

Scanned and converted to PDF by HansO, 2005 Pages 1-280, see part2 for the rest. Edited: January 1985 by Steven M. Ting Graphic design: Mervin Fong.

The information in this document is subject to change without notice. ASCII Corp. makes no warranty with regard to this manual, including but not limited to, implied warranties of merchantability and fitness for a particular purpose. The parties above assume no responsibility for any errors which may appear in this document.

This document is not intended as "Consumer goods" under applicable federal or state law(s).

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of ASCII Corporation and Qest Publishing Inc.

MSX is a registered trademark of Microsoft Corporation, Bellevue, WA. 280 is a registered trademark of Zilog, Inc.

Printed in United States

MSX BIOS

Copyrighted © 1985 by ASCII Corporation of Japan

All rights Reserved

Published by

QEST PUBLISHING INC. 39 W. 32nd Street Suite 800 New York, N. Y. 10001

(212) 564-0749 Telex: 650-190-8083 MCI

TABLE OF CONTENTS

BIOS LISTING	1	-	256
MSX BIOS CROSS REFERENCE	257	-	280
SYMBOL TABLE	281	-	285
APPENDIX A MSX USA & UK OVERLAY PATCHES BIOS CALLS			
APPENDIX B CHARACTER SET & KEYBOARD LAYOUT HOOKS & RAM ROUTINES			

PAGE (MSX ROM BASIC BIOS) Macro-80 3.44 01-Jan-85 -BIOS header- BIOS calls (Basic Interpreter, Slot I/O)

.list		; (C) Copyright by ASCII Corp., 1983	; Proprietary information. All rights reserved.		; File: BIOHDR.MAC	; USE: Restart calls and ROM entries table	; Written by Jey Suzuki, Rick Yamashita	; ASCII Corporation, Japan		; Edit: January, 1985	; Reason: Zilog 280 Mnemonic version and cleanup	; Edited by: Steven M. Ting		••	; Labels referenced in this listing, are the absolute locations	; within the MSX ROM. However, "ONLY" this BIOS entry point table,	; and RAM variables are quaranteed to be permanent.		; All other locations in the ROM, will be changed without notice.		SUBTTL -BIOS header- BIOS calls (Basic Interpreter, Slot I/O)
l	3 2	4	5	6	7	8	6	10	11	12	13	14	15	16	17	18	19	20	21	22	23

-1

PAGE	
01-Jan-85	Slot I/0)
3.44	BIOS calls (Basic Interpreter, Slot I/O)
Macro-80	ls (Basic
(MSX ROM BASIC BIOS)	-BIOS header- BIOS call

							table, ROM.	Data read	write		, HL					
2	(KST U thru KST 5) are reserved for BASIC for inter-slot calls, and RST 7 for	;Fail safe	;Finds all connected RAM ;and cartridges	VDP. **	the VDP hardware directly ddress found here, to be certain	future versions of	;Address of character generator table, ;to allow use of other character ROM.	Current port address for VDP Da	=	the RST	; if equal to the byte pointed by		kead a pyte irom another slot	Fetch next char from BASIC text		;Write a byte to another slot
	Ω Ω		CHKRAM ; F	Special information for the VDP. **	Any program that accesses the VDE should read the 1/0 mort address	is compatib]	CGTABL ;A	98H ;C		SYNCHR ;C				IRGTR	0	WRSLT ;W
Int.	; The IOLLOWING KST'S ; interpreter, RST 6 1 ; hardware interrupt :	BEGIN: DI	ч. 	; ** Special i	; Any program ; should read	tware	DW	DB	DB	JP	ű	8U er	JD B(I	JP	DB	JP DB
) Mu alls			C3 02D7				lbbF	98	98	C3 2683	00	5		C3 2686		C3 01D1 00
ROM BASIC BIOS header- BIOS ca		0000	000				0004	0006	000	0008			000F	0010	0013	0014 0017
X 0 4 0 0	20 23 29 29	30	31 32 33 34	35	36 37	38 39	40 41 42	43	44 45	46	47	0 C	50 4	51	52	53 54

PAGE 2-1 3	or printer be of chara lapanese internation corea te format Y-M-D 1: D-M-Y tre	TD Byte (2) Format: Format: B7 B6 B5 B4 B3 B2 B1 B0 + + + + + + + + + + + + + + Type of Keyboard + + + + + 0.Japanese 2:French + + + + + + 1:Int 3:UK + + + + + 4:DIN
01-Jan-85 Slot I/O)	OUTDO 0 CALSLT 0 DCOMPR 0 GETYPR 0	
3.44 Interpreter,	5, 80 5, 80 5, 80 5, 80 5, 80	DB
) Macro-80 alls (Basic	C3 1B45 00 C3 0217 00 C3 146A 00 C3 25E 00 C3 2689 00	00
ROM BASIC BIOS header- BIOS ca	0018 0015 0017 0020 0023 0028 0028 0028 0028	002C
(MSX RO -BIOS he	2 2 2 7 2 7 2 2 2 2 2 2 2 2 2 2 2 2 2 2	7 7 7 8 8 8 1 8 8 8 8 8 8 8 8 8 8 8 8 8

PAGE 2-2 4	<pre>DB 0,0,0,0 DB 0,0,0,0 DB 0,0,0,0 DB 0,0,0,0,0 DB 0,0,0,0,0 DB 0,0,0,0,0 pB 0,0,0,0,0,0 pB 0,0,0,0,0 pB 0,0,0,0,0 pB 0,0,0,0,0,0 pB 0,0,0,0,0 pB 0,0,0,0,0 pB 0,0,0,0,0 pB 0,0,0,0,0 pB</pre>
01-Jan-85 Slot I/O)	0,0,0 CALLF CALLF 0,0,0,0 0,0,0,0 ince used for I, KEYINT INITIO INIFNK iNIFNK iniFNK
0 3.44 01-Jan Interpreter, Slot I/O)	DB DB JP DB TP JP JP JP JP SUBTTL -BIOS h
(MSX ROM BASIC BIOS) Macro-80 -BIOS header- BIOS calls (Basic In	00 00 00 C3 0205 00 00 00 00 00 C3 0C3C C3 049D C3 139D
OM BASIC I eader- BIC	002D 0030 0033 0037 0038 0038 0038
(MSX R -BIOS h	88 89 90 100 100 100

5 PAGE 3 5	following entry points provides control of the s registers, screen mode settings, and memory block between DRAM and VRAM.	;Disables screen display ;Enables screen display	;Write a byte to any VDP register ;Read VRAM addressed using [HL]	;Write VRAM addressed using [HL] :Sets un VDP for read	;Sets up VDP for write	;Fills VRAM with specified data	Ę			;change Foreground, background, ;border, color			;Handler for non-maskable interrupt		;Init sprite data	; Init VDP for 40 X 24 text mode (SCREEN 0)	; " " " 32 X 24 text mode (SCREEN 1)	; " " High resolution mode (SCREEN 2)	; " " Multi color mode (SCREEN 3)	o display	32 X
01-Jan-85 sor)	ng entry p ters, scre n DRAM and	DI SSCR ENASCR	WRTVDP RDVRM	WRT'VRM SF'TRD	SETWRT	FILVRM	LDIRMV	LDIRVM	CHGMOD	CHGCLR	0		IWN		CLRSPR	INITXT	INIT32	INIGRP	TIMINI	SETTXT	SETT32
3.44 01 splay processor)	The following entry points pr VDP's registers, screen mode move between DRAM and VRAM.	JP JP	df ff	qIJ dŢ,	Ч. Ч.Г	JP	đ	đj (qU ţ	٩L	DB	••	ЧĻ		ዊር	£	ዊ	ዊር	ዊ	ЧĻ	дГ
o-80 Jeo dis	~ ~ ~ ~ ~ ~											••		••							
BIOS) Macro-80 OS calls (Video display		00	C3 057F C3 07D7	C3 07CD C3 07EC			070			C3 U/F/	00		C3 1398				C3 0538	C3 05D2	C3 061F	C3 0594	0
ROM BASIC BIOS header- BIOS ca		0041 0044	0047 004A	004D 0050	0053	0056	0059	0050	005F	0062	0065		0066		0069	006C	006F	0072	0075	0078	007B
(MSX ROM -BIOS hea	102 103 105 105	108 109	110 111	113	114	115	116	717	811	119	121	122 123	124	125	126	127	128	129	130	131	132

_

S

; " " " High-res mode ; " " " Multi color mode	<pre>;Get address of sprite pattern table ; " " " attribute table</pre>	Returns current sprite size	; Print a character on the graphic screen	; SUBTTL -BIOS header- BIOS calls (Programmable Sound Generator control)
JP SETGRP JP SETMLT	JP CALPAT TD CALATR	0	JP GRPPRT	; -BIOS header- BIOS calls
0602 0659	06 E4	0704	1510	SUBTTL
ບ ເ	0084 C3 (ទួញ	ប	
133	135	136 137	138	139 140

.

و

PAGE	
01-Jan-85	l Generator cont
3.44	e Sound
Macro-80	(Programmabl
(MSX ROM BASIC BIOS) N	-BIOS header- BIOS calls

	; Following entry points are used for PSG initialization, : read and write PSG registers, and PLAY statement execution.		JP GICINI ; Init PSG, and static data for PLAY	JP WRTPSG ;Write data to PSG	JP RDPSG ;Read data from PSG	JP STRTMS ;Checks and start background task for PLAY		SUBTTL -BIOS header- BIOS calls (Keyboard, CRT, and Printer)						
			-	C3 1102	• •	•••								
			0600	0093	9600	6600								
141 142	143 144	145	146	147	148	149	150	151						

(MSX ROM BASIC BIOS) Macro-80 3.44 01-Jan-85 -BIOS header- BIOS calls (Keyboard, CRT, and Printer)

to printer, if possible -BIOS header- BIOS calls (Game and Cassette 1/0, Queue handler) Place cursor at Column [H], Row [L] Display Function key, if neccessary Same as ISCNTC, but used by BASIC Stop displaying the Function keys Read line from keyboard to buffer same as above, except in case of Checks status of keyboard status Print a "?", then jump to INLIN Checks for graphic header byte Checks status of line printer Return char typed, with wait Enable Function key display Output character to console Force screen to text mode [Control-STOP] pressed?? [Shift-STOP] pressed?? and convert code AUTFLG is set : Clear screen Buzz General INPUT and PRINT utilities. z TOTEXT OINLIN CKCNTC ERAFNK DSPFNK PINLIN I SCNTC LPTSTT CNVCHR BREAKX LPTOUT POSIT FNKSB INLIN CHSNS CHPUT CHGET BEEP CLS Ч Ч Ę ß f f ß ß 8 8 8 8 8 ß SUBTTL 088E 0B26 0B15 0B2B 083B O3FB 0848 085D 23D5 046F 10F9 1113 10CB 08BC 0884 089D **23BF** 23CC 0D6A 0000000000 ញ ញ 00C6 00C9 00 BA 00BD 0000 00C3 0000 OOCF 00D2 OOAE 009F 00A5 00A8 00AB 00B4 00B7 00A2 00Bl 0090 171 172 178 168 169 170 173 174 175 176 **158** 159 160 162 163 164 165 166 167 177 154 155 156 161 152 153 157

œ

ഹ

PAGE

PAGE	
14 01-Jan- 85	e I/0, Queue hand
3.44	and Cassette
Macro-80	(Game and
C BIOS)	BIOS calls
K ROM BASIC	BIOS header-
XSM)	-BIO

Following are used to read the value from Joysticks, Graphic pad (tablet), and Paddles.	;Return status of joystick ;Read joystick trigger button .Returns status of graphic pad	;Read paddle		Following are used to access the cassette tape,	tor on/off	;Turn on motor and read tape header	;Read tape data	;Stops reading from tape	;Turn on motor and write tape header	;Write data to tape	;Stops writing to tape	;Start, stop cassette motor, or	; flip motor (on to off, off to on)					;Bytes left in queue	;Send a byte to queue	- - -	BLUS CALIS (Generalized graphics)
Following are used to read the val Graphic pad (tablet), and Paddles.	GTTRIG GTTRIG GTPAD	GTPDL		are used to a	read/write, and motor on/off	TAPION	TAPIN	TAPIOF	TAPOON	TAPOUT	TAPOFF	STMOTR				es		LFTQ	PUTQ		-BLUS neader - BLUS
; Following ; Graphic pa	ar ar	с Ц		; Following	; data read/;	JP	ЧĻ	ЧĻ	ЧŲ	ЧĻ	ЧĻ	дГ		••	••	; BASIC queues	••	£	JP		SOIE- TITENS
	El m C) –				e	7)	6	_	•	D	4						В	2		
	C3 11E C3 125 C3 125	127				1A6	C3 1AB	19E	19F	IAl	19D	138						C3 14E	149		
	00D5 00D8 00D8	00DE				00EI	00E4	00E7	00EA	00ED	00100	00F3						00F6	00F9		
179 180 181 182 183	184 185 186	187	188 189	190	191 192	193	194	195	196	197	198	199	200	201	202	203	204	205	206 205	207	208

δ

PAGE 01-Jan-85 (MSX ROM BASIC BIOS) Macro-80 3.44 -BIOS header- BIOS calls (Generalized graphics)

BASIC interpreter's GENGRP and ADVGRP modules use	; Moves one pixel right	; " " Leit	dn = = .		; " " down		;Scales X Y cordinates	;Maps cordinates to physical address	;Get current physical address and	;mask pattern	; Put current physical address and	;mask pattern	;Sets the color attribute byte	;Reads attribute of current pixel	;Sets current pixel to specified attribute	; Sets pixel horizontally	;Returns aspect ratio	;Do paint initialization	;Scan pixels to the right	; " " " left		SUBTTL -BIOS header- BIOS calls (Misc. Entries)
.nterpreter'	RIGHTC	LEFTC	UPC	TUPC	DOWNC	TDOWNC	SCALXY	MAPXYC	FETCHC		STOREC		SETATR	READC	SETC	NSETCX	GTASPC	INITN	SCANR	SCANL		leader- BIOS
; For BASIC i	JP	ЧŲ	JP	ЧЪ	JP	ЧĻ	JP	ЧĻ	ЧĻ		JP		дЪ	ЧŲ	дЪ	đ	ЧЪ	ЧŲ	ЧĻ	đ	••	SUBTTL -BIOS
			C3 175D																C3 18E4			
	00FC	00FF	0102	0105	0108	010B	010E	1110	0114		0117		OIIA	011D	0120	0123	0126	0129	012C	012F		
209 210 211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233

10

PAGE 8		;Turn [CAPSLOCK] light, on/off	;Change status of 1 bit sound port ;Return output of primary slot register	;Write to primary slot register	;Read VDP status register	;Read a specified row in the	;keyboard matrix	;Performs operation for mass storage	;devices (such as disks)	;Initialize mass storage device	;Are we doing device I/O	;Output to line printer	;Used by Music background tasking		;Clear the keyboard buffer	;Performs far-call into BASIC	; RESERVED FOR EXPANSION		er stuff
01-Jan-85		CHGCAP	CHGSND RSLREG	WSLREG	RDVDP	SNSMAT		ОІДҮНД		FORMAT	ISFLIO	OUTDLP	GETVCP	GETVC2	KILBUF	CALBAS	005AH		- Slot handl
cro-80 3.44 Misc. Entries)		JP	JP JP	JP	ЧĻ	JP		дЪ		JP	JP	JP	JP	đ	JP	JP	DS	••	SUBTTL - SLOT - Slot handler stuff
Ma Ls (C3 0F7A C3 144C					C3 148A					C3 1470		-				
MSX ROM BASIC BIOS) BIOS header- BIOS call		0132	0135 0138	013B	013E	0141		0144		0147	014A	014D	0150	0153	0156	0159	015C		
(MSX RC -BIOS h∈	23 4 235	237	238 239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

acro-80 3.44 01-Jan-85 PAGE 9 uff	1 ;A8H read from PPI 1 ;A8H Write to PPI F 5cated at 0000-3FFFH must o	<pre>; this module which are entered via following addresses. ; 000CH RDSLT ; 0014H WRSLT ; 001CH CALSLT ; 0024H ENASLT</pre>	Selects the appropriate slot according to the value given through registers, and read the content of memory from the slot.	<pre>% Input parameters: % - FxxSSPP % % ++ primary slot # (0-3) % ++ l if secondary slot # (0-3) * + l if secondary slot # specified % HL - address of target memory</pre>	 A returned value A - content of memory Note: Interrupts are disabled automatically but never enabled
BASIC BIOS) M	00A8 00A8				
(MSX ROM - SLOT -	256 257 258 259 260	261 263 265 265 266 266 266 266 266 266 266 266	270 271 272 273 274	275 276 277 278 279 281 281 281 283	284 285 286

PAGE 9-1 13			;Calculate bit pattern and mask code	; Expanded slot specified		; Save current setting	; Cancel current setting for target address	;Add new setting	;Call read primitive routine (in system area)	;Return value via [Acc]			;Save target address	; Select secondary slot	;Restore target address and save [HL]			;Restore old slot select register	Write slot)
01-Jan-85	by this routine.		SELPRM	M, RDESLT	A, (PPI.AR)	D,A	C	б	RAMLOW	A,E			НL	SELEXP	(SP), HL	R	RDSLT	WRESED	Slot handler (Write slot)
3.44	by thi		CALL	ЧŲ	IN	ГD	AND	OR	CALL	LD	RET		PUSH	CALL	EX	PUSH	CALL	JR	-SLOT-
08		RDSLT:										RDESLT:							SUBTTL
BIOS) Macro-80 andler stuff			CD 027E		DB A8	57	Al	BO	CD F380	7B	60		ES	CD 02A3	E3	C5	CD 01B6	18 1B	
M BASIC BIOS) Slot handler		01B6	01B6	01B9	01BC	01BE	01BF	01C0	01C1	01C4	01C5	01C6	01C6	01C7	OlCA	OLCB	01CC	OICF	
(MSX ROM - SLOT -	287 288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306

PAGE 10	WRSLT	slot according to the value given vrite to the memory in the specified		<pre>iry slot # (0-3) idary slot # (0-3) secondary slot # specified</pre>	address of target memory	- value to be written	Interrupts are disabled automatically but never enabled by this routine.		; Save data to be written	jearculate bit pattern and mask code ; Expanded slot specified	;Restore data to be written		; Save current setting	; cancer current setting for target address ; Add new setting	
01-Jan-85		the appropriate slot registers, and write	ters: P	<pre> ++ primary slot # ++ secondary slot l if secondary</pre>	I.	- value to	Interrupts are di by this routine.		DE сет ром	M, WRESLT	DE	A, (PPI.AR)	D,A	صر	
3.44		Selects the through reginslot.	Input parameters: A - FxxxSSPP		ΗΓ	ы	Note: Inter by th	, WRSLT:	HSUG	JP	POP	IN		ANU OR	
S) Macro-80 r (Write slot)	•	~ ~ ~ ~			~ ~ •			-	5 0.027E			3 A8			
(MSX ROM BASIC BIOS -SLOT- Slot handler									01D1 D5					01DD B0	
(MSX RC -SLOT-	307 308 309	311 312 313	314 315 316	319 319 320	321 322 323	32 4 325	326 327 328	329	330	332	333	334	335 226	337	

PAGE 10-1 15	;Call write primitive routine (in system area)	;Save target address, get data to be written	; Save data to be written	;Select secondary slot	;Restore data to be written	;Restore target address and save [HL]					;Save target address and get old [HL]	;Save value returned by RDSLT	;Get current setting	;Cancel current setting for 0C000H0FFFFH		;Enable 0C000H0FFFFH of target bank	;Restore old setting of slot register		;Finally restore old primary slot register		;Restore value returned by RDSLT	;Restore target address	
01-Jan-85	WRPRIM	(SP), HL	НL	SELEXP	DE	(SP),HL	BC	WRSLT		BC	(SP),HL	AF	А,В	00111111B	U	(PPI.AW),A	A,L	(OFFFFH),A	Α,Β	(PPI.AW),A	AF	Η	
-80 3.44 slot)	JP WRESLT:	EX	HSUG	CALL	POP	EX	HSUI	CALL	WRESED:	POP	EX	HSUI	ГD	AND	OR	TUO	ΓD	LD	ГD	OUT	POP	POP	RET
Macro (Write	C3 F385	E3	ES	CD 02A3	Dl	E3	C5	CD 01D1		C1	E3	F5	78	E6 3F	Bl	D3 A8	7D	32 FFFF	78	D3 A8	Fl	EI	60
MSX ROM BASIC BIOS) -SLOT- Slot handler	01DE 01E1	OlEl	01E2	01E3	01E6	01E7	01E8	01E9	OLEC	01EC	01ED	OLEE	OLEF	01FO	01F2	01F3	01F5	01F6	01F9	01FA	OlFC	OlfD	OLFE
(MSX RO SLOT-	338 339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361

PAGE 11						;Get return address, save [HL]	; Save working registers		;Get destination slot		; Move it to IYH		;Get destination address			;Prepare true return address		; Move it to IX	;Restore working registers		;Resture [HL], save true return address	
01-Jan-85			IY, (EXPTBL-1)	CALSLT		(SP),HL	AF	DE	A, (HL)	AF	IY	HL	E, (HL)	HL	D, (HL)	HL	DE	IX	DE	AF	(SP), HL	
3.44			LD	JR		EX	HSUG	HSUA	LD	HSUG	POP	INC	LD	INC	LD	INC	PUSH	POP	POP	POP	ΕX	-SLOT-
00 5t)		CALBAS:			CALLF:																	SUBTTL
3IOS) Macro-80 31er (Write slot)			FD 2A FCC0	18 12		E3	F5	D5	7E	F5	FD El	23	5E	23	56	23	D5	DD El	Dl	Fl	E3	
MSX ROM BASIC BIOS SLOT- Slot handler		01FF	01FF	0203	0205	0205	0206	0207	0208	0209	020A	020C	020D	020E	020F	0210	0211	0212	0214	0215	0216	
(MSX ROI -SLOT-	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383

17		ter	
		regis	
12	CALSLT cALSLT cALSLT cALSLT	;Save environments ;Get target slot information ;Get target address ;Call expanded slot ;Save current value of primary slot register	
PAGE	CALSLT cALSLT cALSLT cALSLT	;Save ;Get t ;Call ;Save	
-85	all to ury slc idary s second disabl		
01-Jan-85	t co t co	AF,AF' IY AF AF IX HL SELPRM M,CALESL A,(PPI.AR) AF	
3.44	Performs inter-sl Input parameters: IY - FxxxSSPP ++ + IX - address to Note: Interrupts by this ro You can ne of 280.	.T: EXX EX EX PUSH POP POP CALL JP IN PUSH	
0		; CALSLT:	
Ma cro-8			
BIOS)		D9 FD E5 F1 E5 DD E5 E1 CD 027E FA 022E F3 A8 F5	
(MSX ROM BASIC -SLOT-		0217 0217 0218 0218 0218 0215 0215 0215 0217 0222 0225 0227	
(MSX R -SLOT-	20100987655433333333333333333333333333333333333	403 404 405 405 403 403 403 411 411 413 413	

(MSX ROM -SLOT-	MSX ROM BASIC BIOS SLOT-)S) Macro-80	3.44	01-Jan-85	PAGE 12-1	18
1010						
415		АІ	AND	c	;Cancel current setting for target address	
416		BO	OR	В	;Add new setting	
417	022A I	60	EXX		;Restore environments except PSW	
418		C3 F38C	ЧĿ	CLPRIM	;Jump to primitive routine (in system area)	
419	022E	CALESL:				
420		CD 02A3	CALL	SELEXP	;Select secondary slot register	
421		75 2	PUSH	AF	;Move primary slot # in [IYH]	
422	0232 F	FD El	POP	IΥ		
423		35	PUSH	НL	;Save [B,C,L] which contain information	
424		25	HSUG	BC	; for restoring slot environments	
425		1F	ГD	C,A	;Move primary slot # to [BC]	
426		06 00	ΓD	B,0		
427	0239 7	7D	LD	A,L	;Re-calculate what is currently output	
428		34	AND	Н	;to expansion slot register	
429		32	OR	D		
430		21 FCC5	ГD	HL , SLTTBL	;Calculate address into SLTTBL	
431		90	ADD	HL, BC		
432		77	LD	(HL),A	; Set current value output to expansion	
433					;slot register	
434		ES	PUSH	ΕĹ	;Remember this address	
435		08	EX	AF, AF	;Restore possible arguments passed	
436		90 0	EXX		;via registers	
437	0244 0	CD 0217	CALL	CALSLT	;Call by primary slot #	
438		60	EXX		:Save possible values returned via	
439		38	EX	AF, AF'	;registers	
440		El	POP	円	;Restore address into SLTTBL	
441		1	POP	BC	;Restore information about old slots	
442		DI	POP	DE		
443		78	LD	A, B	;Get current setting	
444	024D F	E6 3F	AND	00111111B	;Cancel current setting for 0C000H0FFFFH	
		31	OR	U		

PAGE 12-2		;Enable OCOOOHOFFFFH of target bank	Restore old setting of slot register		Finally restore old primary slot register		; And change SLTTBL also	Restore possible returned values		
01-Jan-85		(PPI.AW),A	A, E	(OFFFFH),A	A, B	(PPI.AW),A	(HL),E	AF'AF'		
3.44	DI	OUT	ГD	LD	ГD	OUT	ГD	EX	EXX	RET
MSX ROM BASIC BIOS) Macro-80 SLOT-	F3	D3 A8	7B	32 FFFF	78	D3 A8	73	08	D9	60
M BASIC	0250	0251	0253	0254	0257	0258	025A	025B	025C	025D
(MSX ROM -SLOT-	446	447	448	449	450	451	452	453	454	455

01-Jan-85 PAGE 13	ENASLTENASLT	the surveyiste slot scending to the value given	through registers, and permanently enables the slot.		ters:	Q		++ primary slot # (U-3) -+ secondary slot # (D-3)	l if secondary		address of target memory		Interrupts are disabled automatically but never enabled	by this routine.			;Calculate bit	M,ENESLT ;Expanded slot specified	A, (PPI.AR)	C ;Cancel current setting for target address	B ;Add new setting	(PPI.AW), A			·	SELEXP ;Select secondary slot
3.44		Colocts tho	through regi		Input parameters:	A – FxxxSSPP		+ + +			HL - address		Note: Inter	by th		ENASLT:	CALL	дЪ	IN	AND	OR	OUT	RET	ENESLT:	HSUG	CALL
Macro-80							••	•~ •		. •-	••	••	••	••	••	щ	027E	026B	8			8		ш		02A3
BIOS)																		FA 0.	DB A8	Al	BO	D3 A8	ഇ			CD CD
ROM BASIC BI																025E	025E	0261	0264	0266	0267	0268	026A	026B	026B	026C
(MSX R -SLOT-	456 457 458	459	46U 461	462	463 464	465	466	467	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486

PAGE 13-1	<pre>;Move primary slot # to [BC] ;Re-calculate what is currently output ;to expansion slot register ;Calculate address into SLTTBL ;Calculate address into expansion ;Set current value output to expansion ;Set current value output to expansion ;Restore target address ;Restore primary slot # to [Acc] ;Enable by primary slot register</pre>	
01-Jan-85	C,A B,0 A,L H D HL,SLTTBL HL,BC (HL),A HL A,C ENASLT	
3.44	LD LD LD AND OR ADD LD POP LD JR	
MSX ROM BASIC BIOS) Macro-80 SLOT-	F 4F 0 06 00 2 7D 3 A4 4 B2 5 21 FCC5 8 09 77 9 77 B 77 C 18 E0 C 18 E0	
om Basi	026F 0270 0273 0273 0273 0274 0275 0278 0278 0278 0277	
(MSX R -SLOT-	4 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	

	espond to address 0000-3FFF 4000-7FFF 8000-BFFF C000-7FFF 0000-3FFF 4000-7FFF 8000-BFFF 2000-FFFF 2000-FFFF
PAGE 14	<pre>;Save slot address ;Extract upper 2 bits ;Format mask pat. correspond to ;Format mask pat. correspond to 000001100 8000-3F 00011000 8000-BF 110000000 0000-3F 1111100 4000-7F 1111100 4000-7F 1111100111 4000-7F 11111001111 8000-BF ;Save mask pattern i1111100 0000-8F ;Save mask pattern ;Save mask pattern i1111100 1000 ;Save mask pattern ;Save mask pattern pattern ;Save mask pattern pat</pre>
01-Jan-85	AF A,H 00000011B E,A A,0C0H E,A E,A C,A C,A C,A C,A B,A B,A
3.44	RM: PUSH PUSH PUSH RLCA RLCA AND JP LD DEC JP LD LD LD LD PUSH RLCA AND DEC JP LD RLCA RLCA AND DEC JP LD RLCA RLCA LD DEC JP LD RLCA RLCA LD DEC LD LD LD LD LD LD LD LD LD LD
)S) Macro-80	F3 F5 7C 07 07 07 5F 5F 03 5F 03 5F 07 07 07 07 07 07 07 07 2F 4F 5F 603 5F 603 5F 47 57 47 57 47 57 47 57 57 57 57 57 57 57 57 57 57 57 57 57
(MSX ROM BASIC BIOS -SLOT-	027E 027E 027E 027E 027E 028E 028B 0288 0298 0298 0298 0298 0298 0298 0298 0298 0298 0298 0298 0298 0298 0298 0298 0298 0298
(MSX ROM -SLOT-	499 500 500 502 503 503 511 500 511 500 511 500 511 500 511 500 511 500 511 500 511 500 511 500 500

14-1	;Convert slot # to proper bit pattern			n for primary	00000000 slot #0	01010101 slot #1	10101010 slot #2	<pre>11111111 slot #3</pre>	;Extract significant bits	Set it to [B]	;Expanded slot specified?	Set sign flag if so			Save target slot	oit pattern for primary slot	act slot # for 0C000H0FFFFH	it	Restore target slot	Save target slot	;Load [D] with specified slot address		;Save current setting	;Cancel current setting for 0C000H0FFFFH		;Enable 0C000H0FFFFH or target bank	Load slot information			;Extract secondary slot #
PAGE	; Conve			; Save	••	••		••	; Extra	;Set	; Expar	;Set a			; Save	;Get bit	; Extract	;Save it	;Resto	; Save	;Load		;Save	;Cance		; Enab	;Load			; Extra
01-Jan-85	A,10101011B	A,0101010B	SL PRM2	D,A					Е	B, A	AF	A			AF	A, D	1100000B	С,А	AF	AF	D,A	A, (PPI.AR)	B,A	00111111B	U	(PPI.AW),A	A, D	•		00000011B
3.44	ГD	ADD	DJNZ	LD					AND	ГD	POP	AND	RET		HSUG	LD	AND	ГD	POP	HSUG	ГD	IN	LD	AND	OR	OUT	LD	RRCA	RRCA	AND
Ma cr o-80	SL PRM2:													SELEXP:																
<u>^</u>	3E AB	C6 55	10 FC	57					A3	47	F]	A7	62		F5	7.A	E6 C0	4F	Ŀ	F5	57	DB A8	47	E6 3F	Bl	D3 A8	7A	OF	0F	E6 03
MSX ROM BASIC BIOS SLOT-	97 99		9B	9D								02A1 1	C2A2 (02A3		02A4		02A7 4		02A9 1	02AA		02AD 4	02AE 1	02B0				02B5 (
(MSX R -SLOT-	530 531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560

	proper							value														
	slot # to		slot #0 «lot #1		slot #3	be added		strip off old	4		register	•	a	•		register	'n	imary port	1	ary slot	1	
14-2	;Convert secondary	;bit pattern	000000000000000000000000000000000000000	10101010	1111111	Make bit pattern to be	Save this	Make bit pattern to strip off old value	ı	;Save this	;Read expanded slot register	1	Save current setting	Strip off old bits	; And set new bits	Set secondary slot register	1	Restore original primary port	Restore target slot	Fake read from primary slot	a	
PAGE	; Conv	;bit			••	;Make	; Save	; Make		; Save	; Read		; Save	; Stri	; And	;Set		;Rest	;Rest	; Fake		
01-Jan-85	D,A A,1010111B	A,0101010B D	P, SLEXP1			ы	D,A	Α,Ε		Н,А	A, (OFFFH)		L,A	Н	D	(OFFFFH),A	A, B	(PPI.AW),A	AF	00000011B		0 - I/O Module
3.44	LD	ADD DEC	ЧĻ			AND	LD	ГD	CPL	LD	LD	CPL	ĽD	AND	OR	ГD	LD	OUT	POP	AND	RET	- OIXSW -
0-80	SLEXPL																					SUBTTL -
Macr	57 3E AB	C6 55 15	F2 02BB			A3	57	7B	2F	67	3A FFFF	2F	6F	A4	B2	32 FFFF	78	D3 A8	Fl	E6 03	60	
MSX ROM BASIC BIOS) SLOT-	02B8 02B9 02B9	02BB 02BD	02.BE			02C1	02C2	02C3	02C4	02C5	02C6	02C9	02CA	02CB	02CC	02CD	02D0	02D1	02D3	02D4	02D6	
(MSX ROM -SLOT-	561 562 563	564 565	566 567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586

25		color	
15		 ;98H Read/write data VDP ;99H write command to VDP ;99H read status from VDP ;1n text mode, foreground and background color ;0therwise background color 	latch address for PSG write data to PSG read data from PSG A of PSG B of PSG B of PSG B of PSG B of PSG Fread from PPI Port A read from PPI Port A read from PPI Port A
PAGE	g	;98H ;99H ;99H ;1n te ;0ther	on ;A0H ;A2H ;A2H ;Port ;Port ;Port ;A8H ;A9H ;A8H ;A8H
3.44 01-Jan-85	<pre> Port definition Port definition Port definition Port definition Port definition Port definition Port definition Po</pre>	EQU 10011000B EQU 10011001B EQU 10011001B EQU 7	PSG address definition EQU 10100001B ; EQU 10100001B ; EQU 10100010B ; EQU 14 EQU 14 EQU 14 EQU 15 PPI address definition EQU 10101001B ; EQU 10101001B ; EQU 10101001B ;
Ma cr o-80		VDP.DRW VDP.CW VDP.SR ; V.COLR	; PSG.LW PSG.DW PSG.DR ; PSG.PA PSG.PB ; ; PSG.PB ; ; PPI.AR PPI.CR PPI.CR
l BASIC BIOS) I/O Module		8000 9099 9000 7000	00A0 00A1 00A2 000E 000E 00A8 00A8 00A8 00A8
(MSX ROM - MSXIO -	588 590 592 594 594 594 594	596 597 598 600 601	602 603 606 606 608 608 610 612 613 612 615 615 615 615

	PPI Port C PPI command register		SCREEN 0	SCREEN 1 AT	SCREEN 2 AT	SCREEN 3	
15-1	write to] write to]	;Data port ;Strobe output ;Printer status		ode (graphics 1) T32NAM,T32COL,T32CGP,T32ATR,T32PAT	mode GRPNAM , GRPCOL , GRPCGP , GRPATR , GRPPAT	, МГТРАТ	, LINL40
PAGE	; AAH ; ABH finition	;Data port ;Strobe ou ;Printer s	1) rcgP	lics 1) 2COL,T32CGP	PCOL, GRPCGP	color mode MLTNAM, MLTCGP, MLTATR, MLTPAT	size LINLEN,CRTCNT,LINL32,LINL40 al constants
01-Jan-85	10101010B ;A 10101011B ;A Printer port definition	10010001B 10010000B 10010000B	Text mode (40*24) TXTNAM, TXTCGP	Text mode (graphics 1) T32NAM,T32COL,T3	Hires mode GRPNAM,GRI	Multi-color mode MLTNAM,MLT	Screen size LINLEN,CRTCN External constants
3.44	EQU Pr	EQU EQU	Те	Те	Ні	nw	EX CC
Ma cr o-80	PPI.CW PPI.CM ;	LPT.DW LPT.SB LPT.ST					
MSX ROM BASIC BIOS) MSXIO - I/O Module	00AA 00AB	1600 0600					
(MSX R(- MSXIO	618 619 620 621	623 624 625 625	627 628 628	630 631 632 633	63 4 635 636 637	638 639 640 641	6 4 4 6 4 5 6 4 5 6 4 7 6 4 7 8 4 7 8 8 8

MSX ROM BASIC BIOS) MSXIO - I/O Module	Macro-80	3.44 01-Jan-85	PAGE 15-2
649		CGTABL	Character generator table
650	•••		'n
651	••	External variables	
652			
653		FORCLR	Foreground color
654	••	BAKCLR	Background color
655	••	BDRCLR	Border color for PAINT
656		SCRMOD	Current screen mode
657			
658			I
659			1
660			I
661	••	OLDSCR	
662		NAMBAS	Base of current name table
663		CGPBAS	of
664	••	PATBAS	of current sprit
665		ATRBAS	
666	••	JIFFY	ı
667	••	CLIKSW	Click switch
668	••	CLIKFL	Click flag to suppress multiple key clicks
669	••	RG0SAV	VDP register #0 save area
670	••	RG1 SAV	
671	••	STATFL	ter
672	••	PATWRK	Work area for pattern converter
673	••		
674	••	External routines	
675	•••		
676	•••	GETQ	
677	••	PUTQ	
678		INITO	
043			

