<sub>H</sub>  
0101 0101
- 10 UNCLEARED, NOT WRITTEN TO,  
BOTTOM AREA, AREA DISPLAYED  
IS CONTENTS OF RAM AT POWER ON.  
COLORS: BLACK PIXEL 00  
WHITE PIXEL 01  
YELLOW PIXEL 10  
GREEN PIXEL 11

- 6 BLUE BACKGROUND AREA
- 5 MAGENTA BYTE WRITE AT 4027<sub>H</sub>  
0101 0101
- 4 BROWN WRITE, TOP AREA  
FILLED WITH 1010 1010
- 7 BLUE WRITE, CENTER AREA  
FILLED WITH 0000 0000
- 8 MAGENTA BYTE WRITE AT 4FEF<sub>H</sub>  
0101 0101
- 9 SAME AS 10

COLORS:	BLUE	PIXEL 00
	MAGENTA	01
	BROWN	10
	CYAN	11

NOTE: ABOVE IS SetScreen 2's DISPLAY WHEN MOTHERBOARD IS OPERATING NORMALLY. THE ABOVE DISPLAY WILL VARY DEPENDENT ON THE MOTHERBOARD FAILURE, SetScreen 2 ATTEMPTS TO WRITE DATA TO SCREEN RAM AS INDICATED ABOVE, BUT MAY NOT BE ABLE TO BECAUSE OF A MOTHERBOARD FAILURE. YOU MAY ONLY SEE A SPLIT SCREEN WITH BLACK ON LEFT AND BLUE ON RIGHT OR YOU MAY NOT EVEN SEE A DISPLAY. SEE ORIGINAL SetScreen DOC FOR MORE INFO.

ONE LINE OF 40 BYTES IS WRITTEN AT A TIME, THERE IS A TIME DELAY BETWEEN EACH LINE WRITTEN TO SLOW DOWN THE SCREEN WRITE PROVIDING AMPLE TIME FOR THE USER TO WATCH THE TV SCREEN FOR ABNORMALITIES DURING EACH OF THE 4 PASSES. EACH PASS NUMBER IS OUTPUT FOR A HOMEMADE BALCHECK II DISPLAY. AN ORIGINAL BALCHECK UNIT COULD BE MODIFIED SLIGHTLY TO DISABLE THE BALCHECK ROM VIA A MINI-TOGGLE SWITCH ALLOWING SetScreen 2 TO OUTPUT A PASS NUMBER TO THE BALCHECK DISPLAY. IT IS RECOMMENDED THAT THE USER RUN THESE COLUMN 3 WRITE ROUTINES ON A WORKING MOTHERBOARD TO SEE HOW THEY NORMALLY RUN. IF THE TROUBLESHOOTER UNDERSTANDS AT THE CHIP LEVEL WHAT IS GOING ON TO PRODUCE THE PIXEL GRAPHICS RUNNING ON THE TV SCREEN, HIS KNOWLEDGE AND OBSERVATION OF THE TV SCREEN MAY ELIMINATE CERTAIN CHIPS AS "POSSIBLE BAD" CHIPS. FOR EXAMPLE, IF YOU CAN SEE ON THE TV SCREEN EACH HORIZONTAL LINE BEING WRITTEN YOU KNOW THAT THE CUSTOM ADDRESS CHIP, BUFFER CHIP U20 AND THE RAS, CAS AND WE LINES ARE OPERATING PROPERLY.

THE FOLLOWING LISTS THE COLUMN 3 KEY TO PRESS TO EXECUTE THE WRITE PIXEL NUMBER ROUTINE. AS MENTIONED ABOVE THE PIXEL IS WRITTEN WITH 4 COLOR VARIATIONS USING 4 PASSES. EACH ROUTINE CONTINUES REPEATING THE FOUR PASSES UNTIL THE "RESET" BUTTON IS PUSHED. EXECUTE THE ROUTINE BY HOLDING DOWN THE APPROPRIATE COLUMN 3 KEY AND PUSHING THE "RESET" BUTTON.

