

```

LOCATION OBJECT CODE LINE SOURCE LINE
0840 010011 LD BC,XP,BK
0843 09 ADD HL,BC
0844 010008 LD BC,11
0847 7A IF [D,LT,70H]
0848 FE70 LD A,D
084A 3005 CP 70H
084C CD1FD0 JR NC,ELSE5
084F 1802 CALL WRITE_VRAM
0851 ELSE5 JR END5
0851 ED80 ;MOVE TO CPU RAM
0853 DOE5 LDIR
0853 D0E5 ;ENDIF
0855 ED58006 LD DE,[WORK_BUFFER]
0859 210013 LD HL,VP,BK+1
085C 19 ADD HL,DE
085D E8 EX DE,HL
085E 010014 LD BC,BK,PTM-8
0861 09 ADD HL,BC
0862 0609 DO B,9
0864 1A LD B,9
0865 13 LD A,[DE]
0866 D5 INC DE
0867 110008 PUSH DE
086A 19 LD DE,B
0868 E5 ADD HL,DE
086C 5F LD E,A
086D 1600 LD D,0
086F 4F LD C,A
0871 3E09 PUSH BC
0873 90 LD A,9
0874 0600 SUB B
0876 D603 LD B,0
0878 3A03 JR C,PM51
087A 04 INC B
087D 78 JR PM52
087E FD2A8006 LD A,B
0882 FD8612 LD IY,[WORK_BUFFER]
0885 FDC8037E ADD A,[IY+YP,BK]
0889 FD210001 BIT 7,[IY+FLAGS]
0880 2029 LD IY,1
088F 3E03 ;IF [PSW,IS,ZERO]
0891 CD1F8B JR NZ,ELSE6
0894 C1 LD A,3
0895 2A8006 CALL GET_VRAM
0898 C5 POP BC
0899 110084 LD HL,[WORK_BUFFER]
089C 19 PUSH BC
089D 59 LD DE,BK,CLR
089E 19 ADD HL,DE
089D 59 LD E,C

0853 D0E5 PUSH IX ;SAVE GRAPHICS POINTER
0855 ED58006 ;DE := BUFFER BASE
0859 210013 ;DISPLACEMENT TO BEGINNING OF BKGD NAMES
085C 19 ADD HL,DE
085D E8 ;DE POINT TO FIRST BACKGROUND NAME
085E 010014 LD BC,BK,PTM-8
0861 09 ADD HL,BC ;HL POINTS TO BACKGROUND PAT GEN STORAGE AREA -8
0862 0609 DO B,9 ; WILL BE INCREMENTED BEFORE FIRST USE
0864 1A LD B,9
0865 13 LD A,[DE] ;GET NAME
0866 D5 INC DE ;POINT TO NEXT NAME
0867 110008 PUSH DE ;SAVE NAME POINTER
086A 19 LD DE,B ;INCREMENT BUFFER POINTER
0868 E5 ADD HL,DE ;SAVE BUFFER POINTER
086C 5F LD E,A ;INDEX INTO PATTERN TABLE
086D 1600 LD D,0 ;SAVE PATTERN NAME IN C
086F 4F LD C,A ;SAVE COUNTER
0871 3E09 PUSH BC ;ADD [B-9]/3 TO YP_BK TO FIND
0873 90 LD A,9 ; OUT WHICH THIRD OF TABLES ARE IN
0874 0600 SUB B
0876 D603 LD B,0
0878 3A03 JR C,PM51
087A 04 INC B
087D 78 JR PM52
087E FD2A8006 LD A,B
0882 FD8612 LD IY,[WORK_BUFFER]
0885 FDC8037E ADD A,[IY+YP,BK]
0889 FD210001 BIT 7,[IY+FLAGS]
0880 2029 LD IY,1
088F 3E03 ;IF [PSW,IS,ZERO]
0891 CD1F8B JR NZ,ELSE6
0894 C1 LD A,3
0895 2A8006 CALL GET_VRAM
0898 C5 POP BC
0899 110084 LD HL,[WORK_BUFFER]
089E 19 PUSH BC
089D 59 LD DE,BK,CLR
089E 19 ADD HL,DE
089D 59 LD E,C
;DISPLACEMENT TO COLOR GEN AREA
;POINT TO IT
;PATTERN NAME

```

```

LOCATION OBJECT CODE LINE SOURCE LINE
089E CB38 3387 SRL E
08A0 CB38 3388 SRL E ;DIVIDE NAME BY 8
08A2 CB38 3389 SRL E
08A4 1600 3390 LD D,0
08A6 3E09 3391 LD A,9 ;CALC POSITION IN BUFFER TO MOVE GEN TO
08A8 90 3392 SUB B
08A9 4F 3393 LD C,A
08AA 0600 3394 LD B,0
08AC 09 3395 ADD HL,BC
08AD FD210001 3396 LD IY,1
08B1 3E04 3397 LD A,4 ;COLOR GENERATOR TABLE CODE
08B3 CD1FB8 3398 CALL GET_VRAM
08B6 1821 3399 JR EMD6
08B8 3400 ELSE6
08B8 CB2F 3401 SRA A ;MUST BE MODE 11
08BA CB2F 3402 SRA A ;DIVIDE Y_POS BY 8
08BC CB2F 3403 SRA A
08BE FE03 3404 PH7 ;IF [A,LT,3]
08C0 3017 3405 CP 3
08C2 57 3406 JR NC,EMD7
08C3 D5 3407 LD D,A
08C4 E5 3408 PUSH DE
08C5 3E03 3409 PUSH HL
08C7 CD1FB8 3410 LD A,3 ;PATTERN GENERATOR TABLE CODE
08CA E1 3411 CALL GET_VRAM
08CB 110068 3412 POP HL
08CE 19 3413 LD DE,BK CLR-BK_PTN ;HL := PATTERN BUFFER ADDRESS
08CF D1 3414 ADD HL,DE ;DISPLACEMENT BETWEEN PATTERN AND COLOR BUFFERS
08D0 FD210001 3415 POP DE ;HL := POINTER TO COLOR BUFFER
08D4 3E04 3416 LD IY,1
08D6 CD1FB8 3417 LD A,4 ;CODE FOR COLOR TABLE
08D9 3418 CALL GET_VRAM
08D9 EMD7 3419 ;EMD7
08D9 EMD6 3420 ;EMD6
08D9 C1 3421 POP BC ;RESTORE REGISTERS
08DA E1 3422 POP HL
08DB D1 3423 POP DE
08DC 1086 3424 ; ENDD0
3425 DJNZ DLP2
3426
3427 ; NOW THE PATTERN AND COLOR GENERATORS ARE IN THEIR RESPECTIVE BUFFERS
3428 ; SO GET THE FOUR GENERATORS FOR THIS FRAME OF THE OBJECT
3429
08DE DDE1 3430 POP IX ;RESTORE GRAPHICS POINTER
3431
08E0 D9 3432 PM8
08E1 D05603 3433 EXX
08E4 D05E02 3434 LD D,[IX+3]
08E7 D04605 3435 LD E,[IX+2] ;DE' := NEW_GEN
08EA D04E04 3436 LD B,[IX+5]
08ED D9 3437 LD C,[IX+4] ;BC' := PIRN_POINTER
08EE D0E5 3438 PUSH IX
08F0 E1 3439 POP HL ;HL := ADDRESS OF GRAPHICS
08F1 FD2A8006 3440 LD IY,[WORK_BUFFER]
08F5 FD7E04 3441 LD A,[IY+FRM] ;A := FRM #
08F8 87 3442 ADD A,A ;XZ
08F9 4F 3443 LD C,A
    
```