; 0000xxxxx ; ;	; C - 'slot-expanded' Ilag	; DE - lowest address of the biggest RAM block ever found	I	; SPH - slot # of the biggest RAM block	; B - non 0 if we're now checking secondary slot		DE DU A, F DET CM A	D3 A8 00T (FFL.AW),A 35 50 T.D 2 10'	AF XOR A ; Port C - output (mode	D3 AB OUT (PPI.CM), A ; Port B -		; register until the RAM is found.	; subroutine calls 'cause we do not yet know where the available ; RAM exits. Everything has to be done inside ROM and CPU's	; area. Note that we cannot use RAM as work area nor perform	; Look into every slot from OFFFFH to C000H, and set system work		<pre> CHKRAM</pre>	very slot from 0 that we cannot u calls 'cause we Everything has t til the RAM is f A,82H (PPI.CM),A A,P' (PPI.CW),A hing f ve're now chec f we're now chec f we're now chec age: f we're now chec f address of the xpanded' flag	CHKRAM: CHKRAM: CHKRAM: Look into er area. Note i subroutine er RAM exits. J register un UD OUT XOR OUT LD OUT LD OUT Start searcl B - non 0 i SPH - slot SPH - slot SPH - slot DE - lowest C - 'slotee	
<pre>non 0 if we're r - slot # of the - secondary slot lowest address 'slot-expanded'</pre>	ŭ	ă I I	ă I	- non 0 if we're now checking		; Start searching	; Start searching	D3 AA ; CUT (PPI.CW),A ; Start searching ;	D3 A8 0UT (PPI.AW),A 3E 50 LD A,'P' D3 AA 0UT (PPI.CW),A ; Start searching ;	AF XOR A D3 A8 OUT (PPI.AW),A 3E 50 LD A,'P' D3 AA OUT (PPI.CW),A ; Start searching ;	D3 AB OUT (PPI.CM),A AF XOR A D3 A8 OUT (PPI.AW),A 3E 50 LD A,'P' D3 AA ; ; Start searching ;	3E 82 3AB D3 AB AF D3 AB AF D3 AB D3 AB D3 AB D3 A8 D3 A8 D3 AA D3 AA CUT (PPI.CM),A A COT (PPI.CM),A A COT (PPI.CM),A A COT (PPI.CM),A A COT (PPI.CM),A A COT (PPI.CM),A A COT (PPI.CM),A A COT (PPI.CM),A A COT (PPI.CM),A A COT (PPI.CM),A A COT (PPI.CM),A A COT (PPI.CM),A A COT (PPI.CM),A A COT (PPI.CM),A A COT (PPI.CM),A A COT (PPI.CM),A A COT (PPI.CM),A A COT (PPI.CM),A A COT (PPI.CM),A A A (PPI.CM),A A COT (PPI.CM),A A A (PPI.CM),A A A (PPI.CM),A A A (PPI.CM),A A A (PPI.CM),A A (PPI.CM),A A (PPI.CM),A A (PPI.CM),A A A (PPI.CM),A (PPI.CM),A A A (PPI.CM),A (PPI.CM),A A (PPI.CM),A	; register until the RAM is found 3E 82 D3 AB AF D3 AB AF D3 AB D3 AB D3 AB D3 AB D3 AB D3 AB D3 AB D0T CPI.CM),A A,PI D1 A,PI.CM),A A COT CPI.CM),A A COT CPI.CM),A A COT CPI.CM),A A COT CPI.CM),A A COT CPI.CM),A A COT CPI.CM),A A COT CPI.CM),A A COT CPI.CM),A A COT CPI.CM),A A COT CPI.CM),A A COT CPI.CM),A A COT CPI.CM),A A COT CPI.CM),A A COT CPI.CM),A A COT CPI.CM),A A COT CPI.CM),A A COT CPI.CM),A A COT CPI.CM),A A COT CPI.CM),A COT CPI.CM),A A COT CPI.CM),A COT CPI.CM),A COT CPI.CM),A COT CPI.CM),A COT CPI.CM),A COT CPI.CM),A COT CPI.CM),A CPI.CM	<pre>; subroutine calls 'cause we do r ; RAM exits. Everything has to be ; register until the RAM is found] 28 82</pre>	; area. Note that we cannot use 1 ; subroutine calls 'cause we do r ; RAM exits. Everything has to be ; register until the RAM is found ; register until the RAM is found ; D3 AB A,82H OUT (PPI.CM),A A AF XOR A OUT (PPI.CM),A A 3E 50 UT (PPI.AW),A LD A,'P' D3 AA D3 AA CUT (PPI.CW),A ; ; start searching ;	CHKRAM: CHKRAM: CHKRAM: CHKRAM: Look into every slot from OFFF area. Note that we cannot use I subroutine calls 'cause we do I RAM exits. Everything has to be RAM exits. Everything has to be register until the RAM is found TD A,82H D3 AB CUT (PPI.CM),A AF D3 AB D3 AA D3 AA D3 AA CUT (PPI.CW),A CUT		a ge:	; Register us	
- I - D -	j i ng st	r n n t	n n t	gister usage: - non 0 if we're now checking	; Register usage:	; Start searching	; Start searching	D3 AA OUT (PPI.CW),A ; ; Start searching	D3 A8 00T (PPI.AW),A 3E 50 LD A,'P' D3 AA 0UT (PPI.CW),A ; Start searching	AF XOR A D3 A8 OUT (PPI.AW),A 3E 50 LD A,'P' D3 AA OUT (PPI.CW),A ; Start searching	D3 AB OUT (PPI.CM),A AF XOR A D3 A8 OUT (PPI.AW),A 3E 50 LD A,'P' D3 AA OUT (PPI.CW),A ; Start searching	3E 82 3E 82 D3 AB AF D3 AB D3 AB D3 AB D3 A8 D3 A8 D3 A8 D3 A8 D0T CDT CDT CPI.CM),A A COT CPI.CM),A A CPI.CM),A CP	; register until the RAM is found 3E 82 D3 AB AF AF D3 AB AF D3 AB D3 AB D3 AB D3 AB D3 AB D3 AB D3 AB CUT (PPI.CM),A ACM,A CUT (PPI.CM),A ACM,A CUT (PPI.CM),A ACM,A CUT (PPI.CM),A ACM,A CUT (PPI.CM),A ACM,A CUT (PPI.CM),A ACM,A CUT (PPI.CM),A ACM ACM ACM ACM ACM ACM ACM AC	<pre>; subroutine calls 'cause we do r ; RAM exits. Everything has to be ; register until the RAM is found ; 3E 82</pre>	; area. Note that we cannot use 1 ; subroutine calls 'cause we do r ; RAM exits. Everything has to be ; register until the RAM is found 3 RA 3 RA 3 RA 3 RA 3 RA 3 RA 3 RA 3 RA	CHKRAM: CHKRAM: CHKRAM: CHKRAM: Look into every slot from OFFFI area. Note that we cannot use I subroutine calls 'cause we do I subroutine calls 'cause we do I RAM exits. Everything has to be register until the RAM is found The A, 82H D3 AB SE 82 D3 AB D3 AB D3 AB D3 AA D3 AA D3 AA CUT (PPI.CM),A CUT (PPI.CW),A CUT (PPI.CW),A CUT (PPI.CW),A CUT (PPI.CW),A Start searching Start searching			••	
an national states and a states a	- i n n t	r i n t	- nc	gister usage: - non 0 if we're now checking	; ; Register usage:			D3 AA ; OUT (PPI.CW),A	D3 A8 00T (PPI.AW),A 3E 50 LD A,'P' D3 AA . OUT (PPI.CW),A	AF XOR A D3 A8 OUT (PPI.AW),A 3E 50 LD A,'P' D3 AA ; OUT (PPI.CW),A	D3 AB OUT (PPI.CM), A AF XOR A D3 A8 OUT (PPI.AW), A 3E 50 LD A, 'P' D3 AA OUT (PPI.CW), A	; 3E 82 D3 AB AF AF D3 A8 D3 A8 D3 A8 D3 A8 D3 A8 D3 A8 D0T CP1.CM),A A COT CP1.CM),A COT CP1.	; register until the RAM is found 3E 82 3E 82 D3 AB AF D3 AB D3 AB D3 AB D3 AB D3 AB D3 AB D3 AB D3 AB D3 AB D0T CPI.CM),A ACM,A COT CPI.CM),A ACM,A COT CPI.CM),A ACM,A COT CPI.CM),A ACM CPI.CM),A COT CPI.CM),A C	; subroutine calls 'cause we do r ; RAM exits. Everything has to be ; register until the RAM is found ; register until the RAM is found 3 as b b3 AB OUT (PPI.CM),A AF XOR A D3 AB OUT (PPI.CM),A 3E 50 LD A,P' D3 AA . D3 AA . D3 AA . D3 AA . D3 AB . D1 A,P'	; area. Note that we cannot use 1 ; subroutine calls 'cause we do r ; RAM exits. Everything has to be ; register until the RAM is found ; register until the RAM is found 3 LD A,82H D3 AB OUT (PPI.CM),A AF XOR A D3 AB OUT (PPI.CM),A AF 20 D3 AB OUT (PPI.CM),A 3E 50 D3 AA CUT (PPI.CW),A 3B 50 D3 AA CUT (PPI.CW),A 3B 50 D0T (PPI.CW),A 3C 20 D3 AA CUT (PPI.CW),A 2C 20 D3 AB CUT (PPI.CW),A 2C 20 D3 AC	CHKRAM: CHKRAM: CHKRAM: CHKRAM: CHKRAM: Look into every slot from OFFFI area. Note that we cannot use I subroutine calls 'cause we do I RAM exits. Everything has to be RAM exits. Everything has to be register until the RAM is found register until the RAM is found AR D3 AB D3 AB D3 AB D3 AB D3 AB D3 AB CUT (PPI.CM),A CUT (PPI.CW),A CUT (PPI.CW)		hing	; Start seard	
<pre>; Look into every slot from OFFF ; area. Note that we cannot use 1 ; subroutine calls 'cause we do r ; RAM exits. Everything has to be ; register until the RAM is found ; register until the RAM is found aF D3 AB D3 AB CUT (PPI.CM),A A, P' D3 AB D3 AB D3 AB CUT (PPI.CW),A A, P' D3 AB CUT (PPI.CW),A A, P' D0T (PPI.CW),A A, P' OUT (PPI.CW),A A, P' D1 A, P' D1 A, P' CUT (PPI.CW),A A, P' C - 'slot-expanded' flag C - 'slot-expanded' flag</pre>	<pre>; Look into every slot from OFFFFH to C000H, and set ; area. Note that we cannot use RAM as work area nor ; subroutine calls 'cause we do not yet know where th ; RAM exits. Everything has to be done inside ROM and ; register until the RAM is found. 3E 82 LD A,82H ; Port A - output (mod OUT (PPI.CM),A ; Port B - input (mod AF XOR A ; Port C - output (mod OUT (PPI.AW),A ; Select slot 0 for al 3E 50 OUT (PPI.CW),A ; port C - output (mod 0UT (PPI.CW),A ; port B - input (mod B AF XOR A ; Port C - output (mod OUT (PPI.CW),A ; port B - input (mod B AF XOR A ; Port C - output (mod OUT (PPI.CW),A ; port C - output (mod B AF XOR A ; Port C - output (mod B AF XOR A ; PORT B - input (mod B AF XOR A ; PORT B - input (mod B AF XOR A ; PORT B - input (mod B AF XOR A ; PORT B - input (mod B AF Y ; PORT B + input (mod B AF Y ; PORT B AF AF</pre>	<pre>; Look into every slot from OFFF ; area. Note that we cannot use I ; subroutine calls 'cause we do r ; RAM exits. Everything has to be ; register until the RAM is found ; register until the RAM is found 3E 82 0UT (PPI.CM),A AF 0UT (PPI.CM),A AF 0UT (PPI.CM),A 3E 50 0UT (PPI.CW),A 3E 50 0UT (PPI.CW),A 3E 50 0UT (PPI.CW),A 3E 50 0UT (PPI.CW),A ; start searching ; Register usage: ; SPH - slot # of the biggest RAN ; SPL - secondary slot # of the liggest RAN</pre>	<pre>; Look into every slot from OFFF ; area. Note that we cannot use I ; subroutine calls 'cause we do r ; RAM exits. Everything has to be ; register until the RAM is found ; register until the RAM is found] AB AA AA AAA AF AAA AAAAAAAAAAAAAAAAAAA</pre>	<pre>; Look into every slot from OFFF ; area. Note that we cannot use 1 ; subroutine calls 'cause we do r ; RAM exits. Everything has to be ; register until the RAM is found ; register until the RAM is found 3E 82</pre>	<pre>; Look into every slot from OFFF ; area. Note that we cannot use 1 ; subroutine calls 'cause we do r ; RAM exits. Everything has to be ; register until the RAM is found ; register until the RAM is found 3E 82</pre>	; Look into every slot from OFFF ; area. Note that we cannot use I ; subroutine calls 'cause we do r ; RAM exits. Everything has to be ; register until the RAM is found ; register until the RAM is found ; register until the RAM is found 3E 82 LD A,82H D3 AB XOR A D3 AB XOR A D3 AB XOR A D3 AB CUT (PPI.CM),A AF COT CW A	<pre>; Look into every slot from OFFF ; area. Note that we cannot use 1 ; subroutine calls 'cause we do r ; RAM exits. Everything has to be ; register until the RAM is found ; register until the RAM is found 3E 82 LD A,82H D3 AB XOR A D3 AB XOR A D3 AB XOR A D3 AB CUT (PPI.CM),A AF 00T (PPI.CM),A AF 00T (PPI.CM),A</pre>	3E 82 D3 AB AF AF	3E 82 D3 AB	3E 82	 Flook into every slot from OFFFH to C000H, and set system work area. Note that we cannot use RAM as work area nor perform subroutine calls 'cause we do not yet know where the available RAM exits. Everything has to be done inside ROM and CPU's register until the RAM is found. 	 ; Look into every slot from OFFFH to C000H, and set system work ; area. Note that we cannot use RAM as work area nor perform ; subroutine calls 'cause we do not yet know where the available ; RAM exits. Everything has to be done inside ROM and CPU's 	; Look into every slot from OFFFH to C000H, and set system work ; area. Note that we cannot use RAM as work area nor perform	; Look into every slot from OFFFFH to C000H, and set system work		CHKRAM: ; ;			••	
<pre> Look into every slot from OFFF area. Note that we cannot use 1 subroutine calls 'cause we do r register until the RAM is found register usage: Register usage: reft he biggest RAM register usage: register usage:</pre>	<pre> Look into every slot from OFFF area. Note that we cannot use 1 subroutine calls 'cause we do r register until the RAM is found register until the RAM is found ar ar</pre>	<pre> Look into every slot from OFFF area. Note that we cannot use I subroutine calls 'cause we do r subroutine calls 'cause we do r register until the RAM is found register until the RAM is found ar b3 AB b3 AB b0T b1D b1D b1CW),A cUT b1LD A,P' b0T cUT b1LD A,P' b0T cUT cPPI.CW),A b1 cUT cPPI.CW),A cUT cPPI.CW),A cUT cPPI.CW),A cUT cPPI.CW),A cUT cPPI.CW),A cUT cPPI.CW),A cUT cPPI.CW),A cUT cPPI.CW),A cUT cPPI.CW),A cUT cPPI.CW),A cUT cPPI.CW),A cUT cPPI.CW),A cOT cPPI.CW),A cOT cPPI.CW),A cPPI.CW),A cPPI.CW),A cPPI.CW),A cPPI.CW),A cPPI.CW),A cPPI.CW),A cPPI.CW),A cPPI.CW),A cPPI.CW),A cPPI.CW),A cPPI.CW),A cPPI.CW),A cPPI.CW),A cPPI.CW),A cPPI.CW,CW cPPI.CW,CW cPPI.CW,CW cPPI.CW,CW cPPI.CW,CW cPPI.CW,CW cPPI.CW,CW cPPI.CW,CW cPPI.CW,CW cPPI.CM,CW cPPI.CM,CW cPPI.CM,CW cPPI.CM</pre>	<pre> Look into every slot from OFFF area. Note that we cannot use 1 subroutine calls 'cause we do r subroutine calls 'cause we do reckin subroutine calls 'cause row checkin subroutine calls 'cause we can out the biggest RAN </pre>	<pre> Look into every slot from OFFF area. Note that we cannot use 1 subroutine calls 'cause we do r subroutine calls 'cause we do r register until the RAM is found are</pre>	<pre> Look into every slot from OFFF area. Note that we cannot use 1 subroutine calls 'cause we do r subroutine calls 'cause we do r register until the RAM is found ar as a by a by a by a by a by a by a b</pre>	<pre>; Look into every slot from OFFF] ; area. Note that we cannot use I ; subroutine calls 'cause we do r ; RAM exits. Everything has to be ; register until the RAM is found ; register until the RAM is found] AB AB AB AB AB D3 AB XOR A A AB D3 AB XOR A OUT (PPI.CM),A AF D3 A8 LD A,PI.CM),A AF OUT (PPI.CM),A AF OUT (PPI.CM),A AP D3 A8 LD A,PI.CM),A AF OUT (PPI.CM),A AF OUT (PPI.CM),A AP D3 A8 LD A,PI.CM),A</pre>	<pre> Look into every slot from OFFFI area. Note that we cannot use I subroutine calls 'cause we do I subroutine calls 'ca</pre>	3E 82 AF AF AF	3E 82 D3 AB	3E 82	 Look into every slot from OFFFH to C000H, and set system work Look into every slot from 0FFFH to C000H, and set system work area. Note that we cannot use RAM as work area nor perform subroutine calls 'cause we do not yet know where the available RAM exits. Everything has to be done inside ROM and CPU's register until the RAM is found. 	; Jook into every slot from OFFFH to C000H, and set system work ; area. Note that we cannot use RAM as work area nor perform ; subroutine calls 'cause we do not yet know where the available ; RAM exits. Everything has to be done inside ROM and CPU's	; ; Look into every slot from OFFFH to C000H, and set system work ; area. Note that we cannot use RAM as work area nor perform	; ; Look into every slot from OFFFFH to C000H, and set system work			CHKRAM			
<pre>Icook into every slot from OFFF subroutine calls 'cause we do r subroutine calls 'cause we do r subroutine calls 'cause we do r RAM exits. Everything has to be register until the RAM is found as as b3 AB b3 AB b4 b5 b1 b1 b1 b1 b1 b1 b1 b1 b1 b1 b1 b1 b1</pre>	<pre>: Look into every slot from OFFF ; area. Note that we cannot use 1 ; subroutine calls 'cause we do 1 ; RAM exits. Everything has to be ; register until the RAM is found ; register until the RAM is found 3E 82 0UT (PPI.CM),A AF D3 AB D3 AB D3</pre>	<pre>I Look into every slot from OFFF area. Note that we cannot use I subroutine calls 'cause we do r subroutine calls 'cause we do r s register until the RAM is found register until the RAM is found aF D3 AB D3 AB D3 AB D3 AB D3 AB D3 AB D3 AB D3 AB D3 AB D3 AB CUT (PPI.CM),A CUT (PPI.CW),A CUT (PPI.CW),A CUT (PPI.CW),A CUT (PPI.CW),A CUT (PPI.CW),A Start searching B - non 0 if we're now checkin SPH - slot # of the biggest RAN</pre>	<pre> Took into every slot from 0FFF area. Note that we cannot use 1 subroutine calls 'cause we do r subroutine calls 'cause we do reckin subroutine calls 'cause we cannot subroutine calls 'cause we do reckin subroutine calls 'cause we cannot subroutine</pre>	<pre>3E 82 Look into every slot from 0FFF ; area. Note that we cannot use 1 ; subroutine calls 'cause we do r ; RAM exits. Everything has to be ; register until the RAM is found ; register until the RAM is found 3E 82 OUT (PPI.CM),A AF XOR A D3 AB XOR A D3 AB CUT (PPI.CM),A AF 0UT (PPI.CM),A 3E 50 OUT (PPI.CM),A 5 5 5 5 5 7 7 8 7 8 7 8 7 8 7 8 7 8 7 8</pre>	<pre> The second se</pre>	3E 82 3E 82 32 82 33 82 34 82 35 90 36 00T 37 82 38 82 39 82 31 82 32 90 33 82 93 82 93 82 93 82 93 84 93 85 93 84 93 84 93 84 93 84 93 84 93 84 93 93 93 94 93 94 93 94 93 94 93 94 93 94 94 94 95 94 94 94 95 94 94 94 95 94 <td><pre>3E 82 Control con</pre></td> <td>3E 82 AF D3 AB 2</td> <td>л л л л л л л л л л 3Е 82 D3 AB</td> <td>3E 82</td> <td> Look into every slot from OFFFH to C000H, and set system work Look into every slot from OFFFH to C000H, and set system work area. Note that we cannot use RAM as work area nor perform subroutine calls 'cause we do not yet know where the available RAM exits. Everything has to be done inside ROM and CPU's register until the RAM is found. </td> <td><pre> provide the set of the set</pre></td> <td><pre>;</pre></td> <td>; ;</td> <td>; </td> <td></td> <td></td> <td></td> <td>CHKRAM:</td> <td>02D7</td>	<pre>3E 82 Control con</pre>	3E 82 AF D3 AB 2	л л л л л л л л л л 3Е 82 D3 AB	3E 82	 Look into every slot from OFFFH to C000H, and set system work Look into every slot from OFFFH to C000H, and set system work area. Note that we cannot use RAM as work area nor perform subroutine calls 'cause we do not yet know where the available RAM exits. Everything has to be done inside ROM and CPU's register until the RAM is found. 	<pre> provide the set of the set</pre>	<pre>;</pre>	; ;	; 				CHKRAM:	02D7

PAGE 16-1 29		<pre>;Initialize lowest address ever found ;Start from slot #0 .Clear hit nation</pre>	CTEAT DIT PATCETI	;Shift bit mattern	Assume this slot is not expanded	Read from possible expansion slot register			Read back as 00001111?	;Nop, this is not an expanded slot		:Read back as 1111112	:Non. not expanded slot	:We're checking exnanded slot	:Sav this slot is expanded					Select the expanded slot		Start checking from OBFOOH to ROOOH			
01-Jan-85	slot #1 expanded slot #0 expanded	DE,OFFFH A C.A	x (127x TCC)	C C	B,0	HI, OFFFH	(HT),0F0H	A, (HL)	OFH WF CUTTE	UZ,CKRMI5	A. (HL)	A	NZ, CKRMI 5	В	0,0			Start from expansion slot #0		(OFFFH),A		HL, OBFOOH		A, (HL)	
3.44	+ + + + + + + + + + + + + + + + + + +	LD XOR LD		SLA	LD	LD	ΓD	LD	sub t	л. Г.Т	ED E	INC	ЯĻ	INC	\mathbf{SET}			t from es		ГD	:	LD		ГD	CPL
80			CKRM05													CKRM10:	••	;Start	•••		CKRM15:		CKRM20:		
BIOS) Macro-80 available RAM		11 FFFF AF 4F	D3 A8		00 00		36 FO		20 OF	20 UB	7E	3C	20 06	04	CB C1					32 FFFF		21 BF00		7E	2F
1 BASIC - Find		02E2 02E5 02E6	02E7 02E7	02E9	02 EB	02ED	02F0	0262	0285	02F7	02F8	02F9	02FA	02FC	02FD	02FF				02FF	0302	0302	0305	0305	0306
(MSX ROM - MSXIO -	711 712 713	714 715 716	717 718	719	720	721	27/	221 VCL	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741

PAGE 16-2			;RAM not equipped in this page	;Make sure it's not a coincidence	;Check more		;Check next page				;Below the one ever found						;Set possible secondary slot #			;Set primary slot #		;Register these slot #'s			;Are we checking secondary slot	; No			; Prepare to select next secondary slot
01-Jan-85	(HL),A (HL)	(HL),A	NZ , CKRM2 5	Г	NZ, CKRM20	Н	M, CKRM20		г,0	Н	A,L	ы	А,Н	A,D	NC, CKRM30	DE, HL	A, (OFFFFH)		L,A	A,(PPI.AR)	Н,А	SP,HL		А,В	А	Z,CKRM35	A,(0FFFFH)		A,10H
3.44	LD CP	LD	JR	INC	JR	DEC	JP	5.	ГD	INC	LD	SUB	LD	SBC	JR	EX	LD	CPL	LD	IN	ГD	LD		ГD	AND	JR	LD	CPL	ADD
BIOS) Macro-80 available RAM	77 BE 35	27 77	20 07	2C	20 F5	25	FA 0305	CKRM25	2E 00	24	7D	93	7C	9A	30 0A	EB	3A FFFF	2F	6F	DB A8	67	F9	CKRM30:	78	A7	28 0A	3A FFFF	2F	C6 10
BASIC Find	0307 0308	030A	030B	030D	030E	0310	0311	0314	0314	0316	0317	0318	0319	031 A	031B	031D	031E	0321	0322	0323	0325	0326	0327	0327	0328	0329	032B	032E	032F
MOX ROM - MSXIO -	742 743	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772

PAGE 16-3		;Continue if more secondary slots remain			; Prepare to select next slot	;Continue if more primary slots remain
01-Jan-85	0100000B	C,CKRM10		A,(PPI.AR)	A,0101000B	NC, CKRM05
3.44	CP	JR	CKRM35:	NI	ADD	JR
BASIC BIOS) Macro-80 Find available RAM	FE 40	38 CA	U		C6 50	
	0331	0333	0335	0335	0337	0339
(MSX ROM - MSXIO -	773	774	775	776	<i><i>۲<i>TT</i></i></i>	778

PAGE 17			agest one					;Set primary slot register		;Set possible secondary slot register										; Initialize lowest address ever found	<pre>;Start from slot #0</pre>			;Select the slot	;Assume this slot is not expanded	;Shift bit pattern	;This slot is not expanded	;We're checking expanded slot			
01-Jan-85			done select the blagest one		HL,0	HL, SP	А,Н	(PPI.AW),A	A,L	(OFFFFH),A		OC000HOFFFH		A,C					C,A	DE, OFFFH	A,(PPI.AR)	00111111B		(PPI.AW),A	B,0	IJ	NC, CKRM60	В	A, (OFFFH)		00111111B
80 3.44		•••	. Check is don		LD	ADD	LD	OUT	LD	LD	••	; Next, check	••	LD	RLCA	RLCA	RLCA	RLCA	LD	LD	IN	AND	CKRM50:	OUT	LD	RLC	JR	INC	LD	CPL	AND
BIOS) Macro-80 available RAM					21 0000	39	7C	D3 A8	7D	32 FFFF				79	07	07	07	07	4F	11 FFFF		E6 3F				CB 01		04	3A FFFF	2F	E6 3F
BASIC Find					33	033E	33	34	34	34				0346	\sim	\sim	\sim	\sim	034B	\sim	\sim	\sim	\sim	\sim	\sim	\sim	\sim	\sim	\sim	\sim	\sim
(MSX ROM - MSXIO -	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809

PAGE 17-1	:Select the expanded slot		;Start checking from 0FE00H to 0C000H							;RAM not equipped in this page	; Make sure it's not a coincidence	;Check more				;Check next page	14			; Below the one ever found				; No	;Register this address as the lowest	secondary	- - -		;Set primary slot #
01-Jan-85	(OFFFH),A		HL , 0FE00H	A.(HI.)		(HL) , A	(HT)		(HL),A	NZ, CKRM70	Г	NZ, CKRM65	Н	А,Н	0C0H	NC , CKRM6 5		г,0	Н	A,L	ы	А,Н	Α, D	NC, CKRM75	DE, HL	A, (OFFFFH)		L,A	A, (PPI.AR)
-80 3.44	CKRM55: LD	CKRM60:	CKRM65.	LD LD	CPL	LD	CP	CPL	LD	JR	INC	JR	DEC	LD	CP	JR	CKRM70:	LD	INC	ГD	SUB	LD	SBC	JR	EX	ΓD	CPL	ΓD	NI
BIOS) Macro-80 available RAM	32 FFFF		21 FE00	7E	2F	77	BE	2F	77	20 09	2C	20 F5	25	7C	FE CO	30 EF		2E 00	24	7D	93	7C	9 A	30 0A	EB	3A FFFF	2F	6F	DB A8
M BASIC - Find	0362 0362	0365	0365 0368	0368	0369	036A	036B	036C	036D	036E	0370	0371	0373	0374	0375	0377	0379	0379	037B	037C	037D	037E	037F	0380	0382	0383	α	0387	8
(MSX ROM - MSXIO -	810 811	812	813 814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840

PAGE 17-2		;Register these slot #'s			;Are we checking secondary slot	; No			; Prepare to select next secondary slot	;Continue if more secondary slots remain			; Prepare to select next slot	;Continue if more primary slots remain	ite setup
01-Jan-85	Н,А	SP, HL		A, B	A	Z,CKRM80	A, (OFFFFH)		A,0100000B	NC, CKRM55		A, (PPI.AR)	A,0100000B	NC, CKRM50	SUBTTL - MSXIO - Slot attribute setup
3.44	ГD	ГD	175:	ГD	AND	JR	LD	CPL	ADD	JR	480:	IN	ADD	JR	ULL - MSXIO
MSX ROM BASIC BIOS) Macro-80 MSXIO - Find available RAM	038A 67	038B F9	3 8C CKRM7 5				0390 3A FFFF			30		DB	39A C640		SUBT
(MSX ROM BA - MSXIO - F	841 03						847 03					852 03		854 03	855
\sim 1															

-85 PAGE 18		; Check is done, select the biggest one					1),A ;Set primary slot register		1),A ;Set possible secondary slot register		a	ZETO		H ;length of work area		OW ;beginning of work	; init first byte	transfer it to rest of area				;Get 'slot-expanded' flag	;Loop 4 times			;Set carry if LSB is set	;[Acc]=255 if expanded, 0 if not expanded		;Set table for each slot	
01-Jan-85		, select		HL,0	HL, SP	А,Н	(PPI.AW),A	A,L	(OFFFFH),A	A,C		area with zero		BC,0C49H	DE, RAMLOW+1	HL , RAMLOW	(HT)'0					C,A	B,4	HL, EXPTBL+3		с U	A,A	80H	(HL),A	Ħ
3.44		Check is done		LD	ADD	LD	OUT	LD	LD	LD		Clear work are		LD	LD	LD	LD	LDIR		Set EXPTBL		ГD	ГD	ГD	SSLTLP:	RR	SBC	AND	LD	DEC
BIOS) Macro-80 attribute setup	•			21 0000	39	7C	D3 A8	7D	32 FFFF	79		••	••	01 0C49	11 F381	21 F380	36 00	ED BO	••		~	4F	06 04	21 FCC4	SS	CB 19	9F	E6 80	77	2B
MSX ROM BASIC MSXIO - Slot				039E	03A1	03A2	03A3	03A5	03A6	03A9				03AA	03AD	03BO	03B3	03B5				03B7	03B8	03BA	03BD	03BD	03BF	03C0	03C2	03C3
(MSX ROM - MSXIO -	856 857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	æ	886

PAGE 18-2	;IM 1 processing
3 .44 01-Jan-85	IM 1 ;IM 1 JP INIT SUBTTL - MSXIO - Control-[C] processing
3.44	IM JP - MSXIO
MSX ROM BASIC BIOS) Macro-80 • MSXIO – Slot attribute setup	ED 56 C3 2680
om Bas: 0 - S]	03F6 03F8
(MSX R(MSXI(918 919 920

PAGE 19		; Is BASIC text in ROM		; Yes		;Seen any interesting key							; No	;Is it ctrl-stop?	; Yes, execution aborted		pressed		;STOP pressed (pause)			;Display cursor if disabled	;Wait for next interesting key					; Wait for character if SELECT pressed	; Seen?	;Not yet
01-Jan-85		A, (BASROM)	A	NZ	НĹ	HL , INTFLG		A, (HL)	(HT) , 0	НĹ		A	2	с	Z, EXCABO		next STOP is pr		HL	DE	BC	CKDPC0	HL , INTFLG			A, (HL)	(HT) , 0		А	Z,WATINT
3.44 3.44 ing	I SCNTC:	LD	AND	RET	HSUG	ГD	DI	ΓD	LD	POP	EI	AND	RET	CP	JR	••	; Pause until	••	PUSH	HSUG	PUSH	CALL	ГD	WATINT:	DI	ΓD	ГD	EI	AND	JR
BIOS) Macro-80 ol-[C] processing		3A FBB1	A7	C0	ES	21 FC9B	F3	7E	36 00	El	FB	A7	C8	FE 03	28 1C				E5	D5	C5	CD 09DA			F3	7E	36 00	FB	A7	28 F8
BASIC BIOS Control-[03FB	03FB	03FE	03FF	0400	0401	0404	0405	0406	0408	0409	040A	040B	040C	040E				0410	0411	0412	0413	0416	0419	0419	041A	041B	041D	041E	041F
(MSX ROM BASIC BIOS - MSXIO - Control-[921 922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951

39		1000		e t
PAGE 19-1	;Erase cursor if disabled ;Abort?	<pre>;No ;Save text pointer ;Save text pointer ;Cancel any input ;Is STOP trap ON ;No, accept this break ;Request STOP trap .cinco not chance intervent .cinco not chance</pre>	; this must be enclosed by 'DI' and 'EI' ; Restore text pointer ; Make sure we're in text mode ; Make sure BASIC is enabled	;Restore text pointer ;Must return with carry cleared, zero set ;LSPD
01-Jan-85	AF CKERCO AF BC DE HL 3	NZ HL KILBUF CKSTTP NC,EXABOL HL,REQSTP	REQTRP HL TOTEXT A, (EXPTBL) H.010000B	ENASLT HL A SP,(SAVSTK) BC STOP
3.44	PUSH CALL POP POP POP CP CP	RET PUSH CALL JR LD LD		CALL POP XOR LD PUSH JP
0 Dd		EXCABO:	EXABOL:	
Macro-80 Control-[C] processing	F5 CD 0A27 F1 C1 D1 E1 FE 03	C0 E5 CD 0468 CD 0454 30 0A 21 FC6A 53	r 5 CD 0EF1 FB E1 C9 C9 C0 083B 3A FCC1 26 40	
щ	0421 0425 0425 0426 0428 0428 0428	042B 042C 042C 042D 0430 0433 0433	1 m m m m m m m m m m m m m m m m m m m	0447 0448 0448 0446 044C 0450 0451
(MSX ROM - MSXIO -	9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9	959 961 964 963 965 965	900 967 971 973 974 973	976 977 979 980 981

PAGE 19-2	;Is STOP trap ON ;No, accept this break ;Is STOP trap specified ;No, accept this break ;No, accept this break ;No, accept the break ;Xee we in direct mode ;Yes, treat as break ;Yes treat as break ;Set flag to indicate STOP trap active ;Empties ring buffer	
01-Jan-85	CKSTTP: Check for STOP trap Check for STOP trap LD A, (REQSTP) RRCA RET NC LD HL, (REQSTP+1) LD A, H OR L RET Z LD HL, (CURLIN) INC HL LD A, H OR L RET Z LD A, H CURLIN) INC HL LD A, H CURLIN A, H CURLIN A, H CURLIN A, H CURLINA A, H C	1 (1117196)
3.44	for Since Si	RET
-80 sing	CKSTTP: ; Check ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	
ASIC BIOS) Macro-80 Control-[C] processing		22 F3FA C9
I BASIC I Contro	0454 0454 04557 04559 04557 04555 04664 04665 04665 04665 04665 04665 04665 04665 04665 04665 04665 04665 04665 04665 04665 04665 04665 04665 04665 04665 04655 04655 04655 04655 04655 04655 04655 04655 04655 04555 00000000	046E
(MSX ROM BASIC BIOS) - MSXIO - Control-[C]	988 988 986 988 998 999 999 1000 1000 1000 1000 100	900T

PAGE 20	f pressed, return with carry set.	;Leave others unaffected ;Select 6th row	;STOP key is assigned to bit 4 ;O when pressed	;Cancel any input	;STOP pressed, mark as pressed to prevent ; to be doubly recognized zation
01-Jan-85	stop key pressed. If	A,(PPI.CR) 0F0H 7 (PPI.CW),A	A,(PPI.BR) 10H NZ A,(PPI.CR)	A, (PPI.CW), A A, (PPI.BR) 2 NZ HL HL, (PUTPNT) (GETPNT), HL	A, (OLDKEY+7) ;STO 0EFH ;to (OLDKEY+7),A ;to A,0DH (REPCNT),A - PSG Initialization
3.44	BREAKX: ; ; Check if stop	IN AND OR OUT	IN AND RET IN	OUT IN AND RET PUSH LD LD POP	LD AND LD LD LD SCF RET SUBTTL - MSXIO
ASIC BIOS) Macro-80 Control-[C] processing	ш .: к .	DB AA E6 F0 D3 AA D3 AA		31 D3 AA DB A9 E6 02 E5 2A F3F8 2A F3F8 E1 E1	3A FBE1 E6 EF 32 FBE1 3E 0D 32 F3F7 C9 C9
₽ ₩	046F	046F 0471 0473 0473	0477 0479 0479 047B 047C	04/E 047F 047F 0481 0483 0485 0488 048A 048D	048E 0491 0493 0496 0498 049B 049B
(MSX R - MSXIO	1007 1008 1009 1010	1012 1012 1014 1014	1016 1017 1018 1018	1020 1021 1022 1023 1023 1025 1025 1026 1028	1029 1030 1031 1031 1032 1033 1035 1035

PAGE 21	<pre>;Set Port A to input mode ;Port B to output mode ;Port B to output mode ;Dummy write cycle to wake up the PSG ;envelope register ;Any value is OK! ;Any value is OK! disabled ueues, and static data. disabled ved. ;save caller's registers</pre>	
01-Jan-85	ITIO: Initialize I O LD A,7 LD A,7 LD A,7 LD A,0FH CALL WRTPSG ;Set LD A,0FH CALL WRTPSG ;Port LD A,0BH ;Port LD A,0BH ;Port CALL WRTPSG ;Any CALL WRTPSG ;Any CALL WRTPSG ;Any CALL WRTPSG ;Any CALL WRTPSG ;Any LD E,0CFH CALL WRTPSG ;Any CALL WRTPSG ;Any CALL WRTPSG ;Any CALL WRTPSG ;Any LD A,0FH 01000000B LD A,0FH OUT (LPT.SB),A CINI: Initialize GI sound chip, queues, al Entry - Interrupts must be disabled Exit - All registers preserved. PUSH BC PUSH BC PUSH AF	
3.44	INITIO: i Initialize I (i Initialize I (LD LD LD LD LD LD LD LD LD LD	
ASIC BIOS) Macro-80 PSG Initialization	3E 07 1E 80 CD 1102 3E 0F 1E CF CD 1102 3E 0B 5F CD 1102 CD 1102 CD 1102 CD 1102 E6 40 32 FCAD 33 FF D3 90 FF D5 FF	
щ	049D 049D 049F 049F 04A4 04A6 04A6 04A8 04AB 04B1 04B1 04B5 04B5 04B5 04B5 04BD 04BD 04BD 04BD 04BD 04BD 04BD 04BD	
(°MSX ROM - MSXIO -	1037 1039 1040 1041 1042 1044 1044 1044 1051 1052 1053 1053 1053 1053 1053 1053 1053 1053	

PAGE 21-1	E Contraction of the second		:=VCBC + VCBSTZ + MISCIF							eue	. Addrocc of much		; Mask pattern, /r = Music queue len - l	;Queue length		;Save length of queue	; Save address of queue	;Save mask pattern	; Save queue ID	Initialize a guene by [Acc] [R] [DF]		write to reds 8.9.10		;0 out amplitude (turn voice off)	Restore [Acc]	Save queue ID	; OctaveX	:[HL] points to octave for voice [A]		;[HL] points to default value table
01-Jan-85	all static data	HL, MUSICF	B,71H	A		(HL),A	HL	MUSCLL		clear music dynamic queue		D TUDION	D,/FII	ыь , б UR	1	НГ	DE	BC	AF	DTINI	AF	A,8	E,0	WRTPSG	8	AF	L,0FH	GETVC1	DE, HL	HL, MUSITB
3.44	First, clear	ГD	LD	XOR	MUSCLL:	LD	INC	ZNLC		Then clear m	U.1					HCUA	HSUG	PUSH	HSUG	CALL	POP	ADD	LD	CALL	SUB	HSUG	LD	CALL	EX	LD
: BIOS) Macro-80 Initialization	£	21 FB3F	06 71	AF	MUS	77	23	10 FC		H	11 F975	7 12		2000	010 BE	0 J J	ça	C5	F5	CD 14DA	Fl	C6 08	1E 00	CD 1102	D6 08	F5	2E 0F	CD 1477	EB	21 0508
DM BASIC - PSG		04C1	04C4	04C6	04C7	04C7	04C8	04C9			04CB	04CE	0400	0403			04D4	04D5	04D6	04D7	04DA	04DB	04DD	04 DF	04E2	04E4	04E5	04E7	04EA	04EB
(MSX ROM - MSXIO -	1068 1069	1070	1071	1072	1073	1074	1075	1076	1077	1079 1079	1080	1081	1082	1083	1084		COL	1086	1087	1088	1089	1090	1001	1092	1093	1094	1095	1096	1097	1098

PAGE 21-2	;EMSITB - MUSITB ;default variables for this voice	Restore gueue ID	;Restore mask	Restore gueue address	;Restore queue length	;Update queue address		;Next channel		;Loop till done all three voices		; input port A, output port B	;disable noise, enable all 3 tones	;Restore environments			for music variables	idefault octave	.default note length		;default tempo	;default volume	;default envelope period		; end of music table	Utility routines for VDP	
01-Jan-85	BC,6	AF	BC	HL	DE	HL, DE	DE,HL	А	ε	C,GICIN1	A,7	E,0B8H	WRTPSG	POPALL			; table of default values f	ОДН		0410	78H	88H	OFFH	H00			
3.44	LD LDIR	POP	POP	POP	POP	ADD	EX	INC	CP	JR	LD	ГD	CALL	ЧĻ	[TB:		able of def	au		ПВ	DB	DB	DB	DB	EMSITB:	– OIXSW – TLIJ	
BASIC BIOS) Macro-80 PSG Initialization	04 EE 01 0006 04 F1 F1 B0	F1				04F7 19			ŦΕ	04FC 38 D5	ЗE				0508 MUSITB:	••	; ta						050C FF			SUBTTL	
(MSX ROM - MSXIO -	1099 001	1101	1102	1103	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119	1120	1121	1122	1123	1124	1125	

PAGE 22	(40 by 24)									;Set border/foreground/background color		;Initialize character pattern	;Actually set VDP registers	1			(graphics 1)							
01-Jan-85	INITXT: ; i Initialize VDP for text mode (40 by	DISSCR A	(SCRMOD), A	(ULDSCK), A A, (LINL40)	(TINLEN), A	HL, (TXTNAM)	(NAMBAS), HL	HL, (TXTCGP)	(CGPBAS), HL	CHGCLR	CLRTXT	INIPAT	SETTXT	ENASCR			Initialize VDP for text mode (graphics 1)	DISCO	A.1	(SCRMOD) . A	(OLDSCR), A	A, (LINL32)	(LINLEN), A	HL, (T32NAM)
80 3.44 r VDP	INITXT: ; ; Initialize ;	CALL XOR	LD	LD LD	LD	LD	ΓD	ГD	ГD	CALL	CALL	CALL	CALL	JR	INIT32:	•	; initialize V	CALT.	ID	LD	ΓD	LD	LD	LD
BASIC BIOS) Macro-80 Utility routines for VDP		CD 0577 AF		32 FCBU 3A F3AE		F3B			22 F924			07	0	18 38				CD 0577		32 FCAF	32 FCB0	3A F3AF	32 F3B0	2A F3BD
щ	050E	050E 0511	0512	0518	051B	051E	0521	0524	0527	052A	052D	0530	0533	0536	0538			0538	053B	053D	0540	0543	0546	0549
(MSX ROM - MSXIO -	1126 1127 1128 1129 1130	1131 1132	1133	1135 1135	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1149 1149	1150	1151	1152	1153		1155	