COLUMN 3 KEY	PIXEL NUMBER WRITTEN IN 4 COLOR PASSES	SCREEN RAM BYTE								
C	PIXEL 1	<table border="1"><tr><td></td><td></td><td></td><td></td></tr><tr><td>PIXEL 3</td><td>PIXEL 2</td><td>PIXEL 1</td><td>PIXEL 0</td></tr></table>					PIXEL 3	PIXEL 2	PIXEL 1	PIXEL 0
PIXEL 3	PIXEL 2	PIXEL 1	PIXEL 0							
MR	PIXEL 3									
7	PIXEL 0									
4	PIXEL 2									

THE COLOR VARIATIONS ARE DEFINED IN SetScreen 2'S COLOR TABLE. SEE TABLE IN SetScreen CODE LISTING AT 2003H.

YOU CAN SEE ON THE TV SCREEN WHICH PIXEL NUMBER IS BEING WRITTEN WITH THE COLOR VARIATIONS BY NOTING THE POSITION OF A VERTICAL STRIPE WITH RESPECT TO THE MIDDLE VERTICAL SPLIT LINE.

- ② TO REPEATEDLY ATTEMPT TO WRITE (FILL) ENTIRE SCREEN WITH 0000 0000 AND 1111 1111, HOLD DOWN KEY 1 IN COLUMN 3 AND PRESS "RESET" BUTTON.

COLUMN 3 KEY	EXECUTES
1	CLEAR AND FILL ENTIRE SCREEN

# COLUMN 2 SIX BYTE READ ROUTINE

SetScreen2  
PAGE 4

SetScreen2 ATTEMPTS TO WRITE DATA TO THE 4 CORNER BYTES IN SCREEN RAM. SEE PAGES 1 AND 2 FOR DETAILS. IF YOUR TV SCREEN IS LARGE ENOUGH, YOU CAN VISUALLY DETERMINE IF SETSCREEN2 WRITES SUCCESSFULLY TO THESE 4 BYTES.

BY HOLDING DOWN THE APPROPRIATE KEY IN COLUMN 2 AND PRESSING THE "RESET" BUTTON, SETSCREEN2 ATTEMPTS TO READ THE 4 CORNER BYTES WITH 3 VARIATIONS. SETSCREEN2 WILL OUTPUT READ TO A HOMEMADE BALCHECK II DISPLAY CONNECTED TO THE MOTHERBOARD.

COLUMN 2 KEY	ATTEMPT	COMMENT
↑	READ ONCE (4000 <sub>H</sub> ). READ SHOULD BE 55 <sub>H</sub> .	BYTE AT 4000 <sub>H</sub> SHOULD BE ALL WHITE.
MS	WRITE 44 <sub>H</sub> TO 4000 <sub>H</sub> . READ ONCE (4000 <sub>H</sub> ).	BYTE AT 4000 <sub>H</sub> SHOULD BE 0100 0100. WHT BLK WHT BLK
8	READ ONCE (4027 <sub>H</sub> ). READ SHOULD BE 55 <sub>H</sub> .	BYTE AT 4027 <sub>H</sub> SHOULD BE ALL MAGENTA.
5	WRITE 44 <sub>H</sub> TO 4027 <sub>H</sub> . READ ONCE (4027 <sub>H</sub> ).	BYTE AT 4027 <sub>H</sub> SHOULD BE 0100 0100. MAG BLUE MAG BLUE
2	READ ONCE (4FC8 <sub>H</sub> ). READ SHOULD BE 55 <sub>H</sub> .	BYTE AT 4FC8 <sub>H</sub> SHOULD BE ALL WHITE.
0	WRITE 44 <sub>H</sub> TO 4FEF <sub>H</sub> . READ ONCE (4FEF <sub>H</sub> ).	BYTE AT 4FEF <sub>H</sub> SHOULD BE 0100 0100. MAG BLU MAG BLU

# NEW SetScreen 2 LISTING

2000<sub>H</sub> C3 0B 20

JP SetScreen

## SetScreen COLOR TABLE

CLRTBL 2003<sub>H</sub> AC  
 86  
 07  
 00 ← DOG  
 CD  
 79  
 2B  
 F9

GREEN PIXEL 11  
 YELLOW ↓ 10  
 WHITE ↓ 01  
 BLACK ↓ 00  
 CYAN 11  
 BROWN 10  
 MAGENTA 01  
 BLUE 00

LEFT  
 COLORS  
 RIGHT  
 COLORS

COLOR REG 7  
 ↓  
 6  
 5  
 4  
 3  
 2  
 1  
 0

## SetScreen

SetScreen 200B<sub>H</sub> AF  
 011808  
 DOG → ED 79  
 2011<sub>H</sub> 10FC  
 3ECC  
 DOG → D3 0A  
 3E14  
 DOG → D3 09  
 210320  
 DOG → 010B08  
 2021<sub>H</sub> EDB3

