

```

3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712

    11EE
    11EE
    11EF
    11F2
    11F5
    11F8
    11F8
    11FA
    11FB
    11FD
    11FE
    11FF
    1200
    1200
    1203
    1204
    1205
    1206
    1207
    120A
    120C
    120C
    120D
    120F
    1210

    3D
    FA 1200
    CD 120C
    21 1233
    E6 0F
    5F
    16 00
    19
    7E
    C9
    CD 1226
    0F
    0F
    0F
    0F
    21 1243
    18 EC
    47
    3E 0F
    F3
    CD 110E

    GTSTCK:
    ;
    DEC
    JP
    CALL
    LD
    A
    M,KYSTCK
    SLSTCK
    HL,STKTBL
    AND
    LD
    LD
    ADD
    LD
    RET
    CALL
    RRCA
    RRCA
    RRCA
    RRCA
    LD
    JR
    SLSTCK:
    ;
    ; Select proper joystick and read from it
    ;
    LD
    LD
    DI
    CALL
    B,A
    A,PSG.PB
    RDPG

    ;STICK(0) - read cursor keys
    ;Read joystick
    ;Read keyboard
    ;Move cursor status to lower four bits
    GTR0W8
    HL,KSTKTB
    STICK1
    ;Read what is currently output to port B
  
```

```

3713 1213 10 06          DJNZ      SLSTC1
3714 1215 E6 DF          AND       0DFH
3715 1217 F6 4C          OR        4CH
3716 1219 18 04          JR        SLSTC2
3717 121B
3718
3719 121B E6 AF          AND       0AFH
3720 121D F6 03          OR        3
3721 121F
3722 121F D3 A1          OUT      (PSG.DW),A
3723 1221 CD 110C        CALL    INGI
3724 1224 FB           EI
3725 1225 C9           RET
3726 1226
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
1226 F3
1227 DB AA
1229 E6 F0
122B C6 08
122D D3 AA
122F DB A9

          SLSTC1:
          ;
          AND
          OR
          OUT      (PSG.DW),A
          CALL    INGI
          EI
          RET

          SLSTC2:
          ;
          ; Get keyboard's 8th row, bit assignments are as follows.
          ;
          ; RDULxxxxS
          ; ||||| |
          ; ||||| +- space
          ; ||||+----- left
          ; ||+----- up
          ; |+----- down
          ; +----- right
          ;

          DI
          IN      A,(PPI.CR)
          AND    0F0H
          ADD    A,8
          OUT    (PPI.CW),A
          IN      A,(PPI.BR)

```

- MSXIO - Joystick and Paddle interface

```

3744 1231 FB EI
3745 1232 C9 RET
3746
3747 1233 ;
3748 1233 00 STKTBL: DB 0 ;RLBF
3749 1234 05 DB 5 ;RLB
3750 1235 01 DB 1 ;RL F
3751 1236 00 DB 0 ;RL
3752 1237 03 DB 3 ;R BF
3753 1238 04 DB 4 ;R B
3754 1239 02 DB 2 ;R F
3755 123A 03 DB 3 ;R
3756 123B 07 DB 7 ; LBF
3757 123C 06 DB 6 ; LB
3758 123D 08 DB 8 ; L F
3759 123E 07 DB 7 ; L
3760 123F 00 DB 0 ; BF
3761 1240 05 DB 5 ; B
3762 1241 01 DB 1 ; F
3763 1242 00 DB 0 ;
3764 ;
3765 1243 ;KSTKTBL:
3766 1243 00 DB 0 ;RBFL
3767 1244 03 DB 3 ;RBF
3768 1245 05 DB 5 ;RB L
3769 1246 04 DB 4 ;RB
3770 1247 01 DB 1 ;R FL
3771 1248 02 DB 2 ;R F
3772 1249 00 DB 0 ;R L
3773 124A 03 DB 3 ;R
3774 124B 07 DB 7 ; BFL