PAGE 22-1							;Set border foreground background color		;Initialize character pattern	;Clear sprites	;Actually set VDP registers																
01-Jan-85	(NAMBAS), HL HL, (T32CGP)	(CGPBAS), HL	HL, (T32PAT)	(PATBAS), HL	HL, (T32ATR)	(ATRBAS), HL	CHGCLR	CLRTXT	INIPAT	ERASPR	SETT32			n display		A, (RG1SAV)	0100000B	DISSCI			en display		A, (RG1SAV)	0BFH		B,A	c,1
-80 3.44 or VDP	LD	LD	LD	LD	LD	LD	CALL	CALL	CALL	CALL	CALL	ENASCR:	••	; Enable screen display		ГD	OR	JR	DI SSCR:	••	; Disable screen display	••	ΓD	AND	DISSC1:	ΓD	LD
BASIC BIOS) Macro-80 Utility routines for VDP	22 F922 2A F3C1		2A F3C5	22 F926	2A F3C3	22 F928	CD 07F7	CD 077E	CD 071E	CD 06BB	CD 05B4					3A F3E0	4	18 05					3A F3E0	E6 BF		47	0E 01
	054C 054F	0552	0555	0558	055B	055E	0561	0564	0567	056A	056D	0270				0270	0573	0575	0577				0577	057A	057C	057C	057D
(MSX ROM - MSXIO -	1157 1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183	1184

PAGE 23	<pre>register is properly set ;Get data to set ;Get register # ;Get register # ;Remember this value 'cause this is ;a write-only register v 32) v 32) </pre>	
3.44 01-Jan-85 P	WRTVDP: Write data to VDP C = register # C = register # Register save area for the regis Register save area for the regis PUDP.CW),A LD A,B DI OUT (VDP.CW),A C = register # PUSH HL OUT (VDP.CW),A EI PUSH HL DD OUT (VDP.CW),A EI PUSH HL DD OUT (VDP.CW),A EI RET SETTXT: Set VDP for text mode (40 by 32) ADD ADD ADD CD ADD C (HL),A POP C (HL),A C (HL),A	
MSX ROM BASIC BIOS) Macro-80 MSXIO - Utility routines for VDP	057F 057F 057F 0580 0581 0583 0583 0584 0583 0584 0583 0584 0583 0584 0583 0584 0583 0584 0588 0588 0588 0592 0591 77 0592 0591 77 0592 0594 0592 0594 0592 0594 0592 0592 0592 0592 0592 0592 0592 0592	
- MSX ROM - MSXIO -	1185 1186 1187 1188 1188 1189 1199 1199 1200 1200 1200 1200 1200 120)

GE 23-1	;Set register #1	;Set mask pattern ;Set screen mode s 1) ;Set register #0	;Set register #1	;Set screen mode
PAGE	Ŭ 	; Sc ; Sc ; Sc ; Sc	х х	х х х
01-Jan-85	B,A C,O WRTVDP A,(RGISAV) 0E7H 10H B,A B,A WRTVDP HL.TXTNAM	DE;0 ;Set SETSCM ;Set sets mode (graphics 1) A,(RG0SAV) ;Set 1 B,A C,0 WRTVDP	A, (RG1SAV) 0E7H B,A C WRTVDP HL,T32NAM	DE, 0 SETSCM
3.44 DP	LD LD CALL CALL OR CALL LD CALL	ы	LD LD LD CALL CALL	JP JP INIGRP:
BASIC BIOS) Macro-80 Utility routines for VDP	47 0E 00 CD 057F 3A F3E0 E6 E7 F6 10 47 CD 057F 21 F3B3	0000 0677 F3DF 01 00 057F		C3 0677
_	0599 0597 0597 0587 0582 0584 0588 0588 0588	0584 0581 0584 0584 0587 0587 0588	05BF 05C2 05C4 05C5 05C6 05C6	05D2
(MSX ROM - MSXIO -	1216 1217 1218 1219 1220 1222 1223 1223 1223		1237 1238 1239 1240 1241 1242	1245 1244 1246 1246

PAGE 23-2	mode								;Initialize name table									;Clear pattern and color table		; Actually set VDP mode				graphics 2)	•	;Set register #0				
01-Jan-85	Initialize VDP for graphics mode	DISSCR	A,2	(SCRMOD), A	HL , (GRPPAT)	(PATBAS), HL	HL, (GRPATR)	(ATRBAS), HL	HL, (GRPNAM)	SETWRT	A	B,3		(VDP.DRW),A	A	NZ, INIGRI	INIGRI	CLSHRS	ERASPR	SETGRP	ENASCR			graphics mode (graphics 2)	1	A, (RG0SAV)	2	B,A	c,0	WRTVDP
) 3.44 VDP	; Initialize V :	CALL	ĽD	ΓD	LD	LD	ГD	ГD	LD	CALL	XOR	LD	INIGR1:	OUT	INC	ЯĽ	ZNLC	CALL	CALL	CALL	ЧГ	SETGRP:	••	; Set VDP for	••	LD	OR	LD	LD	CALL
ASIC BIOS) Macro-80 Utility routines for VDP		CD 0577	3E 02	32 FCAF	2A F3CF	22 F926	2A F3CD	22 F928	2A F3C7	CD 07DF	AF	06 03		D3 98	3C	20 FB	10 F9	CD 07A1	CD 06BB		C3 0570					3A F3DF	F6 02	47	0E 00	CD 057F
щ		05D2	05D5	05D7	05DA	05DD	05E0	05E3	05E6	05E9	05EC	05ED	OSEF	05EF	05F1	05F2	05F4	05F6	05F9	05FC	05FF	0602				0602	0605	0607	0608	060A
(MSX ROM - MSXIO -	1247 1248	4	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277

ŝ

PAGE 23-3	;Set register #1		or mode	;Initialize name table
01-Jan-85	A,(RGISAV) 0E7H B,A	C WRTVDP HL,GRPNAM DE,7F03H SETSCM	Initialize VDP for multi-color mode CALL DISSCR LD A,3 LD (SCRMOD),A LD (SCRMOD),A LD HL,(MLTPAT) LD HL,(MLTATR) LD HL,(MLTATR) LD (ATRBAS),HL	HL, (MLTNAM) SETWRT DE,6 C,4 A,D B,1 B,1 (VDP.DRW),A A INIML3
-80 3.44 Dr VDP	LD AND LD	INC CALL LD LD LD JR JR INIMLT:	; Initialize V ; CALL LD LD LD LD LD LD LD LD LD	LD CALL CALL LD LD INIML1: LD LD LD LD LD LD LD LD INIML3: OUT INC DJNZ
BASIC BIOS) Macro-80 Utility routines for VDP	3A F3E0 E6 E7 47	0C CD 057F 21 F3C7 11 7F03 18 58	CD 0577 3E 0577 3Z FCAF 2A F3D9 2A F3D7 22 F926 23 F3D7	
_	060D 0610 0612	0613 0614 0617 061A 061D 061F	061F 0622 0624 0627 062A 062D 0630	0633 0636 0636 0637 0637 0637 0637 0637
(MSX ROM - MSXIO -	1278 1279 1280	1281 1282 1283 1284 1285 1285	1288 1289 1291 1291 1293 1295 1295	1297 1298 1298 1300 1301 1303 1303 1305 1306 1308

PAGE 23-4		;Clear pattern table ;Actually set VDP mode		;Set register #0	;Set register #1	;Set mask pattern ;Set name table	
01-Jan-85	C NZ,INIML2 D,A E NZ,INIML1	CL SMLT ERASPR SETMLT	JP ENASCR TMLT: Set VDP for multicolor mode	A, (RG0SAV) 1 B,A C,0 WRTVDP	A, (RGISAV) 0E7H 8 B, A C, 1 WRTVDP HL, MLTNAM	DE,0 BC,SETGRP SETREG	В, ОАН А, D
-80 3.44 Dr VDP	DEC JR DEC JR	CALL CALL CALL	JP SETMLT: ; Set VDP for :	LD AND LD LD CALL	LD AND OR CR LD LD LD	LD SETSCM: LD CALL	LD LJ
BASIC BIOS) Macro-80 Utility routines for VDP	0D 20 F5 1D 20 EF	CD 07B9 CD 06BB CD 0659	c3 0570	3A F3DF E6 01 47 0E 00 CD 057F	3A F3E0 E6 E7 F6 08 47 0E 01 CD 057F 21 F3D1	11 0000 01 0602 CD 0690	06 0A 7A
щ	0646 0647 0649 0648 0648	064D 0650 0653	0656 0659	0659 065C 065E 065F 065F	0664 0667 0669 066B 066B 066E 066E	0674 0677 0677 0677 067A	067D 067F
(MSX ROM - MSXIO -	1309 1310 1311 1312 1313	1314 1315 1316	1317 1318 1319 1320	1322 1323 1324 1325 1326	1327 1328 1329 1330 1331 1332 1333	1334 1335 1336 1337	1338 1339

	le	able	sprite attribute table	ttern table																								
23-5	;Set color table	;Set pattern table	prite at	sprite pa																								
PAGE	;Set o	;Set F	;Set s	;Set s																								
01-Jan-85	SETRG1 B,5 A,E	SETRG1 B,9	SETREG	B,5		A		Н	AF	A, (HL)	HL	Н, (НL)	L,A	A		нг, нг	A, A	SETRG2	L,A	AF	ц	B,A	WRTVDP	HL	用	ΗΓ	U	
3.44	CALL LD LD	CALL LD	CALL	LD		XOR		PUSH	HSUG	LD	INC	LD	LD	XOR		ADD	ADC	ZNLC	LD	POP	OR	ГD	CALL	POP	INC	INC	INC	RET
ASIC BIOS) Macro-80 Utility routines for VDP					SETREG:		SETRG1:								SETRG2:													
) Mac outines	0691 05	0691 09	0690	05		_							_	_			_	FC	_				057F					_
BIOS ity r	CD 06 7B	06 06	CD	90		AF		ES	F5	7 E	23	99	6F	AF		29	8F	10	6F	Fl	B5	47	8	El	23	23	00	60 0
щ	0680 0683 0685	0686 0689	068B	068E	0690	0690	0691	0691	0692	0693	0694	0695	0696	0697	0698	0698	0699	069A	069C	069D	069E	069F	06A0	06A3	06A4	06A5	06A6	06A7
(MSX ROM - MSXIO -	1340 1341 1342	1343 1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370

PAGE 24		;Set register #1	;Clear sprite pattern table ;Length of sprite pattern table	;Load foreground color (default) to [E]	;Set number of sprite plane to [B]	;Erase code (i.e. vertical position) ;Set vertical position	;Load default sprite name	; Prepare for next
01-Jan-85	prites	A, (RGISAV) B, A C, 1 WRTVDP	HL, (PATBAS) BC,0800H A FILVRM	A, (FORCLR) E, A HL, (ATRBAS)	LD BC,2000H CLSPR2: ; default sprite name to [C] ;	A,0D1H WRTVRM HL HL	A, C WRTVRM HL	C A, (RGLSAV)
.0 3 .44 · VDP	CLRSPR: ; ; Clear all sprites ;	LD LD LD CALL	LD LD XOR CALL ERASPR:	U I U I U I U I	LD CLSPR2: ; default spri ;	LD CALL INC INC	LD CALL INC	INC LD RRCA
BASIC BIOS) Macro-80 Utility routines for VDP			2A F926 01 0800 AF CD 0815		01 2000	3E D1 CD 07CD 23 23	79 CD 07CD 23	0C 3A F3E0 0F
MSX ROM BASIC BIOS MSXIO - Utility ro	06A8	06A8 06AB 06AC 06AC 06AE	06B1 06B4 06B7 06B8 06B8	06BB 06BE 06BF	06C2 06C5	06C5 06C7 06CA 06CB	06CC 06CD 06D0	06D1 06D2 06D5
(MSX RC - MSXIO	1371 1372 1373 1374 1375	1376 1377 1378 1378	1380 1381 1382 1383 1383	1385 1386 1387	1388 1389 1390 1391	1392 1393 1394 1395	1396 1397 1398	1399 1400 1401

PAGE 24-1	;16*16? ;No ;Yes, C=C+4	;Load default color	;Assume 8 byte long ;Check size of sprite ;Good assumption ;32 byte long sprite	;Get base address of sprite pattern table ;Form destination/source address ;Get plane number to [L]
01-Jan-85	NC, CLSPR3 C C C	A, E WRTVRM HL CLSPR2 CLSPR2	H, 0 HL, HL HL, HL HL, HL GSPSIZ 8 Z, GSPAD1 HL, HL HL, HL	DE, HL HL, (PATBAS) HL, DE L, A L, A
o-80 3.44 for VDP	RRCA JR INC INC CL.SPR3:	CALPAT: CALL DJNZ RET ; f.D	LD LD ADD ADD ADD CALL CP JR ADD ADD ADD CSPAD1:	EX LD ADD RET CALATR: ; LD
) Macr outines	0F 30 03 0C 0C	7B CD 07CD 23 10 E2 C9 C9	26 00 29 29 29 CD 0704 7E 08 28 02 29 22	EB 2A F926 19 C9 6F
щ	06D6 06D7 06D9 06DA 06DB	0650 0650 0650 0651 0651 0653 0653	0687 0687 0687 0688 0688 0688 0687 0687	06F3 06F7 06F7 06F8 06F9 06F9
(MSX ROM - MSXIO -	1402 1403 1404 1405 1406	1409 1409 1411 1412 1413 1413 1414	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1426 1427 1428 1429 1431 1431

PAGE 24-2	;Sprite attribute consists of 4 bytes ;Load base address ;Calculate target address ;Calculate target address ;Assume 8 byte long ;Assume 8 byte long ;32 byte long sprite
01-Jan-85	H,0 HL,HL HL,HL DE,HL HL,(ATRBAS) HL,DE HL,DE A,R A,R A,R A,8 A,8 A,8 A,8 A,8 A,8 A,8 A,8 A,32 A,32
3.44	LD H ADD H ADD H EX DD H EX DD H LD H ADD H RET RET A RET M RET M RET M
ASIC BIOS) Macro-80 Utility routines for VDP	26 00 29 29 28 F928 19 C9 C9 C9 T T T Cet s ; Cet s ; ; ; Cet s ; ; ; ; ; Cet s ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;
щ	06FA 06FA 06FD 06FE 06FF 06FF 0702 0703 0703 0703 0704 0703 0704 0707 0707
(MSX ROM - MSXIO -	14433 14433 14433 144433 14444 14444 14444 14444 14444 14444 14444 14444 14444 14444 14444 14444 14444 14444 1444 1444 1444 144444 144444 144444 144444 144444 1444444

PAGE 25																			;Get target address of VRAM	;Set VDP for write operation	;Get slot # of character genarator table	;Get address of character genarator table	;Load total length	;Save source slot		;Restore source slot	;Save source slot	;Save counter	
01-Jan-85		SETRD	(SP),HL	(SP),HL		A, (VDP.DRW)	(DE),A	DE	BC	A,C	в	NZ, LDIMVI				Set default character pattern		H.INIP	HL, (CGPBAS)	SETWRT	A, (CGPNT)	HL, (CGPNT+1)	BC,0800H	AF		AF	AF	BC	
o-80 3.44 for VDP	LDIRMV:	; CALL	EX	EX	LDIMV1:	II	ГD	INC	DEC	LD	OR	JR	RET	INIPAT:	••	; Set default	••	CALL	ГD	CALL	ГD	LD	LD	HSUG	INIPT1:	POP	FUSH	FUSH	DI
Macr tines		CD 07FC		E3		DB 98	12	13	0B	79	BO	20 [.] F7	C9					CD FDC7	2A F924	CD 07DF	3A F91F	2A F920	01 0800	F5		Fl	F5	C5	F3
щ	070F	4020	0712	0713	0714	0714	0716	0717	0718	0719	071A	071B	071D	071E				071E	0721	0724	0727	072A	072D	0730	0731	0731	0732	0733	0734
(MSX ROM - MSXIO -	1451 1452	1453 1454	1455	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481

PAGE 25-1	;Read from specified slot	;Restore counter		;Bump character source pointer					;Discard stack																	Get pattern corresponding to ASCII code in [A]		te work area (PATWRK). Entered	to graphic	1	
01-Jan-85	RDSLT	BC	(VDP.DRW),A	HL	BC	A,C	В	NZ, INIPTI	AF				DE, HL	SETWRT		A, (DE)	(VDP.DRW), A	DE	BC	A,C	В	NZ, LDI VML				corresponding to		Pattern is returned to 8 byte work area	by GRPPRT (print a character		
o-80 3.44 for VDP	CALL EI	POP	OUT	INC	DEC	ГD	OR	JR	POP	RET	LDIRVM:	••	EX	CALL	LDI VM1 :	ГD	OUT	INC	DEC	LD	OR	JR	RET	GETPAT:	••	; Get pattern		; Pattern is r	; by GRPPRT (p	••	
[OS) Macr / routines	CD 01B6 FB	cl	D3 98	23	0B	79	BO	20 EF	Fl	60			EB	CD 07DF		IA	D3 98	13	0B	79	BO	20 F7	C9								
BASIC Util	0735 0738	~	\sim	073C	073D	073E	073F	0740	0742	0743	0744		0744	0745	0748	0748	0749	074B	074C	074D	074E	074F	0751	0752							
(MSX ROM - MSXIO -	1482 1483	œ	48	48	48	1488	1489	1490	1491	1492	1493	1494	1495	1496	1497	1498	1499	1500	1501	1502	1503	1504	1505	1506	1507	1508	1509	1510	1511	1512	

PAGE 25-2	troyed	Prepare for calculation							;[HL]:=source address	;Load destination address	;Load total length	;Get slot # of character genarator table		;Save source slot	Save source address	Save destination address	Save counter	Read from specified slot		;Restore counter	;Restore destination address	;Restore source address		;Bump destination pointer	;Bump character source pointer	;Restore source slot				
01-Jan-85 PA	are completely destroyed		L,A	HL, HL	HL, HL	нг, нг	DE, HL	HL, (CGPNT+1)	HL,DE ;[DE, PATWRK ;I	B,8 ;I	A, (CGPNT) ;C		AF ;5	田		BC	RDSLT ;1		BC ;I	DE ;1	HL 11	(DE),A	DE ;	王	AF ;1	GTPAT1			
3.44 DP	; All registers ;	LD	LD	ADD	ADD	ADD	EX	LD	ADD	ГD	ГD	LD	GTPAT1:	HSUG	HSUG	HSUG	PUSH	CALL	EI	POP	POP	POP	LD	INC	INC	POP	ZNĽC	RET	CLSSUB:	
ASIC BIOS) Macro-80 Utility routines for VDP		26 00	6F	29	29	29	EB	2A F920	19	11 FC40	06 08	3A F91F	9	F5	E5	D5	C5	CD 01B6	FB	C1	Dl	El	12	13	23	Fl	10 EF	C9	0	
щ		S	ഫ	0755	0756	0757	0758	0759	075C	075D	0760	0762	0765	0765	0766	0767	0768	0769	076C	076D	076E	076F	0770	0771	0772	0773	0774	0776	7770	
(MSX ROM - MSXIO -	1513 1514	1515	1516	1517	1518	1519	1520	1521	1522	1523	1524	1525	1526	1527	1528	1529	1530	1531	1532	1533	1534	1535	1536	1537	1538	1539	1540	1541	4	1543

PAGE 25-3	;Check current screen mode	;Hires	;Multi-color							;Set address for write	;40 * 24		;32 * 24		;Fill space character code		;Set cursor at home position	; Say all lines are terminated			;Load non 0 value						;Set border color	;Initialize color	; Save this for future use		;Load background color
01-Jan-85	CHKSCR	Z, CLSHRS	NC, CLSMLT			(text mode)		A, (SCRMOD)	A	HL, (NAMBAS)	BC,03C0H	Z,CLRTX1	BC,0300H		A, ' '	FILVRM	CSHOME	HL, LINTTB	B,18H		(HL),B	HL	CLRTX2	FNKSB			CHGBDR	BC,1800H	BC	HL, (GRPCOL)	A, (BAKCLR)
3.44	CALL	JR	JR	: . :		ear screen		LD	AND	ГD	LD	JR	ГD	:1:	ГD	CALL	CALL	ГD	ГD	:2:	ГD	INC	ZNCC	JP	KS:		CALL	LD	HSUG	ГD	ГD
o-80 for VDP				CLRTXT:	••	; Clear								CLRTX1						CLRTX2					CL SHRS:	••					
ASIC BIOS) Macro-80 Utility routines for			30 3B					3A FCAF	A7.		01 03C0	28 03	01 0300		3E 20	CD 0815	_	21 FBB2	06 18		70	23	10 FC	C3 0B26			CD 0832	01 1800	C5	2A F3C9	3A F3EA
щ	0777 2220	0 / / A	077C	077E				077E	0781	0782	0785	0788	078A	078D	078D	078F	0792	0795	0798	079A	079A	079B	079C	079E	07A1		07A1	07A4	07A7	07A8	07AB
(MSX ROM - MSXIO -	1544 1545	54 1	4	54	4	4	S	55	S	55	55	55	55	55	55	55	56	56	56	56	9	9	56	9	9	1569	7		7	~	7

PAGE 25-4	<pre>;Load 6144 ;Set border color ;Set border color ;Set ul pixels to background color ;Set up address for write ;Clear sprites (except sprite pattern)</pre>
01-Jan-85	FILVRM HL, (GRPCGP) BC A FILVRM FILVRM HL, BAKCLR HL, BAKCLR A, A A, A A, A A, A A, A A, A A, A A,
3.44	CALL LD POP YOR YOR JP ADD ADD ADD ADD ADD OR ADD ADD ADD ADD ADD ADD ADD ADD ADD AD
5-80 For VDP	JFLVRM: CLSMLT: ;
(MSX ROM BASIC BIOS) Macro-80 - MSXIO - Utility routines for VDP	CD 0815 2A F3CB Cl AF AF AF AF AF C3 0815 C3 0815 C3 0815 C3 0815 C3 0815 C1 0832 21 F3EA 87 87 87 87 87 87 87 87 2A F3D5 01 0600 18 E9
A BASIC B - Utilit	07AE 07B1 07B4 07B5 07B5 07B6 07B9 07B9 07C1 07C2 07C2 07C3 07C3 07C3 07C3 07C5
(MSX ROM - MSXIO -	1575 1576 1576 1578 1586 1588 1588 1588 1588 1588 1589 1590 1592

26		;Save data to be written	
PAGE		Save di	
01-Jan-85	te to VRAM	AF SETWRT (SP),HL (SP),HL AF (VDP.DRW),A from VRAM from VRAM SETRD (SP),HL (SP),HL (SP),HL (SP),HL (SP),HL (SP),HL A,(VDP.DRW) A,(VDP.DRW)	А,L (VDP.CW),A А,H
-80 3.44 br VDP	WRTVRM: ; ; Write a byte to VRAM :	PUSH CALL EX EX EX EX POP OUT POP OUT RET RET RET RET EX EX EX EX EX EX EX EX EX EX EX EX EX	LD DI LD LD
BASIC BIOS) Macro-80 Utility routines for VDP		F5 CD 07DF E3 F1 D3 98 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9	7D F3 D3 99 7C
	07CD	07CD 07CE 07D1 07D2 07D4 07D7 07D7 07D7 07D7 07D7 07D5 07D5 07D5	07DF 07E0 07E1 07E3
(MSX ROM BASIC BIOS) - MSXIO - Utility rout	1594 1595 1596 1597	1599 1601 1602 1603 1604 1603 1608 1609 1610 1611 1613 1613 1613 1613 1613 1613	1621 1622 1623 1624

PAGE 26-1	;For write, set bit 6 high			VDP												•	- changes foreground, background, and border color	.Are we in text mode		;Yes, change color in 40*24 text mode		;Change border color for all		; No	;We're in 32*24 text mode
01-Jan-85	00111111B 01000000B (VDP.CW),A			for read from VDP	passed to HL		A,L		(VDP.CW),A	А,Н	00111111B	(VDP.CW),A					anges foregroun	(CLEMOD)	A	M, CHCLTX	AF	CHGBDR	AF	NZ	A, (FORCLR)
) 3.44 VDP	AND OR OUT	EI RET	SETRD:	; ; Set address	; ; Address is F	••	ΓD	DI	OUT	LD	AND	OUT	EI		CHGCLR:	••	; CHGCLR - cha			đ	HSUG	CALL	POP	RET	LD
Macro-80 tines for	E6 3F F6 40 D3 99	FB C9					7D	F3	D3 99	7C	E6 3F	D3 99	FB	60				5 K C 5 K C	JA FCAF	52 FA 0824	μS	CD 0832	Fl	C0	3A F3E9
	07E4 07E6 07E8	07EA 07EB	07EC				07EC	07ED	0 7 EE	07F0	07F1	07F3	07F5	07F6	07F7				1 J 1 0	07FB	07FE	07FF	0802	0803	0804
(MSX ROM - MSXIO -	1625 1626 1627	1628 1629	1630	1631 1632	1633 1634	1635	1636	1637	1638	1639	1640	1641	1642	1643	1644	1645	1646	1647	0401	1650	1651	1652	1653	1654	1655

01-Jan-85	А, А А, А А, А	A, A HL, BAKCLR	(нг.) нг., (т32СОГ.)	BC,20H	ļ	AF SETWRT		AF	(VDP.DRW),A	AF	BC	A,C	В	NZ , FLVRM1	AF				A, (FORCLR)	A, A	A, A	A, A	Α,Α	HL , BAKCLR	(HL)	В,А
3.44	ADD ADD ADD	ADD LD	OR L.D	LD LD		CALL		POP	OUT	PUSH	DEC	LD	OR	JR	POP	RET			LD	ADD	ADD	ADD	ADD	ГD	OR	LD
ASIC BIOS) Macro-80 Utility routines for VDP		F3EA	F3BF		FIL VRM:	07DF	FLVRM1:		98					F7			CHCLTX:	••	F3E9					F3EA		
BIOS) ty rou	87 87 87	87 21	B6 2A	01	Ц С	2 8 9		Fl	D3	FΣ	0B	79	BO	20	Fl	60			3A	87	87	87	8-7	21	B6	47
щ	0807 0808 0809	080A 080B	080E 080F	0812	0815	0816	0819	0819	081A	081C	081D	081E	081F	0820	0822	0823	0824		0824	0827	0828	0829	082A	082B	082E	082F
(MSX ROM - MSXIO -	1656 1657 1658	1659 1660	1661 1662	1663	1664 1665	1666 1666	1667	1668	1669	1670	1671	1672	1673	1674	1675	1676	1677	1678	1679	1680	1681	1682	1683	1684	1685	1686

26-2

PAGE

PAGE 26-3	Get border color
PA(ů.
01-Jan-85	CHGBD1 A, (BDRCLR) B, A C, 7 WRTVDP
3.44	JR LD LD LD LD LD LD LD LD LD LD LD LD LD
o-80 for VDP	CHGBDR: ; CHGBD1:
MSX ROM BASIC BIOS) Macro-80 MSXIO - Utility routines for VDP	18 03 3A F3EB 47 0E 07 C3 057F
BASIC Utili	0830 0832 0832 0835 0835 0835 0835 0836
(MSX ROM - MSXIO -	1687 1688 1689 1690 1691 1693 1693

PAGE 27	ext mode	;Check current screen mode ;We're in text mode	;No, change to text mode then		;Statement not ending ;Save text pointer	;Restore text pointer creen	;Change to what mode ;To text mode	;To hires mode ;To multicolor mode points
01-Jan-85	- Force screen to text mode	CHKSCR C	A, (OLDSCR) H.TOTE CHGMOD	S	NZ HL CLSSUB	P HL ;	A M,INITXT Z,INIT32 A	I
-80 3.44 or VDP	TOTEXT: ; ; TOTEXT - FOY	; CALL RET	LD CALL JP CLS:	; ; CLS - clears ;	RET PUSH CALL	POP RET CHGMOD: ; CHGMOD - Cha	JP JP DEC DEC	JP JP SUBTTL - MSXIO
BIOS) Macro-80 .ty routines for VDP			3A FCB0 CD FDBD C3 084F		C0 E5 CD 0777	Е1 С9	3D FA 050E CA 0538 3D	CA 05D2 C3 061F
MSX ROM BASIC MSXIO - Utili	083B	83	083F 0842 0845 0848		ິໝິໝີ	084D 084E 084F	084F 0850 0853 0856	0857 085A
(MSX RC - MSXIO	1695 1696 1697 1698	1699 1700 1701	1702 1703 1704 1705	1706 1707 1708	1709 1710 1711	1712 1713 1714 1715 1716	1717 1718 1719 1720 1721	1722 1723 1724

PAGE 28		;Save character to output	;Check if aborted	;No ;Restore character	;Save it again ;Send to output port ;Generate strobe	;Restore data output	Reset carriage position Send CR even if LPT not active
01-Jan-85	PTOUT: Output a character to printer	H.LPTO AF	BREAKX C , LPTABO LPTSTT	Z , CHPLP1 AF	AF (LPT.DW),A A (LPT.SB),A A (LPT.SB),A	AF A	A (LPTPOS),A A,0DH CHPLP2 AF
3.44	LPTOUT: ; Output a cha: ;	CALL PUSH CHPLP1:	CALL JR CALL	JR POP CHPLP2:	PUSH OUT XOR OUT DEC OUT	POP AND RET LPTABO:	XOR LD LD CALL POP SCF
BIOS) Macro-80 entry points		CD FFB6 F5	CD 046F 38 12 CD 0884	8 F6 1	91 90 90	6 0 0	AF 32 F415 3E 0D CD 086C F1 37
M BASIC - Some	085D	085D 0860 0861	0861 0864 0866	0869 086B 086C	086C 086D 086F 086F 0873 0873	0875 0876 0877 0877 0878	0878 0879 087C 087C 087E 0881 0881
(MSX ROM BASIC - MSXIO - Some	1725 1726 1727 1728	1730 1731 1732	1733 1734 1735	1736 1737 1738	1740 1740 1742 1743	1745 1746 1747 1748 1749	1750 1751 1752 1753 1754

1		if ready					
28-1		0					
PAGE		;LSB is	ON :	position	; OUTCHR	-	
01-Jan-85		H.LPTS A,(90H)	A, A	Position cursor to specified position	А,1БН 18Н А,'Ү' 18Н А.Т	А, L А, lFH 18Н А, H А, lFH 18Н	acter code HL AF
3.44	RET :	CALL IN RRCA RRCA	CCF SBC RET	tion cur	LD RST RST RST	ADD RST LD ADD RST	RET NCHR: Convert character PUSH HL PUSH AF
0-80	LPTSTT:		FOSIT:	; Posit			CNVCHR: ; ; Conve ;
BIOS) Macro-80 entry points	60	CD FFBB DB 90 0F 0F	3F 9F C9	З Е ТВ		C6 1F DF 7C C6 1F DF	ЕБ С9 F5
BASIC Some	0883 0884	0884 0887 0889 088A	088B 088C 088D 088D 088E	088F	0890 0891 0893 0894	0895 0897 0898 0899 0898 0898	089D 089D 089D 089D 089E
(MSX ROM - MSXIO -	1756 1757 1758	1759 1760 1761 1762	1763 1764 1765 1766	1769 1769 1770	1771 1772 1773 1774	1775 1775 1777 1778 1779 1779	1/80 1781 1782 1783 1784 1785 1785

PAGE 28-2 68	;Preceeded by a header byte	;Clear this since seen ;No	;Get rid of offset ;Valid range .ves	;Compensate value	;Set Z flag ;Make sure carry is cleared		<pre>;Graphic header ;No, do not modify ;Set GRPHED flag ;Carry is clear indicating one more byte is ;required ;required</pre>	ILACLEI LU LAI
01-Jan-85	HL, GRPHED A (HL)	(HL),A Z,CNVCH3 AF	0100000B	A,0100000B	A	Η̈́	AF 1 NZ,CNVCH1 (HL),A HL	- MONIU - UULPUL A CHARACLEI LU LAI
-80 3.44	LD XOR	2. 0.1 8. 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.	SUB CP	ADD ADD CNVCH1:	CP SCF CNVCH2:	POP RET CNVCH3:	POP CP JR LD POP RET CTTDATT	ATVOL - HITANC
(MSX ROM BASIC BIOS) Macro- - MSXIO - Some entry points		44 77 45 28 0D 45 81			80 BF B1 37 82	82 El 83 C9 84	B4 F1 B5 FE 01 B7 20 F7 BA E1 BB C9 BB C9	
(MSX ROM BAS - MSXIO - Sc	1787 089F 1788 08A2 1789 08A3		1795 08A8 1795 08A8		1798 08B0 1799 08B1 1800 08B2		1805 08B4 1806 08B5 1807 08B7 1808 08B7 1809 08B3 1809 08B4 1810 08B4 1810 08B4	TTOT

PAGE 29						;Are we in text mode	;No, ignore this	; Erase old cursor if cursor enabled				;Display new cursor if cursor enabled													;Convert character code	;Was a graphic header, wait for next		;Converted code, send as is
01-Jan-85		HL	DE	BC	AF H.CHPU	CHKSCR	NC, POPALL	CKERCS	AF	AF	CHPUTI	CKDPCS	A, (CSRX)	А	(TTYPOS) , A		AF		BC	DE	Ш				CNVCHR	NC	C,A	NZ, CHPUT3
3.44		HSUG	HSUG	HSUG	CALL	CALL	JR	CALL	POP	PUSH	CALL	CALL	ГD	DEC	ГD		POP		POP	POP	POP	RET			CALL	RET	ΓD	ЯĻ
Macro-80 acter to CRT	CHPUT: i	-														POPALL:		PBDHRT:					CHPUTI:	••				
) char		E5	D5 21	C5	CD FDA4	CD 0B9F	30 12	CD 0A2E	Fl	F5	CD 08DF	CD 09E1	3A F3DD	3D	32 F661		Fl		C1	DI	EI	60			CD 089D	D0	4F	20 0D
MSX ROM BASIC BIOS MSXIO - Output a	08BC	08BC	08BD	08BE	08C0	08C3	08C6	08C8	08CB	08CC	08CD	08D0	08D3	08D6	08D7	08DA	08DA	08DB	08DB	08DC	08DD	08DE	08DF		08DF	08E2	08E3	08E4
(MSX RO) - MSXIO	1812 1813 1814	1815	1816	0101 /181	1819	1820	1821	1822	1823	1824	1825	1826	1827	1828	1829	1830	1831	1832	1833	1834	1835	3	1837	1838	ŝ	4	1841	1842

PAGE 29-1	<pre>;Are we executing escape sequence ; ?Yes ;Restore character ;Control code ;Rubout ;Yes ;Rubout ;Yes ;Convert to raw code and write to VRAM ;Advance cursor ;All done if not wrapped to next line ;Advance cursor ;All done if not wrapped to next line ;Duterminate this line ;Unterminate this line ;Go to start of the next line ;Covn cursor ;Exit if not at bottom ;I:=window top line ;Croil up by deleting the first line ;Croil up by deleting the first line ;Scroil up by deleting the first line</pre>	
01-Jan-85	HL, ESCCNT A, (HL) A NZ, INESC A, C C, CNTPUT C, CNTPUT C, CNTPUT A C, CNTPUT A C, CNTPUT A C, CNTPUT A Z, RUBOUT PUTVRM RIGHT NZ A SETTRM H, l H, l DOWN NZ STOCSR L, l DELLNO DELLNO A A A A A A A A A A A A A A A A A A A	
0 3.44 o CRT	LD HL, LD A, (1, LD A, (1, NZ, 1, LD A, C CP NZ, LD HL, CC C, C CP 7FH CC TFH CC TFH CC LL FUT CC LL PUT CC LL PUT CC LL PUT CC LL PUT CC CC TFH LD H, 1 LD H, 1 LD H, 1 LD L, 1 LD L, 1 DOW RET NZ CC LL SET CC LL SET CC LL SET CC LL C CC LL SET CC LL C FO CON CC	
.IOS) Macro-80 a character to	21 FCA7 7E A7 C2 098F 79 FE 20 79 78 77 78 78 77 78 78 77 78 78 78 78 78	
M BASIC BIOS - Output a	08 E6 08 EA 08 EA 08 EA 08 EB 08 EB 08 EF 08 EF 09 01 09 05 09 05 00 00 00 00 00 00 00 00 00 00 00 00 0	
(MSX ROM - MSXIO -	1843 1844 18445 18446 18446 18846 1855 1855 1855 1866 1866 1866 1866 186	

PAGE 29-2 71	<pre>;Make sure carry is cleared ;Make sure carry is cleared ;Undefined function ;Found? ; found found? ; found? found? ; found? ; found? ; found? ; found? ; found? ; found? ; found? ; found? ; found? ; found? ; found? ; found? ; found? ; found? found</pre>	
01-Jan-85	pace feed c home screen age return escape sequence escape sequence c npht hL,JMPBC c,OCH down HL,JMPBC c,OCH HL HL,UMPBC C,OCH HL HL NZ,INDJMP C,(HL) HL NZ,INDJMP C,(HL) HL A M (HL) HL SCRY) HL C,(CR) A A A A A A A A A A A A A A A A A A A	
-80 3.44 to CRT	<pre>% Back space % 9 Tab % 10 Line feed % 11 Cursor home % 12 Clear screen % 13 Carriage return % 30 Cursor left % 30 Cursor left % 31 Cursor up % 31 Cursor down % 31 Cursor down % 10 HL,J INDJMP: % INC HL NNC NC HL NNC HL NNC HL NNC HL NNC HL NNC NC HL NNC NC HL NNC NC HL NNC HL NNC NC NC</pre>	1
Macro- Iracter	21 092D 0E 0C 0E 0C 0D 78 8E 73 23 73 23 74 75 26 73 20 75 75 75 75 75 75 75 75 75 75 75 75 75	,)
(MSX ROM BASIC BIOS) - MSXIO - Output a cha	0914 0914 0915 0919 0918 0919 0922 0923 0928 0928 0928 0928 0928	
(MSX ROM - MSXIO -	1874 1875 1876 1876 1876 18879 18881 18881 18881 18881 18886 18881 18896 18896 18893 18893 18993 1900 1901 1901 1903	

PAGE 29-3					table			; Beep		;Back space		; Tabulation		;Line feed		; Home		;Clear		;Carriage return		;Enter escape sequence		;Cursor right		;Cursor left		;Cursor up		;Cursor down
01-Jan-85		BC			Function dispatch table			7	BEEP	8	BS	6	TAB	10	LF	11	CSHOME	12	CLRTXT	13	CR	27	ENTESC	28	ADVCUR	29	BS	30	đ	31
3.44		PUSH	RET		Fur			DB	ΜQ	DB	DW	DB	DW	DB	DW	DB	ΜŪ	DB	Μ	DB	Μ	DB	ΜQ	DB	DW	DB	ЪW	DB	ΜQ	DB
s) Macro-80 character to CRT	JMPBC:			••	••	••	CNTTBL:																							
BIOS) tacha:		C5	60					07	1113	08	0A4C	60	0A71	0A	0908	0B	0A7F	00	077E	00	0A81	lB	0989	lc	0A5B	lD	0A4C	ΙE	0A57	lF
BASIC BIO9 Output a	092D	092D	092E				092F	092F	0630	0932	0933	0935	0936	0938	0939	093B	093C	093E	093F	0941	0942	0944	0945	0947	0948	094A	094B	094D	094E	0950
(MSX ROM BASIC BIOS) - MSXIO - Output a ch	1905 1906	1907	1908	1909	1910	1911	1912	1913	1914	1915	1916	1917	1918	1919	1920	1921	1922	1923	1924	1925	1926	1927	1928	1929	1930	1931	1932	1933	1934	1935

29-4 PAGE 01-Jan-85 3.44 MSX ROM BASIC BIOS) Macro-80
 MSXIO - Output a character to CRT

NMOD MD	SUBTTL - MSXIO - Escape sequence handler
0A61	
0951	
1936	1937

30			s creen		screen	; To maintain compatibility with	;Erase to end-of-line		;Erase to end-of-page		;Erase entire line		a line		a line		;Locate cursor		dn		down		right		left		home		des		modes
PAGE			;Clear screen		;Clear screen	; To ma	; Erase		; Erase		; Erase		;Insert a line		;Delete		;Locate		;Cursor up		;Cursor down		;Cursor right		;Cursor left		;Cursor home		;Set modes		;Reset modes
01-Jan-85			= :C =	CLRTXT	"E"	CLRTXT	"K"	EOL	"J"	EOP	"I"	ELN	"1"	ILN	"M"	DLN	"⊼"	LOC	"A"	UP	"B"	DOWN	"C"	RIGHT	"D"	LEFT	"H"	CSHOME	"X"	SETMOD	"Y"
3.44		ESCTBL:	DB	DW	DB	DW	DB	DW	DB	DW	DB	DW	DB	DW	DB	ΜQ	DB	ΜQ	DB	DW	DB	DW	DB	DW	DB	DW	DB	DW	DB	DW	DB
.OS) Macro-80 sequence handler		ш	~	7 TE	45	077E	m	OAEE	~	0B05	r)	OAEC	0	0AB4	0	A85	•	09860		A57	2	0A61	43	A44	4	0A55	8	0A7F	8	0860	6
BI De									4 A																						
ROM BASIC [0 - Escal		S	σ	95	δ	0957	σ	095A	0950	095D	095F	0960	0962	0963	0962	0960	0968	0960	096E	0960	096E	096F	1760	0972	97	0975	0977	δ		097B	
(MSX RC - MSXIO	ŝ	1939	1940	94	す	1943	1944	1945	б	6	δ	б	െ	െ	ര	δ	റ	1955	σ	റ	1958	σ	σ	σ	1962	1963	1964	1965	Ō	1967	1968