LOCATION OBJECT CODE LINE SOURCE LINE

```

08FA 0600 3444 LD B,0
08FC 110006 3445 LD DE,6
08FF 19 3446 ADD HL,DE
0C00 09 3447 ADD HL,BC
0C01 5E 3448 LD E,(HL)
0C02 23 3449 INC HL
0C03 56 3450 LD D,(HL)
0C04 2A0006 3451 LD HL,(WORK_BUFFER)
0C07 01007C 3452 LD BC,(OBJ_PTN+24)
0C0A 09 3453 ADD HL,BC
0C0B E5 3454 PUSH HL
0C0C C5 3455 PUSH BC
0C0D 010005 3456 LD BC,5
0C10 7A 3457 IF L,D,LT,70H)
0C11 FE70 3458 LD A,D
0C13 3005 3459 CP 70H
0C15 CD1FE2 3460 JR NC,ELSE8
0C18 1803 3461 CALL READ_VRAM
0C1A 3462 JR END8
0C1A EB 3463 ELSE8
0C1A EB 3464 EX DE,HL
0C1B ED80 3465 LDIR
0C1D FD2A0006 3466 END8
0C21 C1 3467 LD IY,(WORK_BUFFER)
0C22 FD09 3468 POP BC
0C24 FD7E04 3469 ADD IY,BC
0C27 FD2A0006 3470 LD A,(IY+4)
0C28 FD7702 3471 LD IY,(WORK_BUFFER)
0C2E D1 3472 LD [IY+COL],A
0C2F 2A0006 3473 POP DE
0C32 010064 3474 LD HL,(WORK_BUFFER)
0C35 09 3475 LD BC,(OBJ_PTN)
0C36 0604 3476 ADD HL,BC
0C38 1A 3477 DO B,4
0C39 00BE01 3478 LD B,4
0C3C D5 3479 DLP4
3481 CP [IX+1]
3482 PUSH DE
3483 IF (PSW,IS,CARRY)
3484 JR NC,ELSE9
3485 EXX
3486 ADD A,A
3487 ADD A,A
3488 ADD A,A
3489 LD L,A
3490 LD H,0
3491 ADD HL,BC
3492 PUSH HL
3493 EXX
3494 POP DE
3495 EX DE,HL
3496 PUSH BC
3497 LD BC,B
3498 LDIR
3499 POP BC
3500 EX DE,HL
3501 JR END9
;FRAME POINTERS START AT THIS OFFSET
;HL := POINTS TO FRAME_POINTER FOR FRAME
;DE := ADDRESS OF FRAME NAMES
;HL := BUFFER BASE ADDRESS
;USE LOCATION FOR LAST GEN TO STORE NAMES
;SAVE FOR LATER USE
;SAVE OFFSET
;THEN NAMES ARE IN VRAM
;GET THE 4 NAMES
;NAMES IN CPU MEMORY SPACE
;GET COLOR BYTE
;OFFSET TO FIRST NAME
;A := COLOR BYTE
;COLR := COLOR BYTE
;DE := ADDRESS OF FIRST NAME IN BUFFER
;HL := BUFFER BASE ADDRESS
;START OF OBJECT'S PATTERN BUFFER
;GET 4 PATTERNS CORRESPONDING TO THE FOUR NAMES
;GET NAME
;COMPARE TO NUNGEN
;SAVE POINTER TO NAMES
;THEN NAME < NUNGEN, THEREFORE
; PATTERN PART OF GRAPHICS TABLES
;A := 8*NAME
;HL := POINTER TO PATTERN
;DE := " "
;NUMBER OF BYTES TO MOVE

```

```

LOCATION OBJECT CODE LINE SOURCE LINE
00B8 CB3F 3729 SRL A
00BA CB3F 3730 SRL A
00BC CB3F 3731 SRL A
00BE 57 3732 LD D,A
00BF 59 3733 LD E,C
00C0 05 3734 PUSH DE
00C1 01001C 3735 LD BC,BK PTN
00C4 09 3736 ADD HL,BC
00C5 E04B0006 3737 LD BC,(WORK_BUFFER) ;GET BUFFER BASE ADDR
00C9 09 3738 ADD HL,BC
00CA E5 3739 PUSH HL
00CB FD210003 3740 LD IY,3
00CF 3E03 3741 LD A,3
00D1 CD1FBE 3742 CALL PUT_VRAM
00D4 E1 3743 POP HL
00D5 110048 3744 LD DE,BK CLR-BK PTN ;GET POINTER BACK
00D8 19 3745 ADD HL,DE ;OFFSET BETWEEN BUFFERS
00D9 01 3746 POP DE ;HL POINTS TO START OF NEXT 3 COLOR GENERATORS
00DE 3E04 3747 LD IY,3 ;GET INDEX INTO GEN TABLES
00E0 CD1FBE 3748 LD A,4 ;CODE FOR COLOR GENERATOR TABLE
00E3 3750 EMD15 ;EMD15
00E3 C1 3751 POP BC ;RESTORE COUNTER AND INDEX
00E4 04 3752 INC B
00E5 78 3753 CP 3
00E6 FE03 3754 JR NZ,RPT2
00E8 20AE 3755 UNTIL [B,E0,3] ;REPEAT 3 TIMES
00EA 3756 ;EMD16
00EE FD4606 3757 EMD04 ;EMD04
00F1 78 3758
00F2 FE80 3759 ; RESTORE OLD SCREEN IF IT'S Y_PAT_POS AND X_PAT_POS DIFFERS FROM THE
00F4 2821 3760 ; Y_PAT_POS AND X_PAT_POS FOR THE OBJECT
00F6 FD4E07 3761 PM14 LD IY,(WORK_BUFFER)
00F9 FD6611 3762 LD B,[IY+XP_OS] ;TEST FOR VALID OLD SCREEN DATA
00FF 87 3763 IF [B,NE,80H] ;THEN THERE IS VALID DATA
0000 ED42 3764 LD A,B
0001 78 3765 CP 80H
0002 2821 3766 JR Z,EMD16
0003 FD4E07 3767 LD C,[IY+YP_OS]
0004 FD6611 3768 LD H,[IY+XP_BK]
0005 FD6E12 3769 LD L,[IY+YP_BK]
0006 87 3770 OR A
0007 ED42 3771 SBC HL,BC
0008 2813 3772 IF [PSM,IS,MZERO]
0009 2A8006 3773 JR Z,END17
0010 110008 3774 LD HL,(WORK_BUFFER)
0011 19 3775 LD DE,YP_OS+1
0012 FD5E06 3776 ADD HL,DE
0013 FD5607 3777 LD E,[IY+XP_OS]
0014 010303 3778 LD D,[IY+YP_OS]
0015 CD0808 3779 LD BC,0303H
0016 3780 CALL PUTFRAME
0017 3781 EMD17 ;EMD17
0018 3782 EMD16 ;EMD16
0019 FD2A8006 3783 ; PLACE OBJECT ON SCREEN
0020 3784 LD IY,(WORK_BUFFER)
0021 3785 LD HL,(WORK_BUFFER) ;HL := BUFFER BASE ADDR

```

```

;THIS NUMBER / B INDICATES WHICH 1/3 OF
; TABLES TO USE

;DE := INDEX INTO PATTERN AND COLOR TABLES
;SAVE INDEX
;FORM POINTER TO GENERATORS IN HL

;GET BUFFER BASE ADDR

;SAVE THIS POINTER
;NUMBER OF ELEMENTS TO MOVE
;PATTERN GENERATOR TABLE CODE

;GET POINTER BACK
;OFFSET BETWEEN BUFFERS
;HL POINTS TO START OF NEXT 3 COLOR GENERATORS
;GET INDEX INTO GEN TABLES

;CODE FOR COLOR GENERATOR TABLE

;RESTORE COUNTER AND INDEX

;TEST IF OS POSITION SAME AS CURRENT POSITION

;CLEAR THE CARRY
;IS THERE ANY DIFFERENCE?
;THEN POSITION HAS CHANGED

;GET BUFFER BASE
;POINT TO OLD_SCREEN NAMES

;DE := X AND Y PAT POS
;BC := X AND Y EXTENT