```

```
3775 124C 00 DB 0 ; BF
3776 124D 06 DB 6 ; B L
3777 124E 05 DB 5 ; B
3778 124F 08 DB 8 ; FL
3779 1250 01 DB 1 ; F
3780 1251 07 DB 7 ; L
3781 1252 00 DB 0 ;
3782
3783 ;
3784 GTTRIG:
3785 ;
3786 1253 3D DEC A
3787 1254 FA 126C JP M,KEYTRG
3788 1257 F5 PUSH AF
3789 1258 E6 01 AND 1
3790 125A CD 120C CALL SLSTCK
3791 125D C1 POP BC
3792 125E 05 DEC B
3793 125F 05 DEC B
3794 1260 06 10 LD B,10H
3795 1262 FA 1267 JP M,TRIG1
3796 1265 06 20 LD B,' '
3797 1267 A0 AND B
3798 1268 TRIG2:
3799 1268 D6 01 SUB 1
3800 126A 9F SBC A,A
3801 126B C9 RET
3802 126C KEYTRG:
3803 ;
3804 126C CD 1226 CALL GTROW8
3805 126F E6 01 AND 1
```

;STRIG(0), use keyboard

;Read joystick

;Prepare mask pattern for trigger A

;Prepare mask pattern for trigger B

;Extract trigger status

;Return 255 if [Acc]=0, 0 if non-0

;Read keyboard

;Extract space status

( MSX ROM BASIC BIOS ) Macro-80  
 - MSXIO - Joystick and Paddle interface

```

3806      1271      18 F5      JR      TRIG2
3807      1273      GTPDL:
3808      ;
3809      ; Get value of paddle
3810      ;
3811      ; Input parameters (passed via [Acc])
3812      ;
3813      ; 1 - Paddle A connected to joystick port 1
3814      ; 2 - Paddle A connected to joystick port 2
3815      ; 3 - Paddle B connected to joystick port 1
3816      ; 4 - Paddle B connected to joystick port 2
3817      ; 5 - Paddle C connected to joystick port 1
3818      ; 6 - Paddle C connected to joystick port 2
3819      ; 7 - Paddle D connected to joystick port 1
3820      ; 8 - Paddle D connected to joystick port 2
3821      ; 9 - Paddle E connected to joystick port 1
3822      ; 10 - Paddle E connected to joystick port 2
3823      ; 11 - Paddle F connected to joystick port 1
3824      ; 12 - Paddle F connected to joystick port 2
3825      ;
3826      1273      3C      INC      A      ;Force parameter 2 based
3827      1274      A7      AND      A
3828      1275      1F      RRA
3829      1276      F5      PUSH     AF
3830      1277      47      LD       B,A
3831      1278      AF      XOR      A
3832      1279      37      SCF
3833      127A      PDL1:
3834      127A      17      RLA
3835      127B      10 FD   DJNZ
3836      127D      47      LD       B,A      ;Set mask pattern
    
```

; Save port # (carry reset if port 1)  
 ; Form mask pattern  
 ; Set mask pattern