XORA A=0  
 LD BC, 0818<sub>H</sub>  
 OUT (C), A  
 DJNZ-2  
 LDA, 204<sub>D</sub>  
 OUT (0A<sub>H</sub>), A  
 LDA, 20<sub>D</sub>  
 OUT (09<sub>H</sub>), A  
 LD HL, CLRTBL  
 LD BC, 080B<sub>H</sub>  
 OTIR  
 LD HL, 4000<sub>H</sub>  
 LD BC, 0550<sub>H</sub>  
 LD A, 1010 1010  
 LD (HL), A  
 CPI A ← (HL)  
 HL ← HL+1  
 BC ← BC-1

STOP ALL SOUND  
 SET  
 VERTICAL BLANK REGISTER  
 TO 204<sub>D</sub>  
 SET  
 HORIZONTAL COLOR BOUNDARY  
 TO 20<sub>D</sub>, SPLIT SCREEN  
 SET ALL COLORS  
 VIA  
 OUTPUT PORT 0B<sub>H</sub>  
 $\frac{102 \text{ LINES}}{3} = 34 \text{ LINES}$   
 FILL 1/3 OF SCREEN  
 $34 \times 40 = 1360 = 0550<sub>H}</sub> BYTES  
 WITH COLOR AA<sub>H</sub>  
 COLORS, L YEL, R BRN$

MFILL 210040  
 015005  
 3EAA  
 MFILL 1 202B<sub>H</sub> 77  
 ED A1

JP PE, MFILL1  
 LD BC, 02A8<sub>H</sub>  
 LDA, 00<sub>H</sub> NO XORA IN CASE  
 LD (HL), A OF CHANGE  
 CPI  
 JP PE, MFILL2

FILL 17 LINES WITH 00<sub>H</sub>  
 BELOW TOP FILL  
 $17 \times 40 = 680 = 02A8<sub>H}</sub> BYTES  
 COLORS, L BLK, R BLU$

MFILL 2 2031<sub>H</sub> EA 2B20  
 01A802  
 3E00  
 2036<sub>H</sub> 77 DOG  
 ED A1  
 EA 3620  
 3E55  
 320040  
 2041<sub>H</sub> 32 27 40  
 32 C8 4F  
 2047<sub>H</sub> 32 EF 4F

LDA, 55<sub>H</sub> 0101 0101  
 LD (4000<sub>H</sub>), A  
 LD (4027<sub>H</sub>), A  
 LD (4FC8<sub>H</sub>), A  
 LD (4FEF<sub>H</sub>), A

WRITE TO 4 CORNER BYTES  
 4000<sub>H</sub> 4027<sub>H</sub>  
 4FC8<sub>H</sub> 4FEF<sub>H</sub>  
 COLORS, L WHT, R MAGENTA

```

    DOG
204AH DB 17
      A7
    DOG
      2061
2051H DB 16
      A7
      C2 D1 20
      76
      FF FF
      FF FF FF
      FF FF
      FF
      FF FF FF
2061H FF FF FF
    
```

```

INA, (COL 3)
AND A
JR NZ, CHK COL 3 JMP 97D = 61H
INA, (COL 2)
AND A
JP NZ, CHK COL 2
HALT CPU EXECUTES NOPS
DI @ 0001H
    
```

KEY PRESSED IN LEFT MOST COLUMN 3?  
IF SO, JMP TO CHECK COL 3

KEY PRESSED IN COLUMN 2?  
IF SO, JMP TO CHECK COL 2

FUTURE CHK COL 1?

FUTURE CHK COL 0?

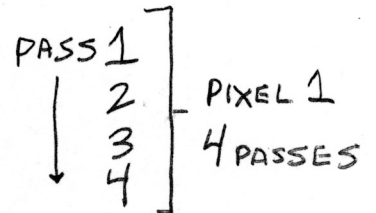
FUTURE JUMP?