VT52

PAGE 30-1												:Sav row is expected next	:'LXI B' instruction		:Tell him we're in escare secuence		
01-Jan-85	RSTMOD			Function dispatch table	4	Α,Ι	1		A, 2	L		A,4	T		A,0FFH	(ESCCNT), A	•
o-80 3.44 andler	DW	SETMOD:	••	; Function di		ГD	DB	RSTMOD:	LD	DB	LOC:	ΓD	DB	ENTESC:	LD	LD	RET
(MSX ROM BASIC BIOS) Macro-80 - MSXIO - Escape sequence handler	0983					3E 01	01		3E 02	01		3E 04	01		3E FF	32 FCA7	60
DM BASIC 1 - Escape	097E	0980				0860	0982	0983	0983	0985	0986	0986	0988	0989	0989	098B	098E
(MSX ROM - MSXIO -	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985

PAGE 31		; Arguments expected	; Exit from escape sequence	; kes tore character	;Number of ESC handler entries				;Set modes?	; Yes	;Reset modes?			;Update ESCCNT	; Assume column expected		;Column expected		;Row expected	;Point CSRY		;Get max limit in B	;Restore character	; 0-xx				;Legal value
01-Jan-85		P, INESCI	(HT),0	A, C HI FSCTBL-2	C,0FH	JMUUNI			A	Z,GOSET	А	Z, GORSET	А	(HL),A	A, (LINLEN)	DE, CSRX	Z, INESC2	(НГ),3	GETLEN	DE		B,A	A,C	-	В	A	(DE),A	υ
-80 3.44 Idler	INESC:	ΔĿ	LD LD	0 J	LD L	JP	INESC1:	••	DEC	JR	DEC	JR	DEC	LD	LD	LD	JR	LD	CALL	DEC	INESC2:	LD	LD	SUB	CP	INC	ΓD	RET
: BIOS) Macro-80 pe sequence handler			36 00	71 0951		C3 0919			3D	28 1E	3D	28 25	3D	77	3A F3B0	11 F3DD	28 06	36 03	CD 0C32	lB		47	79	D6 20	B8	3C	12	D8
M BASIC - Escap	098F	098F	0992	1994 1995	66	099A	0660		0660	099E	09A0	09A1	09A3	09A4	09A5	09A8	09AB	09AD	0.9 AF	09B2	09B3	09B3	09B4	09B5	09B7	09B8	09 B9	09BA
(MSX ROM - MSXIO -	1986 1987 1988	1989	50	თ თ	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009		2011	2012	01	10	2015	01

PAGE 31-1	;Substitute by possible largest value		;Exit from escape sequence	; Block cursor?	; Yes	;Cursor oII? :Yes. reset cursor-enable switch	;Unimplemented feature	•				;Exit from escape sequence	;Restore character	;Underscore cursor?	;No, try next							;Cursor on?	;No, unimplemented feature	
01-Jan-85	A,B (DE),A	modes	(НГ),А АС	- 4 -	Z, STSTYL	A Z.STCSSW				is modes		(HL),A	A,C	.4.	NZ, RSETIO	A		(CSTYLE),A				A	NZ	A
3.44	LD LD RET	Set various modes	01 UI	SUB	JR	JEC.	RET			Reset various modes		ГD	ГD	SUB	ЯĻ	INC		LD	RET			DEC	RET	INC
-80 idler	GOSET:	: Set						GORSET:	••	; Rese	••						STSTYL			RSETIO:	••			
BIOS) Macro-80 e sequence handler	78 12 C9		77	D6 34	28 0B	3D 28 OF						77	79	D6 34	20 05	3C		32 FCAA	C9			3D	CO	3C
A BASIC BIOS - Escape sec	09BB 09BC 09BD 09BE		09BE 09BE	0000	09C2	09C5	09C7	09C8				09C8	09C9	09CA	0900	09CE	0.9 CF	0.9 CF	09D2	09D3		09D3	0904	09D5
(MSX ROM - MSXIO -	2017 2018 2019 2020	2021 2022 2023	2024	2026	2027	2029 2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047

PAGE 31-2				<pre>;Get current cursor position ;Save it for future use ;Get a raw character at cursor ;Remember this code ;Then read pattern for this code</pre>
01-Jan-85	(CSRSW),A or if disabled	A,(CSRSW) A NZ DSPCSR or if enabled	A, (CSRSW) A Z cursor H.DSPC CHKSCR NC	HL, (CSRY) HL GETVRM (CODSAV),A L,A H,0
80 3. 44 dler	STCSSW: LD (CSRSW),A RET CKDPC0: ; Display cursor if disabled	LD A, (CSRSW) AND A RET NZ JR DSPCSR CKDPCS: ; Display cursor if enabled	د د	LD PUSH CALL LD LD LD
3IOS) Macro-80 s sequence handler	32 FCA9 C9	3A FCA9 A7 C0 18 05		2A F3DC E5 CD 0BD8 32 FBCC 6F 26 00
1 BASIC B - Escape	09D6 09D6 09D9 09D A	09DA 09DD 09DE 09DF 09DF 09E1	09E1 09E4 09E5 09E6 09E6 09E9 09E9	09ED 09F1 09F1 09F4 09F8 09F8
(MSX ROM BASIC BIOS - MSXIO - Escape se	2048 2049 2050 2051 2052 2053	2055 2055 2058 2059 2060 2061	2063 2064 2065 2066 2066 2068 2068 2070 2071	2073 2074 2075 2075 2077 2078

PAGE 31-3	; [A] * 8								;Make a complement of this pattern	; Assume full reverse cursor			;Good assumption	; No, reverse bottom 3 lines only							;Assign this pattern to 255				;Restore cursor position	;Get code for cursor	;Set it at cursor position					
01-Jan-85	HL, HL	нг, нг	Ш, П	DE, HL	HL, (CGPBAS)	HL	HL, DE	GET8 B	HL, BUFEND+7	B,8	A, (CSTYLE)	A	Z, DSPCS1	B,3		A, (HL)		(HL),A	HL	DSPCS1	HL	BC,07F8H	HL, BC	PUT8B	Η	C, OFFH	PUTVRM			if disabled		
3.44	ADD	ADD	ADD	ЕX	LD	HSUG	ADD	CALL	LD	LD	LD	AND	JR	LD	DSPCS1:	LD	CPL	LD	DEC	DJNZ	POP	ГD	ADD	CALL	POP	LD	ЧŲ	CKERCO:		Erase cursor if disabled		
c BIOS) Macro-80 ape sequence handler		29						CD		90				90								01 07F8			EI	OE FF	C3 0BE6	CK	•~	•		
M BASIC B. - Escape	09FA	09FB	09FC	09FD	09FE	0A01	0A02	0A03	0A06	0A09	0A0B	0A0E	0A0F	0All	0A13	0A13	0A14	0A15	0A16	0A17	0A19	0ALA	0A1D	OALE	0A21	0A22	0A24	0A27				
(MSX ROM BASIC BIOS - MSXIO - Escape sex	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	

PAGE 31-4																					;Get old code		;Restore old code		len ts
01-Jan-85		A, (CSRSW)	A	NZ	ERACSR			a cursor if enabled		A, (CSRSW)	А	2					H.ERAC	CHKSCR	NC	HL, (CSRY)	A, (CODSAV)	C,A	PUTVRM		- Cursor movements
80 3.44	dler	LD	AND	RET	JR	CKERCS:	••	; Erase a curso	••	ГD	AND	RET	ERACSR:	••	; Erase cursor	••	CALL	CALL	RET	ΓD	LD	ГD	JP	••	SUBTTL - MSXIO -
MSX ROM BASIC BIOS) Macro-80	Escape sequence handler			B CO	C 18 05	Щ					1 A7	12 C8					8		00 6	A 2A F3DC	3 A	0 4F	1 C3 0BE6		
MSX ROM BAS	· MSXIO - Es		2111 0A2A				2115	2116	2117		2119 0A31		2121 0A3	2122	2123	2124		2126 0A36			2129 0A3D	2130 0A4	2131 0A41	2132	2133

PAGE 32		;Are we at the right-end of line? ;Yes, return with Z flag ;Go to next column		;Not at left-end	;'LXI D,' instruction	;Are we at the left-end of line? ;'MVI A,' instruction	;Are we at the top of any window? ;Yes, return with Z flag
01-Jan-85		A, (LINLEN) H H H	STOCSK	LEFT NZ A, (LINLEN) H, A	11H	н , ЗЕН	L Z
3.44	RIGHT: ; Cursor right ;	LD CP RET INC	BS: BS: Back space	CALL RET LD LD	DB LEFT: ; ; ;	DEC DB UP: ; Cursor up	DEC RET
ASIC BIOS) Macro-80 Cursor movements	дост	3A F3B0 BC C8 24	ID	CD 0A55 CO 3A F3B0 67		35 3E	2D C8
BASIC B. Cursor	0A44	0A44 0A47 0A48 0A49	0A4A 0A4C	0A4C 0A4F 0A50 0A53	0A54 0A55	0A55 0A56 0A57	0A57 0A58
(MSX ROM BASIC BIOS - MSXIO - Cursor mo	2134 2135 2136 2137 2138	2139 2140 2141 2142	2143 2144 2145 2146 2147	2148 2149 2150 2151	2152 2153 2154 2155 2155	2157 2158 2159 2160 2161 2162	2163 2164

(MSX ROM	MSX ROM BASIC BIOS)	OS) Macro-80	3.44	01-Jan-85	PAGE	32-1
- OIXSW -	Cursor	Cursor movements				
2165	0A59	18 OE	JR	STOCSR		
2166	0A5B		ADVCUR:			
2167			••			
2168			; Advance cursor	or		
2169						
2170	0A5B	CD 0A44	CALL	RIGHT		
2171	0A5E	c0	RET	NZ		
2172	OA5F	26 01	LD	Н,1		
2173	0A61		DOWN:			
2174						
2175			; Cursor down			
2176						
2177	0A61	CD 0C32	CALL	GETLEN	;Get an	Get an actual
2178	0A64	BD	CP	Ţ	;Are we	e at the
2179	0A65	C8	RET	Z	;Yes, r	;Yes, return v
2180		38 05	JR	C, DOWNL	;We're below	below s
2181		2C	INC	Ц	;Go to	Go to next l
2182	0A69		STOCSR:			
2183	0A69	22 F3DC	LD	(CSRY), HL		
2184	0A6C	C9	RET			
2185	0A6D		: INWOO			
2186			••			
2187	0A6D	2D	DEC	Г		
2188	0A6E	AF	XOR	A		
2189	0A6F	18 F8	JR	STOCSR		
2190	0A71		TAB:			
2191						
2192			; Tabulation			
2193			••			
2194		3E 20	ГD	Α,''		
2195	0A73	CD 08DF	CALL	CHPUTI		

Get an actual bottom of screen; Are we at the bottom of screen? Yes, return with Z flag We're below screen bottom Go to next line

PAGE 32-2	<pre>(CSRX) Z,TAB Line insert and delete of CRT</pre>	
01-Jan-85		
3.44	LD A DEC A DEC A AND 7 JR NN 7 JR N RET RET RET CSHOME: ; ; CUTSOT home ; ; CUTSOT home ; ; CUTSOT home ; ; CUTSOT home ; ; CUTSOT home ; ; CUTSOT home ; ; CUTSOT home ; ; CUTSOT home ; ; CUTSOT home ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	
MSX ROM BASIC BIOS) Macro-80 MSXIO - Cursor movements	3A F3DD 3D F6 07 20 F3 C9 F3 C	
M BASIC I - Cursoi	0A76 0A79 0A7A 0A7E 0A7F 0A7F 0A7F 0A81 0A81 0A81 0A81	
(MSX ROM - MSXIO -	2196 2197 2198 2199 2200 2201 2203 2204 2204 2205 2206 2206 2209 2209 2209 2210 2211 2211 2211	

84									creen			ly		to be moved upward)			n [DE]													
PAGE 33			[T]		top of line				;Get an actual height of screen		; Something is wrong	;Delete the bottom line only	;Save row	;Save counter (# of lines to be moved upward)			;Get address of [LINTTB] in [DE]										;Save counter		;Get l line specified by L	
01-Jan-85			a line specified by [L]	+ + -	d be set at the		CR		GETLEN	Г	U	Z, ELN	Н	AF	C,A	B,0	GETTRM	L,E	Н, D	НĹ		HL , FSTPOS	(HL)	AF	用		AF	ц	GETILN	Ŀ
			; Delete a line		; Cursor should be	••	CALL	DELLNO:	CALL	SUB	RET	JP	FUSH	HSNA	ГD	ΓD	CALL	LD	LD	INC	LDIR	ГD	DEC	POP	POP	DELLN1:	HSUG	INC	CALL	DEC
) Mac rt and							CD 0A81		CD 0C32	95	D8	CA 0AEC	ES	F5	4F	00 90	CD 0C1D	6B	62	23	ED BO	21 FBCA	35	Fl	El		F5	2C	CD 0BAA	2D
ပေပ	0A60						0 A 85	0A88	0A88	0A8 B	0A8C	0A8D	0A90	0A91	0A92	0A93	0A95	0A98	0A99	0A9A	0A9B	0A9D	0AA0	0 AAL	0AA2	0AA3	0 AA 3	0AA4	0AA5	0AA8
- MSXIO 2214	CT77	9177	2217	2218 8122	6177	2220	2221	2222	2223	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239	2240	2241	2242	2243	2244

PAGE 33-1 85	4	; Put l line specified by L		;Restore counter			;Blank bottom line					Cursor should be set at the top of line				;Get an actual height of screen			; Something is wrong::			;Save row to be inserted	;Save # of lines to be moved downward						;Save pointer to [LINTTB] for the bottom line	; Form source address		
01-Jan-85		PUTLN	Г	AF	А	NZ, DELLNI	ELN			e		d be set at 1		CR		GETLEN	Н,А	Г	υ	Z, ELN	Г,Н	Η̈́Γ	AF	C,A	B,0	GETTRM	L,E	Н, D	ΗĹ	Ħ		HL
-80 3.44	of CR1	CALL	INC	POP	DEC	JR	дĿ	ILN:	••	; Insert a line	••	; Cursor shoul	••	CALL	INSLNO:	CALL	LD	SUB	RET	£	ГD	HSUG	HSUA	ΓD	ΓD	CALL	ΓD	ΓD	HSUI	DEC	LDDR	POP
BIOS) Macro-80	σ,	CD 0BC3	2C	Fl	3D	20 F2	C3 0AEC							CD 0A81		CD 0C32	67	95	D8	CA 0AEC	6C	E5	F5	4F	00 00	CD 0C1D	6B	62	ES	2B	ED B8	El
		0AA9	0 AAC	0AAD	OAAE	OAAF	0AB1	0AB4						0AB4	0AB7	0AB7	0ABA	0ABB	0ABC	0ABD	0AC0	0AC1	0AC2	0AC3	0AC4	0AC6	0AC9	0ACA	0ACB	OACC	0ACD	OACF
(MSX ROM BASIC	OIXSM -	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271	2272	2273	2274	2275

PAGE 33-2	<pre>;Make sure the bottom line is terminated ;Save counter ;Restore counter ;Restore counter</pre>
01-Jan-85	(HL),H AF HL AF L GETILN L PUTILN L PUTILN L AF AF AF AF AF AF AF AF AF AF AF AF AF
(MSX ROM BASIC BIOS) Macro-80 3.44 - MSXIO - Line insert and delete of CRT	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
MSX ROM BASIC MSXIO - Line	 7 7 7 7 1 7 1 1<
) – MS	2276 2277 2279 2279 2280 2281 2281 2283 2285 2285 2286 2286 2286 2286 2286 2286

PAGE 34					יא בעם אין	, Halks apace	; We le al uie lop of acteur . Croretrike with a snare	5					d							D D		;Save current position (column)			;Restore current position		;Overstrike with a space		
0 3.44 01-Jan-85	. miroaita			; FIASE DIEVIDUS CHARACLET					ELN:	••	; Erase entire line		; Cursor should remain unchanged	••	LD H,1	EOL:		; Erase to end-of-line		; Cursor should remain unchanged	CALL TERMIN		CALL VADDR	CALL SETWRT	POP HL	EREOL1:	LD A,''	OUT (VDP.DRW), A	I NC H
AASIC BIOS) Macro-80 Character(s) erase	C	53			£			38	1						BC 26 01	3E						E5			F8 E1	65	3E	FB D3 98	
(MSX ROM BASIC BIOS - MSXIO - Character		2293 UAE3	2294	C677				22.73 UAE9 2300 0AE9		2302	2303	2304	2305	2306		2308 0AEE	2309	2310	1107	2312	2314 0AFF						2320 0AF9		2322 0AFD

PAGE 34-1	LD A, (LINLEN) CP H JR NC, EREOLI BEOP: EOP: EOP: EOP: Cursor should remain unchanged Cursor should remain unchanged Cursor should remain unchanged PUSH HL ;Save current position PUSH HL ;Save current position CALL EOL ;Erase to end-of-line POP HL ;Save current position CALL EOL ;Erase to end-of-line POP HL ;Save current position CALL GETLEN ;Get an actual height of CRT CP L RET Z ;All done LD H, l INC L JR EOP
01-Jan-85	LD A, (LINLEN) CP H JR NC, EREOLI RET P: Erase to end-of-page Erase to end-of-page Cursor should remain unchanged PUSH HL CALL EOL POP HL CALL CALL EOL POP HL CALL EOL POP HL
3.44	LD A, (LIN CP H JR NC, ERE RET P: Erase to end-of-page cursor should remain PUSH HL CALL EOL POP HL CALL EOL POP HL CALL GETLEN CP L RET Z LD H, 1 INC L BTTL - MSXIO - FUNC
ASIC BIOS) Macro-80 Character(s) erase	3A F3B0 BC C9 C9 F5 C9 F60 F F F F F F F F F F F F F F F F F F
M BASIC - Chara	0AFE 0B01 0B02 0B04 0B05 0B05 0B05 0B06 0B06 0B06 0B06 0B06
(MSX ROM BASIC - MSXIO - Char	2323 2323 2324 23226 23226 2333 2333 233

PAGE 35		;Say no function key is displayed	;We're not in text mode, just set flag ;Save possible text pointer ;Erase last line	Restore possible text pointer abled	;Now being displayed? ;No	;Say function key is displayed ;We're not in text mode, just set flag ;Save possible text pointer
01-Jan-85	function key	H.ERAF A SETCHK	NC HL HL, (CRTCNT) ELN	POP HL ; RET NKSB: Display function key if enabled	A,(CNSDFG) A Z tion key	H. DSPF A, 0FFH SETCHK NC HL A, (CSRY)
a cro-80 3.44 display/erase.	ERAFNK: ; ; Erase functi ;	CALL XOR CALL	RET PUSH LD CALL	POP POP RET FNKSB: ; Display func ;	LD A, (CNS AND A RET Z DSPFNK: ; Display function key	CALL LD CALL RET PUSH LD
M SY		CD FDB8 AF CD 0B9C	D0 E5 2A F3B1 CD 0AEC		3A F3DE A7 C8	CD FDB3 3E FF CD 0B9C D0 E5 3A F3DC
M BASIC - Funct	0B15	0B15 0B18 0B19	0B1C 0B1D 0B1E 0B1E 0B21	0B26 0B26 0B26	0B26 0B29 0B2A 0B2B 0B2B	0B2B 0B2E 0B30 0B33 0B33 0B34 0B35
(MSX ROM BASIC BIOS) - MSXIO - Function ke	2345 2346 2347 2349 2349	2350 2351 2352	2353 2354 2355 2356	2359 2358 2359 2360 2361 2362	2363 2364 2365 2365 2365 2368 2369	2370 2371 2372 2373 2374 2375

06	;Scroll up if we're at the bottom of screen	s ta tus pressed	ssed	f function key is displayed cination	;Restore temporary destination in [DE] ;Total number of keys ;Calculate (LINLEN-4) / 5 ;Not enough room for function keys	
PAGE 35-1	;Scroll up if we're	;Get current shift status ;Assume shift not pressed	;Good assumption ;Shift is being pressed	<pre>;Mark which part of function key is ;Set temporary destination ;=40</pre>	;Restore temporary des ;Total number of keys ;Calculate (LINLEN-4) ;Not enough room for f	
01-Jan-85	HL, CRTCNT (HL) A, 0AH NZ, NTBOTM 18H	A, (SFTKEY) HL, FNKSTR A, 1	C, DSPFKI HL, FNKSTR+80 A	(FNKSWI),A DE,BUFEND DE B,'(' A,'' (DE),A DE DE DE DE DE	DE C,5 A,(LINLEN) 4 C,DSPFKE B,OFFH	В
3.44 /erase.	LD CP LD JR RST NTBOTM:		JR LD XOR DSPFK1:	LD FUSH LD LD LD SFKCL: LD JNZ DJNZ	POP LID LID SUB JR LD LD	
) Macro-80 eys display	21 F3B1 BE 3E 0A 20 01 DF NTB	BEB 87F 1	04 F8CF	32 FBCD 11 FC18 D5 06 28 3E 20 12 13 13 10 FC	D1 0E 05 3A F3B0 D6 04 38 2B 06 FF DSPFK4	04
щ	0B38 0B3B 0B3C 0B3E 0B40 0B41	0B41 0B44 0B45 0B48 0B48	0B4A 0B4C 0B4F 0B50	0850 0853 0856 0857 0858 0858 0858 0858	0B5F 0B60 0B62 0B65 0B65 0B69 0B6B	0B6B
(MSX ROM - MSXIO -	2376 2377 2378 2379 2380 2381	2382 2383 2384 2385 2385	2385 2387 2388 2389	2390 2391 2393 2394 2395 2395 2398 2398	2399 2400 2401 2402 2403 2403 2405	40

PAGE 35-2					;No enough room	;Skip next byte		;Put separator space	;Save key counter	;Reset # of characters actually fetched		;Get from function key string	; Prepare for next fetch			;This is a graphic header, fetch more	;Converted graphics character, store this	;Printable?	;No, ignore this								;Skip rest		;Restore counter		
01-Jan-85	J.	NC, DSPFK4	А,В	A	Z,DSPFKE	3EH		DE	BC	c,0		A, (HL)	用	с С	CNVCHR	NC, DSPFK5	NZ, DSPFK8	-	C, DSPFK6		(DE),A		DE	DSPFK5	A,10H	U	C,A	HL, BC	BC	C	NZ,DSPFK2
Macro-80 3.44 display/erase.	SUB	JR	LD	AND	JR	DB	DSPFK2:	INC	HSUG	LD	DSPFK5:	LD	INC	INC	CALL	JR	JR	CP	JR	DSPFK8:	LD	DSPFK6:	INC	ZNCO	ΓD	SUB	LD	ADD	POP	DEC	JR
Ys Y		30 FB	78	A7	28 20	3E		13	C5	0E 00		7E	23	00	CD 089D	30 F8		FE 20	38 01		12		13	10 EE	3E 10	91	4F	60	c1	0D	20 El
д	0B6C	0B6E	0B70	0B71	0B72	0B74	0B75	0B75	0B76	0B77	0B79	0B79	0B7A	0B7B	0B7C	0B7F	0B81	0B83	0B85	0B87	0B87	0 B88	0 B8 8	0 B8 9	0 B8 B	0B8D	0 B8 E	0 B8 F	0890	0.B91	0B92
(MSX ROM - MSXIO -	2407	2408	2409	2410	2411	2412	2413	2414	2415	2416	2417	2418	2419	2420	2421	2422	2423	2424	2425	2426	2427	2428	2429	2430	2431	2432	2433	2434	2435	2436	2437

5-3	Display at the lowes
PAGE 35-3	;Display a
01-Jan-85	HL , (CRTCNT) PUTILN
MSX ROM BASIC BIOS) Macro-80 3.44 MSXIO - Function keys display/erase.	DSPFKE: LD CALL
MSX ROM BASIC BIOS) Macro-80 MSXIO - Function keys display.	2A F3B1 CD 0BC3
M BASIC - Funct	0B94 0B94 0B97
(MSX RC - MSXIO	2438 2439 2440

;Display at the lowest line		;Restore possible text pointer			outines
HL, (CRTCNT)	PUTILN	HL			SUBTTL - MSXIO - Low level routines
LD	CALL	POP	RET	••	SUBTTL - MSXIO
2A F3B1	CD 0BC3	El	C9		
0B94	0.B97	0 B9 A	0 B9 B		
2439	2440	2441	2442	2443	2444

		the status	scified by H,L
36	node	ı with	ads uo
PAGE	s cr een 1	;Return with	of positi
01-Jan-85	ETCHK: Set CNSDFG and check current screen mode LD (CNSDFG),A HKSCR:	current screen mode LD A,(SCRMOD) CP 2 RET RET bytes from HL	PUSH HL LD C,8 JR GETILI STILN: Get character and attribute of position specified by H,L Character returned in C PUSH HL LD H,1 CALL VADDR LD A,(LINLEN) LD C,A
3.44	SETCHK: ; ; Set CNSDFG a ; LD CHKSCR: ;	; Check curren ; LD CP CP CP RET GET8B: ; Get 8 bytes ; CP	PUSH HL LD C,8 JR GETILI GETILN: ; Get character and attr ; Character returned in ; PUSH HL LD H,1 CALL VADDR LD A,(LINLE LD C,A
3ASIC BIOS) Macro-80 Low level routines	32 F3DE	3A FCAF 3A FCAF FE 02 C9 C9 C9 C9 FF FF	08 0A 0A 0A 01 01 F3B0 F3B0
H	0B9C 0B9C 0B9C 0B9F	0B9F 0BA2 0BA4 0BA5 0BA5	0BAA 0BAA 0BAA 0BAA 0BAD 0BB0 0BB3
(MSX ROM BASIC BIOS - MSXIO - Low level	2445 2446 2444 2448 2449 2450 2451 2451	2453 2454 2455 2455 2455 2459 2460 2461 2461	2463 2463 24665 24665 2466 2471 2471 2471 2473 2473 2473 2473

PAGE 36-1	;Storage for l line																									;Save coordinate	;Calculate VRAM address	;Set up VDP for read	
01-Jan-85	B,0 DE,BUFEND	LDIRMV	用				Н,	с,8	PUTLLI			ΗL	н,1	VADDR	A, (LINLEN)	C,A		B,0	DE, HL	HL, BUFEND	LDIRVM	HL				Η	VADDR	SETRD	(SP), HL.
o-80 3.44 s	GETILI: LD LD	CALL	POP	RET	PUT8 B:		HSUG	LD	JR	PUTILN:		PUSH	LD	CALL	ΓD	LD	PUTLL:	LD	EX	ΓD	CALL	POP	RET	GETVRM:		HSUG	CALL	CALL	EX
(MSX ROM BASIC BIOS) Macro-80 - MSXIO - Low level routines	06 00 11 FC18		El	C9			E5	0E 08	18 0A			E5	26 01	CD OBF?	3A F3B	4F		00 00	EB	21 FC18	CD 0744	El	60			E5	CD 0BF2	CD 07EC	E3
BASIC LOW]	0BB4 0BB4 0BB6	0 BB9	0BBC	0 BBD	OBBE		0BBE	OBBF	0BC1	0BC3		0 BC 3	0BC4	0BC6	0BC9	0 BCC	0BCD	0 BCD	0 BCF	0 BD0	0BD3	0 BD6	0BD7	0 BD8		0 BD8	0BD9	0 BDC	OBDF
(MSX ROM BASIC BIOS) - MSXIO - Low level rc	2476 2477 2478	2479	2480	2481	2482	2483	2484	2485	2486	2487	2488	2489	2490	2491	2492	2493	2494	2495	2496	2497	2498	2499	2500	2501	2502	2503	2504	2505	2506

PAGE 36-2	;Get character code in C ;Restore coordinate	address out of H,L (column,row) in HL H H H H H
01-Jan-85	(SP), HL A, (VDP, DRW) C, A HL HL VADDR SETWRT A, C (VDP, DRW), A HL	iffer address out rned in 旺 BC B,H 日,田 田,田 田,田 B,H B,H 田,田
3.44	EX IN LD POP POP RET PUTVRM: RET CALL CALL CALL CALL CALL CALL CALL RET	VADDR: Calculate buffer addre ; Calculate buffer addre ; address returned in HL ; PUSH BC LD E,H LD E,H LD H,0 HL,HL ADD HL,HL ADD HL,HL
BIOS) Macro-80 evel routines	E3 DB 98 4F 4F E1 C9 C9 C0 0BF2 C0 07DF 79 C9 C1 D3 98 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9	0
1 BASIC BIOS - Low level	0 BE0 0 BE1 0 BE1 0 BE3 0 BE4 0 BE6 0 BE6 0 BE6 0 BE7 0 BE7 0 BE7 0 BE7 0 BE7 0 BE7 0 BF1	0BF1 0BF2 0BF2 0BF2 0BF4 0BF4 0BF7 0BF7 0BF8 0BF8 0BF7 0BF8 0BF7 0BF7 0BF7 0BF7 0BF7 0BF7
(MSX ROM - MSXIO -	2507 2508 2510 2511 2511 2513 2514 2515 2515 2515 2519 2519 2519	2520 2521 2522 2523 2523 2523 2523 2533 2533

5 PAGE 36–3		()											ds)							line-terminator-table and affect flags			ss of corresponding terminator byte.	
01-Jan-85	HL, HL HL, DE A, (SCRMOD)	A A, (LINLEN Z, VADDRl	VADDR2	1	HL, BC 41+1			А		E,A	HL, DE	DE, HL	HL, (NAMBAS	HL, DE	HL	BC						the	the address	affected.
5-80 3.44	ADD ADD LD	AND LD JR	SUB JR	; ;	ADD SUB	VADDR2:	CPL	AND	RRA	ΓD	ADD	EX	LD	ADD	DEC	POP	RET	GETTRM:	••	; Get value of	•-	Ч	it: DE h	; Z TLAG IS AI
: BIOS) Macro-80 level routines	29 19 3A FCAF		D6 22 18 03		09 D6 2A		2F	A7	lF	5F	19	EB	2A F922	19	2B	cı	C9							
BASIC LOW 1	0BFE 0BFF 0C00	0C03 0C04 0C07	0C09 0C0B	UCUD	0C0D 0C0E	0C10	0C10	0C11	0C12	0C13	0C14	0C15	0C16	0C19	0C1A	0C1B	0C1C	0C1D						
(MSX ROM - MSXIO -	2538 2539 2540	2541 2542 2543	2544 2545	2547	2548 2549	2550	2551	2552	2553	2554	2555	2556	2557	2558	2559	2560	2561	2562	2563	2564	2565	2566	2567	2568

PAGE 36-4	;Save HL		;Move address to DE ;Restore HL ;Affect flags		;Load non 0 value in Acc	;Get address of terminator byte in	;Change table	leen	;0 or -1
01-Jan-85	HL DE, BASROM	н, 0 НL, DE А, (HL) DF HT	DE, HL HL A		3EH A	AF GETTRM AF	(DE),A	Get an actual height of screen	A, (CNSDFG) HL HL, CRTCNT A, (HL) HL
3.44	UT HSNA		EX POP AND RET		DB XOR	PUSH CALL POP	LD RET	n actua]	LD PUSH LD ADD POP RET
80	••			TERMIN: ;	UNTERM: SETTRM:		GETLEN:	; ; Get a ;	
BIOS) Macro-80 evel routines	E5 11 FBB1 26 00	20 00 19 7E	El A7 C9	ŗ	ЗЕ АF	F5 CD 0ClD F1	12 C9		3A F3DE E5 21 F3B1 86 E1 C9
MSX ROM BASIC BIOS MSXIO - Low level	0C1D 0C1E	0C21 0C23 0C24	0C26 0C26 0C27 0C28	0C29	0C29 0C2A 0C2A 0C2B	0C2B 0C2C 0C2F	0C30 0C31 0C32		0C32 0C35 0C36 0C36 0C38 0C38
(MSX ROM - MSXIO -	2569 2570 2571	2573 2574 2574	2576 2577 2578 2578	2579 2580 2581	2581 2582 2583 2584	2585 2586 2587	2588 2589 2590	с 592 2593 2593	2594 2595 2596 2597 2598 2599

DE

36-5		
PAGE		
01-Jan-85		
3.44		
(MSX ROM BASIC BIOS) Macro-80	MSXIO - Low level routines	
BIOS	evel	
BASIC	Low l	
ROM	। ०	
MSX]	MSXI	
_	T	

	Keyboard encoding routines
	SUBTTL - MSXIO -
2600	2601

PAGE 37					-	;Save all registers												TO allow other interrunts than four times		Interrupt requested by VDP2	:No. skip the rest	To allow timer interrupt to be	;used elsewhere.	;Now that it became obvious that VDP	generated the interrupt, we re-enable	interrupt here to allow RS232C's	surcertupe of something like that.	;Collision detected?
01-Jan-85		ard		upt routine		用	DE	BC	AF		AF, AF'	Н	DE	BC	AF	IY	IX	H.KEYI	A, (VDP.SR)	A	P, INTRET	H.TIMI					(STATFL), A	-
ro-80 3.44 g routines	KEYINT: :	; Encode keyboard	•	; Timer interrupt routine	••	HSUG	HSUG	PUSH	HSUG	EXX	EX	HSUG	PUSH	HSUG	HSUG	HSUG	HSUG	CALL	NI	AND	ЛР	CALL		EI			LD	AND
s) Mac encodin						ES	D5	C5	F5	D9	08	ES	D5	C5	F5	FD E5	DD E5	CD FD9A	DB 99	A7		CD FD9F		FB			32 F3E7	E6 20
1 BASIC BIOS • Keyboard	0C3C					0C3C	0C3D	0C3E	0C3F	0C40	0C41	0C42	0C43	0C44	0C45	0C46	0C48	0C4A	0C4D	0C4F	0C50	0C53		0C56			0C57	0C 5A
(MSX ROM - MSXIO -	2602 2603 2604	2605	2606	2607	2608	2609	2610	2611	2612	2613	2614	2615	2616	2617	2618	2619	2620	2621	2622	2623	2624	2625	2626	2627	2629 2629	2630	2631	2632

PAGE 37-1	;Assume so .Permest tran if so	+				;Count down interval count				;Not yet reached 0	;Request trap		;Load initial value		;Update interval count										;Check music flag		;Start with queue 0		;C7=carry, carry=C0, [C]=[C]/2	; Save queue ID	;Save MUSICF
01-Jan-85	HL, TRPTBL+33 N7 DECTED	N 5 / NEQ INF		aı trap		HL, (INTCNT)	НĹ	А,Н	Г	NZ , NTINTT	HL,TRPTBL+3*17	REQTRP	HL, (INTVAL)		(INTCNT), HL		ffy count		HL, (JIFFY)	HL	(JIFFY), HL		gueue		A, (MUSICF)	C,A	А		C	AF	BC
3.44	LD	TTRO		Check interval trap		ГD	DEC	ГD	OR	JR	ГD	CALL	LD	гт:	ГD		Increment jiffy count		ГD	INC	ГD		; Check music queue		ГD	LD	XOR	NT:	RR	HSUG	PUSH
Macro-80 oding routines				: Che	••									TTNITU		•~	; Inc	••				••	; Che	••				*TNISUM			
OS) d enco	21 FC6D	C4 OFFT				2A FCA2	2B	7C	B5	20 09	21 FC7F	CD 0EF1	2A FCA0		22 FCA2				2A FC9E	23	22 FC9E				3A FB3F	4F	AF		CB 19	F5	C5
M BASIC BI - Keyboar	0050	ACON.				0C62	0065	0066	0C67	0C68	0C6A	0C6D	0C70	0C73	0C73				0C76	0C79	0C7A				0C7D	0C80	0C81	0C82	0C82	0C84	0C85
(MSX ROM - MSXIO -	2633	2034	2635	2636	2637	2638	2639	2640	2641	2642	2643	2644	2645	2646	2647	2648	2649	2650	2651	2652	2653	2654	2655	2656	2657	2658	2659	2660	2661		2663

PAGE 37-2				;Next queue	; All done?	;Not yet		;Need to scan?	;No, return soon	;Time delay of first repeat	1	joy sticks			;Read joystick A															; Save this	
01-Jan-85	C, ACTION	BC	AF	A	ε	C, MUSINT	HL, SCNCNT	(HL)	NZ , INTRET	(旺),3		Check trigger button of joy		A	SLSTCK	0011000B	AF	A,1	SLSTCK	.0.			BC	В	AF	GTROW8	1	BC	B	C,A	HL , TRGFLG
3.44	CALL	POP	POP	INC	CP	JR	LD	DEC	Яŗ	ГD		k trigge:		XOR	CALL	AND	HSUG	LD	CALL	AND	RLCA	RLCA	POP	OR	PUSH	CALL	AND	POP	OR	LD	LD
BASIC BIOS) Macro-80 Keyboard encoding routines	DC 113B	c1	Fl	3C	FE 03	38 F2	21 F3F6	35	20 6C	36 03	••	; Chec	••	AF	CD 120C	E6 30	F5	3E 01	CD 120C	E6 30	07	07	cl	BO	F5	CD 1226	E6 01	C1	BO	4 F	21 F3E8
щ	0C86	0C89	0C8A	0C8B	0C8C	0C8E	0C90	0C93	0C94	0C96				0C98	0C99	0C9C	0C9E	0C9F	0CA1	0CA4	0CA6	0CA7	0CA8	0CA9	0CAA	0CAB	0CAE	0CB0	0CB1	0CB2	0CB3
(MSX ROM - MSXIO -	2664	2665	2666	2667	2668	2669	2670	2671	2672	2673	2674	2675	2676	2677	2678	2679	2680	2681	2682	2683	2684	2685	2686	2687	2688	2689	2690	2691	2692	2693	σ

PAGE 37-3 102	;Any transition? ;Is this transition negative ;Update trigger status ;Check trigger 4 ;Check trigger 2 ;Check trigger 3 ;Check trigger 1 ;Check trigger 1	<pre>;Enable first key click ;Detect valid key transition and check buffer ;Some characters still remain, don't repeat ;Need to enter repeat mode ;No ;Set short time repeat ;Clear OLDKEY status</pre>
01-Jan-85	(HL) (HL) (HL),C (HL),C C,A HL,TRPTBL+3*12 C,REQTRP C HL,TRPTBL+3*16 C,REQTRP C HL,TRPTBL+3*14 C,REQTRP C HL,TRPTBL+3*15 C HL,TRPTBL+3*15 C HL,TRPTBL+3*13 C HL,TRPTBL+3*13 C HL,TRPTBL+3*13	
3.44	XOR AND LD LD LD RRCA RL LD CALL LD RL LD CALL LD LD LD LD LD LD LD LD	CALL CALL Scan keyboard XOR XOR LD JR LD JR LD LD LD LD LD
BASIC BIOS) Macro-80 Keyboard encoding routines	0CB6 AE 0CB7 A6 0CB8 71 0CB9 4F 0CB9 4F 0CB3 0F 0CB4 0F 0CB5 0F 0CB4 0F 0CB5 0F 0CB1 DC 0CC1 CB 0CC3 21 0CC3 21 0CC4 DC 0CC5 DC 0CC6 DC 0CC7 21 0CC6 DC 0C01 CB 0CC6 DC 0C03 21 0C03 21 0C04 DC 0C05 DC 0C05 DC 0C06 DC 0C09 CC 0CD9 CB 0CD9 CB 0CD9 CB 0CD9 CB 0CD9 CB 0CD9	DC 0EF1 AF AF 32 FBD9 CD 0D12 20 18 21 F3F7 35 20 12 36 01 21 FBDA
(MSX ROM - MSXIO -	2695 2695 2696 2699 2700 2703 2709 2709 2709 2710 2710 2711 2711	2713 2714 2715 2715 2716 2719 2721 2721 2723 2723 2723 2723