3837	127E	F1	POP	AF		
3838	127F	0E 10	LD	C,10H	;Assume port 1	
3839	1281	11 03AF	LD	DE,03AFH		
3840	1284	30 05	JR	NC,PDLP1	;Good assumption	
3841	1286	0E 20	LD	C, ' '		
3842	1288	11 4C9F	LD	DE,4C9FH		
3843	128B					PDLP1:
3844	128B	3E 0F	LD	A,PSG.PB		
3845	128D	F3	DI			
3846	128E	CD 110E	CALL	RDP5G		
3847	1291	A3	AND	E	;Get current port B content	
3848	1292	B2	OR	D		
3849	1293	B1	OR	C		
3850	1294	D3 A1	OUT	(PSG.DW),A	;Set trigger high	
3851	1296	A9	XOR	C		
3852	1297	D3 A1	OUT	(PSG.DW),A	;Set trigger low again	
3853	1299	3E 0E	LD	A,0EH		
3854	129B	D3 A0	OUT	(PSG.IW),A		
3855	129D	0E 00	LD	C,0	;Initialize counter	
3856	129F					PDL2:
3857	129F	DB A2	IN	A,(PSG.DR)		
3858	12A1	A0	AND	B	;End of pulse?	
3859	12A2	28 05	JR	Z,PDL3	;Yes	
3860	12A4	0C	INC	C	;Bump counter	
3861	12A5	C2 129F	JP	NZ,PDL2	;No overflow yet	
3862	12A8	0D	DEC	C	;Make it 255	
3863	12A9					PDL3:
3864	12A9	FB	EI			
3865	12AA	79	LD	A,C	;Return counted value	
3866	12AB	C9	RET			
3867	12AC					GTPAD:

```

3868 ;
3869 ; Read touch pad (NEC PC-6051 compatible)
3870 ;
3871 ; Input parameter (passed via [Acc])
3872 ;
3873 ; 0 - sense touch pad status ---
3874 ; 1 - return X coordinate |for touch pad connected
3875 ; 2 - return Y coordinate |to joystick port 1
3876 ; 3 - return switch status -----
3877 ;
3878 ; 4 - sense touch pad status ---
3879 ; 5 - return X coordinate |for touch pad connected
3880 ; 6 - return Y coordinate |to joystick port 2
3881 ; 7 - return switch status -----
3882 ;
3883 ; Result is returned via [Acc]. As for status, 255 is returned
3884 ; if true, 0 if false.
3885 ;
3886 FE 04 CP 4 ;Read pad connected to port 1
3887 12AC 11 0CEC LD DE,0CECH ;Assume so
3888 12AE 38 05 JR C,GTPDP1 ;Good assumption
3889 12B1 11 03D3 LD DE,03D3H ;Connected to port 2
3890 12B3 D6 04 SUB 4
3891 12B6 GTPDP1:
3892 12B8 3D DEC A ;Argument=0?
3893 12B9 FA 12C5 JP M,GTPAD0 ;If so, read pad and return status
3894 12BC 3D DEC A
3895 12BD 3A FC9D LD A,(PADX) ;Assume PAD(1) - X coordinate
3896 12C0 F8 RET M ;Good assumption
3897 12C1 3A FC9C LD A,(PADY) ;Return Y coordinate
3898 12C4 C8 RET Z

```

```

3899 12C5          GTPAD0:
3900 12C5          F5          PUSH
3901 12C6          EB          EX
3902 12C7          22 F866     LD
3903 12CA          9F          SBC
3904 12CB          2F          CPL
3905 12CC          E6 40       AND
3906 12CE          4F          LD
3907 12CF          3E 0F       LD
3908 12D1          F3          DI
3909 12D2          CD 110E     CALL
3910 12D5          E6 BF       AND
3911 12D7          B1          OR
3912 12D8          D3 A1       OUT
3913 12DA          F1          POP
3914 12DB          FA 12E8     JP
3915 12DE          CD 110C     CALL
3916 12E1          FB          EI
3917 12E2          E6 08       AND
3918 12E4          D6 01       SUB
3919 12E6          9F          SBC
3920 12E7          C9          RET
3921 12E8
3922
3923 12E8          0E 00       LD
3924 12EA          CD 1332     CALL
3925 12ED          CD 1332     CALL
3926 12F0          38 28       JR
3927 12F2          CD 1320     CALL
3928 12F5          38 23       JR
3929 12F7          D5          PUSH

```

```

AF
DE,HL
(RUNFLG),HL
A,A
01000000B
C,A
A,PSG.PB
RDPSPG
0BFH
C
(PSG.DW),A
AF
M,TRYAGN
INGI
8
1
A,A
C,0
REDPAD
REDPAD
C,PADX1
REDCOD
C,PADX1
DE
;
;inz
;sense Panel input and select X
;branch if no input
;read first coordinate
;branch if input released
;save for comparison

```

```

;0 if port 1 specified, 100 octal if port 2
;disable interrupt till done
;Select proper port
;PAD(0) specified

```



- MSXIO - Joystick and Paddle interface

```

3930 12F8 CD 1320 CALL REDCOD
3931 12FB C1 POP BC
3932 12FC 38 1C JR C,PADXL
3933 12FE 78 LD A,B
3934 12FF 92 SUB D
3935 1300 30 02 JR NC,NONEGL
3936 1302 2F CPL
3937 1303 3C INC A
3938 1304 NONEGL:
3939 1304 FE 05 CP 5
3940 1306 30 E0 JR NC,TRYAGN
3941 1308 79 LD A,C
3942 1309 93 SUB E
3943 130A 30 02 JR NC,NONEG2
3944 130C 2F CPL
3945 130D 3C INC A
3946 130E NONEG2:
3947 130E FE 05 CP 5
3948 1310 30 D6 JR NC,TRYAGN
3949 1312 7A LD A,D
3950 1313 32 FC9D LD (PADX),A
3951 1316 7B LD A,E
3952 1317 32 FC9C LD (PADY),A
3953 131A PADXL:
3954 131A FB EI
3955 131B 7C LD A,H
3956 131C D6 01 SUB 1
3957 131E 9F SBC A,A
3958 131F C9 RET
3959 1320 REDCOD:
3960 ;