```

```

LOCATION OBJECT CODE LINE SOURCE LINE
0E1E 110013 3786 LD DE,YP,BK+1 ;POINT TO NAMES FOR OBJECT
0E21 19 3787 ADD HL,DE
0E22 FD5E11 3788 LD E,[Y+XP,BK]
0E25 FD5612 3789 LD D,[Y+YP,BK]
0E28 010303 3790 LD BC,0303H
0E28 CD0808 3791 CALL PUTFRAME
0E2E C9 3792 ***** END OF PUT_MOBILE *****
3793 RET
3794
3795 ; REGS A, H AND L CONTAIN 24 BIT PATTERN TO BE COMBINED WITH BACKGROUND GENERATORS
3796 ; IX POINTS TO THE FIRST OF THREE GENERATOR BYTES TO BE COMBINED WITH A, H AND L
3797 COM_PAT_COL
3798 BIT 0,[Y+FLAGS]
3799 ; IF (PSW,IS,ZERO)
3800 JR NZ,ELSE1B
3801 ;OR' GENS OR REPLACE
3802 ;THEN 'OR'
3803 ;OR' LEFT BYTE WITH BACKGROUND
3804 ;AND SUBSTITUTE FOR THAT GENERATOR BYTE
3805 ;NOW DO MIDDLE BYTE
3806 ;RIGHT HAND BYTE
3807
3808 ;REPLACE BACKGROUND WITH NON-ZERO BYTES
3809 ;IS BYTE NON-ZERO
3810 ELSE1B ;ELSE
3811 OR A
3812 IF (PSW,IS,ZERO)
3813 JR Z,END19
3814 LD [IX+0],A
3815 ;ENDIF
3816 LD A,H
3817 OR A
3818 ; IF (PSW,IS,ZERO)
3819 JR Z,END20
3820 LD [IX+8],A
3821 ;ENDIF
3822 LD A,L
3823 OR A
3824 ; IF (PSW,IS,ZERO)
3825 JR Z,END21
3826 LD [IX+16],A
3827 ;ENDIF
3828 END18
3829 PM15
3830 ;
3831 ;
3832 ;
3833 ;
3834 ;
3835 ;
3836 ;
3837 ;
3838 ;
3839 ;
3840 ;
3841 ELSE23
3842 ;ELSE
3843 LD C,0
3844 ;MASK REPLACE COLORU WITH TRANSPARENT
3845 ;MASK REPLACE COLORU WITH TRANSPARENT
3846 ;MASK REPLACE COLORU WITH TRANSPARENT
3847 ;MASK REPLACE COLORU WITH TRANSPARENT
3848 ;MASK REPLACE COLORU WITH TRANSPARENT
3849 ;MASK REPLACE COLORU WITH TRANSPARENT
3850 ;MASK REPLACE COLORU WITH TRANSPARENT
3851 ;MASK REPLACE COLORU WITH TRANSPARENT
3852 ;MASK REPLACE COLORU WITH TRANSPARENT
3853 ;MASK REPLACE COLORU WITH TRANSPARENT
3854 ;MASK REPLACE COLORU WITH TRANSPARENT
3855 ;MASK REPLACE COLORU WITH TRANSPARENT
3856 ;MASK REPLACE COLORU WITH TRANSPARENT
3857 ;MASK REPLACE COLORU WITH TRANSPARENT
3858 ;MASK REPLACE COLORU WITH TRANSPARENT
3859 ;MASK REPLACE COLORU WITH TRANSPARENT
3860 ;MASK REPLACE COLORU WITH TRANSPARENT
3861 ;MASK REPLACE COLORU WITH TRANSPARENT
3862 ;MASK REPLACE COLORU WITH TRANSPARENT
3863 ;MASK REPLACE COLORU WITH TRANSPARENT
3864 ;MASK REPLACE COLORU WITH TRANSPARENT
3865 ;MASK REPLACE COLORU WITH TRANSPARENT
3866 ;MASK REPLACE COLORU WITH TRANSPARENT
3867 ;MASK REPLACE COLORU WITH TRANSPARENT
3868 ;MASK REPLACE COLORU WITH TRANSPARENT
3869 ;MASK REPLACE COLORU WITH TRANSPARENT
3870 ;MASK REPLACE COLORU WITH TRANSPARENT
3871 ;MASK REPLACE COLORU WITH TRANSPARENT
3872 ;MASK REPLACE COLORU WITH TRANSPARENT
3873 ;MASK REPLACE COLORU WITH TRANSPARENT
3874 ;MASK REPLACE COLORU WITH TRANSPARENT
3875 ;MASK REPLACE COLORU WITH TRANSPARENT
3876 ;MASK REPLACE COLORU WITH TRANSPARENT
3877 ;MASK REPLACE COLORU WITH TRANSPARENT
3878 ;MASK REPLACE COLORU WITH TRANSPARENT
3879 ;MASK REPLACE COLORU WITH TRANSPARENT
3880 ;MASK REPLACE COLORU WITH TRANSPARENT
3881 ;MASK REPLACE COLORU WITH TRANSPARENT
3882 ;MASK REPLACE COLORU WITH TRANSPARENT
3883 ;MASK REPLACE COLORU WITH TRANSPARENT
3884 ;MASK REPLACE COLORU WITH TRANSPARENT
3885 ;MASK REPLACE COLORU WITH TRANSPARENT
3886 ;MASK REPLACE COLORU WITH TRANSPARENT
3887 ;MASK REPLACE COLORU WITH TRANSPARENT
3888 ;MASK REPLACE COLORU WITH TRANSPARENT
3889 ;MASK REPLACE COLORU WITH TRANSPARENT
3890 ;MASK REPLACE COLORU WITH TRANSPARENT
3891 ;MASK REPLACE COLORU WITH TRANSPARENT
3892 ;MASK REPLACE COLORU WITH TRANSPARENT
3893 ;MASK REPLACE COLORU WITH TRANSPARENT
3894 ;MASK REPLACE COLORU WITH TRANSPARENT
3895 ;MASK REPLACE COLORU WITH TRANSPARENT
3896 ;MASK REPLACE COLORU WITH TRANSPARENT
3897 ;MASK REPLACE COLORU WITH TRANSPARENT
3898 ;MASK REPLACE COLORU WITH TRANSPARENT
3899 ;MASK REPLACE COLORU WITH TRANSPARENT
3900 ;MASK REPLACE COLORU WITH TRANSPARENT
3901 ;MASK REPLACE COLORU WITH TRANSPARENT
3902 ;MASK REPLACE COLORU WITH TRANSPARENT
3903 ;MASK REPLACE COLORU WITH TRANSPARENT
3904 ;MASK REPLACE COLORU WITH TRANSPARENT
3905 ;MASK REPLACE COLORU WITH TRANSPARENT
3906 ;MASK REPLACE COLORU WITH TRANSPARENT
3907 ;MASK REPLACE COLORU WITH TRANSPARENT
3908 ;MASK REPLACE COLORU WITH TRANSPARENT
3909 ;MASK REPLACE COLORU WITH TRANSPARENT
3910 ;MASK REPLACE COLORU WITH TRANSPARENT
3911 ;MASK REPLACE COLORU WITH TRANSPARENT
3912 ;MASK REPLACE COLORU WITH TRANSPARENT
3913 ;MASK REPLACE COLORU WITH TRANSPARENT
3914 ;MASK REPLACE COLORU WITH TRANSPARENT
3915 ;MASK REPLACE COLORU WITH TRANSPARENT
3916 ;MASK REPLACE COLORU WITH TRANSPARENT
3917 ;MASK REPLACE COLORU WITH TRANSPARENT
3918 ;MASK REPLACE COLORU WITH TRANSPARENT
3919 ;MASK REPLACE COLORU WITH TRANSPARENT
3920 ;MASK REPLACE COLORU WITH TRANSPARENT
3921 ;MASK REPLACE COLORU WITH TRANSPARENT
3922 ;MASK REPLACE COLORU WITH TRANSPARENT
3923 ;MASK REPLACE COLORU WITH TRANSPARENT
3924 ;MASK REPLACE COLORU WITH TRANSPARENT
3925 ;MASK REPLACE COLORU WITH TRANSPARENT
3926 ;MASK REPLACE COLORU WITH TRANSPARENT
3927 ;MASK REPLACE COLORU WITH TRANSPARENT
3928 ;MASK REPLACE COLORU WITH TRANSPARENT
3929 ;MASK REPLACE COLORU WITH TRANSPARENT
3930 ;MASK REPLACE COLORU WITH TRANSPARENT
3931 ;MASK REPLACE COLORU WITH TRANSPARENT
3932 ;MASK REPLACE COLORU WITH TRANSPARENT
3933 ;MASK REPLACE COLORU WITH TRANSPARENT
3934 ;MASK REPLACE COLORU WITH TRANSPARENT
3935 ;MASK REPLACE COLORU WITH TRANSPARENT
3936 ;MASK REPLACE COLORU WITH TRANSPARENT
3937 ;MASK REPLACE COLORU WITH TRANSPARENT
3938 ;MASK REPLACE COLORU WITH TRANSPARENT
3939 ;MASK REPLACE COLORU WITH TRANSPARENT
3940 ;MASK REPLACE COLORU WITH TRANSPARENT
3941 ;MASK REPLACE COLORU WITH TRANSPARENT
3942 ;MASK REPLACE COLORU WITH TRANSPARENT
3943 ;MASK REPLACE COLORU WITH TRANSPARENT
3944 ;MASK REPLACE COLORU WITH TRANSPARENT
3945 ;MASK REPLACE COLORU WITH TRANSPARENT
3946 ;MASK REPLACE COLORU WITH TRANSPARENT
3947 ;MASK REPLACE COLORU WITH TRANSPARENT
3948 ;MASK REPLACE COLORU WITH TRANSPARENT
3949 ;MASK REPLACE COLORU WITH TRANSPARENT
3950 ;MASK REPLACE COLORU WITH TRANSPARENT
3951 ;MASK REPLACE COLORU WITH TRANSPARENT
3952 ;MASK REPLACE COLORU WITH TRANSPARENT
3953 ;MASK REPLACE COLORU WITH TRANSPARENT
3954 ;MASK REPLACE COLORU WITH TRANSPARENT
3955 ;MASK REPLACE COLORU WITH TRANSPARENT
3956 ;MASK REPLACE COLORU WITH TRANSPARENT
3957 ;MASK REPLACE COLORU WITH TRANSPARENT
3958 ;MASK REPLACE COLORU WITH TRANSPARENT
3959 ;MASK REPLACE COLORU WITH TRANSPARENT
3960 ;MASK REPLACE COLORU WITH TRANSPARENT
3961 ;MASK REPLACE COLORU WITH TRANSPARENT
3962 ;MASK REPLACE COLORU WITH TRANSPARENT
3963 ;MASK REPLACE COLORU WITH TRANSPARENT
3964 ;MASK REPLACE COLORU WITH TRANSPARENT
3965 ;MASK REPLACE COLORU WITH TRANSPARENT
3966 ;MASK REPLACE COLORU WITH TRANSPARENT
3967 ;MASK REPLACE COLORU WITH TRANSPARENT
3968 ;MASK REPLACE COLORU WITH TRANSPARENT
3969 ;MASK REPLACE COLORU WITH TRANSPARENT
3970 ;MASK REPLACE COLORU WITH TRANSPARENT
3971 ;MASK REPLACE COLORU WITH TRANSPARENT
3972 ;MASK REPLACE COLORU WITH TRANSPARENT
3973 ;MASK REPLACE COLORU WITH TRANSPARENT
3974 ;MASK REPLACE COLORU WITH TRANSPARENT
3975 ;MASK REPLACE COLORU WITH TRANSPARENT
3976 ;MASK REPLACE COLORU WITH TRANSPARENT
3977 ;MASK REPLACE COLORU WITH TRANSPARENT
3978 ;MASK REPLACE COLORU WITH TRANSPARENT
3979 ;MASK REPLACE COLORU WITH TRANSPARENT
3980 ;MASK REPLACE COLORU WITH TRANSPARENT
3981 ;MASK REPLACE COLORU WITH TRANSPARENT
3982 ;MASK REPLACE COLORU WITH TRANSPARENT
3983 ;MASK REPLACE COLORU WITH TRANSPARENT
3984 ;MASK REPLACE COLORU WITH TRANSPARENT
3985 ;MASK REPLACE COLORU WITH TRANSPARENT
3986 ;MASK REPLACE COLORU WITH TRANSPARENT
3987 ;MASK REPLACE COLORU WITH TRANSPARENT
3988 ;MASK REPLACE COLORU WITH TRANSPARENT
3989 ;MASK REPLACE COLORU WITH TRANSPARENT
3990 ;MASK REPLACE COLORU WITH TRANSPARENT
3991 ;MASK REPLACE COLORU WITH TRANSPARENT
3992 ;MASK REPLACE COLORU WITH TRANSPARENT
3993 ;MASK REPLACE COLORU WITH TRANSPARENT
3994 ;MASK REPLACE COLORU WITH TRANSPARENT
3995 ;MASK REPLACE COLORU WITH TRANSPARENT
3996 ;MASK REPLACE COLORU WITH TRANSPARENT
3997 ;MASK REPLACE COLORU WITH TRANSPARENT
3998 ;MASK REPLACE COLORU WITH TRANSPARENT
3999 ;MASK REPLACE COLORU WITH TRANSPARENT
4000 ;MASK REPLACE COLORU WITH TRANSPARENT