103			
PAGE 37-4	;Check if currently pressed key is valid ;Restore all registers	;Get what is currently output to Port C ;Leave higher 4 bits unaffected	Move current key status to NEWKEY Select row Get column information of selected row
01-Jan-85	DE, OLDKEY+1 BC, OAH (HL), OFFH KEYCK4 IX IX AF BC DE HL AF, AF' BC BC BC BC DE HL HL	A, (PPI CR) 0F0H C,A B,0BH	HL, NEWKEY A, C (PPI.CW), A A, (PPI.BR)
3.44	LD LD LD LDIR CALL POP POP POP POP POP POP POP POP RET RET	IN AND LD	LLD LLD IN
-80 routines	INTRET: KEYCHK:		KEYCK1:
BIOS) Macro-80 ard encoding rout	11 FBDB 01 000A 36 FF ED B0 ED B0 CD 0D4E FD E1 F1 C1 C1 C1 C1 C1 C1 C1 C1 C1 C1 C1 C1 C1	DB AA E6 F0 4F 06 0B	
BASIC BIOS Keyboard	0CF5 0CF8 0CFB 0CFB 0CFD 0D02 0D02 0D04 0D04 0D06 0D08 0D08 0D06 0D06 0D06 0D07 0D06 0D07 0D01 0D11 0D12	0D12 0D14 0D16 0D17	0019 001C 001C 001D 001F
(MSX ROM BASIC BIOS) - MSXIO - Keyboard en	2726 2728 2729 2729 2729 2739 2739 27440 27440 27440 27440 27440 27440 27440 27440 27440 27440 27440 27440 27440 27440 27440 27440 27733 27732 277732 277732 277732 277772 27772 27772 27772 27772 27772 27772 277772 277772 277772 2777777	2748 2749 2750 2751	2752 2753 2754 2755 2755

PAGE 37-5	;Move it	;Select next row		;Loop until all rows are sensed	;Warm start enabled?		; No	;Get current status of the 6th row	;Check if KANA, GRAPH, CTRL and SHIFT	;are pressed simultaneously					;[OLDKEY] + 11					;Get OLDKEY status	;Compare with NEWKEY status	;Changed, set long repeat interval		; No change						;Set number of rows	
01-Jan-85	(HL),A	C	HL	KEYCKI	A, (ENSTOP)	А	Z, NOSTOP	A, (SFTKEY)	0E8H	NZ, NOSTOP	IX , READYR	CALBAS			DE, NEWKEY	B,0BH		DE	HL	A, (DE)	(王)	NZ, KEYCK3	KEYCK2	KEYCK4			A,0DH	(REPCNT), A		B,0BH	HL , OLDKEY
acro-80 3.44 ing routines	LD	INC	INC	DJNZ	ΓD	AND	JR	LD	CP	К	ГD	JP	NOSTOP:	••	LD	ΓD	KEYCK2:	DEC	DEC	ΓD	СЪ	ЯĻ	DJNZ	JR	KEYCK3:	••	ГD	LD	KEYCK4:	LD	ГD
) M en cod	77	00	23	10 F6	3A FBBO	A7	28 OE	3A FBEB	FE E8	20 07	DD 21 409B	C3 OIFF			11 FBE5			lB	2B	lA	BE	20 04	10 F8				3E 0D	32 F3F7			21 FBDA
MSX ROM BASIC BIOS MSXIO - Keyboard	0D21	0D22	0D23	0D24	0D26	0D2 9	0D2A	0D2C	0D2F	0D31	0D33	0D37	0D3A		0D3A	0D3D	0D3F	0D3F	0D40	0D41	0D42	0D43	0D45	0D47	0049		0D49	0D4B	0D4E	0D4E	0D50
(MSX ROI - MSXIO	2757	2758	2759	2760	2761	2762	2763	2764	2765	2766	2767	2768	2769	2770	2771	2772	2773	2774	2775	2776	2777	2778	2779	2780	2781	2782	2783	2784	2785	2786	2787

Ч					ជ្គ																									
PAGE 37-6		;Get current key status		ıgeö		old status	;Active transition, go find it						not		;Load GETPNT	;Load lower 8 bit of PUTPNT	;Check if same				;Make sure interrupts are enabled	; Save environments			;Are we in text mode?	;No, do not flip function keys		;Get current function key display		;Function key displayed at all?
01-Jan-85	DE, NEWKEY	A, (DE)	C,A	(HT)	(HT)	(旧),C	NZ, KEYANY	DE	HL	KEYCK5			; Check if buffer is empty or not		HL, (GETPNT)	A, (PUTPNT)	Г					НĹ	DE	BC	CHKSCR	NC, CHSNSI	A, (FNKSWI)	HL, SFTKEY	(HT)	HL , CNSDFG
3.44	LD	ГD	LD	XOR	AND	LD	CALL	INC	INC	ZNLC			t if buf		LD	ГD	SUB	RET			EI	HSUG	HSUG	HSUG	CALL	ĥ	LD	ГD	XOR	ГD
ro-80 routines	KEYCK5:										CHKBUF:	••	; Check	••					CHSNS:											
) Macr ncoding	11 FBE5	IA	4F	AE	A6	71	C4 0D89	13	23	10 F4					2A F3FA	3A F3F8	95	C9			FB	ES	D5	C5	CD 0B9F	30 OF	3A FBCD	21 FBEB	AE	21 F3DE
(MSX ROM BASIC BIOS MSXIO - Keyboard en	0D53 0D56	0D56	0D57	0D58	0D59	0D5A	0D5B	0D5E	0D5F	0D60	0D62				0D62	0D65	0D68	0D69	0D6A		0D6A	0D6B	0D6C	0D6D	0D6E	0D71	0D73	0D76	0D79	0D7A
(MSX R - MSXIO	2788 2789	56	7	2792	2793	\sim	2795	2796	2797	2798	2799	2800	2801	2802	2803	2804	2805	2806	2807	2808	2809	2810	2811	2812	2813	2814	2815	2816	2817	2818

PAGE 37-7		Update display	:Restore environments								Save environmente			:Save nressed hit		falmilate base ando					:Set up counter for 8 hit	, c					If pressed hit coll boundary	TT FLESSEN DIL, CALL KEY CODEL.
01-Jan-85	(IHI)		BC	DE	HL				SUBROUTINE 'KEYANY']]]		HL	DE	BC	AF	A,0BH	E	A, A	A, A	A, A	C,A	B,8	AF			BC	AF	C.KEYCOD	AF
-80 3.44 routines	AND RRCA CALL	CHSNS1:	POP	POP	POP	RET	KEYANY:		; [[[SUBROUTI		HSUA	PUSH	PUSH	HSUG	ΓD	SUB	ADD	ADD	ADD	LD	ΓD	POP	KYANY1:	RRA	PUSH	HSUT	CALL	POP
BIOS) Macro-80 ard encoding routines	A6 0F DC 0B2B		cl cci	Dl	El	C9					E5	D5	C5	F5	3E 0B	90	87	87	87	4F	06 08	Fl		lF	C5	F5	DC 0E3B	Fl
OM BASIC - Keybo	0D7D 0D7E 0D7F	0D82 0D82	0085	0D86	0D87	0D88	0D89				0D89	0D8A	0D8B	0D8C	0D8D	0D8F	0600	1600	0D92	0D93	0D94	0096	0097	0D97	0D98	0D99	0D9A	0000
(MSX R(- MSXIO	2819 2820 2821	000	i m	òn	à.	2	2	2	~	~	~	22	<u> </u>	<u>(7)</u>		c	<u>m</u>	\sim	4	4	4	4	4	4	4	4	-	

	are checked																								
37-8	Try next code Loop until all bits are checked Restore erv ironments		if valid											;Capital lock		lock		;Function key				key			
PAGE	;Try r ;Loop ;Resto	KEYCOD']]	in buffer if valid			;09				;AZ				;Capit		;Kana lock		; Funct				; Stop key			
01-Jan-85	BC C KYANYI PBDHRT	SUBROUTINE 'KEYCOD']	Return key-code		10	KYNUM	22	KYCODI	48	KYALP	51	KYEASY	52	KYLOCK	53	KYKLOK	58	KYFUNC	60	KYEASY	61	KYSTOP	65	KYEASY	66
3.44	POP INC DJNZ JP]]]	Ret		DB	DW	DB	ΜQ	DB	DW	DB	DW	DB	DW	DB	ΜQ	DB	DW	DB	DW	DB	DW	DB	DW	DB
DS) Macro-80 1 encoding routines	C1 0C 10 F5 C3 08DB		~ ~	; KV.TMAR•		0E67	16	DEAL	30	0E7E	33	0F10	34	0F36	35	OFLF	3A	OEBB	30	0F10	3D	0F46	41	0F10	42
BASIC BIOS) Keyboard end	0D9E 0009F 0000A0 1000A0 1000A0 1000A0 1000A0 1000A0 1000A0 1000A0 1000A0 1000A0 1000A2 000A2 000A2 000A2 000A2			20025			0DA8	0DA9 (0DAB 0	0DAC (0DAE	ODAF (0DB1	0DB2 (0DB5 (0DB7	0DB8 (0DBA	0DBB (0DBD	0DBE (0DC0	0DC1 (0DC3
(MSX ROM - MSXIO -	2850 2851 2852 2853 2853	2854 2855	2856 2857	858 859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	2880

PAGE 37-9	;CLS/HOME key	;Double quote	; CTRL+sh i ft ; CTRL ; SHIFT ;	; CTRL+SHIFT ; CTRL ; SHIFT ;	;Less than sign
01-Jan-85	KYCLS 255 KYEASY 255		PUTCHR PUTCHR KEYSFT KEYNOM	KYl SFC-10 KYl CNT-10 KYl SFT-10 KYl NOM-10 ''-^\@[;:],./"	255 "=~ `{+*}" 00111100B
0 3.44 utines	DW DB DW ; NMSFTB: DB		ALPJMP: DW DW jW ; KYClTB:	DW DW DW DW KYI NOM: DB	DB KYLSFT: DB DB
BASIC BIOS) Macro-80 Keyboard encoding routines	OFO6 FF OF10 FF	21 22 23 24 25 26 27 28 29	0F55 0F55 0E93 0E95	0DFD 0DF1 0DE5 0DD9 2D 5E 5C 40 5B 3B 3A 5D 2C 2E 2F	FF 3D 7E 7C 60 7B 2B 2A 7D 3C
щ	0DC4 0DC6 0DC7 0DC7 0DC9 0DC9	0DCB 0DCB 0DCC 0DD0	0DD3 0DD5 0DD5 0DD7 0DD9 0DD9	000B 000D 000F 00E1 00E3 00E3 00E3 00EB	0DEE 0DEF 0DEF 0DF3 0DF3 0DF7
(MSX ROM - MSXIO -	2881 2882 2883 2884 2885 2885 2885	2887 2888 2889 2891 2891	2892 2893 2895 2895 2897 2898 2898	2899 2900 2901 2903 2903 2905 2905	2907 2908 2910 2911

0	n sign																ign	than sign					(48)	(49)	(20)	(21)	(22)
PAGE 37-10	;Greater than																Less than sign:	Greater that					; Shift	; Control	;Graph	;Cap lock	;Kana lock
đ	,																	ī.					••	ī.	ī.	ī.	
01-Jan-85	00111110B "?_"	n_n "∕n_n@"	n⊖n_n∕ n			"]"-"@"	",./"	255		u == =	"6"-"∧"		"0"-"0"	"6"-"]"		" ິງ"–າຄູ	00111100B	00111110B	= Č =	"	l		0	0	0	0	0
3.44	DB DB	DB DB	DB	DB	DB	DB	DB	DB		DB	DB	DB	DB	DB	DB	DB	DB	DB	DB	DB			DB	DB	DB	DB	DB
:) Macro-80 encoding routines	KYLCNT								KY1 SFC:													EASYTB					
encodir	5 F	0.53			3 3A		2E 2F	6-		~	61	•	_	~	3 2A	~	•	~	6.	6			_	_	_	_	_
BASIC BIOS Keyboard e	3E 3F	2D 1E	IC	1B 1B	3B	lD	2C	FF		3D	LΕ	IC	00	lb	2B	ID	30	3E	3F	ΓF			00	00	00	00	00
	0DF8 0DF9 0DFB	0DFB 0DFC	ODFD	ODFF	0E00	0E02	0E03	0E06	0E07	0E07	0E08	0E09	0E0A	0E0B	0E0C	0E0E	OEOF	0E10	0Ell	0E12		0E13	0E13	0E14	0E15	0E16	0E17
(MSX ROM - MSXIO -	2912 2913 2914	2915 2916	2917	2919 2919	2920	2921	2922	2923	2924	2925	2926	2927	2928	2929	2930	2931	2932	2933	2934	2935	2936	2937	2938	2939	2940	2941	2942

(MSX ROM BASIC BIOS) M - MSYIO - Keuhoard encod	;) Macro-80 3.44 01-Jan-85 encoding routines	-85 PAGE 37-11
00	DB 0	; F1
00	DB 0	;F2
00	DB 0	; F3
00	DB 0	; F4
00	DB 0	; F5
lB	DB 27	; Es ca pe
60	DB 9	; Tab
00	DB 0	; Stop
08	DB 8	; Back space
18	DB "X"-"@"	" ;Select
0D	DB 13	; En ter
20	DB 32	; Space
00	DB 12	;Clear
12	DB "R"-"@"	"; Insert
7 F	DB 127	; Rubout
lD	DB 29	;Left
le	DB 30	dn :
lf	DB 31	; Down
lC	DB 28	;Right
ĸ		
••	For additic	additional key matrix
ĸ		
01	DB "A"-"@	••
04	DB "D"-"@	
OF	DB "O"-"ê"	••
10	DB "P"-"8"	
11		
12		
13		
14		•
00		
	•	

	(81)	(82)	(83)	(84)	(85)	(86)	(87)
37-12							
PAGE			••		••	••	••
01-Jan-85	c	0	0	0	0	0	0
3.44	DB						
MSX ROM BASIC BIOS) Macro-80 MSXIO - Keyboard encoding routines	4 00		5 00		_		V 00
BASI Key	0E34	0E35	0E36	0E37	0E38	0E39	0E3A
(MSX ROM - MSXIO -	2974	2975	2976	2977	2978	2979	2980

PAGE 38			if valid	;Get raw code	;Just for fail safe				; Possibly a KANA or graphic character	; No	Get shift key status	;Control pressed?	I	;Yes, this supersedes everything	;How about graphic shift	; Yes, this has the 2nd priority	;KANA lock active		;Yes				;Compare range	•	;Get jump address in [DE]	
01-Jan-85		[[[SUBROUTINE 'KEYCOD']]]	Return key-code in buffer if valid	A,C	OFFH	2	HL , KY JTAB	H.KEYC	48	NC, KYCLAS	A, (SFTKEY)			NC, KYCLA0		NC , KYGRAP	A, (KANAST)	A	NZ, KYKANA		A,C		(田)	HL	E, (HL)	НГ
acro-80 3.44 ing routines	; КЕҮСОD:	; [[[SUBROUT] :	; Return key-c	, LD	CP	RET	ГD	CALL	CP	JR	LD	RRCA	RRCA	ЯĻ	RRCA	ЧĻ	LD	AND	JP	KYCLA0:	LD	KYCLAS:	CP	INC	ΓD	INC
) M encod				79	FE FF	C8	21 0DA5	CD FDCC	FE 30	30 13	3A FBEB	OF	OF	30 0B	OF	D2 107D	3A FCAC	A7	C2 0F83		79		BE	23	5E	23
MSX ROM BASIC BIOS MSXIO - Keyboard	0E3B			0E3B	0E3C	0E3E	0E3F	0E42	0E45	0E47	0E49	0E4C	0E4D	0E4E	0E50	0E51	0E54	0E57	0E58	0E5B	0E5B	OESC	0E5C	0E5D	OE5E	0E5F
(MSX RO - MSXIO	2981 2982 2983 2984	2985 2986 2986	2987	2989 2989	2990	2991	2992	2993	2994	2995	2996	2997	2998	2999	3000	3001	3002	3003	3004	3005	3006	3007	3008	3009	3010	3011

PAGE 38-1	<pre>;Assume matched ;Good assumption ;Discard stack ;Check next possibility ;Assume no shift ;Save code ;Check shift status ;Restore code ;Check shift status ;Restore code ;Good assumption ;Good assumption ;This must not be 'DADF' ;Get code for shift-number ;Shift '0'? ;Yes, ignore this ;Put this in buffer</pre>
01-Jan-85	D, (HL) HL DE C C C KYCLAS A, "0' B, A A, (SFTKEY) A, (SFTKEY) A, (SFTKEY) A, (HL) 0FFH B, 0 HL, MSFTB HL, BC A, (HL) 0FFH B, 0 2 PUTCHR PUTCHR PUTCHR HL, ALPJMP HL, DE
3.44	LD LD PUSH RET POP PUSH PUSH POP PUSH JR LD JR LD JR LD JR LD JR LD JR LD ADD ADD ADD ADD ADD LD ADD LD ADD LD ADD LD ADD JR JR ADD JR JR JR JR JR JR JR JR JR JR JR JR JR
2-80 routines	KYNUM: ; JPUTCH: ; ;
IC BIOS) Macro-80 yboard encoding routines	56 23 23 05 08 08 18 75 47 47 47 78 06 00 09 09 09 09 09 09 21 00 21 00 21 00 21 00 21 00 21 00 21 00 21 00 21 00 21 00 21 00 21 00 21 00 21 00 21 21 00 21 21 00 21 21 00 21 21 21 20 21 20 21 20 21 20 20 20 20 20 20 20 20 20 20 20 20 20
BASIC	0E60 0E61 0E63 0E63 0E64 0E64 0E65 0E67 0E67 0E67 0E73 0E73 0E73 0E73 0E73 0E73 0E73 0E7
(MSX ROM - MSXIO -	3012 3013 3015 3015 3015 3016 3015 3019 3024 3028 3024 3028 3024 3028 3024 3028 3023 3023 3033 3033 3033 3033 3034 3034

PAGE 38-2	;Get code ;Make it a control character (1 - 26)	; Save code	;Bit 5 is on if CAP lock not active	;Extract shift and control status
01-Jan-85	A, (旺) 旺 H, (旺) L, A A, C 15H (旺)	A,''' B,A A,(CAPST)	00100000B B A,01000000B JPUTCH HL,KYCITB A,(SFTKEY)	3 A,A E,A D,0 HL,DE A,(HL) HL H,(HL) L,A
-80 3.44 coutines	LD INC LD LD LD SUB XEYSFT:	; ADD KEYNOM: LD LD CPL	AND XOR XOR ADD JR KYCOD1: ; LD	ADD ADD LD LD LD LD LD LD LD LD LD
BIOS) Macro-80 bard encoding routines	7E 23 66 67 79 E9 E9 E9	C6 20 47 3A FCAB 2F	E6 20 A8 C6 40 18 DA 21 0DDB 3A FRFR	E6 03 5F 16 00 19 23 23 6F
I BASIC BIO · Keyboard	0E8B 0E8C 0E8D 0E8E 0E8F 0E8F 0E90 0E92 0E92	0E93 0E95 0E95 0E96	0E9A 0E9A 0E9F 0E9I 0EAI 0EAI	0EA7 0EA7 0EA8 0EA8 0EA8 0EA8 0EA8 0EA8 0E80 0E80
(MSX ROM BASIC BIOS - MSXIO - Keyboard	3043 3044 3045 3046 3047 3048 3050	3051 3052 3053 3053 3055 3055	3059 3059 3059 3060 3062 3063	3065 3065 3066 3068 3070 3071 3073

PAGE 38-3					;Should generate some code?	;Yes	;No code should be generated					; Is shift pressed?		; No					;[DE] is (5665)		; Check if this function key is an event device				;Request trap if not in direct mode							
01-Jan-85	E.C	HI DE		A, (HL)	OFFH	NZ, PUTCHR				S		A, (SFTKEY)		C, KYFNCI	A,C	A,5	C,A		E,C	D,0	HL, FNKFLG-53	HL, DE	A, (HL)	А	NZ , FNKINT		DE, HL	Ш, Ш	нг, нг	нг, нг	нг, нг.	DE_FNKSTR-53*16
3.44	L,D		TUR .	ГD	CP	JP	RET	Ü		Function keys		ГD	RRCA	JR	LD	ADD	ГD	••	ГD	LD	ГD	ADD	ГD	AND	JR	2:	ЕX	ADD	ADD	ADD	ADD	U'I
BIOS) Macro-80 bard encoding routines	59		T	7E		C2 0F55	C9	KYFUNC:	••	; Fund		3A FBEB	OF	38 04	79	C6 05	4F	KYFNC1	59	16 00	21 FB99	19	7E	A7	20 13	KYFNC2	EB	29	29	29	29	11 F52F
I BASIC BIOS Keyboard	OEB2	0502	Caru	0EB4	0EB5	0EB7	0 EBA	0 EBB				0EBB	0 EBE	OEBF	0EC1	0EC2	0EC4	0EC5	0EC5	0EC6	0EC8	0ECB	0ECC	OECD	0 ECE	0ED0	0ED0	0ED1	0ED2	0ED3	0ED4	ORD5
(MSX ROM BASIC BIOS) - MSXIO - Keyboard end	3074	1000	C/ NS	3076	3077	3078	3079	3080	3081	3082	3083	3084	3085	3086	3087	3088	3089	3090	3091	3092	3093	3094	3095	3096	3097	3098	3099	3100	3101	3102	3103	3104

PAGE 38-4	;Get function key string address	; Move address to DE		;Get from function key string	; End of string	;Yes	;Put this character in buffer	;Check next character				; Are we in direct mode (CURLIN=65535)				;Yes, treat as normal function key				
01-Jan-85	HL , DE	DE, HL		A, (DE)	А	2	PUTCHR	DE	KYFNC3			HL, (CURLIN)	HL	А,Н	ц	Z, KYFNC2	HL, TRPTBL-53*3	HL, DE	HL, DE	HL, DE
3.44 cines	ADD	EX	KYFNC3:	LD	AND	RET	CALL	INC	JR	FNKINT:		LD	INC	LD	OR	JR	LD	ADD	ADD	ADD
(MSX ROM BASIC BIOS) Macro-80 - MSXIO - Keyboard encoding routines	19	EB	Я	IA	A7	C8	CD 0F55	13	18 F7	н		2A F41C	23	7C	B5	28 E5	21 FBAD	19	19	19
l BASIC Keybo	0ED8	0ED9	OEDA	0 EDA	0EDB	0 EDC	0EDD	0EEO	0 EE I	0EE3		0 E E 3	0 EE6	0 EE 7	0EE8	0EE9	0 E E B	OEEE	OEEF	0 EFO
(MSX ROM - MSXIO -	3105	3106	3107	3108	3109	3110	3111	3112	3113	3114	3115	3116	3117	3118	3119	3120	3121	3122	3123	3124

PAGE 39			Request trap (called to request trap for event devices)		Since REQTRP is mostly called from within an interrupt routine.	interrupt mask through DI or EI.		;Trap on?	; TRAP NOT ON		;Trap request	1	; No change		:Trap on + Trap request								Set carry if shift not pressed		;Load code for CLS		
01-Jan-85			(called to requ		is mostly calle	the interrupt me	A, (HL)	1	Z	A, (HL)	4	(HL)	Z	(HT),A	5	NZ	A, (ONGSBF)	A	(ONGSBF), A				A, (SFTKEY)		A,0CH	A,0	PUTCHR
2-80 3.44 routines		REQTRP: ;	; Request trap	• •	; Since REQTRP	; don't touch the	ΓD	AND	RET	LD	OR	CP	RET	LD	XOR	RET	LD	INC	ΓD	RET		KYCLS:	LD	RRCA	ΓD	SBC	JR
BIOS) Macro-80 ard encoding routines							7E	E6 01	C8	7 E	F6 04	BE	C8	77	EE 05	C0	3A FBD8	3C	32 FBD8	C9			3A FBEB	0F	3E 0C		18 45
(MSX ROM BASIC BIOS - MSXIO - Keyboard		T.430					OEFI	0 EF2	0 EF4	OEF5	0 EF6	0EF8	0EF9	OEFA	OEFB	OEFD	OEFE	0F01	0F02	0F05		0F06	0F06	0F09	0F0A	OFOC	OFOE
(MSX R(- MSXIO	3125 3126 3126	312/ 3128	3129 3130	3131	3132	3133 3134	3135	3136	3137	3138	3139	3140	3141	3142	3143	3144	3145	3146	3147	3148	3149	3150	3151	3152	15	3154	3155

156 0F10 157 0F10 158 0F10 159 0F10 161 0F13 162 0F14 163 0F14 164 0F13 165 0F14 166 0F13 165 0F14 166 0F15 165 0F16 166 0F17 167 0F18 169 0F17 1170 0F17 1171 0F22 1173 0F17 1170 0F23 1171 0F23 1172 0F17 1173 0F17 1174 0F23 1175 0F24 1176 0F23 1177 0F23 1178 0F23 1179 0F29 1170 0F29 1171 0F29 1172 0F29 1173 0F29 1174 0F29 1175 0F29<
--

PAGE 39-2		;Toggle capital status	;Update capital status	;Assume 'turn off' ;Good assumption ;Change to 'turn on'		;Move CTRL status to carry ;Assume CTRL pressed also ;Good assumption ;CTRL not pressed, just treat as pause
01-Jan-85	A A A	L, CAPST A, (HL)	(НГ.),А	A A, OCH Z, CGCAP1 A	(FFI.CM),A A,(SFTKEY)	A, 3 NC,KYSTP1 A (INTFLG),A
Macro-80 3.44 ding routines	NOKEY: RET KYLOCK: ; : Capital lock kev		CPL LD CHGCAP:	AND LD JR INC CGCAP1:	00T RET KYSTOP: ; STOP key ; LD	RRCA RRCA LD JR INC KYSTP1: LD
) en co	60	21 FCAB 7E 2F	77 2F		U3 AB C9 3A FBEB	0F 0F 3E 03 3C 3C 32 FC9B
M BASIC BIOS - Keyboard	0F35 0F35 0F36	0F36 0F39 0F33	0F3B 0F3C 0F3D	0F3D 0F3E 0F40 0F42 0F43	0F45 0F45 0F46 0F46 0F46 0F46	0F49 0F4A 0F4B 0F4D 0F4F 0F50 0F50 0F50
(MSX ROM - MSXIO -	3187 3188 3189 3191	3192 3193 3194 3195	3196 3197 3198	3199 3201 3201 3203 3203	3204 3205 3206 3207 3208 3209 3210	3211 3212 3213 3214 3214 3215 3215 3215

PAGE 39-3	;Only generate click if pause fer.	<pre>;Load PUTPNT in [HL] ;Save the character to buffer ;Increment PUTPNT</pre>	Compare it with new PUTPNT FIF same skip next step Save HL in PUTPNT	;Key click enabled? ;No ;Already generated?	;Yes, don't click any more ;Set flag to disable more clicks	;Assume 'turn off' ;Good assumption ;Change to 'turn on'
01-Jan-85	JR C,GENCLK ;0 PUTCHR: ; Put one character in key buffer.	HL, (PUTPNT) (HL), A UPDATE	A, NGELFNI) L Z PUTPNT), HL	A, (CLIKSW) A Z A, (CLIKFL) A	NZ A,0FH (CLIKFL),A (PPI.CM),A A,0AH	A NZ,CLICKW A A,OEH Z,CGSND1 A
3.44 nes	JR PUTCHR: ; Put one char:	LD LD CALL	CP LD	GENCLK: LD AND RET LD AND	RET LD LD OUT LD CLICKW:	DEC JR JR AND LLD JR INC
310S) Macro-80 ird encoding routines	38 OF PU		3A F3FA BD C8 22 F3F8	GE 3A F3DB A7 C8 3A FBD9 A7		3D 20 FD A7 3E 0E 28 01 3C
(MSX ROM BASIC BIO - MSXIO - Keyboard	0F53 0F55	0F55 0F58 0F59	0F5F 0F60 0F61	0F64 0F64 0F67 0F68 0F69 0F69	0F6D 0F70 0F73 0F73 0F75 0F77	0F77 0F78 0F7A 0F7A 0F7B 0F7D 0F7D 0F7F
(MSX RO - MSXIO	3218 3219 3220 3221	3222 3224 3224 3225	3226 3227 3228 3229	3230 3231 3232 3233 3233 3234 3234	3236 3237 3238 3238 3240 3241	3242 3243 3244 3244 3246 3246 3248

PAGE 39-4		; KANA key pressed while KANA lock is active	Affect Z flag	;Check shift key :Affect Carry flag							;Assume shift not pressed	; Good assumption					; Push jump address		;Capital lock (katakana) active?			;active	;Special characters?
01-Jan-85	(PPI.CM),A	ssed while KANA	A, (KANAMU) A	A, (SFTKEY)	Z,KAIUEO	HL , KANJNO	C, KYKANI	HL, KANJSF KVEDNI	TARGATIA		HL , KANANO	C, KYKAN1	HL , KANASF		B,0	HL, BC	BC, PUTCHR	BC	A, (CAPST)	A	A, (HL)	NZ	165+1
3.44	OUT RET	tey pre	AND	LD RRCA	JR	LD	JR L				ГD	JR	LD		LD	ADD	ГD	HSUG	LD	AND	LD	RET	CP
:) Macro-80 encoding routines	CGSND1: KYKANA: ;	; KANA ;							KAIUEO:	••				KYKAN1:									
BIOS) Macı ard encodin	D3 AB C9		3A FCAD A7	3A FBEB OF	28 0A		38 0D	21 104D 18 08			21 0FBD	38 03	21 OFED		00 90	60	01 0F55	C5	3A FCAB	A7	7E	C 0	FE A6
l BASIC BIO Keyboard	0F80 0F80 0F82 0F83		0F86	0F87 0F8A	0F8B	0F8D	0F90	0F92 0F95	0F97		0F97	0F9A	0F9C	0F9F	0F9F	OFAl	0FA2	0FA5	0FA6	0FA9	0 F A A	OFAB	OFAC
(MSX ROM BASIC BIOS) - MSXIO - Keyboard en	32 4 9 3250 3251 3252 3253	3254 3255 2255	3257	3258 3259	3260	3261	3262	3263 3764	3265	3266	3267	3268	3269	3270	3271	3272	3273	3274	3275	3276	3277	3278	3279

03.4401-Jan-85PAGE39-5outinesRETC;Yes, no conversion necessaryRETC0B0H;Really first halfRETZ;Really first halfRETC;91-32+1;Really first halfRETC;0191-32+1;fscod asumptionRETC;0191-32+1;fscod asumptionRETC;0191-32+1;fscod asumptionRETC;0191-32+1;fscod asumptionRETC;compensateRETC;compensateRETC;compensateRETC;compensateRETC;compensateRETDB0C9H,0B1H,0B2H,0B3H,0B4H,0B5H,0C5HDB0C9H,0B1H,0B2H,0B3H,0B4H,0B5H,0C5HDB0C9H,0D3H,0DEH,0D7H,0D8H,0D5H,0D5HDB0D9H,0D3H,0DEH,0D7H,0D8H,0D5H,0D5HDB0D9H,0D3H,0DEH,0D7H,0D9H,0D5H,0D5HDB0D9H,0D3H,0D2H,0D9H,0D2H,0D5HDB0D9H,0D3H,0D2H,0D9H,0D5H,0D5HDB0D9H,0D3H,0D2H,0D2H,0D5HDB0D9H,0D3H,0D7H,0C2H,0D9H,0D5HDB0C9H,0C3H,0D3H,0D1H,0D2H,0D5HDB0C9H,0C3H,0D3H,0C1H,0CAH,0C0Hfiftedififtedififtedififtedififtedififtedififtedififtedififtedififtedififtedififtedififtedi <t< th=""></t<>
3.44 01-Jan-85 RET C CP 0B0H RET Z CP 0B0H RET Z CP 0DEH RET NC SUB ' ' CP 191-32+1 RET C ADD A, 32+32 RET C ADD A, 32+32 RET C ADD A, 32+32 RET C ADD A, 32+40 CP 00BH, 0B2H, 0 CP 00BH, 00BH, 0 CP 0DBH, 0D3H, 0DEH, 0 DB 0D0H, 0D3H, 0DEH, 0 DB 0D0H, 0BBH, 0C4H, 0 DB 0D0H, 0C3H, 0C7H, 0 CCH, 0C7H, 0C7H, 0C6H, 0 DB 0D0H, 0C8H, 0C7H, 0C8H, 0 DB 0C6H, 0C7H, 0C8H, 0 DB 0C6H, 0C7H, 0C8H, 0 DB 0C6H, 0C7H, 0C8H, 0 DB 0D0H, 0C8H, 0C7H, 0C8H, 0 DB 0C6H, 0C7H, 0C8H, 0 DB 0D0H, 0C8H, 0C7H, 0C8H, 0 DB 0C6H, 0C7H, 0C8H, 0 DB 0D0H, 0C8H, 0C7H, 0C8H, 0 DB 0D0H, 0C8H, 0C7H, 0C8H, 0 DB 000H, 008H, 0C7H, 0C8H, 0 DB 000H, 0C8H, 0C7H, 0C8H, 0 DB 000H, 008H, 0C7H, 008H, 0 DB 000H, 008H, 0C7H, 0 DB 000H, 008H, 0C7H, 008H, 0 DB 000H, 008H, 007H, 0 DB 000H, 007H, 007H, 0 D1 000H, 007H, 007H, 007H, 0 D1 000H, 000
3.44 RET RET RET RET RET SUB SUB SUB SUB RET ADD RET ADD RET ADD RET ADD RET ADD RET ADD RET SUB SUB SUB SUB SUB SUB SUB SUB SUB SUB
3.44 RET RET RET RET RET SUB SUB RET ADD RET ADD RET DB DB DB DB DB DB DB DB DB DB DB DB DB
utines KANANO ; ; KANASF KANASF
m ()
05) Macro-80 d encoding routines FE B0 C8 FE DE D0 C6 20 FE A0 D6 20 C6 40 C9 B1 B2 B3 B4 B5 C5 C4 C2 C8 D1 B2 C4 D2 D7 D8 D3 D5 D7 D8 D3 D6 D7 C8 D7 C9 B1 B2 C5 C6 C7 C8 D7 C9 B1 C2 C6 C7 C8 D7 D8 D3 D6 D7 C9 C3 C6 C3 C6 C3 C6 C3 C8 C3 C9 C4 C9 C4 C9 C5 C9 C
BI
 (MSX ROM BASIC MSXIO - Keybo 3280 0FAE 3281 0FAE 3281 0FAE 3283 0FB1 3284 0FB4 3284 0FB4 3285 0FB5 3289 0FB5 3291 0FB0 3291 0FB0 3293 0FB0 3293 0FB0 3294 0FB0 3293 0FB0 3303 0FB0 3304 0FB0 3306 0FB0 3309 0FB0 3300 0FB0

01-Jan-85 PAGE 39-6	0С6Н,0С7Н,0С8Н,0D7Н,0D8Н,0D9Н,0DАН	0A2H,0D3H,0B0H,0A3H,0AEH,0A4H,0A1H	0A5H,0BBH,0C4H,0AFH,0BDH,0B8H,0BEH	ОВҒН, ОСҒН, ОССН, ОДОН, ОДІН, ОД2Н, ОАДН	0АСН , 0СDН , 0СЕН , 0В6Н , 0В9Н , 0ВСН , 0ВАН	0CBH,0C3H,0B7H,0C1H,0CAH.0C0H			table JIS order, un-shifted	0DCH,0C7H,0CCH,0B1H,0B3H,0B4H,0B5H	тело того паро про про про про		0DFH,0DAH,0B9H,0D1H,0C8H,0D9H,0D2H		ОДВН,ОСІН,ОВАН,ОВҒН,ОВСН,ОВ2Н,ОСАН	0B7H,0B8H,0C6H,0CFH,0C9H,0D8H,0D3H		0D0H,0D7H,0BEH,0C0H,0BDH,0C4H,0B6H		0С5Н,0СВН,0С3Н,0ВВН,0DDH,0С2Н		
3.44	DB	DB	DB	DB	DB	DB			Kana t	DB	au	2	DB		DB	DB		DB		DB		
) Macro-80 encoding routines								KANJNO:	~ ~												KANJSF:	
Macro-80 ding rou	C5 C8 D7	BO A3 Al	C4 AF BF		CE B6	A 7 Cl				C BI	د م لغ		10 6		A BF	6 CF	~	E CO		3 BB		
) M Icod	AB C C7 C D9 D					BC BA C3 B7	co			_	194 B5 D5 D6				CI BA B2 CA		D8 D3			CB EB EB	Ņ	
	AA 7 C6 0 D8 1					C B C B C B	CA				н Г 197				DB BC BC							
BASIC Keybc	0FF1 0FF4 0FF8	OFFB OFFF	1002	1000	1010	1014 1017	101B	101D		1010	1024	1028	102B	102F	1032 1036	1039	103D	1040	1044	1047	104D	
(MSX ROM - MSXIO -	3311 3312 3313	3314 3315	3316 3317	3319 3319	3320	3321 3322	3323	3324	3325 3326	3327	3329 3329	3330	3331	3332	3333 3334	3335	3336	3337	3338 3228	2320 2340	3341	

3.44 (MSX ROM BASIC BIOS) Macro-80 - MSXIO - Keyboard encoding routines

39-7 PAGE 01-Jan-85

Shifted		з 0A6H, 0C7H, 0CCH, 0A7H, 0A9H, 0AAH, 0ABH		3 0ACH, 0ADH, 0AEH, 0CEH, 0CDH, 0B0H, 0DEH		3 0A2H,0DAH,0B9H,0A3H,0A4H,0A1H,0A5H		3 0DBH,0C1H,0BAH,0BFH,0BCH,0A8H,0CAH		3 0B7H,0B8H,0C6H,0CFH,0C9H,0D8H,0D3H		3 0D0H,0D7H,0BEH,0C0H,0BDH,0C4H,0B6H		3 0C5H,0CBH,0C3H,0BBH,0DDH,0AFH	
; Sh	••	14D A6 C7 CC A7 DB	151 A9 AA AB	154 AC AD AE CE DB	158 CD B0 DE	15B A2 DA B9 A3 DB	15F A4 A1 A5	1062 DB CI BA BF DB	166 BC A8 CA	169 B7 B8 C6 CF DB)6D C9 D8 D3	170 D0 D7 BE C0 DB	174 BD C4 B6	177 C5 CB C3 BB DB	7B DD AF
3342	3343							3350 10					3355 10	3356 10	3357 10

PAGE 40							;Get from graphic key table	;Should generate some code		;1 byte code?		; Assume not	; Was 2 byte code, put header byte					4FH,47H,41H,42H,43H,44H,45H		46H,4DH,4EH,57H,00H,49H,00H		84H,82H,81H,85H,5FH,5DH,80H		83H,00H,5BH,5AH,54H,58H,55H		53H,4AH,56H,00H,00H,5EH,4BH		00Н,00Н,50Н,00Н,52Н,4СН,59Н
01-Jan-85			acters	с С	L, CRPTAB	HL, BC	A, (HL)	А	Z	80H	AF	A,1	C, PUTCHR	AF	PUTCHR			4FH,47H,41H,4		46H,4DH,4EH,5		84H,82H,81H,8		83H,00H,5BH,5		53H,4AH,56H,C		00Н,00Н,50Н,С
0 3.44 utines	; ; ;		; Graphic characters		LD LD	ADD	ΓD	AND	RET	CP	HSU	ГD	CALL	POP	JP		GRPTAB:	DB		DB								
BIOS) Macro-80 ard encoding routines				00 00		60	7E	A7	C8	FE 80	F5	3E 01	DC 0F55	Fl	C3 0F55			4F 47 41 42	43 44 45	46 4D 4E 57	00 49 00	84 82 81 85	5F 5D 80	83 00 5B 5A	54 58 55	53 4A 56 00	00 5E 4B	00 00 20 00
ROM BASIC BIOS) O - Keyboard enc	0701			4501	LOTE	1082	1083	1084	1085	1086	1088	1089	108B	108E	108F		1092	1092	1096	1099	109D	10A0	10A4	10A7	10AB	loae	10B2	10B5
(MSX RC - MSXIO	3358 3359 3360	3361	3362	3363	3365	3366	3367	3368	3369	3370	3371	3372	3373	3374	3375	3376	3377	3378	3379	3380	3381	3382	3383	3384	3385	3386	3387	3388