WRITE PIXEL STRIPES COLOR TABLES

```

PIX01 2064H 00 0000 0000 ZERO
              04 0000 0100
              08 0000 1000
              0C 0000 1100
    
```

L=BLK, R=BLUE  
L=WHT, R=MAGENTA  
L=YEL, R=BRN  
L=GRN, R=CYAN



```

PIX03 2068H 00 0600 0000
              40 0100 0000
              80 1000 0000
              C0 1100 0000
    
```

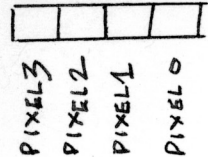
PIXEL 3  
4 PASSES \*

```

PIX00 206CH 00 0000 0000
              01 0000 0001
              02 0000 0010
              03 0000 0011
    
```

PIXEL 0  
4 PASSES \*

\* SAME PIXEL COLORS AS ABOVE PIXEL 1  
SCREEN RAM BYTE



```

PIX02 2070H 00 0600 0000
              10 0001 0000
              20 0010 0000
              30 0011 0000
    
```

PIXEL 2  
4 PASSES \*

```

2074H 00 0000 0000
      FF 1111 1111
      00 0000 0000
      FF 1111 1111
    
```

L=BLK, R=BLUE  
L=GRN, R=CYAN  
CLEAR AND FILL

WRITE PIXEL STRIPES

ENTER WITH L = COLOR TABLE LOW ORDER BYTE

```

1 STRIPE 2078H 06 04
              0E 01
              79
              D3 FC
              7D ← DOG
2080H D9
2081H 6F
    
```

```

LDB, 4D
LDC, 1D
LDA, C
OUT(FCH), A
LDA, L
EXX
LDL, A
    
```

B = INITIAL PASS COUNTER  
C = INITIAL PASS NUMBER  
OUTPUT PASS NUMBER TO BALCHECK II DISPLAY  
A = COLOR TBL LO BYTE  
SAVE PASS CTR, PASS# IN BC!  
COLOR TBL LO BYTE IN L!  
L = TBL LO BYTE

WSTRIPE 2 2082H 26 20  
7E  
11 00 40  
0E 66  
WLINE 1 06 28  
WLINE 2 12  
13  
10 FC

LD H, 20H  
LDA, (HL)  
LD DE, 4000H  
LD C, 102D  
LD B, 40D  
LD (DE), A  
INC DE  
DJNZ, WLINE 2

A = PASS COLORS TO WRITE  
DE = SCREEN START ADR  
C = "LINES TO WRITE" COUNTER  
B = "BYTES PER LINE" COUNTER

WRITE A BYTE  
BUMP SCRN ADR  
-4 0000 0100  
1101 1011  
+1  
1111 1100

WRITE ONE LINE

DELAY 2090H 47  
DELAY 1 3E 60  
08  
3E FF  
DELAY 2 3D ← DOG  
1000 0011 -3 →  
1111 1100  
+1  
1111 1101  
F D  
20 FD  
08 ← DOG  
3D ← DOG  
20 F6

LDB, A  
LDA, 60H  
EX AF, AF'  
LDA, FFH  
DECA  
JR NZ, DELAY 2  
EX AF, AF'  
DECA  
JR NZ, DELAY 1

SAVE "PASS COLORS TO WRITE" IN B  
A = WRITE LINE SPEED = 1ST LOOP CTR  
SAVE "1ST LOOP CTR" IN A'  
A = 2ND LOOP CTR  
1ST DELAY LOOP.  
A = 1ST LOOP CTR AGAIN  
-10 0000 1010  
1111 0101  
+1  
1111 0110

EXECUTE DELAY BEFORE WRITING NEXT LINE.