```

LOCATION	OBJECT CODE	LINE	SOURCE LINE
0E7B		3043	END23 ;EMDIF
0E7C	7B	3044	LD A,E
0E7C	87	3045	OR A
		3046	IF [PSW,IS,NZERO]
0E7D	2808	3047	JR Z,END24
0E7F	D07E00	3048	LD A,[IX+0]
0E82	A1	3049	AND C
0E83	B0	3050	OR B
0E84	D07700	3051	LD [IX+0],A
0E87	7C	3052	END24 ;EMDIF
0E88	87	3053	LD A,H
		3054	OR A
		3055	IF [PSW,IS,NZERO]
0E89	2808	3056	JR Z,END25
0E8B	D07E08	3057	LD A,[IX+8]
0E8E	A1	3058	AND C
0E8F	B0	3059	OR B
0E90	D07708	3060	LD [IX+8],A
0E93	7D	3061	END25 ;EMDIF
0E94	87	3062	LD A,L
		3063	OR A
		3064	IF [PSW,IS,NZERO]
0E95	2808	3065	JR Z,END26
0E97	D07E10	3066	LD A,[IX+16]
0E9A	A1	3067	AND C
0E9B	B0	3068	OR B
0E9C	D07710	3069	LD [IX+16],A
0E9F		3070	END26 ;EMDIF
0E9F	D0E1	3071	POP IX
0EA1	C9	3072	END22 ;EMDIF
		3073	RET
		3074	EMO
		3075	PROG ;PUT_MOBILE

;GET FIRST OBJECT'S PATTERN BYTE  
;ARE THERE ANY '1' BITS ?

;GET BACKGROUND COLOR GEN  
;MASK OUT COLOR1  
;ADD OBJECT COLOR1  
;UPDATE COLOR GENERATOR

;SAME FOR MIDDLE BYTE

;RIGHT HAND BYTE

;RESTORE BACKGROUND POINTER

LOCATION OBJECT CODE LINE      SOURCE LINE

```

3877 ; .IDENT PUTCOMP
3878 ; .ZOP
3879 ; .EPOP
3880 ; .IF1 ,.INSERT B:SPZBO.ASH
3881 ; .COMMENT )
3882 ;
3883 ;
3884 ; ***** PUT_COMPLEX *****
3885 ;
3886 ; DESCRIPTION: THE POSITION AND FRAME NUMBER OF EACH OF A COMPLEX OBJECT'S
3887 ; COMPONENT OBJECTS IS UPDATED. THEN PUT_OBJECT IS CALLED FOR
3888 ; EACH OF THE COMPONENT OBJECTS.
3889 ;
3890 ; INPUT:
3891 ; IX = ADDRESS OF OBJECT TO BE PROCESSED
3892 ; HL = ADDRESS OF OBJECT'S GRAPHICS TABLES IN ROM
3893 ; B = SELECTOR FOR METHODE OF COMBINING OBJECT GENERATORS
3894 ; WITH BACKGROUND GENERATORS
3895 ;
3896 ; 1 = OBJECT PATTERN GENS Ored WITH BACKGROUND PATTERN GENS
3897 ; COLOR1 OF BACKGROUND CHANGED TO MOBILE OBJECT'S COLOR
3898 ; IF CORRESPONDING PATTERN BYTE NOT ZERO
3899 ;
3900 ; 2 = REPLACE BACKGROUND PATTERN GENS WITH OBJECT PATTERN GENS
3901 ; TREAT COLOR SAME AS #1
3902 ;
3903 ; 3 = SAME AS #1 EXCEPT COLORO CHANGED TO TRANSPARENT
3904 ;
3905 ; 4 = SAME AS #2 EXCEPT COLORO CHANGED TO TRANSPARENT
3906 ;
3907 ; C = OBJECT TYPE, AND NUMBER OF COMPONENTS
3908 ; *****
3909 ; )
3910 ; EXT PUTOBJ
3911 ; GLB PUTCOMPLEX
3912 ;
3913 ; PUTCOMPLEX
3914 ; UPDATE THE FRAME NUMBER AND THE X AND Y LOCATION IN EACH OF THE COMPONENT
3915 ; OBJECT'S STATUS AREAS
3916 ; PUSH BC
3917 ; EXX
3918 ; LD H, [IX+3]
3919 ; LD L, [IX+2]
3920 ; LD A, [HL]
3921 ; INC HL
3922 ; LD C, [HL]
3923 ; INC HL
3924 ; LD B, [HL]
3925 ; INC HL
3926 ; LD E, [HL]
3927 ; INC HL
3928 ; LD D, [HL]
3929 ; EXX
3930 ; ADD A,A
3931 ; ADD A,A
3932 ; LD E,A
3933 ; LD D,0

```

DEA2

```

DEA2 C5
DEA3 D9
DEA4 D06603
DEA7 D066E02
DEAA 7E
DEAB 23
DEAC 4E
DEAD 23
DEAE 46
DEAF 23
DEB0 5E
DEB1 23
DEB2 56
DEB3 09
DEB4 87
DEB5 8F
DEB6 5F
DEB7 1600

```

```

;SAVE SELECTOR AND COMPONENT COUNT [COMP_CNT]
;USE PRIMED REGS FOR X_LOC AND Y_LOC
;HIGH BYTE OF STATUS
;LOW BYTE OF STATUS
;A := FRAME
;BC' := X_LOCATION
;DE' := Y_LOCATION
;FRAME := 4*FRAME
;FORM POINTER TO FRAME AND OFFSET POINTERS