PAGE 40-1	СН,48Н,00Н							;Check buffer boundary						oard						;Character already there?	;Yes, do not touch cursor	;Display cursor if disabled		;Any character in buffer?	;No, wait	; Erase cursor if disabled		
01-Jan-85	00Н,51Н,00Н,5СН,48Н,00Н			ter		用	A,L	18H	NZ	HL , KEYBUF				racter from keyboard		HL	DE	BC	H.CHGE	CHSNS	NZ, CHGET2	CKDPC0		CHSNS	Z, CHGET1	CKERC0		HL , INTELG
3.44 cines	DB	; 11DNATR.		; Update pointer		INC	ГD	CP	RET	LD	RET	CHGET:		: Get one character		PUSH	HSUI	FUSH	CALL	CALL	ЯŲ	CALL	CHGET1:	CALL	JR	CALL	CHGET2:	LD
ASIC BIOS) Macro-80 Keyboard encoding routines	52 4C 59 00 51 00 5C 48 00	•• -	,			23	7D	FE 18	C0	21 FBF0	60	0				ES	D5	C5	CD FDC2	CD 0D6A	20 0B	CD 09DA	0	CD 0D6A	28 FB	CD 0A27	-	21 FC9B
I BASIC · Keybo	10B9 10BC 10C0	1002	1			10C2	10C3	10C4	10C6	10C7	10CA	10CB				10CB	10CC	10CD	10CE	1001	1004	10D6	10D9	10D9	10DC	10DE	10E1	10E1
(MSX ROM BASIC - MSXIO - Keyb	3389 3390 3391	3392 2302	3394	3395	3396	3397	3398	3399	3400	3401	3402	3403	3404	3405	3406	3407	3408	3409	3410	3411	3412	3413	3414	3415	3416	3417	3418	3419

PAGE 40-2	<pre>;Code for pause? ;No ;Clear this ;Clear this ;Save pressed key ;Update [GETPNT] ;Set new [GETPNT] ;Pass result to Acc ;Pass result to Acc ;To disable CONTinuing</pre>	
01-Jan-85	LD A, (HL) CP 4 JR NZ, CHGET3 LD (HL),0 (HL),0 (HL),0 LD HL, (GETPNT) LD C, (HL) CALL UPDATE LD C, (HL) CALL UPDATE LD A,C JP PBDHRT CKCNTC: ; Check ctl-C ; PUSH HL LD HL,0 CALL ISCNTC ; PUSH HL LD HL,0 SUBTT - MSXIO - Music routines	
-80 3.44 routines	LD CP JR JR LD CHGET3: LD LD LD LD CALL LD LD CALL LD CALL LD CALL CALL	
(MSX ROM BASIC BIOS) Macro-80 - MSXIO - Keyboard encoding routines	7E 7E 7E 20 220 36 36 36 36 36 37 38 48 48 20 21 79 22 79 22 79 22 79 23 79 23 79 23 79 23 79 23 79 23 24 25 79 21 0000 21 03FB 21 20 21 22 23 24 25 26 27 28 29 21 200 21	
(MSX ROM BASI - MSXIO - Key	3420 10E4 3421 10E5 3422 10E5 3424 10E6 3425 10E8 3426 10E8 3426 10E8 3426 10E8 3427 10E7 3429 10F5 3430 10F5 3431 10F5 3433 10F9 3434 10F5 3435 10F9 3436 10F9 3437 10F9 3436 10F9 3437 10F9 3436 10F9 3437 10F9 3438 10F9 3439 10F0 3436 10F0 3437 10F0 3438 10F0 3439 1100 3430 3431	

voice A volume (0..15 = volume, 16=use envelope) voice B volume (0..15 = volume, 16=use envelope) voice C volume (0..15 = volume, 16=use envelope) ; Write data to specified register of GI sound chip B5, B4, B3 = voice C, B, A noise enable (0=enabled) B2, B1, B0 = voice C, B, A tone enable (0=enabled) LATCH ADDRESS ; OUTPUT DATA = Reg 14,15 Input Output flags Entry - (E)=data,(A)=register number ervelope shape (0..15) voice C coarse tune voice B coarse tune Exit - All regs preserved voice A coarse tune (PSG.LW),A (PSG.DW),A voice B fine tune voice A fine tune voice C fine tune ; 11-12 envelope period joystick l port joystick 2 port Α,Ε AF GI Reg# - usage DI OUT PUSH OUT ; 7 B7,B6 ГD WRTPSG: ; 10 13 ; 14 6 æ 5 4 M N H O (MSX ROM BASIC BIOS) Macro-80 Music routines F3 D3 A0 F5 7B D3 A1 1105 1102 1103 1106 1107 1102 - OIXSW -3459 3460 3449 3456 3458 3465 3466 3468 3469 3443 3444 3445 3446 3447 3448 3450 3451 3452 3453 3454 3455 3457 3461 3462 3463 3464 3467 3470 3472 3442 3471

128

41

PAGE

01-Jan-85

3.44

PAGE 41-1		d des troyed	<pre>;[A]=fine tune register for voice A ;data to be written on R0 ;0 to coarse tune register ;Rl coarse ;enable voice [A] tone ;[A]=voice enable register ;R7 ;set volume to 7 ;[A]=voice A volume register ;R8</pre>
01-Jan-85	AF from PAD	A, PSG.PA (PSG.LW), A A, (PSG.DR) a 'bell' sound registers are des	A E,01010101B WRTPSG E,A A WRTPSG E,10111110B A,7 WRTPSG E,A A MRTPSG E,A A BC,07D0H
3.44	EI POP RET INGI: 7 Input data f	<pre> LD RDPSG: DUT IN IN RET BEEP: BEEP causes Exit - all r ; </pre>	XOR LD CALL LD LD LD LD LD LD LD LD LD LD LD LD L
3IOS) Macro-80 routines	FB C9 C9	3E 0E DB A2 C9 A2 C9 A2 C9 A2	AF 1E 55 CD 1102 5F 3C CD 1102 1E BE 3E 07 CD 1102 5F CD 1102 01 07D0
BASIC F Music	1109 110A 110B 110C	110C 110E 110E 1112 1113	1113 1114 1116 1116 1118 1118 11120 1122 1125 1125 1125 1127
(MSX ROM - MSXIO -	3473 3474 3475 3476 3476 3477 3477	94 8 9 34 8 9 34 8 8 34 9 0 34 9 0 34 9 0 34 9 0 34 9 0 34 9 0 34 9 0 34 9 0 34 9 1 34 9 1 34 9 1 34 9 1 34 9 1 34 9 1 34 9 34 8 34 8 34 8 34 8 34 8 34 8 34 8 34 8	3491 3492 3494 3495 3499 3500 3500 3500 3500 3500 3500 3500 35

PAGE 41-2	reset GI sound chip;													Get action information from specified music queue. Perform	. Called by interrupt routine							+ Number of bytes that follow this item				f time		BYTES
01-Jan-85	CSDLY1 GICINI			BC	(SP),HL	(SP),HL	А,В	U U	NZ, CSDLY1					information from	synchronization.	I		Formation -		2 BYTES		Number of bytes	1	NNNTTTTTTTTTT		+Period of time		- FROM 1 TO 5
3.44	CALL JP CSDLY1:	Delay by [BC]	1	DEC	EX	EX	ГD	OR	ЯĊ	RET		ACTION:		Get action i	action with	in time.		- Action information		ITEM 1 -		+		NNN				ITEM 2, 3, 4
BIOS) Macro-80 c routines	CD 1133 C3 04BD CS			0B	ЕЗ	E3	78	Bl	20 F9	60		AC	••		••		••	••	••			••	••	••	••	••	••	••
MSX ROM BASIC B MSXIO - Music	112D 1130 1133			1133	1134	1135	1136	1137	1138	113A		113B																
(MSX RO - MSXIO	3504 3505 3506	3507 3508	3509	3510	3511	3512	3513	3514	3515	3516	3517	3518	3519	3520	3521	3522	3523	3524	3525	3526	3527	3528	3529	3530	3531	3532	3533	3534

85 PAGE 41-3	= 0 then this is the HO byte of the tone period.	= 2 then this is just a volume control byte.	IS ON, envelope control is in effect, and bits	shape number of envelope.	_ 1	= 3 THEN CAIS DYTE WILL DE TOLLOWED DY A 2 DYTE	envelope period, HO first.		count number (02)		;Save channel number	;Get pointer into vcb of channel				;[DE]=countdown timer for voice	;Decrement timer	; Put it back lo first					;No action if not zero	;Voice 0 uses queue 0					; SAVE IN [D]	;Get number of following items
01-Jan-85							pe period,		- (A)=Channel cc		B,A	GETVCP	HL	D, (HL)	нг	E, (HL)	DE	(HL),E	HL	(HT),D	Α, D	ы	NZ	A, B	(QUEUEN), A	XGETQ	0FFH	Z, VOICOF	D,A	0E0H
3.44	IF HO 2 BITS	IF HO 2 BITS	IF BIT 4	0-3 give		STIE 2 DH TI	envelc		ENTRY - $(A) =$		LD	CALL	DEC	LD	DEC	ГD	DEC	LD	INC	ГD	ГD	OR	RET	ГD	ГD	CALL	CP	JR	ГD	AND
ASIC BIOS) Macro-80 Music routines		••	••	•~ •	•~ '	••		••	••		47	CD 1470	2B	56	2B	5.E	lB	73	23	72	7 A	B3	C0		32 FB3E	CD 11E2	도니	28 5B	57	E6 E0
A NO											113B	113C	113F	4	1141	1142	1143	1144	1145	4	1147	1148	4	4		4	1151	S	1155	1156
(MSX R - MSXIO	3535 3536	53	23	с С	ч С П	7" 0	54	54	54	54	ß	54	54	54	5	5	52	5	5	52	50	പ്പ	പ്പ	5	56	56	56	90	Q	

132				es from queue
41-4	in [c]	;GET LO 5 BITS OF [D] ;Set MSB of new countdown ;Get LSB of new countdown	<pre>;bet it ;Done all items? ;Yes ;Get next item from queue ;Save this to [D] ;Get HO 2 bits ;Execute volume action</pre>	Get low byte for tone Get back voice number X 2 ;0utput fine tune register ;Point to coarse tune register ;Restore saved value ;Output coarse tune reg ;Output coarse tune reg ;Decrement since we took 2 bytes from queue
PAGE	;Save in [C]	;GET LO ;Set MSI ;Get LSI .set it	; Done a ; Yes ; Get ne ; Save ; Get Hu ; Execut	;Get lo ;Get ba ;X 2 ;Nutput ;Point ;Restont ;Output
01-Jan-85	C,A	А, D 1 FH (HL), A XGETQ HL (HL) д	C C Z XGETQ D,A 0C0H NZ,XVOL	XGETQ E,A A,B WRTPSG A E,D WRTPSG C MORACT
3.44	RLCA RLCA RLCA LD	LD AND LD CALL DEC	MORACT: MORACT: DEC RET CALL LD AND JR	Set tone CALL LD LD RLCA RLCA CALL LD LD DEC JR
Ma cr o-80 s			X K	
) tine	07 07 4F	7A E6 1F 77 CD 11E2 2B 27	00 00 00 00 00 00 11 20 11 20 11	CD 11E2 5F 78 07 07 CD 1102 5A CD 1102 5A CD 1102 01 18 E5
l BASIC F Music	1158 1159 115A 115B	115C 115D 115F 1160 1163	1165 1166 1166 1166 1167 1168 1168 1168	1170 1173 1174 1175 1175 1176 1177 1177 1178 1178
(MSX ROM BASIC BIOS - MSXIO - Music roui	3566 3567 3568 3569	3570 3571 3572 3574 3574	3576 3577 3578 3578 3580 3581 3582 3583 3583	3585 3586 3586 3588 3593 3593 3593 3594 3594 3595 3595

-		t matter			
PAGE 41-5	;save it in [H] ;BIT 7 SET?	<pre>;[A] has junk in ho which shouldn't matter ;Get back voice number ;Regs 8,9,10 ;Output amplitude reg</pre>	;Check envelope generate bit ;Reg 13 for shape ;Set envelope shape if enabled	;See if set envelope period ;No ;Get ho byte of envelope period ;Get low byte of envelope period	;Register 11 for fine tune ;Point to coarse tune
01-Jan-85	H , A 80H Z , XEPER	E,D A,B A,8 WRTPSG	<u>д</u>	A,H 0100000B Z,MORACT XGETQ D,A XGETQ F.A	A, 0BH WRTPSG A E, D
3.44	OL: LD AND JR Set volume	LD LD ADD CALL	en	LD AND JR CALL LD CALL	LD CALL INC LD
co-80	XVOL: ; ; Set		XEPER: ; Set		
BASIC BIOS) Macro Music routines	67 E6 80 28 0F	5A 78 C6 08 CD 1102 78	, B E6 10 3E 0D C4 1102	7C E6 40 28 CC CD 11E2 57 CD 11E2 CD 11E2	3E 0B 3E 0B 3C 3C 5A
-	1181 1181 1182 1184	1186 1187 1188 1188	118E 1190 1192 1195	1195 1196 1198 119A 119D 119E	11A2 11A4 11A4 11A7 11A8
(MSX ROM - MSXIO -	3597 3598 3599 3600 3601 3603	3604 3605 3606 3608 3608	3610 3611 3612 3613 3614 3615 3615	3617 3618 3619 3620 3621 3622	3624 3625 3625 3627

PAGE 41-6	CALL WRTPSG DEC C DEC C JEC C JECOMES here when an EOF mark has been found for a specified channel LD A,B ADD A,B ADD A,B ADD A,B JEC CALL WRTPSG JEC HIL, MUSICF CALL WRTPSG JEC HIL, MUSICF SCF LD HL, MUSICF SCF SCF LD HL, MUSICF SCF LD HL, MUSICF SCF LD HL, MUSICF SCF LD HL, MUSICF SCF LD HL, MUSICF SCF LD HL, MUSICF SCF LD (HL), A SCF LD (HL), A SCF LD (HL), A SCF SCF LD (HL), A SCF LD (HL), A SCF SCF LD (HL), A SCF SCF LD (HL), A SCF SCF LD (HL), A SCF SCF LD (HL), A SCF SCF LD (HL), A SCF SCF LD (HL), A SCF LD (HL), A SCF LD (HL), A SCF LD (HL), A SCF SCF LD (HL), A SCF LD (HL), A SCF SCF LD (HL), A SCF SCF LD (HL), A SCF LD (HL), A SCF SCF LD (HL), A SCF LD (HL), A SCF SCF LD (HL), A SCF LD (HL), A SCF LD (HL), A SCF LD (HL), A SCF SCF LD (HL), A SCF SCF LD (HL), A SCF SCF LD (HL), A SCF SCF SCF SCF SCF SCF SCF SCF	
01-Jan-85	CALL WRTPSG DEC C DEC C JR MORACT JR MORACT JR MORACT HIL MORACT SE DEC C LD A,B ADD A	c
3.44	CALL DEC DEC DEC JR VOICOF: ; Comes here w ; channel ; channel ; channel ; channel LD LD LD LD LD LD CALL INC LD SCF SCF SCF SCF SCF AND SCF INC LD DJNZ SCF AND CALL INC LD DJNZ SCF SCF SCF SCF SCF SCF SCF CALL INC CALL INC CALL INC CALL INC CALL INC CALL INC INC CALL INC INC CALL INC INC CALL INC CALL INC SCF SCF SCF SCF SCF SCF SCF SCF SCF SC	~~~
BIOS) Macro-80 c routines	CD 1102 0D 0D 18 B6 18 B6 18 B6 78 77 77 77 77 86 85 77 77 86 85 77 77 86 85 85 87 87 87 87 87 87 87 87 87 87 87 87 87	·
BASIC Musi	11A9 11AC 11AC 11AE 11BE 11B8 11B8 11B8 11B8 11B8 11B8 11B	
(MSX ROM - MSXIO -	8 8 9 9 9 9 9 9 9 9 9 9 9 9 9	}

PAGE 41-7	return if background task is active:	<pre>;return if nothing for it to do ;l less thing for it to do</pre>	;start it playing now	;Trigger!	;Get queue ID	;Get a byte from a specified queue ;pop H, D, B and return and Paddle interface
01-Jan-85	NZ HL, PLYCNT A, (HL) A	Z (HL) HL,1	(VCBA),HL (VCBB),HL (VCBC),HL	A,0111B (MUSICF),A	A, (QUEUEN) HL DE BC	GETQ PBDHRT Joystick
3.44	RET LD OR	RET DEC LD	U L D L D		HSU4 HSU4 U1	1
Ma cr o-80 s				XGETQ: ;		; SUBTTL
) ine	C0 21 FB40 7E B7	00(FR			3A FB3E E5 D5 C5	CD 14AD C3 08DB
(MSX ROM BASIC BIOS - MSXIO - Music rout:	11C8 11C9 11CC 11CC	11CE 11CF 11D0	11D6 11D6	11DE 11DE 11E1 11E2	11E2 11E5 11E6 11E7	11E8 11EB
(MSX RC - MSXIO	3659 3660 3661 3662	3663 3664 3665 3666	3667 3668	3669 3670 3671 3672 3673	3674 3675 3676 3677	3678 3679 3680 3681

PAGE 42	;STICK(0) - read cursor keys ;Read joystick	GTROWB ;Read keyboard ;Move cursor status to lower four bits HL,KSTKTB STICK1 joystick and read from it B,A A,PSG.PB	Read what is currently output to port B
01-Jan-85	A M,KYSTCK SLSTCK HL,STKTBL HL,STKTBL OFH E,A D,0 HL,DE HL,DE A,(HL)	GTROW8 GTROW8 HL,KSTKTB STICK1 STICK1 r joystick and B,A A,PSG.PB	RDPSG
cro-80 3.44 ddle interface	GTSTCK: ; DEC JP CALL LD STICK1: AND LD LD LD LD LD LD LD LD KBT RET	KYSTCK: ; CALL RRCA RRCA RRCA RRCA RRCA ILD JR SLSTCK: ; Select proper ; Select proper ; LD	DI CALL
) Ma and Pa	3D FA 1200 CD 120C 21 1233 21 1233 21 1233 21 1233 5F 5F 16 00 19 7E 7E 7E	CD 1226 0F 0F 0F 0F 21 1243 18 EC 18 EC 47 3E 0F	F3 CD 110E
(MSX ROM BASIC BIOS - MSXIO - Joystick a	11EE 11EE 11EF 11F2 11F8 11F8 11F8 11F8 11F8 11F8 11F	1200 1200 1203 1204 1205 1206 1207 1200 1200 1200	120F 1210
(MSX R(- MSXIO	3682 3682 3684 3685 3686 3692 3692 3693 3693 3693 3693 3693	3696 3697 3698 3699 3700 3701 3702 3703 3705 3705 3706 3706 3708 3708 3709	3711 3712

137	is low state ick 2, enable P6,P7	;Select joystick l, make sure P8 is low state ;Enable P6,P7 ;Read status of joystick port ;Read status af joystick sort	
42-1	;STICK(1) ;Make sure P8 is ;Select joystick	;Select joysti ;Enable P6,P7 ;Read status c ;Read status c	
PAGE	;STIC ;Make ;Sele	; Sele ; Enab ; Read ; Read	
01-Jan-85	SL STC1 0DFH 4CH SLSTC2	AND 0AFH ;Select jo OR 3 ;Enable P CALL INGI (PSG.DW),A ;Read state EI RET ROW8: Get keyboard's 8th row, bit assignments RDULXXXS	space left up down right A,(PPI.CR) 0F0H A,8 (PPI.CW),A A,(PPI.BR)
-80 3.44 • interface	DJNZ AND OR JR SLSTC1:	AND SLSTC2: OR SLSTC2: OUT CALL EI EI EI RET CALL EI RET CALL EI RET CALL	ADD ADD ADD ADD ADD ADD ADD ADD ADD ADD
BASIC BIOS) Macro-80 Joystick and Paddle in	10 06 E6 DF F6 4C 18 04	E6 AF F6 03 D3 Al CD 110C FB C9 C9	F3 DB AA DB AA E6 F0 C6 08 D3 AA DB A9
M BASIC BIOS - Joystick	1213 1215 1217 1219 121B	121B 121J 121F 121F 1221 1224 1225 1225	1226 1226 1229 1228 1228 1228
(MSX ROM - MSXIO -	3713 3714 3715 3716 3717	3719 3720 3721 3723 3725 3725 3725 3726 3726 3728 3728 3729 3729 3729 3730 3731	3733 3734 3734 3735 3736 3738 3741 3741 3743

PAGE						; RLBF	; RLB	; RL F	; RL	;R BF	;R B	;R F	;R	; LBF	; LB	; L F	; L	; BF	; B	н				; RBFL	; RBF	;RB L	; RB	;R FL	,R F	;R L	; R	; BFL
01-Jan-85						0	5	1	0	e	4	2	°	7	9	8	7	0	5	1	0			0	ε	5	4	I	2	0	с	7
3.44	ce	EI	RET			DB	DB	DB	DB	DB	DB	DB	DB	DB	DB	DB	DB	DB	DB	DB	DB		:	DB	DB	DB	DB	DB	DB	DB	DB	DB
Ma cr o-80	Paddle interface			••	STKTBL																		KSTKTB									
BASIC BIOS)	ick and	FB	60			00	05	01	00	03	04	02	03	07	90	08	07	00	05	01	00			00	03	05	04	01	02	00	03	07
BASIC	Joystick	1231	1232		1233	1233	1234	1235	1236	1237	1238	1239	12 3A	123B	123C	123D	123E	123F	1240	1241	1242		1243	1243	1244	1245	1246	1247	1248	1249	124A	124B
(MSX ROM	- OIXSW -	3744	3745	3746	3747	3748	3749	3750	3751	3752	3753	3754	3755	3756	3757	3758	3759	3760	3761	3762	3763	3764	3765	3766	3767	3768	3769	3770	3771	3772	3773	3774

42-2

PAGE 42-3	; BF ; B L	; B : FL	1 Eu 	л						;STRIG(0), use keyboard	1		;Read joystick	1			;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
01-Jan-85	0 9 1	റയ	1	7	0				A	M, KEYTRG	AF	1	SLSTCK	BC	В	В	B,10H	M, TRIGI	B, ' '		В		1	A, A				GTROW8	1
Macro-80 3.44 Paddle interface	DB DB	DB	DB	DB	DB	••	GTTRIG:	••	DEC	đ	HSUG	AND	CALL	POP	DEC	DEC	LD	JP	LD	TRIG1:	AND	TRIG2:	SUB	SBC	RET	KEYTRG:		CALL	AND
) and	00 06 05	cn 80	01	07	00				3D	FA 126C	F5	E6 01	-	C1	05	05	06 10				A0		D6 01	9F	C9			CD 1226	E6 01
BASIC BIOS Joystick	124C 124D	124F	1250	1251	1252		1253		1253	1254	1257	1258	125A	125D	125E	125F	1260	1262	1265	1267	1267	1268	1.2.68	126A	126B	126C		126C	126F
(MSX ROM - MSXIO -	3775 3776 3777	3778	3779	3780	3781	3782	3783	3784	3785	3786	3787	3788	3789	3790	3791	3792	3793	3794	3795	3796	3797	3798	3799	3800	3801	3802	3803	3804	3805

PAGE 42-4			[Acc])		joystick port 2 joystick port 1			Joystick port 2 invistick port 1					joystick port 2				.ct # (rarry reset if mort 1))) Li				.How most nottorn	LOTIN NASY PACKET II	.Cot mack rattorn	Set IIasy barneti
4 01-Jan-85	L.	e ot paddle	Input parameters (passed via	A connected to	A connected to B connected to	B connected to	C connected to	C connected to	D connected	E connected	ធ	dle F connected to	dle F connected to		·	DA			B, A		Ŀ			ZN	B,A
Macro-80 3.44 Paddle interface	JR GTPDL: ;	; Get value ;	; Input par	; ; 1 - Paddle	; 2 - Paddle · 3 - Paddle	; 4 - Paddle	I	; 6 - Paddle	; / - Padale	i I	1 0	; 11 - Paddle	; 12 - Paddle	••	INC	AND	RRA	PUSH	LD	XOR	SCF	PDL1:	RLA	DU	LD
) and	18 F5														3C	A7	1F	F5	47	AF	37			10 FD	47
BASIC BIO Joystick	1271 1273														1273	1274	1275	1276	1277	1278	1279	127A	127A	127B	127D
(MSX ROM BASIC BIOS - MSXIO - Joystick a	3806 3807 3808	3809 3810	3811	3812 3813	3814	3816 3816	3817	3818	3819	3820	3822	3823	3824	3825	3826	3827	3828	3829	3830	3831	3832	3833	3834	3835	3836

PAGE 42-5	;Assume port l ;Good assumption	;Get current port B content ;Set trigger high	;Set trigger low again ;Initialize counter	;End of pulse? ;Yes ;Bump counter ;No overflow yet ;Make it 255	;Return counted value
01-Jan-85	λ.F C,lOH DE,03AFH NC,PDLP1 C,'' DE,4C9FH A,PSG.PB	RDPSG E D C (PSG.DW),A	C (PSG.DW),A A,0EH (PSG.LW),A C,0	A, (PSG.DR) B Z, PDL3 C NZ, PDL2 C	A,C
Macro-80 3.44 Paddle interface	POP LLD LLD LLD LLD LLD PDLP1: LD	CALL AND OR OR OUT	XOR OUT LD OUT LD PDL2:	IN AND JR JP JP DEC PDL3:	EI LD RET GTPAD:
BASIC BIOS) Macro-80 Joystick and Paddle ii	F1 0E 10 0E 10 30 05 0E 20 11 4C9F 3E 0F F3	CD 110E A3 B2 B1 D3 A1		DB A2 A0 28 05 0C C2 129F 0D	FB 79 C9
	127E 127F 1281 1284 1286 1288 1288 128B 128D	128E 1291 1292 1293 1294	1296 1297 1299 129B 129B 129F	1295 1281 1282 1284 1285 1288 1288	12A9 12AA 12AB 12AB
MSX ROM - MSXIO -	3837 3838 3839 3840 3841 3842 3843 3843 3845	m m m m m		3859 3858 3860 3861 3862 3862	0000

42-6 PAGE 01-Jan-85 3.44 Joystick and Paddle interface Macro-80 (MSX ROM BASIC BIOS) - MSXIO -

;If so, read pad and return status ; Result is returned via [Acc]. As for status, 255 is returned ;Read pad connected to port 1 ;Assume PAD(1) - X coordinate for touch pad connected for touch pad connected Return Y coordinate Connected to port 2 to joystick port l to joystick port 2 Good assumption Good assumption ;Argument=0? Assume so Read touch pad (NEC PC-6051 compatible) : Input parameter (passed via [Acc]) 4 - sense touch pad status ---0 - sense touch pad status : 7 - return switch status - return switch status 5 - return X coordinate 6 - return Y coordinate 1 - return X coordinate
2 - return Y coordinate
3 - return switch status A, (PADY) Z DE,03D3H M, GTPAD0 A, (PADX) DE, OCECH C,GTPDP1 ; if true, 0 if false. A 4 4 Σ 4 DEC JP DEC LD LD LD RET SUB JR LD ΓD СP GTPDP1: •• 3A FC9D F8 3A FC9C C8 FE 04 11 0CEC 11 03D3 D6 04 3D FA 12C5 3D 38 05 12AE 12B1 L2AC 12B3 12B6 12B8 12B8 12B9 12BC 12BD 12C0 12C1 12C4 3883 3884 3886 3888 3889 3890 3893 3898 3875 3876 3877 3878 3879 3880 3882 3885 3887 3892 3894 3895 3896 3868 3869 3870 3872 3874 3881 3891 3897 3873 3871

143	3) brt 2
PAGE 42-7	<pre>;Save status (minus if PAD(0) specified) ;[L]=bits that are not to be modified ;[H]=bits that are to be added ;[H]=bits that are to be added ;0 if port 1 specified, 100 octal if port ;disable interrupt till done ;flow the state to be added ;select proper port ;select proper port ;PAD(0) specified ;PAD(0) specified ; ;panel input and select x ; ; inz ;sense Panel input and select x ; ;read first coordinate ;branch if input released ;save for comparison</pre>
01-Jan-85	AF DE,HL (RUNFLG),HL A,A (RUNFLG),HL A,A 01000000B C,A A,PSG.PB RDPSG 0BFH C (PSG.DW),A AF M,TRYAGN INGI (PSG.DW),A AF M,TRYAGN INGI 1 A, AF M,TRYAGN INGI 1 A, AF M,TRYAGN C C C C C C C C C C C C C C C C C C C
-80 3.44 e interface	GTPADO: PUSH EX LID SBC CPL SBC CPL AND LD LD LD DI LD DI CALL EI AND OR OR OR OR OR CALL EI SUB SBC SBC SUB SBC CALL TTRYAGN: TRYAGN: TRYAGN: SUB SBC CALL AND AND CALL AND AND CALL AND AND AND CALL AND AND AND CALL AND AND AND CALL AND AND AND CALL AND AND CALL AND AND CALL AND AND CALL AND AND CALL AND AND CALL AND CALL AND AND CALL AND AND CALL AND CALL AND CALL AND CALL AND CALL AND CALL AND CALL AND CALL AND CALL AND CALL AND CALL AND CALL AND CALL AND CALL AND CALL AND CALL AND CALL AND CALL AND CALL AND AND CALL CALL CALL CALL CALL CALL CALL CAL
BIOS) Macro-80 cick and Paddle i	F5 EB 22 F866 35 F866 35 F866 46 46 47 73 E6 40 73 E6 10 73 A1 73 12E8 73 12E8 74 12E8 75 110C 75 110C 75 110C 75 110C 73 28 73 23 73 24 74 74 74 74 74 74 74 74 74 74 74 74 74 7
MSX ROM BASIC BIOS MSXIO - Joystick	12C5 12C5 12C5 12C5 12C7 12C8 12C8 12C8 12C8 12C8 12D8 12D8 12D8 12D8 12D8 12C8 12C8 12C8 12C8 12C8 12C8 12C8 12C
(MSX RC - MSXIO	3899 3901 3902 3903 3904 3905 3906 3906 3913 3916 3914 3918 3918 3918 3918 3918 3922 3922 3922 3922 3922 3922 3922 392

PAGE 42-8	<pre>;read another input ;restore previos coord ;branch if input released ;[A]=ABS(X0-X1)) ;[a]=ABS(X0-X1)) ;[a]=ABS(Y0-Y1)) ;[A]=ABS(Y0-Y1))</pre>	
01-Jan-85	REDCOD BC C, PADX1 A, B D NC, NONEG1 A 5 NC, TRYAGN A,C E NC, NONEG2 B NC, NONEG2 A A A C, PADX), A A, B A, D (PADX), A A, B A, B A, B A, C A A A A A A A A A A A A A A A A A A A	
Macro-80 3.44 Paddle interface	CALL POP JR TLD SUB JR CPL INC NONEGL: CPL INC CPL INC CPL JR JR CPL INC CPL I	
BIOS) Macro-80 tick and Paddle i	CD 1320 C1 38 1C 38 1C 38 1C 32 78 30 02 33 02 33 02 33 02 33 05 33 02 33 FC9D 78 78 78 78 78 78 78 70 77 70 77 70 77 72 77 72 72 72 72 72 72 72 72 72 72	
BASIC E Joysti	12F8 12F8 12F5 12F5 12F5 12F5 1306 1306 1306 1306 1306 1316 1316 1316	
(MSX ROM BASIC - MSXIO - Joyst	3930 3930 3931 3932 3933 3934 39343 39455 39443 39455 39443 39455 39443 39455 39443 394555 394555 394555 394555 394555 394555 394555 3945555 3945555 3945555555555	

42-9	
PAGE	
01-Jan-85	
3.44	face
BIOS) Macro-80	tick and Paddle interface
MSX ROM BASIC BIOS	- Joystick
(MSX ROM	- MSXIO -

[D,E]	; change to channel to [Y] when done	;read [X]	;return if input released	4		; change to [X] after read	;read [Y]		;store Y read out	;clear carry	force input is OK	4			nto [L]	Carry set if input released during read	'n	;make sure AD completed	; input 8 bits	; input channel# after done	4	;serial clock(SCK)=l			read PAD	;save SENSE status			
Read X,Y coordinate into [D,E]	C,0AH	REDPAD	U	D,L	DE	c,0	REDPAD	DE	E,L	A	H,A				anel input i	input releas	I	CHKEOC	B,8	D,C		0 , D	2 , D	OUTGI	INGI	H,A			
; Read X,Y coc ;	LD	CALL	RET	ГD	HSUT	LD	CALL	POP	LD	XOR	ГD	RET	REDPAD:	••	; Read touch p	; Carry set if	••	CALL	ΓD	LD	REDLOP:	RES	RES	CALL	CALL	ΓD	RRA	RRA	RRA
	0E 0A	CD 1332	D8	55	D5	0E 00	CD 1332	DI	5D	AF	67	C9						CD 135B	06 08	51		CB 82	CB 92	CD 136D	CD 110C	67	lF	lF	lF
	1320	1322	1325	1326	1327	1328	132A	132D	132E	132F	1330	1331	1332					1332	1335	1337	1338	1338	133A	133C	133F	1342	1343	1344	1345
3961 3962	3 96 3	3964	3965	3966	3967	3968	3969	3970	3971	3972	3973	3974	3975	3976	3977	3978	3979	3980	3981	3982	3983	3984	3985	3986	3987	3988	3989	3990	399I

PAGE 42-10	<pre>;bit 2 to LSB of [L] ;SCK=0 ;initiate another AD ;LSB=SENSE status ;SENSE status to carry ;OK if no carry ;reset CS ;test EOC ;test EOC ;set CS and return</pre>	
01-Jan-85	L C,D OUTGI REDLOP 4,D 5,D OUTGI A,00110101B C D,A OUTGI 1NGI 2 2,EOCCHK 4,D 5,D 5,D 1NGI 2 1NGI 2 1NGI 2 2,EOCCHK 4,D 5,D 0 1NGI 1NGI 1NGI 1NGI 2 1NGI 1	1
-80 3.44 e interface	RL L SET 0,D SET 2,D SET 2,D CALL OUTGI DJNZ REDLOP SET 4,D SET 5,D CALL OUTGI LD A,H RRA REA REA REA REA REA CALL OUTGI LD A,001101 OR C LD D,A CALL OUTGI CALL OUTGI	
(MSX ROM BASIC BIOS) Macro-80 - MSXIO - Joystick and Paddle interface	CB 15 CB 22 CB 22 CB 136D 10 E7 CD 136D CD 136D 7C CB EA CD 136D CD 13	2
I BASIC Joyst	1346 1348 1348 1348 1348 1348 1355 1355 1355 1358 1358 1358 1358 1368 1368 1368 1368 1368 1368 1368 136	1
(MSX ROM - MSXIO -	3993 3999 3999 39999 39999 4000 4000 400	3 3 7 7

PAGE 42-11	;Also known as [PADWRK] es for MSXIO
01-Jan-85	HL, (RUNFLG) ;Also know A,L D,A D,A A,PSG.PB (PSG.LW),A A,(PSG.DR) L A,(PSG.DR) L H (PSG.DW),A DE HL (PSG.DW),A DE HL
MSX ROM BASIC BIOS) Macro-80 3.44 MSXIO - Joystick and Paddle interface	2A F8666 LD 2F CPL LD 2F CPL LD 3E 0F LD 57 LD LD 3E 0F LD 03 AO OUT DB A2 AND DB A2 LD DB A2 ND DB A2 OUT DB A2 OUT DB A2 OUT DB A3 OUT D1 POP POP D1 POP POP C9 SUBTTL MSXIO -
I BASIC B Joysti	136F 1372 1372 1374 1376 1376 1376 1377 1377 1377 1381 1382 1383
(MSX ROM I - MSXIO -	4023 4024 4025 4025 4026 4028 4028 4031 4031 4033 4033 4033 4033 4033 4033

	Flip motor switch Stop motor Skip next 2 bytes ('JN2' instruction) ;RETN
	switch bytes
43	Flip motor switch Stop motor Skip next 2 bytes ;RETN
PAGE	;Flip ;Stop ;Skip ;SkiP
01-Jan-85	A M,FLPMOT NZ,MOTRON A,00001001B 0C2H A,8 (PPI.CM),A A,(PPI.CR) 10H STMOT1 STMOT1 H.NMI
3.44	AND JP JR LLD DB LLD OUT NB LLD AND JR AND JR AND AND AND ARETI RETIN
cro-80 for MSXIO	STMOTR: AND JP STMOT1: JR LD DB MOTRON: LD DB DB DB DB DB DB DB DB DB ND JR IN AND JR NMI: FLPMOT: ; NMI handler ; cALL
Ma tes	A7 FA 1392 20 03 3E 09 3E 08 3E 08 3E 08 03 AB C9 B1 AA E6 10 18 F0 18 F0 CD FDD6 ED 45 ED 45
BASIC Misc.	1384 1384 1384 1385 1388 1388 1388 1388 1388 1398 1398 1398
(MSX ROM BASIC BIOS) - MSXIO - Misc. routir	4040 4041 4042 40443 40444 4044 4044 4051 4055 4055 4055 405

PAGE	sóu									
01-Jan-85	function key strings	BC,0A0H DE,FNKSTR HL,FKTABL		"color "	10	"auto "	ll "goto "	ll "list "	11 "run" 13	12 "color 15,4,7"
3.44	alize	LD LD LD	LDIR RET	EF: DB	DS	DB	DS DB	DS DB	DS DB DB	DS DB
Macro-80 Misc. routines for MSXIO	; INIFNK: ; Initi			; FNKDEF: 6F		6F	6 F	74		6F 35 37
Mac nes f		00A0 F87F 13A9	0	6C	0	74	74	73	9 EE	6C 31 2C
IOS) coutin		01 00 11 F8 21 13	ED B0 C9	63 6F	72 20	61 75 20	67 6F 20	6C 69 20	72 75 0D	63 6F 72 20 2C 34
щ	139D	139D 13A0 13A3	13A6 13A8	13A9 13A9	13AD 13AF	13B9 13BD	13BE 13C9 13CD	13CE 13D9 13DD	13DE 13E9 13EC	13ED 13F9 13FD 1401
(MSX ROM - MSXIO -	4063 4064 4065 4066 4067 4068	4069 4070 4071	4072 4073	40/4 4075 4076	4077 4078	4079 4080	4 U81 4082 4083	4084 4085 4086	4087 4088 4089	4090 4091 4093 4093

01-Jan-85	13 3 "cload"	3 4 10	"cont" 13 11	"list." 13,30,30	8 12 "run"	13 11	A, (VDP.SR)	A,(PPI.AR)	A,(WA.IPI)
3.44	DB DS DB	DB DS	DB DB DB	DB DB	DS DB DB	DB DS	IN RET	IN RET	OUT RET
Macro-80 s for MSXIO	-		ধ	ধ		; RDVDP: ;	RSLREG:	WSLREG: ;	SNSMAT:
đ	6F 61		6E 74	73 74 1E	6 E				
) N ines	ec ec		6F (69 7 1E 1			66	A8	A8
BIOS) M . routines	0D 63		63 0D	6C 2E 0D	0C 72	OD	C9 C9	DB C9	D3 C9
BASIC Misc.	1405 1406 1409	140D 140E	1419 1410 141D	1429 1420 142E	1431 1439 143A	143D 143E 1449	1449 144B 144C	144C 144E 144F	144F 1451 1452
(MSX ROM - MSXIO -	4094 4095 4096	4097 4098 4099	4100 4101 4101	4104 4104 4105	4106 4107 4108	4109 4110 4111 4112 4113	4114 4115 4116 4117	4 118 4119 4120 4121	4122 4123 4124

PAGE 44-2	LD C,A DI N, (PPI.CR) ;Get what is currently output to Port C AND OFOH ;Leave higher 4 bits unaffected ADD A,C OUT (PPI.CW),A ;Select row IN A,(PPI.BR) ;Get column information of selected row EI RET A.C. ;Select row IN A,(PPI.BR) ;Get column information of selected row CHL H.ISFL ;Get column information of selected row RET BET ;Get file pointer DUSH HL ;Save [H,L] LD HL,(PTRFLL) ;Get file pointer COR HL ;Save [H,L] LD HL,(PTRFLL) ;Get file pointer CMPR HL ;Save [H,L] LD HL,(PTRFLL) ;Get file pointer CMPR HL ;Save [H,L] COMPR compares [H,L] with [D,E] unsigned [H,L] less than [D,E] set carry [H,L] less than [D,E] set carry	
01-Jan-85	LD C,A DI ND C,A ND OFOH AND OFOH AND A,C OUT (PPI.CW),A IN A,(PPI.CW),A IN A,(PPI.BR) EI RET RET RET RET RET RET RET RET RET RET	А,Н
3.44	LD LD DI IN AND AND AND OUT IN EI RET PUSH LD PUSH LD CALL PUSH LD OR PUSH LD S the o	ĽD
Macro-80 es for MSXIO	н – <u>Ники</u> – <u>А</u> иниин,	
ě	4F 4F 63 103 AA 103 AA 81 81 81 70 70 70 70 70 70 70 70 70 70 70 70 70	2/
	11111111111111111111111111111111111111	14 6A
(MSX ROM) - MSXIO -	4 4 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Ω

