

# "AN IN-DEPTH LOOK AT..." SERIES SAMPLES

ATTACHED ARE 2 SAMPLES OF THE KIND OF INFO YOU CAN EXPECT TO SEE IN MY UPCOMING SERIES. INFO YOU WILL NOT FIND IN THE NUTTING MANUAL OR "THE BETTER BALLY BOOK".

FIRST, IS A PAGE FROM MY SENTRY SUB#66 DOC, A SENTRY STATUS BREAKDOWN LISTING. COLUMN 1 LISTS ALL THE DEVICES CHECKED BY SENTRY WHICH REPORTS (OUTPUTS) <sup>CHANGE IN</sup> THE DEVICE NUMBER IN THE Z80 REGISTER A AT EXIT. COLUMN 2 LISTS THE PRIORITY CHECK FROM TOP 1 TO BOTTOM 32. COLUMN 4 SHOWS ANY ADDITIONAL INFO, DURING A CHANGE REPORT, WHICH IS PLACED IN REGISTER B FOR A SINGLE CALL OR REGISTER B (CONTEXT BLOCK) IN A UPI MULTIPLE CALL. THE LAST COLUMN LISTS THE DATA BASES (BYTES) WHERE STATUS INFO ARE SAVED. THIS COLUMN LISTS HI-RES BASES. FOR LOW-RES, SIMPLY CHANGE THE 7 TO A 4.

SECOND, IS A 3 PAGE DESCRIPTION OF WRITE ABSOLUTE SUB#38. WHY THIS ROUTINE, WHICH IS THE FINAL LINK IN THE LINKED PATTERN WRITE ROUTINES 30 THRU 38, WAS INCLUDED AS A UPI CALL IS UNCLEAR TO ME. WRITE IS NEARLY USELESS BY ITSELF. IT WRITES ONLY TO SCREEN RAM, NOT MAGIC RAM. ITS USE IS VERY LIMITED. PERHAPS, WHOEVER PROGRAMMED THE UPI NEEDED JUST ONE MORE SUBROUTINE TO COMPLETE THE UPI SUBCALL TABLE AND CHOSE TO ADD "WRITE ABSOLUTE" TO THE TABLE.

# SENTRY STATUS BREAKDOWN

DEVICE NUMBER REG A (HEX)	DEVICE NAME (S=SWITCH)	PRIORITY (DECIMAL)	EXTENDED INFO REG B <sub>CB</sub> (HEX)	MEANING	DATA BASE USED								
0	SNUL	32	MEANINGLESS	NOTHING CHANGED	NONE								
1	SCT0	8	MEANINGLESS	CT0 FLAG SET (CT0 DEC TO 0)	7FDD <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr></table> C ← → C T 7            0 FLAGS SET BY DECCTS P.6-1 A FLAG FOUND SET IS RESET BY SENTRY	7	6	5	4	3	2	1	0
7	6	5	4	3		2	1	0					
2	SCT1	7		CT1 (CT1 )									
3	SCT2	6		CT2 (CT2 )									
4	SCT3	5		CT3 (CT3 )									
5	SCT4	4		CT4 (CT4 )									
6	SCT5	3		CT5 (CT5 )									
7	SCT6	2		CT6 (CT6 )									
8	SCT7	1		CT7 (CT7 )									
9	SFO	16	MEANINGLESS	SEMAPHORE 0 SET	7FDE <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr></table> S ← → S F 7            0 A FLAG FOUND SET IS RESET BY SENTRY	7	6	5	4	3	2	1	0
7	6	5	4	3		2	1	0					
A	SF1	15		1									
B	SF2	14		2									
C	SF3	13		3									
D	SF4	12		4									
E	SF5	11		5									
F	SF6	10		6									
10	SF7	9		7									
11	SSEC	21	MEANINGLESS	SSEC SET (4 SEC ELAPSED SINCE LAST SSEC)	SET BY TIMEZ, P FLAG FOUND SET IS RESET BY SENTRY								
12	SKYU	22	00 <sub>H</sub>	KEY WENT FROM DOWN TO UP	7FE3 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr></table> S ← → S C            C KEY VALUE SEE P.	7	6	5	4	3	2	1	0
7	6	5	4	3		2	1	0					
13	SKYD	23	NEW KEY VALUE SEE P.	KEY IS DOWN *									
14	STO	24	NEW TRIG VALUE 00 <sub>H</sub> =OFF 10 <sub>H</sub> =ON	TRIGGER 0 CHANGED*	7FE4 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr></table> LAST JOY/TRIG STATUS TRIG RIGHT LEFT DOWN UP	7	6	5	4	3	2	1	0
7	6	5	4	3	2	1	0						
15	SJO	25	NEW JOY VALUE	JOYSTICK 0	7FE5 SAME LAST AS JOY/TRIG ABOVE STATUS 7FE6 7FE7 7FDF LAST POT STATUS 7FE0 7FE1 7FE2								
16	ST1	26	TRIG	TRIGGER 1									
17	SJ1	27	JOY	JOYSTICK 1									
18	ST2	28	TRIG	TRIGGER 2									
19	SJ2	29	JOY	JOYSTICK 2									
1A	ST3	30	TRIG	TRIGGER 3									
1B	SJ3	31	JOY	JOYSTICK 3									
1C	SPO	17	NEW POT VALUE	POT 0 CHANGED*									
1D	SP1	18		1									
1E	SP2	19		2									
1F	SP3	20		3									