```

```

LOCATION OBJECT CODE LINE      SOURCE LINE
0EB9 23      INC HL      ;POINT TO FIRST OF FRA_OFFSET_PNTR PAIRS
0EBA 19      ADD HL,DE      ;POINT TO FRAME POINTER
0EBB 4E      LD C,[HL]
0EBC 23      INC HL
0EBD 46      LD B,[HL]      ;BC := FRAME POINTER (POINTER TO LIST OF FRAME #'S)
0EBE 23      INC HL
0EBF 5E      LD E,[HL]
0EC0 23      INC HL
0EC1 56      LD D,[HL]      ;DE := OFFSET POINTER (PNTR TO LIST OF OFFSETS)
0EC2 60      LD H,B
0EC3 69      LD L,C      ;HL := FRAME POINTER
3945      ;
3946      ; DE := Y_LOC, BC := X_LOC, HL = PNTR TO FRAME LIST, DE = PNTR TO OFFSET LIST
3947      ; IX = ADDR OF OBJ, [SP] = COMP CNT & SELECTOR
3948      ; FOR N=0 TO COMP_CNT-1: COMP_OBJ(N) FRAME := FRAME#(N) FROM FRAME LIST
3949      ;      COMP_OBJ(N) X_LOCATION := CHPLX OBJ X_LOCATION + X_OFFSET(N)
3950      ;      COMP_OBJ(N) Y_LOCATION := CHPLX OBJ Y_LOCATION + Y_OFFSET(N)
3951      ;
3952      POP BC
3953      LD A,C
3954      LD C,B
3955      SRL A
3956      SRL A
3957      SRL A
3958      LD B,A
3959      PUSH BC
3960      PUSH IX
3961      PUSH HL
3962      PUSH DE
3963      LD L,[IX+4]
3964      LD H,[IX+5]
3965      INC IX
3966      INC IX
3967      INC HL
3968      INC HL
3969      LD E,[HL]
3970      INC HL
3971      LD D,[HL]
3972      PUSH DE
3973      POP IY
3974      POP DE
3975      POP HL
3976      LD A,[HL]
3977      BIT 7,[IY+0]
3978      JR Z,IBLO
3979      SET 7,A
3980      IBLO:
3981      LD [IY+0],A
3982      INC HL
3983      LD A,[DE]
3984      EXX
3985      LD L,A
3986      LD H,0
3987      ADD HL,BC
3988      LD [IY+1],L
3989      LD [IY+2],H
3990      ;
0EC4 C1      ;GET COMPONENT COUNT INTO B
0EC5 79      ;SAVE SELECTOR IN C
0EC6 48      ;GET COUNT INTO LOW NIBBLE
0EC7 C83F      ;B := component count
0EC8 C83F      ;SAVE COUNTER AND SELECTOR ON STACK
0EC9 C83F      ;SAVE ADDR OF OBJ
0ECB C83F      ;SAVE PNTR TO FRAME LIST
0ECC C83F      ;SAVE PNTR TO OFFSET LIST
0ECD 47      ;HL := ADDR OF COMPONENT OBJ
0EDE C5      ;Point to next object pointer
0ED1 D0E5      ;HL POINTS TO STATUS POINTER
0ED3 E5      ;DE := ADDR OF STATUS FOR COMPONENT OBJECT
0ED4 05      ;IY := ADDR OF STATUS FOR COMPONENT OBJECT
0ED5 D06E04      ;DE := PNTR TO OFFSET LIST
0ED6 D06605      ;HL := PNTR TO FRAME LIST
0ED8 D023      ;GET FRAME NUMBER
0ED9 D023      ;Preserve bit 7 of frame
0EDF 23      ;used by mobile objects to indicate which
0EE0 23      ;VRAH tables in use.
0EE1 5E      ;MOVE TO COMPONENTS STATUS AREA
0EE2 23      ;POINT TO NEXT FRAME NUMBER
0EE3 56      ;GET X_OFFSET
0EE4 05      ;HL := X_OFFSET
0EE5 FDE1      ;HL := X_OFFSET + X_LOCATION
0EE7 D1      ;COMPONENT'S X_LOCATION := X_OFFSET + X_LOCATION
0EE8 E1      ;
0EE9 7E      ;
0EEA FDC8007E      ;
0EEE 2802      ;
0EF0 C8FF      ;
0EF2      ;
0EF2 FD7700      ;
0EF5 23      ;
0EF6 1A      ;
0EF7 D9      ;
0EF8 6F      ;
0EF9 2600      ;
0EFB 09      ;
0EFC FD7501      ;
0EFF FD7402      ;

```

LOCATION	OBJECT CODE	LINE	SOURCE	LINE
0F02	D9	3991	EXX	
0F03	13	3992	INC DE	
0F04	1A	3993	LD A, [DE]	
0F05	D9	3994	EXX	
0F06	6F	3995	LD L, A	
0F07	2600	3996	LD H, 0	
0F09	19	3997	ADD HL, DE	
0F0A	FD7503	3998	LD (1Y+3), L	
0F00	FD7404	3999	LD (1Y+4), H	
0F10	D9	4000	EXX	
0F11	13	4001	INC DE	
0F12	108F	4002	DJNZ LP1	
		4003		
4004		4004	; CALL PUT OBJECT FOR EACH OF THE COMPONENT OBJECTS, PASS SELECTOR IN B	
		4005	; GET OBJECT ADDRESS BACK	
0F14	FDE1	4006	POP IY	
0F16	010004	4007	LD BC, 4	
0F19	FD09	4008	ADD IY, BC	
0F1B	D1	4009	POP DE	
0F1C	FD6E00	4010	LD L, (1Y+0)	
0F1F	FD6601	4011	LD H, (1Y+1)	
0F22	FD23	4012	INC IY	
0F24	FD23	4013	INC IY	
0F26	E5	4014	PUSH HL	
0F27	D0E1	4015	POP IX	
0F29	FDE5	4016	PUSH IY	
0F2B	D5	4017	PUSH DE	
0F2C	43	4018	LD B, E	
0F2D	CD1FFA	4019	CALL PUTOBJ	
0F30	D1	4020	POP DE	
0F31	FDE1	4021	POP IY	
0F33	15	4022	DEC D	
0F34	20E6	4023	JR NZ, LP2	
0F36	C9	4024	RET	
				PROG

;POINT TO Y\_OFFSET

;HL := Y\_OFFSET  
;HL := Y\_OFFSET + Y\_LOCATION

;COMPONENT'S Y\_LOCATION := Y\_OFFSET + Y\_LOCATION

;POINT TO NEXT OFFSET PAIR

;CALL PUT OBJECT FOR EACH OF THE COMPONENT OBJECTS, PASS SELECTOR IN B  
;GET OBJECT ADDRESS BACK

;IY POINTS TO POINTER TO FIRST COMPONENT OBJECT  
;DE := COUNTER AND SELECTOR

;HL := ADDRESS OF COMPONENT OBJECT

;IY POINTS TO NEXT COMPONENT OBJECT POINTER

;IX := ADDRESS OF COMPONENT OBJECT

;SAVE POINTER

;SAVE COUNTER AND SELECTOR

;B := SELECTOR

;GET COUNTER AND SELECTOR

;GET ADDRESS OF NEXT COMPONENT OBJECT POINTER





KACATION	OBJECT CODE	LINE	SOURCE	LINE
0F86	ZB	4140	DEC	HL
0F87	77	4141	LD	(HL),A
0F88	ZB	4142	DEC	HL
0F89	E1	4143	POP	HL
0F8A	E5	4144	PUSH	HL
0F8B		4145	SET	7,(HL)
0F8B	CBFE	4146	SET	7,(HL)
0F8D		4147	POP	HL
0F8D	E1	4148	POP	
0F8E	C9	4149	RET	
		4150		
		4151		
		4152		
0F8F		4153	LD	(HL),D
0F8F	72	4154	LD	HL
0F90	ZB	4155	DEC	(HL),E
0F91	73	4156	LD	TIMER_EXIT
0F92	18F9	4157	JR	
		4158		
		4159		
		4160		
		4161		
		4162		
		4163		
		4164		
		4165		
		4166		
		4167		
		4168		
		4169		
		4170		
		4171		
		4172		
0F94		4173		
0F94	00020002	4174		
0F98	0002	4175		
		4176		
		4177		
		4178		
		4179		
		4180		
		4181		
		4182		
		4183		
		4184		
		4185		
		4186		
		4187		
		4188		
		4189		
		4190		
		4191		
		4192		
		4193		
		4194		
		4195		

;Store given base address for timer table  
 ;Set first byte in timer table to free and last timer  
 ;Store given base address for data block

;Procedure Free Signal  
 ;Acc has signal number to be freed  
 ;No output is generated  
 ;SIGNAL\_NUM:  
 ; DEFS 1  
 ;