AF ← DOG  
0D  
B9

XOR A, A  
DEC C  
CP C

A = 0  
DEC "LINES TO WRITE" CTR

20A0H 78  
DOG → 20 E7  
D9  
OC  
10 02  
18 CF

LD A, B  
JR NZ, WLINE 1  
EXX  
INC C  
DJNZ, NXPASS  
JR WSTRIPE

A = "PASS COLORS TO WRITE"  
WRITE ANOTHER LINE? -25  
LOOP BACK IF SO.  
0001 1001  
1110 0110  
+1  
1110 0111

NXPASS 20A9 79  
D3 FC  
DOG → D9  
23  
20AE 18 D4  
H

EXX  
INC C  
DJNZ, NXPASS  
JR WSTRIPE  
LDA, C  
OUT (FC, HL), A  
EXX  
INCHL  
JR WSTRIPE 2

B = PASS CTR, C = PASS #, L = INITIAL COLOR TBL ADR  
INCREMENT PASS NUMBER  
WRITE ANOTHER PASS? IF SO, LOOP BACK  
LOOP BACK AND REPEAT 4 PASSES  
OUTPUT PASS NUMBER TO BALCHECK II DISPLAY  
HL = LAST COLOR TBL ADR  
B = PASS CTR, C = PASS #, L = INITIAL COLOR TBL ADR  
POINT HL AT NEXT PASS COLOR  
-44 0010 1100  
1101 0011  
+1  
1101 0100

NOTE: YOU CAN SPEED UP OR SLOW DOWN PASS TIME BY VARYING THE "1ST LOOP COUNTER" AT 2092H. TO SPEED UP PASS, DECREASE VALUE OF COUNTER. THIS COUNTER WAS INITIALIZED TO 60H TO MAKE IT EASIER TO OBSERVE ANY ABNORMALITIES NEAR THE HORIZONTAL LINE THAT IS BEING WRITTEN.

CHECK COLUMN 3 COLOR TABLE POINTER ENTER WITH A=COL3 STATUS

```
CHK COL3 20BDH 2E64
          FE01
          28C2
          2E68
          FE02
          28BC
          2E6C
          FE04
20C0H 28B6
          2E70
          FE08
          28B0
          2E74
          FE10
          28AA
          76
20CFH  FF FF
```

```
LDL, 64H
CP 1
JRZ, WSTRIPE
LDL, 68H
CP 2
JRZ, WSTRIPE
LDL, 6CH
CP 4
JRZ, WSTRIPE
LDL, 70H
CP 8
JRZ, WSTRIPE
LDL, 74H
CP 16
JRZ, WSTRIPE
HALT
```

-62 0011 1110  
1100 0001  
+1  
1100 0010

"C" KEY PRESSED?  
IF SO, STRIPE WITH PIXEL 1,  
"MR" KEY PRESSED?  
IF SO, STRIPE WITH PIXEL 3,  
KEY 7 PRESSED?  
IF SO, STRIPE WITH PIXEL 0,  
KEY 4 PRESSED?  
IF SO, STRIPE WITH PIXEL 2,  
KEY 1 PRESSED?  
IF SO, CLEAR + FILL REPEATED

CPU EXECUTES NOP'S, DI @ 0001H

READ 4 CORNER RAM BYTES

ENTER WITH A=COL2 STATUS

```
CHK COL2 20D1H 110040
          FE01
          281A
          FE02
          2813
          1E27
          FE04
20E0H 2810
          FE08
          2809
          11C84F
          FE10
          2805
          1E EF
DSPRD 3E H4
20F1H 12
          1A
          D3 FC
20F5H 76
```

```
LD DE, 4000H
CP 1
JRZ, DSPRD1
CP 2
JRZ, DSPRD
LD E, 27H
CP 4
JRZ, DSPRD1
CP 8
JRZ, DSPRD
LD DE, 4FC8H
CP 10H
JRZ, DSPRD1
LD E, EFH
LD A, 0100 0100
LD (DE), A
LD A, (DE)
OUT (FC), A
HALT
```

UP KEY PRESSED?  
IF SO, READ ONCE (4000H).  
BALCHECK II DISPLAY 55  
"MS" KEY PRESSED?  
IF SO, READ NEW BYTE (4000H).  
BALCHECK II DISPLAY 44  
KEY 8 PRESSED?  
IF SO, READ ONCE (4027H)  
BALCHECK II DISPLAY 55  
KEY 5 PRESSED?  
IF SO, READ NEW BYTE (4027H)  
BALCHECK II DISPLAY 44  
KEY 2 PRESSED?  
IF SO, READ ONCE (4FC8H).  
BALCHECK II DISPLAY 55  
LAST KEY 0 WAS PRESSED  
READ NEW BYTE (4FEFH).  
WRITE CORNER BYTE WITH 0100 0100  
BALCHECK II DISPLAY 44  
OUTPUT READ TO  
BALCHECK II DISPLAY.

CPU EXECUTES NOP'S, DI @ 0001H.