44-4	ize nes
PAGE	; VCB size
01-Jan-85	A Z,GETVCX DE,25H ;VCB size A NZ,GETVCL DE H.PHYD H.FORM I Queue utility routines
3.44	OR A JR Z, LD DE LD DE ADD HI DEC A JR NZ JR NZ POP DE RET H. RET H. RET H. RET H. RET - QUEUTL -
ro-80 Dr MSXIO	GETVCL: GETVCX: PHYDIO: ; ; SUBTTL SUBTTL
ASIC BIOS) Macro-80 Misc. routines for MSXIO	07 0025 055 057 557 557 557
BIOS rou	B7 28 11 19 20 20 20 20 20 20 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9
BASIC Misc.	147E 147E 1481 1481 1484 1485 1488 1488 1488 1488
(MSX ROM BASIC BIOS) Macro-80 - MSXIO - Misc. routines for MS	4187 4188 4189 4190 4191 4195 4199 4199 4199 4200 4201 4203 4203 4203

3.44 01-Jan-85 PAGE 45	Copyright (C) 1980 by Microsoft Corporation Written by Marc Wilson This utility provides for multiple queues with the following capabilities: Queues of varying length - 1,3,7,15,31,63,127,255 Each queue can be any of the possible lengths the queues can be initialized at any time and be located anywhere a single pointer (QUEUES) provides the address of the queue table. The queue table has all information for each queue, 6 bytes per queue. A single non-zero character can be pushed back on top of the queue. The entry for each queue is as follows: +1 GET OFFET +1 GET OFFET +1 GET OFFET +1,+5 QUEUE ADDRESS The utility assumes that the queue table is valid for all queue numbers passed to the routines ROUTINES: All routines assume that [A] equals the queue table.
(MSX ROM BASIC BIOS) Macro-80 - QUEUTL - Queue utility routines	4205 4206 4209 4209 4210 4211 42110 42111 42115 42115 4219 4219 4229 4229 4229 4229 4229 4229

3.44 01-Jan-85 PAGE 45-1	Other requirements follow.	GETQ - Returns current top of queue in [A],	zero flag set if queue empty	PUTQ - Puts byte in [E] reg on end of queue,	zero set if queue is full		; NOTE:	The routines are designed to be reentrant, however	there are some restrictions for cases involving a	single queue (in any case operating on different	queues is alright). The first restriction is that	the same routine cannot be reentered. The second	is that INITQ and POPQ do not allow PUTQ,	GETQ or BCKQ to be entered.		LFTQ - Returns unused number of bytes in queue in [A] reg	INITQ - Initialize queue to empty state,	B reg=length, (DE)=ADDR	*** All routines destroy the registers ***		SUBTTE - QUEUTE - Queue routines	
(MSX ROM BASIC BIOS) Macro-80 - QUEUTL - Queue utility routines	4237 ; (4238 ;	4239 ;	4240 ;	4241 ;	4242 ;	4243 ;NC	4244 ;	4245 ;	4246 ;	4247 ;	4248 ;	4249 ;	4250 ;	4251 ;	4252 ;	4253 ;	4254 ;	4255 ; *	4256 ;	4257 SUF	

		υ			put new pointer	ess	put char	
46		;Get queue pointers ;Bump PUT	around full		c t	;(HL) = QUEUE address	;(HL) = Address to put	;set new pointer
PAGE		;Get queu ;Bump PU ^T	;Wrap around ;OUEUE full	8	;Save place ;Pointer in	; (HL)	; (HL)	;set no
01-Jan-85	queue	GETPTR A, B A HL	(HL) C Z	· 봄 봄 봄 8	нц (SP),НЦ НЦ С,А А.(HL)	нг. Н. (нг.) Г. А В,0	HL, BC (HL), E HL	(旺),C
3.44	data on	CALL LD INC INC	AND CP RET	PUSH DEC DEC	EX INC LD		ADD LD POP	LD RET
0-80	PUTQ: ; ; Put							GETQ: ;
C BIOS) Macro-80 eue routines		CD 14FA 78 3C 23	A6 B9 C8	55 21 21 21 21 21 21 21 21 21 21 21 21 21	2 B 1 E B 2 3 2 B 2 B 2 E 2 B 2 E 2 E 2 B 2 E 2 B 2 B 2 B 2 B 2 B 2 B 2 B 2 B 2 B 2 B	23 66 67 06 00	09 73 E1	71 C9
	1492	1492 1495 1496 1497	· 4 4 4	• 4 4 4 •	1495 1495 1480 1481	14A3 14A3 14A4 14A5 14A6	14A8 14A9 14AA	14AB 14AC 14AD
(MSX ROM BASI - QUEUTL - Qu	4258 4259 4260 4261	4265 4263 4265 4265	$\square \square \square \square$	27272	4275 4275 4275 4276 4277	0 0 0 V	ထထထ	4285 4286 4287 4288

PAGE 46-1		.Cat dualla mointare					; QUEUE empty:	4	;Bump GET offset	;wrap around			; Save place to store pointer				;offset in C				;[HL] = QUEUE address			;qet char from OUEUE						
44 01-Jan-85	from QUEUE	.Т. СЕТРИВ		NZ, GETBAK	A,C		r z	E E	C A) (HT)	C HL	C HL	SH HL	C HL	L HL	HL HL	C,A			н, (нг.)		B,0) HL, BC	A, (HL)		(HL),C				A,0
0-80 3.44	; Get data	CALL	LD	JR	LD	CP	RET	INC	INC	AND	DEC	DEC	FUSH	INC	INC	INC	LD	LD	INC	LD	LD	LD	ADD	LD	POP	LD	OR	RET	INC	LD
BIOS) Macro-80 le routines		CD 14FA	36 00	20 ID	79	B8	C8	23	3C	A6	2B	2B	ES	23	23	23	4F	7E	23	66	6F	00 90	60	7E	El	71	B7	C0	3C	3E 00
MSX ROM BASIC QUEUTL - Queu		14AD	14B0	14B2	14B4	14B5	14B6	14B7	14B8	14B9	14BA	14 BB	14BC	14BD	14BE	14 BF	14C0	14C1	14C2	14C3	14C4	14C5	14C7	14C8	14C9	14CA	14CB	14CC	14CD	14CE
(MSX RC - QUEUTL	4289 4290	29	6	4293	4294	4295	4296	4297	4298	4299	4300	4301	4302	4303	4304	4305	4306	4307	4308	4309	4310	4311	Г	4313	4314	4315	4316	4317	4318	4319

PAGE 46-2	RET LD C,A LD H,,QUEBAK-1 ADD HL,R ADD HL,R ADD HL,R RET ADD HL,B RET INTTQ: INTTQ: INTTQ: INTTQ - Initialize QUEUE RET PUSH BC 5tart of QUEUE table entry PUSH BC 5tart of QUEUE table entry CALL SSTART 5ter bur offset ID (HL),B ;Clear BUT offset ID (HL),B ;Clear BUT offset INTC HL ID (HL),B ;Clear BUT offset INTC HL ID (HL),B ;Clear BUT offset INC HL ID (HL),B ;Clear BUT offset ID (HL),B ;Clear BUT offset INC HL ID (HL),B ;Clear BUT offset ID (H	
01-Jan-85	RET LD C,A LD C,A LD HL,QUEBAK-1 ADD HL,BC LD A,(HL) RET A,(HL) RET A,(HL) A,(HL) RET A,(HL) A,(HL) RET QUEUE PUSH BC CALL QSTART LD (HL),B INC HL LD (HL),B INC HL LD (HL),B INC HL LD (HL),B INC HL LD (HL),B INC HL LD (HL),A INC HL LD (HL),A INC HL LD (HL),A INC HL LD (HL),A INC HL LD (HL),A INC HL LD (HL),A ST INC HL LD (HL),A ST INC HL (HL),A ST INC (HL),A	
3.44	RET LD LD ADD LD ADD LD PUSH CALL LD TNC LD TNC LD TNC LD TNC LD TNC LD TNC LD TNC LD LD LD LD LD LD LD LD LD LD LD LD LD	
) Macro-80 utines	1504 00 1504 1504	
MSX ROM BASIC BIOS) Ma QUEUTL - Queue routines	4D0 4D1 4D1 4D1 4D1 4F 4D1 4F 4D2 00 4D4 00 4D4 00 4D5 00 4D6 00 4D7 00 4D8 7E 4D8 7E 4D8 70 4D8 70 4D8 70 4D8 70 4D8 70 4D8 70 4E1 70 4E1 23 4E1 23 4E1 23 4E1 23 4E8 23 4E8 23 4E8 70 4E9 70 4E9 70 4E9 70 <	
ROM BAS JTL - Q	14D0 14D1 14D1 14D1 14D2 14D2 14D3 14D4 14D5 14D6 14D7 1407 1407 1407 1407 1407 1407 1407 1407 1407 1407 1407 1407 1407 1407 1407 1408 1408 1408 1408 1408 1408 1408 1408 1408 1408 1408 1408 1408 1408 1408 1408 <t< td=""><td></td></t<>	
(MSX RO - QUEUTL	4 4 4 4 4 4 4 4 4 4 4 4 4 4	

PAGE 46-3	Get QUEUE ptrs	;B=PUT PTR+1 ;subtract PUT from GET ;make it positive UNSIGNED INTEGER		<pre>;Get start of QUEUE TABLE entry ;B = PUT OFFSET ;C = GET OFFSET</pre>	;*2 ;*4 ;*6
01-Jan-85	CETPTR A, B A HL (HL)	B, A A, C B (HL)	LD H,0 LD H,0 RET general routines	QSTART B, (HL.) HL C, (HL.) HL	А, В, А
3.44	CALL LD INC INC AND	LD LD SUB AND		CALL LD LD LD INC	
-80			GETPTR: ; ; ; ;		; QSTART :
BIOS) Macro-80 e routines	CD 14FA 78 3C 23 A6	47 79 90 8F	26 00 C9	CD 1504 46 23 4E 23 7F	B7 C9 47 80
MSX ROM BASIC BIO QUEUTL - Queue r	14EB 14EE 14EF 14FO 14F1	14F2 14F3 14F4 14F5 14F5	14F7 14F9 14FA 14FA	14FA 14FD 14FE 14FE 14FF 1500 1501	1502 1503 1504 1504 1505 1505 1507
(MSX RO) - QUEUTL	4351 4352 4353 4354 4355	4356 4357 4358 4358 4359	4361 4362 4363 4364 4365 4365 4366	4368 4369 4370 4371 4373	4374 4375 4376 4377 4377 4378 4378 4379 4380 4381

an-85 PAGE 46-4			HL, (QUEUES)	υ υ		SUBTTL - MSXGRP - Graphic driver (Print a character on GRP screen)
01-J	C,A	в,0	НГ, (HL, BC		ю 1
3.44 01-Jan-85	LD	LD	LD	ADD	RET	SUBTTL - MSXGRF
MSX ROM BASIC BIOS) Macro-80 QUEUTL - Queue routines	4F	00 00	2A F3F3	60	60	
BASIC] - Queu	1508	1509	150B	150E	150F	
(MSX ROM - QUEUTL	4382	4383	4384	4385	4386	4387

PAGE 47		graphic screen					;Convert code	Graphic header byte, return soon:	;Converted graphic code	;CR?	;Do not ignore CR even on graphic screen	;Control character?	;Yes, ignore this	n	;Get character pattern in PATWRK	;Set color of character			;Current Y coordinate in [DE]		;Do the scaling	; Do not print if already out of screen	; Map to CLOC and CMASK	•	;Row counter		;Column counter	;Get current CLOC and CMASK
01-Jan-85 on GRP screen		on the	t1	DE	BC	AF	CNVCHR	NC, JPPPAL	NZ,GPRT05	HC 0	Z, GRPCR	-	C, JPPPAL		GETPAT	A, (FORCLR)	(ATRBYT), A	HL, (GRPACY)	DE, HL	BC, (GRPACX)	SCALXY	NC, JPPPAL	MAPXYC	DE, PATWRK	с,8		B,8	FETCHC
3.44 t a character	GRPPRT:	; ; Print a character	; DIICU	HSUT	HSUA	HSUG	CALL	JR	JR	CP	JR	CP	JR	GPRT05:	CALL	LD	LD	LD	EX	LD	CALL	JR	CALL	LD	LD	GPRT10:	LD	CALL
) Mao driver			۲ ل	D5	C5	F5	CD 089D	9	20 08	0	28 5F	E 2	38 58		CD 0752	3A F3E9	32 F3F2	2A FCB9	EB	ED 4B FCB7	CD 1599	30 42	15D	11 FC40	0E 08		06 08	CD 1639
M BA -	1510		1510	1511	1512	1513	1514	1517	1519	ഹ	151D	151F	1521	1523	1523	1526	1529	152C	152F	1530	1534	1537	1539	153C	153F	4	4	1543
(MSX RO – MSXGRP	4388 4389 4200	noo	4392	39	4395	4396	4397	4398	4399	4400	4401	4402	4403	4404	4405	4406	4407	4408	4409	4410	4411	4412	4413	-	4415	4416	4417	4418

PAGE 47-1	;Save these	Get pattern for a row		; Uneck each bit		;Set It II 1	; Move I pixel right	;Assume out of screen	Good assumption, skip the rest			;Loop till done all columns		;Restore CLOC and CMASK		;Set these	; Move l pixel down	;Out of screen, skip rest and return	;Point to next row		;Loop till done all rows		;Check current screen mode		;We're in high-resolution mode		;We're going out of screen			
01-Jan-85 on GRP screen	HL AF	A, (DE)		A, A	AF'	C, SETC	TRIGHT	HL	C,GPRT30	HL	AF	GPRT20		AF	HL	STOREC	TDOWNC	C,GPRT40	DE	υ	NZ,GPRT10		CHKMOD	A, (GRPACX)	Z,GPRT50	A, ' '	C, GRPCR	GPRT60		
3.44 a character	HSUA	LD	GPRT20:	ADD	PUSH	CALL	CALL	POP	Я	HSUA	POP	ZNLC	GPRT30:	POP	POP	CALL	CALL	JR	INC	DEC	JR	GPRT40:	CALL	ΓD	JR	ADD	JR	JR	GPRT50:	••
310S) Macro-80 Nic driver (Print	ES			87	F5	DC 167E	CD 16AC	E1	38 04	E5	Fl	10 Fl		Fl	El		CD 170A		13	0D	20 DB			щ	0	2	38 OC	0		
MSX ROM BASIC BIOS MSXGRP - Graphic	1546	1548 1548	1549	1549	S	154B	- 10	- i O	ഹ	- LO	ഹ	1556	ഹ	ഹ	ഹ	ហ	U)	- U)	ц)	ц)	ųγ	1566						1572		
(MSX RO - MSXGRP	4419	4420 4421	4	4	4424	P	- 7	4427	- -	4429	- V'	4431	· · ·	~	4434	~	- -	43	4438	43	4440	4441	44	44	4444	44	44	4447	4448	4449

PAGE 47-2	<pre>;Update cursor position ;Reset X position ;Reset Y position also for general graphics)</pre>
01-Jan-85 on GRP screen	A,8 C,GRPCR),A (GRPACX),A POPALL A (GRPACX),A CHKMOD A,6 CHKMOD A,4*8 1 3,GPRT70 A,4*8 1 3,GPRT70 A,4*8 1 A,8 0C0H C,GPRT80 A,8 0C0H C,GPRT80 A,8 0C0H C,GPRT80 A,8 0C0H C,GRPACY),A C,GPRT80 A,8 0C0H C,GRPACY),A C,GRPACY),A C,GRPACX),A
3.44 t a character	ADD A,8 JR C,GRPCR GPRT60: LD (GRPACX),A ;Update (JPPPAL: JPPPAL: JPPPAL: JPPPAL: JP POPALL GRPACX),A ;Update (GRPACX),A ;Reset X LD (GRPACY),A ;Reset X LD (GRPACY) ADD A,4*8 DB 1 GPRT70: ADD A,4*8 DB 1 CGC 0C0H JR C,GPRT80 ADD A,4*8 DB 1 CGC 0C0H JR C,GPRT80 ADD A,4*8 DB 1 CGC 0C0H JR C,GPRT80 ADD A,4*8 DB 1 C,GPRT80 ADD A,4*8 DB 1 C,GPRT80 CCP 0C0H JR C,GPRT80 CP 0C0H CP 0C0H JR C,GPRT80 CP 0C0H CP
(MSX ROM BASIC BIOS) Macro-80 - MSXGRP - Graphic driver (Print	C6 08 38 06 32 FCB7 C3 08DA C3 08DA C3 08DA C3 08DA C6 15D9 32 FCB9 33 FCB9 33 FCB9 33 FCB9 33 FCB9 38 01 AF C0 33 FCB9 32 FCB9 32 FCB9 18 E2
DM BASIC ? - Grap	1574 1578 1578 1578 1578 1578 1588 1588 1588
(MSX ROM F - MSXGRP -	4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4

(MSX ROM BASIC BIOS) Macro-80 3.44 01-Jan-85 PAGE - MSXGRP - (Routines for general graphics)

			the set of	SCALXY - Clips X,Y to max values in physical size and itage out of range values.		[DE]	[DE] = Y clipped	CARRY is reset if one of the value was out of bound	J	;save [HL]	;save [BC] - X coorginate	;no-error ilag	;Y coordinate to [HL]	;IS Y coordinate negative?	-	;No, positive	; Substitute by 0 is negative	; And set out of bound flag			; Maximum Y+L	;Test [HL] with [DE]	; if carry, not out of bound	;[HL] = 192	;Y = 191 ,maximum Y coordinate		;set out of bound ilag		; save Y and get A to [hu]	; IS X COOLDINATE NEGALIVE?
				ps X,Y to max va ues.		ENTRY [BC] = $X (0 \dots \max X)$,	X clipped,	set if one of th		Η	BC	B,1	DE,HL	А,Н	Α,Α	NC, YPOSTV	нг,0	YNEGTV			DE,0C0H	2 OH	C, SCLYOK	DE, HL	HL		B,0		(SP),HL	А,Н
e 5		SCALXY:	••	; SCALXY - Clips X ; of range values.		; ENTRY [BC] .=	; EXIT [BC] = X clipped,	; CARRY is re	••	HSUG	PUSH	ΓD	EX	ГD	ADD	JR	LD	JR	Y POSTV:		LD	RST	JR	EX	DEC	YNEGTV:	LD	SCLYOK:	EX	ΓD
)										E5	C5	06 01	EB	7C	87	30 05	21 0000	18 08			11 00C0	E7	38 04	EB	2B		00 90		E3	7C
		1599								1599	159A	159B	159D	159E	159F	15A 0	15A2	15A5	15A7		15A7	15AA	15AB	15AD	15AE	15AF	l5AF	15Bl	15Bl	15B2
	4474	4475	4476	4477 4478	4479	4480	4481	4482	4483	4484	4485	4486	4487	4488	4489	4490	4491	4492	4493	4494	4495	4496	4497	4498	4499	4500	4501	4502	4503	4504

PAGE 48-1	;No, positive :Substitute by 0 if necetive	And set out of bound flag	;max X +1	;Test [HL] with [DE]	;[III] = 256	;[HL] = 255 - max X coordinate		;error flag		restore [DE] = Y	:We're in high-resolution mode	Divide both X and V by 4 because and vo						set carry if no error	[BC] = X		restore [HI.]				
01-Jan-85	A, A NC , XPOSTV HL , 0	XNEGTV	DE,0100H	20H C.SCLXOK	DE, HL	HL		B,0	Ц Ц	CHKMOD	Z, HRSSCL	· IJ	Г	ы	ы		A, B		B,H	C,L	ΗL				c screen mode
-80 3.44 eral graphics)	ADD JR LD	JR XPOSTV: ;	ΓD	RST JR	EX	DEC	XNEGTV:	LD COT VOV	DOD TOTA	CALL	JR	SRL	SRL	SRL	SRL	HRSSCL:	LD	RRCA	LD	LD	POP	RET	CHKMOD:	••	; Check current screen mode
SIC BIOS) Macro-80 (Routines for general	87 30 05 21 0000		0010 11	E/ 38 04	EB	2B	(06 00		CD 15D9	28 08	CB 3D	CB 3D	CB 3B	CB 3B		78	0F	44	4D	El	C9			
M BA -	15B3 15B4 15B6	15B9 15BB	15BB	15BF	15C1	15C2	1203	15C3	1505	1506	15C9	15CB	15CD	15CF	15D1	15D3	15D3	1504	15D5	15D6	15D7	15D8	15D9		
(MSX RO - MSXGRP	4505 4506 4507	4508 4509 4510	4511 4513	4513	4514	4515		4518 4518	4519	4520	4521	4522	4523	4524	4525	4526	4527	4528	4529	4530	4531	4532	ŝ	4534	4535

PAGE 48-2	;In what mode are we now? ;Return with the condition flag	s to "C" (address, mask)	Video Ram address Mask Node]	bits, max=255) bits, max=191)	XXX 210	
3.44 01-Jan-85 ics)	LD A, (SCRMOD) SUB 2 RET	MAPXYC - Maps X,Y coordinates Entry: [BC] = X, [DE] = Y	CLOC = [HL] = [A] Bit n-resolution n	<pre>X coord - XXXXXXX (8 76543210 Y coord - YYYYYYY (8 76543210</pre>	CLOC = YYYYXXXXYYY 7654376543210	<pre>= 10000000 000 01000000 001 00100000 010 000100000 011 000010000 011 000010000 100</pre>
) Macro-80 3.4 for general graphics)	; FCAF 02 MAPXYC:	; ; MAPXYC ; Entry:	; Exit: ; CMASK ; [High			CMASK
	15D9 3A F 15DC D6 C 15DE C9 15DF C9					
(MSX ROM BASIC BIOS - MSXGRP - (Routines		4541 4542 4543 4544	4545 4546 4547 4548 4549		4557 4557 45558 4559 8559	4562 4563 4565 4565

PAGE 48-3	101 110 111		(6 bits, max=63)	(6 bits, max=47)	XX	10		if XO=0 (even) if XO=1 (odd)		The boundary check has already been done by a call	so no range checking is needed.	X e Xes:	Check current screen mode	;Multi-color mode	; Save X to D also				;Table of power of two	ı	;read bit mask CMASK	
01-Jan-85	00000100 00000010 00000001	r mode]	d - XXXXXX 543210	I.	λλ	54354321210				boundary ch	so no range	BC	CHKMOD	NZ, MMPXYC	D,C	A ,C	7	C,A	HL , TWOPWR	HL, BC	A, (HL)	
3.44 graphics)		[Multi-color mode	X coord	Y coord	CLOC =			CMASK = 11110000 $CMASK = 00001111$		Note: The l	to SCALXY, s	HSUG	CALL	ЯŲ	LD	LD	AND	LD	LD	ADD	LD	
SIC BIOS) Macro-80 (Routines for general		•• ••								•••	•• •	C5	1 CD 15D9				E6 07				7E	
M BA												15DF	15E0	15E3	15E5	15E6	15E7	15E9	15EA	15ED	15EE	
(MSX RO - MSXGRP	ທິທິທີ	57	4572 4573 4574	4575 4576 4577	4578 4579	4580	4581	4582 4583	4584	ഹ	4586 4587	ഹ	4589	59	59	50	4593	4594	ŝ	ഹ	4597	

PAGE 48-4	;Get Y coordinate			;Get X coordinate	:Get Y coordinate					;Set pattern generator address												Get X position
01-Jan-85	(CMASK),A A,E		00011111B B,A	A,D 11111000B	С,А А.F	00000111B	U	C,A	HL, (GKPCGP) HL, BC	(CLOC), HL	R				power of two	80H.40H.20H.10H	08H,04H,02H,01H			; Map XY for multi-color mode		A,C
) 3.44 al graphics)	LD LD	RRCA RRCA RRCA	AND	LD AND	01 1	AND	OR	0; ;		LD	POP	RET	TWOPWR:		; Table of pow	DB	DB	••	MMPXYC:	; Map XY for m	••	LD
BIOS) Macro-80 tines for general	32 F92C 7B	0F 0F	E6 1F 47	7A E6 F8	4F 7B	E6 07	Bl		ZA F3CB 09	22 F92A	c1	60				80 40 20 10	04 02					79
MSX ROM BASIC MSXGRP - (Rou	15EF 15F2	15F3 15F4 15F5	15F6 15F8	15F9 15FA	15FC 15FD	15FE	1600	1601	1605 1605	1606	1609	160A	160B			160B	160F		1613			1613
(MSX RO - MSXGRP	4598 4599	4600 4601 4602	4603 4604	4605 4606	4607 4608	4609	4610	4611	4612 4613	4614	4615	4616	4617	4618	4619	4621	4622	4623	4624 1625	4626	4627	4628

PAGE 48-5	: Firen or odd?	:Assume even	: Good assumption	: Odd		;Set up mask pattern	1				;Get lower byte	1									:Get higher byte	Load start address of pattern table				
01-Jan-85		A,11110000B	NC, MMPXY1	A,00001111B		(CMASK), A	A,C	A, A	Α,Α	11111000B	C,A	Α,Ε	01118	C	C,A	A, E				0111B	B,A	HL, (MLTCGP)	HL, BC	(CLOC), HL	BC	
3.44 L graphics)	RRCA	LD	JR	LD	:IXYAM	LD	LD	ADD	ADD	AND	ГD	LD	AND	OR	LD	ГD	RRCA	RRCA	RRCA	AND	LD	LD	ADD	LD	POP	RET
SIC BIOS) Macro-80 (Routines for general	OF		30 02		I	32 F92C	79	87	87	E6 F8	4F	7B	E6 07	Bl	4F	7B	0F	OF	0F	E6 07	47	2A F3D5	60	22 F92A	c1	60
4 BA -	1614	1615	1617	1619	161B	161B	161E	161F	1620	1621	1623	1624	1625	1627	1628	1629	162A	162B	162C	162D	162F	1630	1633	1634	1637	1638
(MSX RON - MSXGRP	4629	4630	4631	4632	4633	4634	4635	4636	4637	4638	4639	4640	4641	4642	4643	4644	4645	4646	4647	4648	4649	4650	4651	4652	4653	4654

49 PAGE 01-Jan-85 (MSX ROM BASIC BIOS) Macro-80 3.44 - MSXGRP - (Routines for general graphics)

4655					
4656	1639		FETCHC:		
4657			••		
9			; FETCHC - Rea	FETCHC - Reads the value of the graphics	he graphics accumulater
4659			••		
4660			; Exit: [HL] =	; Exit: [HL] = CLOC, [A] = CMASK	SK
4661			••		
4662	1639	3A F92C	LD	A, (CMASK)	
4663	163C	2A F92A	LD	HL, (CLOC)	
4664	163F	60	RET		
4665	1640		STOREC:		
4666			••		
4667			; STOREC - Set	STOREC - Sets the graphics accumulater	cumulater
4668			••		
4669			; Entry: [HL]	Entry: [HL] = CLOC, [A] = CMASK	ASK
4670			••		
4671	1640	32 F92C	ΓD	(CMASK), A	
4672	1643	22 F92A	LD	(CLOC), HL	
4673	1646	60	RET		
4674	1647		READC:		
4675			••		
4676			; READC - Get	the attribute of	READC - Get the attribute of the current graphics accumulater
4677			; position		
4678			••		
4679	1647	C5	HSUG	BC	
4680	1648	E5	PUSH	用	
4681	1649	CD 1639	CALL	FETCHC	;Get CLOC and CMASK
4682	164C	47	LD	B,A	; Save CMASK
4683	164D	CD 15D9	CALL	CHKMOD	;Check current screen mode
4684	1650	20 IA	JR	NZ, MREADC	;Multi-color mode
4685	1652	CD 07D7	CALL	RDVRM	;Read VDP's VRAM (pattern)

tor denerat draphitcs/		
AND B		;Extract specified pixel
PUSH AF		Save whether the pixel is on or off
2000 LD BC,G	BC, GRPDIF	1
ADD	, BC	
07D7 CALL RDVRM	VRM	;Read VDP's VRAM (color)
LD B,A	Ā	; Save this to B
POP AF		;Restore condition
LD A,B	В	;Restore color
04 JR Z,RE	Z, READC1	;Specified dot is off, return
READCO:		
RRCA		;Specified dot is on, return foreground color
RRCA		
RRCA		
RRCA		
READC1:		
0F AND 0FH	F	;Make it a legal value
POP HL		'n
POP BC		
RET		
MREADC:		
••		
07D7 CALL RDVRM	/RM	;Read VRAM
INC B		;Check if specified pixel is even or odd
DEC B		
1667 JP P, RE	P, READC1	;Odd, return lower nibble
	READOD	

PAGE 50	Sets the attribute (color, reverse, etc) to be uture actions. A] = Attribute rry set if illegal value	;Must be less than 16	the point indicated by the graphics accumulater s except AF must be preserved.	;Check current screen mode	;Multi-color mode
3.44 01-Jan-85 graphics)	SETATR: SETATR - Sets the attribute (colo sused in future actions. Entry - [A] = Attribute Exit - carry set if illegal value	C CP 16 CCF C RET C LD (ATRBYT),A RET SETC:	SETC - Sets the point ind to ATTRBYT All registers except AF m	EE .J.J	JR NZ,MSETC PUSH DE CALL PATWRT POP DE POP BC POP HL
MSX ROM BASIC BIOS) Macro-80 MSXGRP - (Routines for general	1676 1	1676 FE 10 1678 3F 1679 D8 167A 32 F3F2 167D C9 167E S	~ ~ ~ ~ ~ ~ ~ ~	E5 CD CD CD	1686 20 08 1688 D5 1689 CD 186C 168D D1 168D C1 168E E1
(MSX RO - MSXGRP	4713 4714 4715 4715 4716 4718 4719 4720 4720	22222222	4729 4730 4731 4732 4733	4734 4735 4736 4737	4738 4739 4740 4741 4742 4743

PAGE 50-1		mode	;Save CMASK in [B]	;Read VRAM			;Leave another unaffected			;Get specified color	Check if even or odd		: Odd						;Form new color	Write new pattern	4			(Graphic cursor movements)
01-Jan-85		pixel in multi-color mode	B,A	RDVRM	C,A	A, B		U	C,A	A, (ATRBYT)	В	В	P, MSETCI	A,A	A, A	Α,Α	A, A		U	WRTVRM	BC	НĹ		i
3.44 l graphics)		; Set a pixel ;	ГD	CALL	LD	ГD	CPL	AND	LD	LD	INC	DEC	дГ	ADD	ADD	ADD	ADD	MSETC1:	OR	CALL	POP	POP	RET	SUBTTL - MSXGRP
SIC BIOS) Macro-80 (Routines for general	6J		47	CD 07D7	4F	78	2F	Al	4 F	3A F3F2	04	05	F2 16A5	87	87	87	87		Bl	CD 07CD	CI	El	C9	
BA	168F 1690		1690	1691		1695	1696	1697	1698	1699	169C	169D	169E	16A1	16A2	16A3	16A4	16A5	16A5	16A6	16A9	16AA	16AB	
(MSX ROM - MSXGRP	4744 4745 4746	4/4/ 4748	4749	4750	4751	4752	4753	4754	4755	4756	4757	4758	4759	4760	4761	4762	4763	4764	4765	4766	4767	4768	4769	4770

51		
PAGE		
01-Jan-85		
3.44		
(MSX ROM BASIC BIOS) Macro-80	- MSXGRP - (Graphic cursor movements)	4771

UPC, DOWNC, RIGHTC, LEFTC These are the C relative movement routines. They adjust the current graphics accumulater in the indicated direction without checking boundary conditions.		- move l pixel right	Keturn carry set it alleauy on bolger	臣	CHKMOD	GT	FETCHC ;Get CLOC, CMASK	;Move l pixel right	NC,HRZMV1 ; Within byte, just change CMASK	A,L ;Get low byte of CLOC	0F8H	0F8H ;On right edge?	A,80H ;Assume not	NZ, RGHTCl ; Goot assumption	ONBRD1 ; On border, set carry and return			; RIGHTC - move l pixel right	
; ; UPC, DOWNC, RIGHTC, LEFTC ; ; These are the C relative ; adjust the current graphi ; direction without checkin	;	; TRIGHT - move	; keturn carry ;	HSUG	CALL	JL	CALL	RRCA	JR	ГD	AND	CP	LD	JR	đ	RIGHTC:	••	; RIGHTC - move	•
				E5			CD 1639	0F	30 4B	7D	E6 F8	FE F8	3E 80	20 10	C3 175A				
	16AC			16AC	16AD	16B0	16B3	16B6	16B7	16B9	16BA	16BC	16BE	16C0	16C2	16C5			
4771 4772 4773 4775 4775 4777 4777	4779 4780 4781	4782 4783	4785 4785	4786	4787	4788	4789	4790	4791	4792	4793	4794	4795	4796	4797	4798	4799	4800	4801

PAGE 51-1						;within byte, just change CMASK			;Load offset to new position	;Change CLOC also				y on border					;Get CLOC and CMASK	;Move 1 pixel left	;Within byte boundary, just change CMASK	;Check if we're on left edge		;Assume not	;Good assumption	;We're on border, set carry and return							
01-Jan-85	甩	CHKMOD	NZ, MRGTC	FETCHC		NC, HRZMV1		DE	DE,8	HRZMOV			- move l pixel left	set if already on border		甩	CHKMOD	NZ, MTLFT	FETCHC		NC, HRZMV1	A,L	0F8H	A,1	NZ, LEFTCI	ONBRDI		×	- move l pixel left		НĹ		
ro-80 3.44 movements)	HSUA	CALL	đ	CALL	RRCA	ЯĻ	RGHTC1:	HSUG	ΓD	ЯĊ	TLEFT:	••	; TLEFT - move	; Return carry		HSUA	CALL	JP	CALL	RLCA	ЯĽ	ГD	AND	ΓD	ЯĻ	ĥ	LEFTC:	••	; LEFTC - move	••	HSUG		
) Macr cursor	E5	CD 15D9		CD 1639	0F	30 32		D5		18 27						ES		-	CD 1639	07	30 IF		E6 F8		20 OF	18 6C					E5		
1 BA -	16C5	16C6	16C9	16CC	16CF	16D0	16D2	16D2	16D3	16D6	16D8					16D8	16D9	16DC	16DF	16E2	16E3	16E5	16E6	16E8	16EA	16EC	16EE				16 EE		
(MSX RO - MSXGRP	4802	4803	4804	4805	4806	4807	4808	4809	4810	4811	4812	4813	4814	4815	4816	4817	4818	4819	4820	4821	4822	4823	4824	4825	4826	4827	82	82	83	m	4832		

PAGE 51-2		;move left l pixel ;within byte boundary, just change CMASK	;Load offset to new position	;Add offset to new position ;Update pattern address	;Update CMASK ;Clear carry	Return carry set if already on screen border. PUSH HL	
01-Jan-85	CHKMOD NZ , MLFTC FETCHC	NC, HRZMV1	DE, OFFF8H	HL, DE (CLOC), HL DE	JD (CMASK),A AND A OP HL XET *ET - move l pixel down.	' set if already HL	DE HL, (CLOC) CHKMOD NZ, MTDNC HL HL, (GRPCGP) DE, 1700H
3.44	CALL JP CALL		PUSH LD	ADD LD POP	U M M U	rn carry PUSH	PUSH LD JP PUSH LD LD
) Macro-80 cursor movements)	15D9 17AC 1639	19 LEFTC1:	FFF8 HRZMOV:	92A	F92C II I I I I I I I I I I I I I I I I I	. Retu	F92A 15D9 17C6 F3CB 1700
		07 30 0	D5 11 F	19 22 F D1	32 F A7 E1 C9	E5	ロック 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
A BA -	16EF 16F2 16F5	16F8 16F9 16FB	16FB 16FC 16FF	16FF 1700 1703	1704 1704 1708 1709 1709	170A	170B 170F 1712 1712 1715 1715 1716 1719
(MSX ROM - MSXGRP	4833 4834 4835	4836 4837 4838	4840 4840 4841	4842 4843 4844 4844	4 8 8 4 4 4 4 8 8 4 4 4 4 4 8 8 4 4 4 4	4853 4854 4855 4855	4857 4858 4858 4859 4860 4861 4862 4863

PAGE 51-3		;Test [HL] with [DE] ;Looks like on border?	; No : Possibly on border		;Really?	; No	Yes, set carry and return											;move down l pixel	; Prepare for boundary check	Load possible offset to new location					already on screen border.	
01-Jan-85	HL, DE DE, HL HL	2 0H	C, DWNC10 A.L	A	7	NZ, DWNC10	ONBRDR			- movel pixel down	8	HL	DE	HL, (CLOC)	CHKMOD	NZ, MDNC		HL	A,L	DE,0F8H	VRTMOV					
3.44 S)	ADD EX POP	RST	ЯГ U Л	INC	AND	JR	JR	•0		DOWNC - move		PUSH	HSUA	LD	CALL	ЧĻ	10:	INC	LD	LD	ЯÇ			PC - move	Return carry set if	
OS) Macro-80 ic cursor movements)	19 EB El	E7	38 13 7D	ßC	E6 07		18 2F	DOWNC		DO .	••	25	D5	F92	CD 15D9	C2 17DC	DWNC10:	23	D	11 00F8	831	TUPC:	•••		; Re	
MSX ROM BASIC BIOS MSXGRP - (Graphic	171C 171D 171E	171F	1720 1722				1728	172A								1732 0				2	3A	173C				
(MSX RO - MSXGRP	999	4867 4868	4869 4870	4871	4872	4873	4874 4875	4876	4877	4878	4879	4880	4881	4882	4883	4884	4885	4886	4887	4888	4889	50	יע	690	4893 4894	

PAGE 51-4	<pre>;Test [HL] with [DE] ;Iooks like on border? ;No ;Possibly on border ;Really? ;No ;Set carry indicating we're on border</pre>	;get current position
01-Jan-85	HL DE HL, (CLOC) CHKMOD NZ, MTUPC HL, (GRPCGP) DE, 0100H HL, DE DE, HL HL 20H NC, UPC10 A, L 7 NZ, UPC10 DE HL HL	pixel up HL DE HL,(CLOC) CHKMOD
30 3.44 rements)	PUSH PUSH LD CALL JP PUSH LD PUSH LD LD LD LD LD LD ADD EX RST ADD ADD SCF POP ONBRDI: SCF POP VPC: RET	; UPC - move 1 ; PUSH PUSH LD CALL
SIC BIOS) Macro-80 (Graphic cursor movements)	E5 D5 C2 15D9 C2 15D9 C2 15D9 C2 17E3 E5 E3 E1 E1 E1 E1 E1 D1 00 E1 20 0F C9 C9 C9 C9 C9 C9 C1 C1 C1 C1 C1 C1 C1 C1 C1 C1 C1 C1 C1	E5 D5 2A F92A CD 15D9
MSX ROM BASIC MSXGRP - (Gra	1122 1223 1223 1223 1223 1223 1225 12555 12555 12555 12555 12555 12555 12555 12555 12555 125	175D 175E 175F 1762
(MSX RO) - MSXGRP	4895 4895 48995 49003 49008 49008 49110000000000	4921 4921 4923 4923 4925

					urn
PAGE 51-5	Prepare for boundary check move up 1 pixel Load possible offset to new location	;Crossed boundary? ;No, it's okay ;Get new location	;Update pattern address ;Clear carry	in multi-color mode ;Assume CMASK is even ;Within byte, just change CMASK	;On right edge? ;No, move to next pixel ;We're on right edge, set carry and return
01-Jan-85	NZ, MUPC A, L HL DE, OFF08H	7 NZ,VRTMV1 HL,DE	(СLOC), НL А DE НL	/ement ents] HC I ZMV1	0F8H NZ , MRGTC1 ONBRD1
3.44	JP LD DEC	AND JR ADD	LD AND POP POP RET	ics cur izontal CCALL AND JP LD AND	an R
) Macro-80 cursor movements)	UPC10:	VRTMOV: VRTMV1:	MTRGT:	Graph ; [Hor ;	MRGTC:
	C2 17F8 7D 2B 11 FF08		22 F92A A7 D1 E1 C9		FE F8 20 0B 18 CF
(MSX ROM BASIC BIOS - MSXGRP - (Graphic	1765 1768 1768 1769 176A	176D 176D 176F 1771 1772	1772 1775 1776 1777 1778 1778	1779 1776 1776 1776 1775 1782	1785 1787 1789 178B
(MSX ROM - MSXGRP	4926 4927 4928 4929 4930	4931 4932 4934 4934	4936 4937 4938 4939 4940	44444444444444444444444444444444444444	4952 4953 4954 4955 4956