LOCATION OBJECT CODE LINE SOURCE LINE

```

4196
4197 FREE_SIG_PAR:
4198 0FB4 00010001 DEFM 1,1
4199
4200 FREE_SIGNALQ:
4201 0FB8 010FB4 LD BC, FREE_SIG_PAR
4202 0FB8 1173BE LD DE, SIGNAL_NUM
4203 0FBE C00098 CALL PARAM
4204 0FC1 3A73BE LD A, [SIGNAL_NUM]
4205
4206 FREE_SIGNAL :
4207 0FC4 4F LD C,A
4208 0FC5 2A7303 HL, [TIMER_TABLE_BASE]
4209 0FC8 47 LD B,A
4210 0FC9 110003 LD DE,3
4211 0FCC B7 OR A
4212 0FDD 2808 JR Z, FREE_MATCH
4213
4214 FREE:
4215 0FCF CB66 BIT EOT, [HL]
4216 0FD1 206A JR NZ, FREE_EXIT
4217 0FD3 19 ADD HL, DE
4218 0FD4 00 DEC C
4219 0FD5 20F8 JR NZ, FREE1
4220
4221 FREE_MATCH:
4222 0FD7 CB6E BIT FREE, [HL]
4223 0FD9 2062 JR NZ, FREE_SET
4224 0FD8 CBEE SET FREE, [HL]
4225 0FD0 CB76 BIT REPEAT, [HL]
4226 0FDF 285C JR Z, FREE_SET
4227 0FE1 CB5E BIT LONG, [HL]
4228 0FE3 2858 JR Z, FREE_SET
4229 ; CALL FREE_COUNTER_
4230
4231 ;
4232 ; FREE (DELETE) COUNTER
4233 ; KENL 3/82
4234 ;
4235 ; NEEDS "TO-DELETE":COUNTER ADDR. IN DE
4236
4237 FREE_COUNTER_
4238 0FE5 23 IMC ;HL should contain timer to delete addr.
4239 0FE6 5E LD E, [HL] ;Get next after mode byte
4240 0FE7 23 IMC HL ; into DE.
4241 0FE8 56 LD D, [HL]
4242 0FE9 05 PUSH DE ;Save them for later.
4243 0FEA 2A7303 LD HL, [TIMER_TABLE_BASE] ;Save beginning of table.
4244 0FEE E5 PUSH HL
4245 0FEE NEXT
4246 0FEE CB66 BIT ;End of table?
4247 0FF0 202E JR ;Free?
4248 0FF2 CB6E BIT ;If so we don't want it.
4249 0FF4 2023 JR NZ, GET_NEXT
4250 0FF6 7E LD A, [HL]
4251 0FF7 E648 AND 48H ;Repeating and long?
4252 0FF9 FE48 CP 48H

```

;Put Free Code into C register  
;Get Timer Base Address  
;Timer offset already available  
;Setup offset to subsequent timers  
;See if the first timer is requested  
;If so we have a match

;Loop to find requested signal  
;Check if end of table  
;If so go no further  
;If not then offset to next timer  
;Decrement signal requested  
;If requested signal not zero then go try the next

;Here when the requested signal matches current signal  
;Check if already free  
;If so then just reset LONG  
;Set current timer to free  
;Check for repeating timer  
;If not then go free it  
;Check also for long timer  
;If not long then go free it  
;Release timer data before freeing timer

;HL should contain timer to delete addr.  
HL ;Get next after mode byte  
E, [HL] ; into DE.  
HL  
D, [HL]  
DE ;Save them for later.  
HL, [TIMER\_TABLE\_BASE] ;Save beginning of table.  
HL  
EOT, [HL]  
NZ, MOVE IT ;End of table?  
FREE, [HL] ;Free?  
NZ, GET NEXT ;If so we don't want it.  
A, [HL]  
48H ;Repeating and long?  
48H