```

```

;read another input
;restore previous coord
;branch if input released
;[A]=ABS(X0-X1)

;less than 5?
;no, try again
;[A]=ABS(Y0-Y1)

;less than 5
;no, try again
;update coordinate [X]
;update coordinate [Y]
;finally enable interrupt
;get SENSE input value
;return value

```

```
3961 ; Read X,Y coordinate into [D,E]
3962 ;
3963 1320 0E 0A LD C,0AH ;change to channel to [Y] when done
3964 1322 CD 1332 CALL REDPAD ;read [X]
3965 1325 D8 RET C ;return if input released
3966 1326 55 LD D,L
3967 1327 D5 PUSH DE
3968 1328 0E 00 LD C,0 ;change to [X] after read
3969 132A CD 1332 CALL REDPAD ;read [Y]
3970 132D D1 POP DE
3971 132E 5D LD E,L ;store Y read out
3972 132F AF XOR A ;clear carry
3973 1330 67 LD H,A ;force input is OK
3974 1331 C9 RET
3975 1332
3976
3977
3978
3979
3980 1332 CD 135B CALL CHKEOC ;make sure AD completed
3981 1335 06 08 LD B,8 ;input 8 bits
3982 1337 51 LD D,C ;input channel# after done
3983 1338
3984 1338 CB 82 RES 0,D ;serial clock(SCK)=1
3985 133A CB 92 RES 2,D
3986 133C CD 136D CALL OUTGI
3987 133F CD 110C CALL INGI ;read PAD
3988 1342 67 LD H,A ;save SENSE status
3989 1343 1F RRA
3990 1344 1F RRA
3991 1345 1F RRA
```

```

3992 1346 CB L5 RL L
3993 1348 CB C2 SET 0,D
3994 134A CB D2 SET 2,D
3995 134C CD 136D CALL OUTGI
3996 134F 10 E7 DJNZ REDLOP
3997 1351 CB E2 SET 4,D
3998 1353 CB EA SET 5,D
3999 1355 CD 136D CALL OUTGI
4000 1358 7C LD A,H
4001 1359 1F RRA
4002 135A C9 RET
4003 135B
4004
4005
4006
4007 135B 3E 35 LD A,00110101B
4008 135D B1 OR C
4009 135E 57 LD D,A
4010 135F CD 136D CALL OUTGI
4011 1362
4012 1362 CD 110C CALL INGI
4013 1365 E6 02 AND 2
4014 1367 28 F9 JR Z,EOCHK
4015 1369 CB A2 RES 4,D
4016 136B CB AA RES 5,D
4017 136D
4018
4019
4020
4021 136D E5 PUSH HL
4022 136E D5 PUSH DE

```

```

;bit 2 to LSB of [L]
;SCK=0

;initiate another AD
;LSB=SENSE status
;SENSE status to carry
;OK if no carry

CHKEOC:
;
; Check and wait for EOC
;
EOCHK:
OUTGI:
;
; Output [D] to PAD
;
PUSH HL
PUSH DE

```

```

;reset CS
;test EOC
;set CS and return

```

4023	136F	2A F866	LD	HL, (RUNFLG)	
4024	1372	7D	LD	A,L	
4025	1373	2F	CPL		
4026	1374	A2	AND		;Also known as [PADWRK]
4027	1375	57	LD	D	
4028	1376	3E 0F	LD	D,A	
4029	1378	D3 A0	LD	A,PSG.PB	
4030	137A	DB A2	OUT	(PSG.LW),A	
4031	137C	A5	IN	A,(PSG.DR)	
4032	137D	B2	AND	L	
4033	137E	B4	OR	D	
4034	137F	D3 A1	OR	H	
4035	1381	D1	OUT	(PSG.DW),A	
4036	1382	E1	POP	DE	
4037	1383	C9	POP	HL	
4038			RET		
4039					

; SUBTTTL - MSXIO - Misc. routines for MSXIO