\* NEW VALUE IN REGISTER B<sub>CB</sub> IS SAVED IN DATA BASE SHOWN. THE POTS ARE DEBOUNCED. A NO CHANGE FOR A POT IS DEFINED AS OLD < NO CHANGE < OLD+7.

## WRITE ABSOLUTE

ENTER WITH: DE = SCREEN ADDRESS\*

\* THE SCREEN LOCATION IN WHICH THE PATTERN WRITING BEGINS

C = X SIZE (PATTERN BYTE WIDES)  
B = Y SIZE (PATTERN LINES HIGH)  
A = MAGIC REGISTER VALUE

7	6	5	4	3	2	1	0
X	X	X	X	X	X	X	X

X = DON'T CARE

→ EXPAND

→ FLOP

HL = PATTERN ADDRESS

SUBROUTINE NAME: WRITA

### DESCRIPTION

THE PATTERN IS PLOPPED TO SCREEN RAM NOT MAGIC RAM. NO OUTPUT TO THE MAGIC REGISTER (PORT 0CH) IS MADE. THE MAGIC REGISTER VALUE IS USED ONLY TO REQUEST A FLOP ONLY, FLOP, EXPAND OR EXPAND WITH A FLOP WRITE.

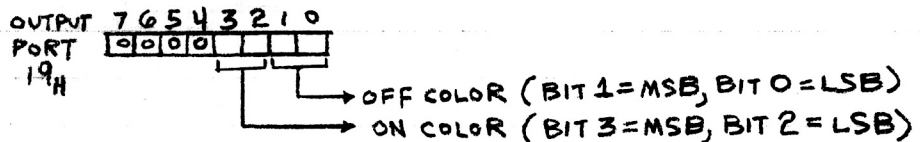
### WRITE PATTERN COMMENTS

- FLOP ONLY PATTERN ① MAGIC REGISTER VALUE AT ENTRY = 00H.  
THE PATTERN IS PLOPPED BYTE BY BYTE INTO SCREEN RAM. ONE BYTE 00H TO THE RIGHT OF EACH PATTERN LINE WRITTEN IS ALSO PLOPPED INTO SCREEN RAM.
- FLOP, PLOP PATTERN ② MAGIC REGISTER VALUE AT ENTRY = 40H.  
THE PLOPPING OF THE PATTERN IS FLOPPED INTO SCREEN RAM. THE SCREEN PATTERN ITSELF IS NOT FLOPPED. THE USER IS THEREFORE REQUIRED TO SUPPLY THE CORRECT PATTERN. ONE BYTE 00H TO THE LEFT OF EACH PATTERN

LINE WRITTEN IS ALSO PLOPPED INTO SCREEN RAM.

EXPAND, PLOP PATTERN ③ MAGIC REGISTER VALUE AT ENTRY =  $08_H$ .

THE PATTERN IS EXPANDED BIT BY BIT. A PATTERN BIT SET (1) IS EXPANDED INTO A PIXEL DEFINED BY BITS 3 AND 2 OF THE EXPAND REGISTER (OUTPUT PORT  $19_H$ ). A PATTERN BIT NOT SET (0) IS EXPANDED INTO A PIXEL DEFINED BY BITS 1 AND 0 OF THE EXPAND REGISTER. THE EXPAND REGISTER IS USER SET AS DESIRED PRIOR TO ENTRY OF WRITA.



WRITA ASSUMES THAT THE EXPAND FLIP-FLOP IS RESET AT ENTRY (REFERENCE "SYSTEM DESCRIPTION P"). THE EXPAND REQUEST IS USED TO STORE A TWO COLOR PATTERN IN HALF THE NORMAL MEMORY SPACE. TWO BYTES OF  $00_H$  TO THE RIGHT OF EACH PATTERN LINE WRITTEN ARE ALSO PLOPPED INTO SCREEN RAM.

EXPAND, FLOP, PLOP PATTERN ④ MAGIC REGISTER VALUE AT ENTRY =  $48_H$

THE PATTERN IS EXPANDED AND FLOPPED AS DESCRIBED IN THE ABOVE ③ AND ② RESPECTIVELY. TWO BYTES OF  $00_H$  TO THE LEFT OF EACH PATTERN LINE WRITTEN ARE ALSO PLOPPED INTO SCREEN RAM.

DATA BASES USED: NONE