LOCATION    OBJECT CODE    LINE    SOURCE LINE

```

OFFB 201C    4253    JR
OFFD 23    4254    INC
OFFE 23    4255    INC
OFFF 7E    4256    LD
1000 BA    4257    CP
1001 3016    4258    JR
1003 2008    4259    JR
1005 28    4260    DEC
1006 7E    4261    LD
1007 BB    4262    CP
1008 300F    4263    JR
100A 2831    4264    JR
100C 23    4265    INC
1000    4266    SUBTRACT_4
1000 56    4267    LD
100E 28    4268    DEC
100F 5E    4269    LD
1010 1B    4270    DEC
1011 1B    4271    DEC
1012 1B    4272    DEC
1013 1B    4273    DEC
1014 73    4274    LD
1015 23    4275    INC
1016 72    4276    LD
1017 1800    4277    JR
1019    4278    GET_NEXT
1019 E1    4279    POP
101A 23    4280    INC
101B 23    4281    INC
101C 23    4282    INC
101D E5    4283    PUSH
101E 18CE    4284    JR
1020    4285    MOVE_IT
1020 0600    4286    LD
1022 B7    4287    OR
1023 E1    4288    POP
1024 01    4289    POP
1025 E5    4290    PUSH
1026 2A7305    4291    LD
1029 ED52    4292    SBC
102B 40    4293    LD
102C 68    4294    LD
102D 62    4295    LD
102E 23    4296    INC
102F 23    4297    INC
1030 23    4298    INC
1031 23    4299    INC
1032 ED80    4300    LDJR
1034 010008    4301    LD
1037 ED42    4302    SBC
1039 227305    4303    LD
103C E1    4304    POP
1030    4305    EXIT
1036    4306    ;
1037    4307    ;
1038    4308    ;
1039    4309    ;
1040    4310    ;
1041    4311    ;
1042    4312    ;
1043    4313    ;
1044    4314    ;
1045    4315    ;
1046    4316    ;
1047    4317    ;
1048    4318    ;
1049    4319    ;
1050    4320    ;
1051    4321    ;
1052    4322    ;
1053    4323    ;
1054    4324    ;
1055    4325    ;
1056    4326    ;
1057    4327    ;
1058    4328    ;
1059    4329    ;
1060    4330    ;
1061    4331    ;
1062    4332    ;
1063    4333    ;
1064    4334    ;
1065    4335    ;
1066    4336    ;
1067    4337    ;
1068    4338    ;
1069    4339    ;
1070    4340    ;
1071    4341    ;
1072    4342    ;
1073    4343    ;
1074    4344    ;
1075    4345    ;
1076    4346    ;
1077    4347    ;
1078    4348    ;
1079    4349    ;
1080    4350    ;
1081    4351    ;
1082    4352    ;
1083    4353    ;
1084    4354    ;
1085    4355    ;
1086    4356    ;
1087    4357    ;
1088    4358    ;
1089    4359    ;
1090    4360    ;
1091    4361    ;
1092    4362    ;
1093    4363    ;
1094    4364    ;
1095    4365    ;
1096    4366    ;
1097    4367    ;
1098    4368    ;
1099    4369    ;
1100    4370    ;
1101    4371    ;
1102    4372    ;
1103    4373    ;
1104    4374    ;
1105    4375    ;
1106    4376    ;
1107    4377    ;
1108    4378    ;
1109    4379    ;
1110    4380    ;
1111    4381    ;
1112    4382    ;
1113    4383    ;
1114    4384    ;
1115    4385    ;
1116    4386    ;
1117    4387    ;
1118    4388    ;
1119    4389    ;
1120    4390    ;
1121    4391    ;
1122    4392    ;
1123    4393    ;
1124    4394    ;
1125    4395    ;
1126    4396    ;
1127    4397    ;
1128    4398    ;
1129    4399    ;
1130    4400    ;
1131    4401    ;
1132    4402    ;
1133    4403    ;
1134    4404    ;
1135    4405    ;
1136    4406    ;
1137    4407    ;
1138    4408    ;
1139    4409    ;
1140    4410    ;
1141    4411    ;
1142    4412    ;
1143    4413    ;
1144    4414    ;
1145    4415    ;
1146    4416    ;
1147    4417    ;
1148    4418    ;
1149    4419    ;
1150    4420    ;
1151    4421    ;
1152    4422    ;
1153    4423    ;
1154    4424    ;
1155    4425    ;
1156    4426    ;
1157    4427    ;
1158    4428    ;
1159    4429    ;
1160    4430    ;
1161    4431    ;
1162    4432    ;
1163    4433    ;
1164    4434    ;
1165    4435    ;
1166    4436    ;
1167    4437    ;
1168    4438    ;
1169    4439    ;
1170    4440    ;
1171    4441    ;
1172    4442    ;
1173    4443    ;
1174    4444    ;
1175    4445    ;
1176    4446    ;
1177    4447    ;
1178    4448    ;
1179    4449    ;
1180    4450    ;
1181    4451    ;
1182    4452    ;
1183    4453    ;
1184    4454    ;
1185    4455    ;
1186    4456    ;
1187    4457    ;
1188    4458    ;
1189    4459    ;
1190    4460    ;
1191    4461    ;
1192    4462    ;
1193    4463    ;
1194    4464    ;
1195    4465    ;
1196    4466    ;
1197    4467    ;
1198    4468    ;
1199    4469    ;
1200    4470    ;
1201    4471    ;
1202    4472    ;
1203    4473    ;
1204    4474    ;
1205    4475    ;
1206    4476    ;
1207    4477    ;
1208    4478    ;
1209    4479    ;
1210    4480    ;
1211    4481    ;
1212    4482    ;
1213    4483    ;
1214    4484    ;
1215    4485    ;
1216    4486    ;
1217    4487    ;
1218    4488    ;
1219    4489    ;
1220    4490    ;
1221    4491    ;
1222    4492    ;
1223    4493    ;
1224    4494    ;
1225    4495    ;
1226    4496    ;
1227    4497    ;
1228    4498    ;
1229    4499    ;
1230    4500    ;
1231    4501    ;
1232    4502    ;
1233    4503    ;
1234    4504    ;
1235    4505    ;
1236    4506    ;
1237    4507    ;
1238    4508    ;
1239    4509    ;
1240    4510    ;
1241    4511    ;
1242    4512    ;
1243    4513    ;
1244    4514    ;
1245    4515    ;
1246    4516    ;
1247    4517    ;
1248    4518    ;
1249    4519    ;
1250    4520    ;
1251    4521    ;
1252    4522    ;
1253    4523    ;
1254    4524    ;
1255    4525    ;
1256    4526    ;
1257    4527    ;
1258    4528    ;
1259    4529    ;
1260    4530    ;
1261    4531    ;
1262    4532    ;
1263    4533    ;
1264    4534    ;
1265    4535    ;
1266    4536    ;
1267    4537    ;
1268    4538    ;
1269    4539    ;
1270    4540    ;
1271    4541    ;
1272    4542    ;
1273    4543    ;
1274    4544    ;
1275    4545    ;
1276    4546    ;
1277    4547    ;
1278    4548    ;
1279    4549    ;
1280    4550    ;
1281    4551    ;
1282    4552    ;
1283    4553    ;
1284    4554    ;
1285    4555    ;
1286    4556    ;
1287    4557    ;
1288    4558    ;
1289    4559    ;
1290    4560    ;
1291    4561    ;
1292    4562    ;
1293    4563    ;
1294    4564    ;
1295    4565    ;
1296    4566    ;
1297    4567    ;
1298    4568    ;
1299    4569    ;
1300    4570    ;
1301    4571    ;
1302    4572    ;
1303    4573    ;
1304    4574    ;
1305    4575    ;
1306    4576    ;
1307    4577    ;
1308    4578    ;
1309    4579    ;
1310    4580    ;
1311    4581    ;
1312    4582    ;
1313    4583    ;
1314    4584    ;
1315    4585    ;
1316    4586    ;
1317    4587    ;
1318    4588    ;
1319    4589    ;
1320    4590    ;
1321    4591    ;
1322    4592    ;
1323    4593    ;
1324    4594    ;
1325    4595    ;
1326    4596    ;
1327    4597    ;
1328    4598    ;
1329    4599    ;
1330    4600    ;
1331    4601    ;
1332    4602    ;
1333    4603    ;
1334    4604    ;
1335    4605    ;
1336    4606    ;
1337    4607    ;
1338    4608    ;
1339    4609    ;
1340    4610    ;
1341    4611    ;
1342    4612    ;
1343    4613    ;
1344    4614    ;
1345    4615    ;
1346    4616    ;
1347    4617    ;
1348    4618    ;
1349    4619    ;
1350    4620    ;
1351    4621    ;
1352    4622    ;
1353    4623    ;
1354    4624    ;
1355    4625    ;
1356    4626    ;
1357    4627    ;
1358    4628    ;
1359    4629    ;
1360    4630    ;
1361    4631    ;
1362    4632    ;
1363    4633    ;
1364    4634    ;
1365    4635    ;
1366    4636    ;
1367    4637    ;
1368    4638    ;
1369    4639    ;
1370    4640    ;
1371    4641    ;
1372    4642    ;
1373    4643    ;
1374    4644    ;
1375    4645    ;
1376    4646    ;
1377    4647    ;
1378    4648    ;
1379    4649    ;
1380    4650    ;
1381    4651    ;
1382    4652    ;
1383    4653    ;
1384    4654    ;
1385    4655    ;
1386    4656    ;
1387    4657    ;
1388    4658    ;
1389    4659    ;
1390    4660    ;
1391    4661    ;
1392    4662    ;
1393    4663    ;
1394    4664    ;
1395    4665    ;
1396    4666    ;
1397    4667    ;
1398    4668    ;
1399    4669    ;
1400    4670    ;
1401    4671    ;
1402    4672    ;
1403    4673    ;
1404    4674    ;
1405    4675    ;
1406    4676    ;
1407    4677    ;
1408    4678    ;
1409    4679    ;
1410    4680    ;
1411    4681    ;
1412    4682    ;
1413    4683    ;
1414    4684    ;
1415    4685    ;
1416    4686    ;
1417    4687    ;
1418    4688    ;
1419    4689    ;
1420    4690    ;
1421    4691    ;
1422    4692    ;
1423    4693    ;
1424    4694    ;
1425    4695    ;
1426    4696    ;
1427    4697    ;
1428    4698    ;
1429    4699    ;
1430    4700    ;
1431    4701    ;
1432    4702    ;
1433    4703    ;
1434    4704    ;
1435    4705    ;
1436    4706    ;
1437    4707    ;
1438    4708    ;
1439    4709    ;
1440    4710    ;
1441    4711    ;
1442    4712    ;
1443    4713    ;
1444    4714    ;
1445    4715    ;
1446    4716    ;
1447    4717    ;
1448    4718    ;
1449    4719    ;
1450    4720    ;
1451    4721    ;
1452    4722    ;
1453    4723    ;
1454    4724    ;
1455    4725    ;
1456    4726    ;
1457    4727    ;
1458    4728    ;
1459    4729    ;
1460    4730    ;
1461    4731    ;
1462    4732    ;
1463    4733    ;
1464    4734    ;
1465    4735    ;
1466    4736    ;
1467    4737    ;
1468    4738    ;
1469    4739    ;
1470    4740    ;
1471    4741    ;
1472    4742    ;
1473    4743    ;
1474    4744    ;
1475    4745    ;
1476    4746    ;
1477    4747    ;
1478    4748    ;
1479    4749    ;
1480    4750    ;
1481    4751    ;
1482    4752    ;
1483    4753    ;
1484    4754    ;
1485    4755    ;
1486    4756    ;
1487    4757    ;
1488    4758    ;
1489    4759    ;
1490    4760    ;
1491    4761    ;
1492    4762    ;
1493    4763    ;
1494    4764    ;
1495    4765    ;
1496    4766    ;
1497    4767    ;
1498    4768    ;
1499    4769    ;
1500    4770    ;
1501    4771    ;
1502    4772    ;
1503    4773    ;
1504    4774    ;
1505    4775    ;
1506    4776    ;
1507    4777    ;
1508    4778    ;
1509    4779    ;
1510    4780    ;
1511    4781    ;
1512    4782    ;
1513    4783    ;
1514    4784    ;
1515    4785    ;
1516    4786    ;
1517    4787    ;
1518    4788    ;
1519    4789    ;
1520    4790    ;
1521    4791    ;
1522    4792    ;
1523    4793    ;
1524    4794    ;
1525    4795    ;
1526    4796    ;
1527    4797    ;
1528    4798    ;
1529    4799    ;
1530    4800    ;
1531    4801    ;
1532    4802    ;
1533    4803    ;
1534    4804    ;
1535    4805    ;
1536    4806    ;
1537    4807    ;
1538    4808    ;
1539    4809    ;
1540    4810    ;
1541    4811    ;
1542    4812    ;
1543    4813    ;
1544    4814    ;
1545    4815    ;
1546    4816    ;
1547    4817    ;
1548    4818    ;
1549    4819    ;
1550    4820    ;
1551    4821    ;
1552    4822    ;
1553    4823    ;
1554    4824    ;
1555    4825    ;
1556    4826    ;
1557    4827    ;
1558    4828    ;
1559    4829    ;
1560    4830    ;
1561    4831    ;
1562    4832    ;
1563    4833    ;
1564    4834    ;
1565    4835    ;
1566    4836    ;
1567    4837    ;
1568    4838    ;
1569    4839    ;
1570    4840    ;
1571    4841    ;
1572    4842    ;
1573    4843    ;
1574    4844    ;
1575    4845    ;
1576    4846    ;
1577    4847    ;
1578    4848    ;
1579    4849    ;
1580    4850    ;
1581    4851    ;
1582    4852    ;
1583    4853    ;
1584    4854    ;
1585    4855    ;
1586    4856    ;
1587    4857    ;
1588    4858    ;
1589    4859    ;
1590    4860    ;
1591    4861    ;
1592    4862    ;
1593    4863    ;
1594    4864    ;
1595    4865    ;
1596    4866    ;
1597    4867    ;
1598    4868    ;
1599    4869    ;
1600    4870    ;
1601    4871    ;
1602    4872    ;
1603    4873    ;
1604    4874    ;
1605    4875    ;
1606    4876    ;
1607    4877    ;
1608    4878    ;
1609    4879    ;
1610    4880    ;
1611    4881    ;
1612    4882    ;
1613    4883    ;
1614    4884    ;
1615    4885    ;
1616    4886    ;
1617    4887    ;
1618    4888    ;
1619    4889    ;
1620    4890    ;
1621    4891    ;
1622    4892    ;
1623    4893    ;
1624    4894    ;
1625    4895    ;
1626    4896    ;
1627    4897    ;
1628    4898    ;
1629    4899    ;
1630    4900    ;
1631    4901    ;
1632    4902    ;
1633    4903    ;
1634    4904    ;
1635    4905    ;
1636    4906    ;
1637    4907    ;
1638    4908    ;
1639    4909    ;
1640    4910    ;
1641    4911    ;
1642    4912    ;
1643    4913    ;
1644    4914    ;
1645    4915    ;
1646    4916    ;
1647    4917    ;
1648    4918    ;
1649    4919    ;
1650    4920    ;
1651    4921    ;
1652    4922    ;
1653    4923    ;
1654    4924    ;
1655    4925    ;
1656    4926    ;
1657    4927    ;
1658    4928    ;
1659    4929    ;
1660    4930    ;
1661    4931    ;
1662    4932    ;
1663    4933    ;
1664    4934    ;
1665    4935    ;
1666    4936    ;
1667    4937    ;
1668    4938    ;
1669    4939    ;
1670    4940    ;
1671    4941    ;
1672    4942    ;
1673    4943    ;
1674    4944    ;
1675    4945    ;
1676    4946    ;
1677    4947    ;
1678    4948    ;
1679    4949    ;
1680    4950    ;
1681    4951    ;
1682    4952    ;
1683    4953    ;
1684    4954    ;
1685    4955    ;
1686    4956    ;
1687    4957    ;
1688    4958    ;
1689    4959    ;
1690    4960    ;
1691    4961    ;
1692    4962    ;
1693    4963    ;
1694    4964    ;
1695    4965    ;
1696    4966    ;
1697    4967    ;
1698    4968    ;
1699    4969    ;
1700    4970    ;
1701    4971    ;
1702    4972    ;
1703    4973    ;
1704    4974    ;
1705    4975    ;
1706    4976    ;
1707    4977    ;
1708    4978    ;
1709    4979    ;
1710    4980    ;
1711    4981    ;
1712    4982    ;
1713    4983    ;
1714    4984    ;
1715    4985    ;
1716    4986    ;
1717    4987    ;
1718    4988    ;
1719    4989    ;
1720    4990    ;
1721    4991    ;
1722    4992    ;
1723    4993    ;
1724    4994    ;
1725    4995    ;
1726    4996    ;
1727    4997    ;
1728    4998    ;
1729    4999    ;
1730    5000    ;
1731    5001    ;
1732    5002    ;
1733    5003    ;
1734    5004    ;
1735    5005    ;
1736    5006    ;
1737    5007    ;
1738    5008    ;
1739    5009    ;
1740    5010    ;
1741    5011    ;
1742    5012    ;
1743    5013    ;
1744    5014    ;
1745    5015    ;
1746    5016    ;
1747    5017    ;
1748    5018    ;
1749    5019    ;
1750    5020    ;
1751    5021    ;
1752    5022    ;
1753    5023    ;
1754    5024    ;
1755    5025    ;
1756    5026    ;
1757    5027    ;
1758    5028    ;
1759    5029    ;
1760    5030    ;
1761    5031    ;
1762    5032    ;
1763    5033    ;
1764    5034    ;
1765    5035    ;
1766    5036    ;
1767    5037    ;
1768    5038    ;
1769    5039    ;
1770    5040    ;
1771    5041    ;
1772    5042    ;
1773    5043    ;
1774    5044    ;
1775    5045    ;
1776    5046    ;
1777    5047    ;
1778    5048    ;
1779    5049    ;
1780    5050    ;
1781    5051    ;
1782    5052    ;
1783    5053    ;
1784    5054    ;
1785  
```

LOCATION OBJECT CODE LINE SOURCE LINE

```

4310 ; RES LONG,[HL]
4311 FREE_EXIT: RET
4312
4313
4314
4315 ;Procedure Request Signal
4316 ;HL pair has length of timer
4317 ;Acc has zero for repeating timer any other value for a non_repeating type
4318 ;Signal number is returned in the Accumulator
4319 ; DOWN
4320 ;REPEAT_SIG_CODE:
4321 ; DEFS 1
4322 ;TIMER_LENGTH:
4323 ; DEFS 2
4324
4325 PROG
4326
4327 REQUEST_SIG_PARAM:
4328 DEFM 2,1,2

103E 00020001
1042 0002

4329
4330 REQUEST_SIGNALQ:
4331 LD BC,REQUEST_SIG_PARAM
4332 LD DE,REPEAT_SIG_CODE
4333 CALL PARAM
4334 LD HL,(TIMER_LENGTH)
4335 LD A,[REPEAT_SIG_CODE]
4336

1053
4337 REQUEST_SIGNAL_:
4338 LD C,A
4339 EX DE,HL
4340 LD HL,(TIMER_TABLE_BASE)
4341 XOR A
4342 LD B,A
4343 TIMER1:
4344 BIT FREE,[HL]
4345 JR Z,NEXT_TIMER1
4346 PUSH HL
4347 ;
4348 LD A,[HL]
4349 AND 10H
4350 OR 20H
4351 LD [HL],A
4352 XOR A
4353 POP AF
4354 OR D
4355 JR NZ,LONG_TIMER
4356 ; RES FREE,[HL]
4357 ; OR D
4358 ; JR NZ,LONG_TIMER
4359 ; RES REPEAT,[HL]
4360 ; RES LONG,[HL]
4361 ; LD A,C
4362 ; OR C
4363 ; JR Z,NOT_A_REPEAT_TIMER
4364 ; SET REPEAT,[HL]
4365 ; NOT A_REPEAT_TIMER:

```

```

;Reset repeat bit just in case
;Return

```

```

;Put Repeat Code into C register
;Put length of timer into DE
;Get Timer Base Address
;Init offset to first Table value

```

```

;See if current timer free
;If not go get the next timer

```

```

;Reset free bit
;Check for zero
;If non zero then its a long timer
;Set for a NON_Repeating timer

```

```

;Check for a short repeating timer

```

```

;Don't reset repeat bit in mode byte if non_repeating
;Set repeat bit

```

LOCATION OBJECT CODE LINE SOURCE LINE

```

106E 23 4366 INC HL
106F 73 4367 LD [HL],E
1070 23 4368 INC HL
1071 73 4369 LD [HL],E
1072 1842 4370 JR INIT_TIMER_EXIT
1074 4371 LONG_TIMER:
1074 CB0E 4372 SET LONG,[HL]
1076 79 4373 LD A,C
1077 B7 4374 OR A
1078 2818 4375 JR Z,MOT_A_LONG_REPEAT
107A D5 4376 PUSH DE
107B EB 4377 EX DE,HL
107C 2A7305 4378 LD HL,(NEXT_TIMER_DATA_BYTE)
107E EB 4379 EX DE,HL
1080 CBF6 4380 SET REPEAT,[HL]
1082 23 4381 INC HL
1083 73 4382 LD [HL],E
1084 23 4383 INC HL
1085 72 4384 LD [HL],D
1086 EB 4385 EX DE,HL
1087 D1 4386 POP DE
1088 73 4387 LD [HL],E
1089 23 4388 INC HL
108A 72 4389 LD [HL],D
108B 23 4390 INC HL
108C 73 4391 LD [HL],E
108D 23 4392 INC HL
108E 72 4393 LD [HL],D
108F 23 4394 INC HL
1090 227305 4395 LD (NEXT_TIMER_DATA_BYTE),HL
1093 1821 4396 JR INIT_TIMER_EXIT
1095 23 4397 MOT_A_LONG_REPEAT:
1095 4398 INC HL
1096 4399 TIMER2:
1096 73 4400 LD [HL],E
1097 23 4401 INC HL
1098 72 4402 LD [HL],D
1099 23 4403 INC HL
109A 181A 4404 JR INIT_TIMER_EXIT
109A 4405
109C 4406 NEXT_TIMER1:
109C CB66 4407 BIT EOT,[HL]
109E 2006 4408 JR NZ,MAKE_NEW_TIMER
10A0 23 4409 INC HL
10A1 23 4410 INC HL
10A2 23 4411 INC HL
10A3 04 4412 INC B
10A4 1884 4413 JR TIMER1
10A4 4414
10A6 4415 MAKE_NEW_TIMER:
10A6 D5 4416 PUSH DE
10A7 E5 4417 PUSH HL
10A8 23 4418 INC HL
10A9 23 4419 INC HL
10AA 23 4420 INC HL
10AB 04 4421 INC B
10AC 3630 4422 LD [HL],30H

```

;Go to next table location  
;Store timer length

;Store timer length again in case of repeat  
;All done so let's exit

;Set long timer bit  
;Check for a long repeat timer

;If zero then go to section for non\_repeating timer  
;Store timer length temporarily  
;Swap registers  
;To get free space in long timer table  
;Then swap back  
;Set mode byte to repeating

;Store low byte of timer address into the value word  
;Store high byte of timer address  
;Move address of data area into HL  
;Get back the length of timer  
;Store that in the data table

;Store it again

;Store the next available data area for future use

;Store it again

;Go to next mode byte

;Count to next offset  
;Go back up to init. timer

;Maximum of 255 signals allowed

;Save DE for a work register  
;Save current timer address

;Go to next available memory location in the timer table

;Increment the signal count  
;Set to free and last timer