PAGE 51-6 18	;Assume CMASK is even ;Good assumption	;Next pixel is 8 byte far ;from the current position	;Assume CMASK is odd	;Good assumption, just change CMASK ;On left edge? ;No ;We're on left edge, set carry and return	;Assume CMASK is odd ;Good assumption, just change CMASK
01-Jan-85	FETCHC A A,0FH M,MHZMVI	DE DE,8 A,0F0H MHCMOV	FETCHC A A,0F0H	P, MHZMV1 A, L 0F8H NZ, MLFTC1 0NBRD1 0NBRD1	FETCHC A,0F0H P,MHZMV1 DE DE,0FFF8H A,0FH
3.44	CALL AND LD JP	PUSH LD LD JR	CALL AND LD	JP LLD JR JR JR	CALL AND JP PUSH LD LD
ro-80 movements)	MRGTC1:		, ;	MLFTC:	MLFTC1: MHCMOV:
) Macr cursor	CD 1639 A7 3E 0F FA 17C0	D5 11 0008 3E F0 18 1F	CD 1639 A7 3E F0	17C F8 AE	CD 1639 A7 3E F0 F2 17C0 D5 11 FFF8 3E 0F
M BASIC - (Gra	178B 178E 178F 1791 1791	1794 1795 1798 1798	179C 179C 179F 17A0	17A5 17A6 17A6 17A8 17A8 17AA	17 AC 17 AF 17 B0 17 B0 17 B5 17 B5 17 B6 17 B9 17 B8
(MSX ROM BASIC BIOS - MSXGRP - (Graphic	4957 4958 4959 4960 4961	4962 4963 4964 4965	4967 4968 4969 4970 4971	L66 L66 L66 L66 L66 L66	4979 4980 4981 4983 4984 4985 4985

PAGE 51-7	;Clear carry ;Possibly on border? ;No ;Check if least 3 bits are all 1's ;No ;Check if least 3 bits are all 1's	;Move down 1 byte
01-Jan-85	ADD HL, DE LD (CLOC), HL, DE ZMV1: LD (CLOC), HL BD (CMASK), A AND A AND A POP HL RET NC: NC: NC: NC: NC: NC: NC: NC: NC: NC:	HL
3.44	ADD LD FOP POP PUSH LD PUSH LD PUSH LD TR SCF POP POP POP POP POP POP POP	INC
-80 Wements)	MHZMV1: MTDNC: ; [Ver ; ; MDNC:	
31OS) Macro-80 phic cursor movements)	19 22 F92A D1 32 F92A 87 87 81 19 23 F3D5 11 0500 11 0500 11 0500 11 10 20 20 20 20 20 20 20 20 20 20 20 20 20	23
MSX ROM BASIC BIOS MSXGRP - (Graphic	1785 1786 1786 1786 1760 1700 1700 1705 1705 1706 1706 1706 1709 1709 1709 1709 1709 1709 1709 1700	17DC
(MSX ROM - MSXGRP	4 4 4 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9	5018

182				turn	
PAGE 51-8	;Load possible offset to next block ;Check	;Possibly on border?	;Test [HL] with [DE] ;No ;Check if we're top of a block	;We're on top border, set carry and return	;Move up l byte ;Load possible offset to next block ;Wrapped to next block? ;No ;Yes, add up offset to next block ;Clear carry
01-Jan-85	A,L DE,0F8H MVTMOV	НL НL, (MLTCGP) DE,0100H НL,DE HL	20H NC, MUPC A, L 7	Ne, MUFC DE HL A, L	HL DE, 0FF08H 7 NZ, MVTMV1 HL, DE (CLOC), HL A
3.44	LD LD RL	PUSH LD LD ADD POP	RST JR LD AND	JR SCF POP RET LD	
ro-80 movements)	MTUPC:			MUPC:	MVTMOV:
) Macı cursor	7D 11 00F8 18 1A	E5 2A F3D5 11 0100 19 F1		zu 04 37 E1 C9 7D	2B 11 FF08 E6 07 20 01 19 22 F92A A7
1 BASIC BIOS - (Graphic	17DD 17DE 17E1 17E3	17E3 17E4 17E7 17E7 17EA	17EC 17EC 17ED 17EF 17F0	1762 1764 1765 1766 1767 1768 1768	17F9 17FA 17FD 17FF 17FF 1801 1802 1802 1802
(MSX ROM BASIC BIOS - MSXGRP - (Graphic	5019 5020 5021 5023 5023	5024 5025 5026 5027 5028	5029 5030 5031 5032	5033 5034 5035 5037 5038 5039 5040	5041 5042 5043 5044 5045 5045 5046 5047 5049

51-9	
PAGE	Misc.)
	and
-85	fill
01-Jan-85	DE HL (Box
3.44	POP DE POP HL RET SUBTTL -MSXGRP- (Box fill and Misc.)
MSX ROM BASIC BIOS) Macro-80 MSXGRP - (Graphic cursor movements)	SUBTTL
BIOS phic	D1 E1 C9
DM BASIC	1806 1807 1808
(MSX RC - MSXGRI	5050 5051 5052 5053

PAGE 52			; NSETCX - Performs SETC, RIGHTC [HL] times		In fact, SETC and RIGHTC are never called to increase speed, and for the reason described below.		Since only 2 colors can be displayed in a byte, some special	handling is required when a full-byte is set when writing left	or right extras. In this case, we can completely ignore the	background color for that byte, allowing 2 colors displayed			coyed.			;Multi-color mode	; Save count	;Get CLOC and CMASK	;Reget count, save CLOC	;Beginig at leftmost position?	;Yes, no extra dots at the left	;Save mask pattern*2				;Decrement pixel count	;The whole dots are within a byte		
01-Jan-85			forms SETC, RI		C and RIGHTC a reason describ		colors can be	required when	ras. In this c	olor for that			; All registers may be destroyed.		CHKMOD	NZ, MNSTCX	НĹ	FETCHC	(SP),HL	A,A	C,NSTC20	AF	BC,0FFFH			HL, BC	NC , NSTCSP		NC, NSTC10
3.44	NSETCX:		NSETCX - Per		In fact, SET and for the		Since only 2	handling is	or right ext	back ground c	in a byte.		All register		CALL	JP	HSUG	CALL	EX	ADD	JR	HSUG	LD	RRCA	NSTC10:	ADD	ĥ	RRCA	ξ
Macro-80 Misc.)	N	••		••			••	••		••	••	••	••		•			•					Fr.		Z				
l and															CD 15D9	C2 18BB	ES	CD 1639	E3	87	38 18	F5	01 FFFF	OF		60	30 45	0F	30 FA
(Box	1809														1809	180C	180F	1810	1813	1814	1815	1817	1818	181B	181C	181C	181D	181F	1820
-MSXGRP-	5055	5056	5057	5058	5059 5060	5061	5062	5063	5064	5065	5066	5067	5068	5069	5070	5071	5072	5073	5074	5075	5076	5077	5078	5079	5080	5081	5082	5083	5084

PAGE 52-1	:Restore mask pattern*2	PORT left-extra nattern	Repet CLOF says count		Write to VRAM (nattern and color)	Restore (T.OC	Load an offset to next byte	Update pattern address	:Reget count, save CLAC		:Get low byte of count	[A] = count mod 8	save count after byte boundary						; [HL,] = [HI,] / 8		;Reget CLOC	;[B]=counter	:No dots in this part		Make specified color a back ground color.	Write to VRAM (nation)		Calculate address of color table	Get specified color	Write to VRAM (color)	Load an offset to next byte
01-Jan-85	AF	А	(SP), HL	ΗL	PATWRT	Η	DE,8	HL , DE	(SP), HL		A,L	7	C,A	A,H		Α,Γ				00111111B	Ħ	B,A	Z,NSTC40		A	WRTVRM	DE, GRPDIF	HL , DE	A, (ATRBYT)	WRTVRM	DE, GRPDIF+8
3.44	POP	DEC	EX	HSNA	CALL	POP	LD	ADD	EX	NSTC20:	LD	AND	LD	ГD	RRCA	LD	RRA	RRCA	RRCA	AND	POP	LD	JR	NSTC30:	XOR	CALL	LD	ADD	LD	CALL	LD
5) Macro-80 and Misc.)		0	~	10	0 186C		0008			SN		07	_		_		_			3F			14	LSN			2000				2008
BASIC BIOS (Box fill	1822 Fl	1823 3D	1824 E3	1825 E5	1826 CD	1829 El	182A 11		182E E3	182F	182F 7D		1832 4F			1835 7D						4	183D 28	183F	ſĿ.	0	1843 11	6	47	84A CD	84D 11
(MSX ROM BASIC BIOS -MSXGRP- (Box fill a	2		5087]											5098 I		5100 1		5102 1			5105 1				5109 I		111	5112 1	5113 I.	5114 1	5115 1

186	table	1)
	pattern table	111111) 1111000) (001
	ne char boundary ght extra for 'right-extra'	Get mask pattern for the right (11111100) ;Get mask pattern for the left (00011111) ;Get mask pattern for the left (00011100) ;Make a pattern to write (00011100) ;Restore CLOC ex.
	e lar boun lt extra or 'righ	rn for t rn for t to writ x.
52-2	l do in ri ess rn	det mask pattern ;Get mask pattern ;Save it ;Get mask pattern t ;Make a pattern t
PAGE	;Bump CLOC ;Loop until do ;dot count in ;No dots in ri ;Save CLOC ;Load address ;Get pattern	;Get mas) ;Get mas) ;Get mas] ;Make a]
35	HL, DE ; Bu NSTC30 ; LG C ; dd M ; N HL, RGTEXT ; LG HL, BC ; S A, (HL) ; G NSTC50 ; G	сн, оғен, с
01-Jan-85	HL, DE NSTC30 C M HL, RGTEXT HL, BC A, (HL) NSTC50 R0H OC0H	BUR, UCUN, UEUN, R, A B, A B, A AF A B HL
3.44	ADD DJNZ DJNZ DEC RET PUSH LD JR JR	DB DB ADD DEC CPL LD POP AND POP
0	NSTC40: RGTEXT:	NSTCSP: ; NSTC50:
Macro-80 Misc.)		0년 1911년 1911년
BIOS) fill and	ы <u>н</u> о (80 C0 F8 FC 37 87 87 87 87 87 87 87 87 87 87 87 87 87
MSX ROM BASIC BIOS) MSXGRP- (Box fill and	1850 1851 1853 1853 1855 1855 1855 1855 1855	185D 1861 1864 1864 1865 1865 1868 1868 1868 1868 1868
(MSX RO) -MSXGRP-	5116 5117 5118 5119 5120 5122 5122 5123 5124 5125 5126	5128 5130 5130 5131 5133 5133 5135 5136 5136 5137 5138 5139 5139

JE 53	Write a pattern to high-resolution screen - Pattern to be written - Address of pattern table - Color of this pattern	<pre>;Save pattern to be added ;Read VRAM (pattern) ;Save current pattern ;Form address of color table ;Read from VRAM (color) ;Read from VRAM (color) ;Extract background color ;Save background color ;Save background color ;Restore foreground and background color ;Restore foreground and background color ;Set foreground color in the upper 4 bit ;[B] has the specified pattern, ;[C] has the specified pattern, ;[D] has the current pattern, ;[D] has the current background color ; shifted left 4 times, ;[E] has the current background color</pre>	;[HL] has the address of color table. ;Get specified color ;Same with current background? ;Yes
PAGE	attern to high-red to be written of pattern table f this pattern	;Save ;Read ;Form ;Form ;Form ;Read ;Save ;Resto ;[B] h ;[C] h ;[D] h ;Set f	; [HL ; Get ; Sam ; Yes
01-Jan-85	ite a pattern to high Pattern to be written Address of pattern ta Color of this pattern	B, A RDVRM C, A DE, GRPDIF HL, DE RDVRM AF E, A AF E, A AF D, A	A, (ATRBYT) E Z, SAMEBG A, A
3.44	PATWRT: PATWRT - Wr PATWRT - Wr PATWRT - Wr PATWRT - ML PATWRYT - 0	LD CALL LD LD ADD CALL PUSH PUSH POP SUB LD	LD CP JR ADD
BASIC BIOS) Macro-80 (Box fill and Misc.)		47 CD 07D7 4F 11 2000 19 CD 07D7 F5 F1 57 57	3A F3F2 BB 28 19 87
1 BASIC (Box f	186C	186C 186D 1870 1871 1874 1878 1878 1878 1878 1878 1878	187F 1882 1883 1885
(MSX ROM BASIC BIOS -MSXGRP- (Box fill a	5141 5142 5143 5144 5145 5145 5146 5148 5149	5150 5151 5152 5153 5155 5155 5155 5160 5162 5162 51663 51663 51663 51663 51663 51663 51663 51663 51663 51663 51663 51664 51663 51664 51664 51664 51664 51664 51664 51666 51667 51666 51667 5167 51	5168 5169 5170 5171

188	with current foreground?	;Save new foreground color ;All pixels are going to be set? ;Yes, Spock will use a new repair technique ;logically	Save current background color Write to VRAM (pattern) Restore current background in [E] Restore color table address Restore new foreground color in upper 4 bits of [Acc] Form new foreground and background color Write to color table	
53-1	with cu	Save new for All pixels a Yes, Spock w Jogically	Write to VRAM (Restore current ba Restore culor t Restore color t Restore new for 4 bits of [Acc] Form new foregr Write to color	
PAGE	; Same	; Save ; All F ; Yes, ; logic	Nurite Nurite Resto Resto Nurite	
01-Jan-85	A, A A, A A, A D Z, SAMEFG	AF A, B C C OFFH Z, PATWRL HL	DE SAMEFG DE HL AF JMPWRT A, B C C	A, B C DE, GRPDIF
3.44	ADD ADD ADD CP JR	PUSH LD OR JR PUSH		
0			SAMEBG:	SAMEFG: WTPTAB:
Macro-80 3 Misc.)	Q		18A2 1A	2 0 0 0
BASIC BIOS) (Box fill and		• [4,]	D1 CD1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1	
	1886 1887 1888 1888 1889 188A	1880 1885 1885 1885 1885 1891 1893	.ଚରରରର ରଚର ରଚୟ 🕯	18A2 18A2 18A4 18A4 18A4
(MSX ROM -MSXGRP-	5172 5173 5174 5175 5176	5177 5177 5178 5179 5180 5181 5182 5183	5184 5185 5186 5185 5189 5192 5193 5193 5193	5198 5199 5200 5202

PAGE 53-2	<pre>;Write to pattern table ;Discard new foreground color ;Reget specified pattern ;Forget current background color, 'cause ;There's no background, we display ;there's no background color. ;Write to pattern table ;Write to pattern table ;Get new color (this will be the ;background color) ;Add current foreground color ;Write to VRAM (color)</pre>	
01-Jan-85	HL, DE JMPWRT AF A, B HL DE WTPTAB DE HL A, (ATRBYT) D WRTVRM	
3.44	ADD JR POP LD CPL PUSH PUSH PUSH POP POP POP CALL CD OR OR	
Macro-80 Misc.)	PATWRL: ; JMPWRT:	
BASIC BIOS) M (Box fill and M	19 18 0E F1 78 2F 78 25 25 25 25 25 21 84 21 82 82 82 82 82	
BASIC (Box f	1887 1888 1888 1888 1888 1888 1888 1888	
(MSX ROM BASIC BIOS) -MSXGRP- (Box fill and	5203 5204 5205 5206 5208 5209 5212 5213 5214 5214 5214 5215 5215 5215 5216 5216 5218	

PAGE 54	;Save counter ;Set pixel ;Move to right ;Restore counter r CIRCLE
01-Jan-85	MNSTCX: NSETCX for multicolor screen PUSH HL ;Save of CALL SETC ;Save of CALL RIGHTC ;Move to POP HL ;Restor DEC L JR NZ,MNSTCX RET GTASPC: GTASPC - load aspect ratio for CIRCLE CALL HL, (ASCPCT1) EX DE,HL LD HL, (ASCPCT2) RET SUBTTL -MSXGRP - (Routines for paint)
3.44	MNSTCX: MNSTCX for n NSETCX for n PUSH CALL POP DEC JR CALL POP DEC JR RET CALL CALL CALL CALL CALL CALL POP DEC JR RET SUBTTL - MSXGRI
MSX ROM BASIC BIOS) Macro-80 MSXGRP- (Box fill and Misc.)	E5 E5 CD 167E E1 E1 20 F5 C9 E3 E3 E3 E3 E3 E3 E3 E3 E3 E3 E3 E3 E3
BASIC] (Box f	18 BB 18 BB 18 BC 18 BE 18 BE 18 C2 18 C3 18 C3 18 C4 18 C4 18 C5 18 C7 18 C6 18 C7 18 C6 18 C7 18 C7
(MSX ROM -MSXGRP-	5220 5221 5221 5223 5224 5223 5233 5233 5233 5233 5233

PAGE 55	<pre>er color ;Save specified color ;In what mode are we now? High-resolution mode ;Legal value? ;Legal value? ;Carry means illegal ;Carry means illegal ;Discard specified color ;Always igmore specified border ;Always legal ;Set border color ;Always legal ;Set border color ;Return with the condition ;Return with the condition ;Return with the condition ;Initialize PNTCNT ;Initialize PNTCNT ;Initialize PNTCNT ;Initialize PNTCNT ;Multi-color mode ;Multi-color mode</pre>	
01-Jan-85	NTINI: PNTINI - Initialize border color PUSH AF Save s CALL CHKMOD ;In who JR Z, PNTHRS ;High POP AF ; Dum ;Legal CCF 10H ;Legal CCF 10H ;Legal CCF 20P AF ;Discar NTHRS: POP AF ;Discar POP AF ;Discar LD A, (ATRBYT) ;Always AND A A, (ATRBYT) ;Always AND A ;ATRBYT) ;Always AND A ;ATRBYT ;ATRBYT ;ATRBYT ;ATRBYT ;ATRBYT ;ATRAYS AND A ;ATRBYT ;ATRBYT ;ATRBYT ;ATRAYS AND A ;ATRBYT ;ATRBYT ;ATRBYT ;ATRAYS AND A ;ATRBYT ;ATRAYS ;ATRAY	3
3.44	PNTINI: FNTINI - IN PUSH PUSH CALL JR POP CP CP CP CP CP CP CP CP CP C	
BIOS) Macro-80 ines for paint)	F5 CD 15D9 28 06 F1 FE 10 3F 18 05 3A F3 A7 A7 A7 A7 A7 A7 A7 A7 A7 A7 A7 A7 A7	
MSX ROM BASIC BIOS MSXGRP - (Routines	18CF 18CF 18CF 18D0 18D5 18D6 18D6 18D6 18D6 18D6 18D6 18D6 18D7 18D7 18C7 18E4 18E4 18E4 18E4 18E4 18E4 18E6 18E6 18E6 18E6 18E6 18E6 18E6 18E6	
(MSX RC -MSXGRP	5241 5241 5243 5245 5245 5245 5249 5254 5254 5255 5255	

192	<pre>d painting, l otherwise. e for 'suspend painting' xel count to draw right ixel changed' flag ;Remember to suspend or not ;Clear 'pixel changed' flag ;Set border color to [B] for comparison ;Read current color ;Still on border? ;Still on border? ;Still on border? ;No, start painting ;All pixels tested? ;Yes ;Not yet out of screen, continue ;Xes ;Not yet out of screen, continue ;All pixels has border attribute on ;this row, let BRDCNT be 0, and return it is found. Start painting ;Check if pixel changed ;Save BRDCNT ;Get current CLOC, CMASK</pre>
PAGE 55-1	Pi o o o o o o o o o o o o o o o o o o o
01-Jan-85	
3.44	<pre>; [B] set to 0 is need to ; Workl = Temporary Work2 = Save area Work3 = Save area Work3 = Save area LD A,B LD (RUNFLG), XOR A LD A,B LD A,B LD A,B CNR3),A LD A,B CNR3),A LD A,B CNR3),A LD A,B CNR3),A LD A,B CNR3),A LD A,B CNR3 LD A,C B A,C B A,D B,A B,A CANR2 CP B JR NZ,SCANR2 DE CP B JR NZ,SCANR2 DE CP B JR NZ,SCANR2 DE CP B JR NZ,SCANR2 DE CP B JR NZ,SCANR2 CANR1 LD A,D CR E B CANR1 LD A,D C CP B JR NZ,SCANR2 DE CP B CALL TRIGHT JR NC,SCANR1 LD DE,0 RET SCANR2 CALL NC,SCANR1 JR NC,SCANR</pre>
BIOS) Macro-80 ines for paint)	78 32 F866 AF 32 F866 AF 33 FCB2 47 47 E8 20 0D 18 7A 20 0D 18 7A E3 C0 16AC 30 F1 11 0000 C9 C1 16AC 30 F1 11 0000 C9 C0 1639 CD 1639 CD 1639
<pre>A BASIC BIOS - (Routines</pre>	18ED 18ED 18EE 18F5 18F5 18F5 18F5 18F5 18F5 18F5 18F5
(MSX ROM BASIC BIOS -MSXGRP - (Routines	5272 5273 5274 5274 5275 5279 5279 5281 5285 5288 5288 5288 5288 5288 5289 5299 529

193	/alue	
PAGE 55-2		;Restore BRDCNT
01-Jan-85	(CSAVEA), HL (CSAVEA), A DE, 0 DE TRIGHT C, SCANR4 READC B Z, SCANR4 CHKCHG B Z, SCANR4 CHKCHG CHKCHG C Z, SCANR4 CHKCHG C Z, SCANR4 CHKCHG C Z, SCANR4 CHKCHG C Z, SCANR4 CHKCHG C Z, SCANR4 CHKCHG C Z, SCANR4 CHKCHG C Z, SCANR4 CHKCHG C Z, SCANR4 C CHKCHG C Z, SCANR4 C CHKCHG C Z, SCANR4 C CHKCHG C Z, SCANR4 C CHKCHG C Z, SCANR4 C CHKCHG C Z, SCANR4 C CHKCHG C Z, SCANR4 C CHKCHG C Z, SCANR3 C Z, SCANR4 C CHKCHG C Z, SCANR3 C Z,	DE
3.44		404
-80 t)	SCANR4 :	
BIOS) Macro-8 ines for paint)	22 F942 32 F944 11 0000 13 6AC CD 1647 CD 1647 CD 1647 B8 05 CD 19AE 19AE 19 EF 19 EF 19 EF 28 F944 CD 1640 EB 22 F867 33 F944 CD 1640 EB 22 F867 33 F944 CD 1640 EB 22 F867 22 F867 21 1640 EB E1 E1 E1 E1	DT
MSX ROM BASIC BIOS MSXGRP - (Routines	1913 1916 1916 1916 1910 1920 1928 1928 1928 1928 1928 1928 1928 1928	ť
(MSX ROI -MSXGRP	5303 5304 5305 5305 5306 5305 5310 5311 5311 5311 5311 5311 5312 5322 5322	n n

55-3		
PAGE		
01-Jan-85		
3.44		
) Macro-80	for paint)	
MSX ROM BASIC BIOS) Macro-80	- (Routines	
(MSX R	-MSXGRP	

5334 194E C3 19A9 JP SCANL4

3E 56 195			;Is it border color?	No, start painting	All pixels tested?	4		Q	:Advance to right, and check if out of screen	Mot vet out of screen continue	Ont of screen let BDDOWT to and roturn	a at act and the product be al and teratil			SGet CT.DC.CMASK	Save VRAM address	Save mask pattern	:Initialize pwmcwm		:Increment PNPCNP	Advance to right, and check if out of some	Going ont of screen	:Reached border color?	:Not vet. continue	
PAGE		mode	; Is	N.	;Al			:Yes	PA:		Ē				-9-9-1- 	Sar :	: Sar	Tr		:In	- Ack	ę	Rea	No.	
01-Jan-85		; Scan to right in multi-color mode	MTSBRD	NC, MSCNR1	DE	A,D	ы	2	TRIGHT	NC, MSCANR	DE,0	•			FETCHC	(CSAVEA), HL	(CSAVEM), A	HL,0		HL	TRIGHT	U	MTSBRD	NC, MSCNR2	
3.44		o righ	CALL	Ŕ	DEC	9	OR	ЕT	CALL	ЯÇ	LD	RET			CALL	LD	Q	ГD		INC	CALL	RET	CALL	JR	RET
0	MSCANR:	; Scan to		.]	Π	П	U	щ	0		н	ц	MSCNR1:	••		Ч	ц	ц	MSCNR2:	П	U	æ	U	IJ	R
ASIC BIOS) Macro-8 (Routines for paint)			CD 19C7	30 0D	lB	7A	B3				11 0000					22 F942		21 0000		23	CD 16AC	D8	CD 19C7	30 F6	60
BASIC (Rou	1951		1951	1954	1956	1957	1958	1959	195A	195D	195F	1962	1963		1963	1966	1969	196C	196F	196F	1970	1973	1974	1977	1979
(MSX ROM BASIC BIOS -MSXGRP - (Routines	5335 5336 5337	5338 5339	5340	5341	5342	5343	5344	5345	5346	5347	5348	5349	5350	5351	5352										5362

PAGE 57 196	eft ;Initialize PNTCNT ;Initialize PNTDFL .Chack current screen mode	; Multi-color mode olution mode	;Clear 'pixel changed' flag ;Set border color to [B] for comparison	;Advance to left, and check if out of screen ;On left edge ;Read color of target pixel ;Reached border? ;Yes ;Update PNTCNT	;'C' must specify 'last pixel painted' ;Save PNTCNT
01-Jan-85	Scan pixels to left HL,0 C,L T	JR NZ,MSCANL ;Multi to left in high-resolution mode	A (WORK3),A A,(BRDATR) B,A	TLEFT C, SCANL3 READC B Z, SCANL2 CHKCHG HL SCANL1	RIGHTC HL
3.44	SCANL: ; ; SCANL - Scan ; LD LD	JR Scan to left	XOR LD LD LD LD SCANT.1:	SCANLL: CALL JR CALL JR CALL INC JR SCANL2:	; CALL SCANL3: PUSH
IOS) Macro-80 nes for paint)	21 0000 44D 670	i m	AF 32 F869 3A FCB2 47	CD 16D8 38 0F CD 1647 B8 CD 1647 B8 28 06 CD 19AE 23 18 EF 18 EF	CD 16C5 E5 E5 E5 E5 E5
MSX ROM BASIC BIOS -MSXGRP - (Routines	197A 197A 197D	1981	1983 1984 1987 1988	1988 1988 1990 1994 1998 1999 1990	199C 199F 199F
(MSX RO -MSXGRP	5363 5364 5366 5366 5367 5367	5371 5372 5373 5374	5375 5376 5377 5378 5378	53 / 9 53 80 53 81 53 84 53 84 53 84 53 86 53 84 53 86 53 87	5389 5390 5391 5392

197	s cr een
PAGE 57-1	<pre>co the right traw [HL] pixel from current 'C' testore PNTCNT testore PNTCNT on 0 if pixels changed attribute et specified paint attribute ame? es, no change of attribute oad non 0 to [Acc] enember this temporarily enember this temporarily on 0 to facc screen dvance to left, and check if out of oing out of screen acted border color? es, adjust CLOC, CMASK and return actement PNTCNT ontinue -color mode</pre>
01-Jan-85	HL, DE NSETCX HL NSETCX HL A, (WORK3) ; N C, A N, (WORK3) ; N HL (HL) HL (HL) ; S HL (HL) ; S HL (HL) ; S HL (HL) ; S HL (HL) ; S ML (MORK3), A ; N N (WORK3), A ; N N M C M TLEFT S M C M TLEFT S M C M SCAN SCAN S C M SCAN SCAN S M SCAN SCAN S M SCAN SCAN SCAN SCAN SCAN SCAN SCAN SCAN
3.44	ADD CALL CALL POP LLD RET PUSH PUSH LLD CCP POP RET INC LD CD LD CALL JP INC JP USH CALL CD JP CALL CD CD CD CD CD CD CALL CALL CALL C
-80 t)	SCANL4: SCANL4: ; ; MSCANL: ; ; Scan ; ; rest h ; ; rest h
) Macro-80 for paint)	1809 F869 F3F2 16D8 19C7 16C5 F3
ASIC BIOS (Routines	日本 19 18 33 18 33 18 33 18 33 18 33 19 18 34 19 19 19 19 19 19 19 19 19 19
l BASI((Rou	1984 1985 1985 1986 1986 1987 1987 1986 1986 1986 1986 1986 1986 1986 1986
(MSX ROM BASIC BIOS -MSXGRP - (Routines	5 5 5 5 5 5 5 5 5 5 5 5 5 5

PAGE 57-2	;Get the color of target pixel	;Load specified border color ;Reached border?	Assume so Yes, return with carry flag set	Is current pixel same as ATRBYT?	;Yes, no changes made. ;Return with carry reset	;Set this pixel to ATRBYT	Fet pixel-unanyed itay Fell caller that we plot a dot	s stuff
01-Jan-85	READC B,A	A, (BRDATR) B	27	A, (ATRBYT) B	2	SETC	C, L A	RET SUBTTL -CASET- Cassette drivers stuff
3.44	CALL	LD SUB	SCF RET	C D	RET	CALL	AND	RET L -CASET-
MSX ROM BASIC BIOS) Macro-80 MSXGRP - (Routines for paint)				19D1 3A F3F2 19D4 B8			19D9 0E 01 19DB A7	60
(MSX ROM -MSXGRP -	5425 5426	5427 5427 5428	5429 5429 5430	5431 5431	5433 5433 5434	5435	5436 5437	5438 5439

55 PAGE 58	uff	es that T cycle is 279.365 nS			TO read from cassette	Routine to check for [STOP] key pressed								:Test BC							01B :Stop motor						
01-Jan-85	₫/write st	iver assum	e for on cod						BC	AF	BC,0		R	A, B	ົບ	NZ, CTWOF1	AF	BC		AF	A,00001001B	(PPI.CM),A	AF	•			
3.44	Cassette read/write stuff	Following driver assumes	Variahles referenced	DPT CM	PSG.DR	BREAKX			PUSH	PUSH	LD		DEC	LD	OR	Я	POP	POP	OF:	HSUG	ГD	OUT	POP	EI	RET		
Macro-80 stuff	. · ·	 Fo	•~ •	- - •			, TADOPE.		-			CTWOF1:							TAPIOF:							TAPOON:	••
BIOS) e drivers									C5	FS	01 0000		0B	78	Bl	20 FB	Fl	cı		ΕS	3E 09	D3 AB	Fl	FB	60		
MSX ROM BASIC ASET- Cassette							1900		19DD	19DE	19DF	19E2	19E2	19E3	19E4	19E5	19E7	19E8	19E9	19E9	19EA	19EC	19EE	19EF	19F0	19F1	
(MSX R -CASET-	5440 5441 5442	5443	5444 5445	5446	5447	5448 5449	5450	5451	5452	5453	5454	5455	5456	5457	5458	5459	5460	5461	5462	5463	5464	5465	46	5467	5468	5469	5470

PAGE 58-1	Write out header, if [A]=0 then write short header otherwise write long header (5sec)	;set flag for length of header	;save flag	; Motor on							;get back header length flag	;get length of header	;short header					;set up counter	;Don't disturb during writing to cassette		;Write enough marks	; compensate overhead				;loop till counter exhausts	; check control-stop and return	
01-Jan-85	Write out header, if [A]=0 then wri otherwise write long header (5sec)	A	AF	A,8 (DDT CM)_A	HL,0		HL	А,Н	ц	NZ, MOTRWT	AF	A, (HEADER)	Z, SYNCWI	A, A	A, A		B,A	с,0			BITLOT	RETRET	BC	A, B	U	NZ, SYNLP1	BREAKX	
80 3.44 :f	; Write out he ; otherwise wr ;	OR	HSUG	LD	LD LD	MOTRWT:	DEC	ΓD	OR	ĥ	POP	ΓD	JR	ADC	ADL	SYNCW1:	LD	ΓD	DI	SYNLP1:	CALL	CALL	DEC	ΓD	OR	ĥ	円	TAPOUT:
BIOS) Macro-80 drivers stuff		B7		3E 08 D3 AB			2B	7C	B5	20 FB	Fl		28 02	87	87		47	0E 00	F3		CD 1A4D	CD 1A3F	0B	78	Bl	20 F5	C3 046F	
(MSX ROM BASIC BIOS -CASET- Cassette driv		19F1	19F2	19F3 19F5	19F7	19FA	19FA	19FB	19FC	19FD	19FF	1A00	1A03	1A05	1A06	1A07	1A07	1A08	LAOA	1A0B	1A0B	LAOE	INII	1A12	1A13	1A14	1A16	1A19
(MSX RC -CASET-	5471 5472 5473	5474	5475	5476	5478	5479	5480	5481	5482	5483	5484	5485	5486	5487	5488	5489	5490	5491	3492	5493	5494	5495	5496	5497	5498	5499	5500	5501

PAGE 58-2				; get time constants for space	4		;compensate loss time since last stop bit		;output start bit		; Initialize counter		;next bit to carry	;output mark if the bit is 1		;Loop until 8 bits sent	; Output stop bit	a a	;Check if break pressed and return
01-Jan-85			Ð	HL, (LOW)	AF	A,L	0 EH	L,A	BITOUT	AF	B,8			C,BIT1	NC, BITO	DATAWL	BITI	BITI	BREAKX
Macro-80 3.44 stuff	DATAW:	••	; Output a byte ;	LD	HSUG	LD	SUB	LD	CALL	POP	LD	DATAWL:	RRCA	CALL	CALL	DJNZ	CALL	CALL	JP
(MSX ROM BASIC BIOS) Macro-				2A F406	F5	7D	D6 0E	6F	CD 1A50	Fl	06 08		0F					CD 1A40	
MSX ROM BASIC BIOS) CASET- Cassette drivers	1A19			1A19	laic	lald	lale	1A20	1A21	1A24	1A25	1A27	1A27	1A28	1A2B	1A2E	1A 30	1A33	1A36
(MSX R	5502	5503	5505	5506	5507	5508	5509	5510	5511	5512	5513	5514	5515	5516	5517	5518	5519	5520	5521

	(17 T)			(18 T) (11 T) + 19.8usec (17 T)
PAGE 59	bit to cassette jumps are used to improve accuracy HL,(LOW) ;Output 0 (space)		; ;compensate overhead ; ; ;	;To compensate time ;Don't change this =16 x [L] + 16 x [H] + 71 =4.47uS x [L] + 4.47uS x [H] . ;
01-Jan-85	bit to cassette jumps are used to HL,(LOW)	BITOUT BITLOT	(SP),Щ(SP),Щ	s (H
80 3.44	BITO: ; ; Output a bit ; Absolute jum ; LD	CALL RETRET: BIT1: ; CALL	EX EX NOP NOP	CALL BITIOT RET BITIOT: ; output a single cycle ; Total number of states ; LD HL,(HIGH BITOUT:
BIOS) Macro-80 drivers stuff	2A F406		臣3 臣3 00 00 00	CD 1A4D C9 2A F408
(MSX ROM BASIC BIOS) -CASET- Cassette drive	1A39 1A39 1A39	1A35 1A35 1A35 1A40 1A40 1A40	1A43 1A44 1A45 1A45 1A46 1A47 1A47	1A49 1A4C 1A4D 1A4D 1A4D 1A50
(MSX RC -CASET-	5522 5523 5524 5525 5526 5528 5528 5529		5536 5537 5533 5539 5540 5541	5542 5543 5544 5545 5546 55549 5551 5551 5551

•

	(12 T)		(2 L)		(T 8)	(11 T)		(5T)	(11 L)	(8T)	(11 L)	(12 T)		(11 L)												_•			detect header	
PAGE 59-1			;Keep low level			;Output high level		;keep high level			;Output low level	Restore data							; Motor on			;Select PSG port A				good pulses are found.		e counter	;Number of pulse to d	
PA	••		; K	••	••	õ		ž	••		0	÷.		••					Ϋ́							of go		:	ž	
01-Jan-85	AF		Г	NZ, KEEPL	A,0BH	(PPI.CM),A		Н	NZ, KEEPH	A,0AH	(PPI.CM),A	AF					Detect header block		A,8	(PPI.CM),A		A, 0EH	(PSG.LW),A			a series		HL,0457H		
3.44	HSUG		DEC	JP	LD	OUT		DEC	JP	LD	OUT	POP		RET			st head		LD	OUT	DI	LD	OUT			First, wait until		LD		
о-80 ff		KEEPL:					KEEPH:						••		TAPION:	••	; Deteo	••						SYNO5:		; First,	••		:01NXS	
BIOS) Macro-80 drivers stuff	ፑ5		2D	C2 1A51	3E 0B	D3 AB		25	C2 1A59	3E 0A	D3 AB	Fl		60					3E 08	D3 AB	F3	3E 0E	D3 A0					21 0457		
MSX ROM BASIC BIOS CASET- Cassette driv	1A50	1 A 51	1A51	1A52	1A55	1A57	1A59	1A59	1A5A	1A5D	LA5F	1A61		1A62	1A63				1A63	1A65	1A67	1A68	1A6A	1A6C				1A6C	1A6F	
(MSX ROM -CASET- C	5553 5554	5555	5556	5557	5558	5559	5560	5561	5562	5563	5564	5565	5566	5567	5568	5569	5570	5571	5572	5573	5574	5575	5576	5577	5578	5579	5580	5581	5582 5583	

			er of pulses	ber of pulses	as a good pulse		necate it		ce?	pulse ever seen			h good pulses				of [BC] pair	
Е 59-2	Remember last value Count full cycle Aborted	Get count ODE = Max count		o short, reset number	compare with last pulse width and approve this his is similar to last one.	;current - last	result was negative, negate it		;within a wow allowance?	;no, reset number of			;Loop till seen enough		pulse.	Initialize sum	;Initialize high byte of [BC] pair	;Loop 256 times
PAGE	; Rei ; Coi	;Get : ODE	100 11 10 1	; Too	pulse width last one.	tro:	:re	-	; wi	ou:			;L00		width of	In:	; In	;Loc
01-Jan-85	D,C CNTFUL C	A, C ODEH	NC, SYN05 5	C, SYN05	with last pu similar to la	D	NC, SYNLL	A	4	NC, SYN05	HL A,H	Г	NZ, SYN10		calculate the mean width of pulse.	НГ,0	B,L	D,L
3.44	LD CALL RET	C D C D	C JY	JR	Now compare if this is s	SUB	JR CPL		SYN11: CP	К	DEC	OR	JR	SYN20:	; Next, calcul	LD	LD	LD
0-80 ff				••		-			SYI				•	SYI				
BIOS) Macro-80 drivers stuff	51 CD 1B34 D8	79 FE DE	ĿО	E		Ċ	30 02 2F	3C	FE 04	30 E6	2B 7C		20 E4			21 0000	45	55
(MSX ROM BASIC BIOS -CASET- Cassette driv	1A6F 1A70 1A73	1A74 1A75	1A77 1A79	1 A7B		1A7D	1A/E 1A80	1481	1A82 1A82	1A84	1A86 1A87	1A88	1A89	1A8B		1A8B	1A8E	1A8F
(MSX R -CASET-	5584 5585 5586	5587 5588	5589 5590	5591 5592	5593 5594 5595	5596	5598 /	5599	5600 5601	5602	5604	5605	5606 5607	5608	5610 5610	5612	5613	5614

PAGE 59–3			;compensate over head	1	Set various values for read routine. Those are.	· · · ·	LOWLIM - lower limit of the width of start bit. [H]*1.5	co count the transition.		;[H] has mean pulse width			;[D]=[mean]/2		;[A]=[mean]x2	;[A]=[mean]x1.5	; save	; compensate overhead at DATAR	1		[D]	a loop, RDBIT takes 60T for loop	er than single short pulse ([mean]/2)	1/60		
01-Jan-85	CNTFUL C	HL,BC D	NZ, SYN30 BC,06AEH	HL, BC	values for rea		ver limit of th	WINWID - width of window to		А,Н		7FH	D,A	HL, HL	А,Н	Q	D,A	9	(LOWLIM), A		Set width of window 'WINWID'	for	as 3 times wider	[WINWID]=[mean] x 1.5 x 40/60	=[D] X 7/3	
0 3.44	SYN30: CALL RET	ADD DEC	JP LD	ADD	; : Set various		; LOWLIM - low	; WINWID – wid	••	LD	RRA	AND	ГD	ADD	ГD	SUB	ГD	SUB	LD		; Set width of	; CNTFUL takes 40T for	; set WINWID as 3	= [((IMNTM] = [we	[n]= :	••
BIOS) Macro-80 drivers stuff	CD 1B34 D8	09 15	C2 1A90 01 06AE	60						7C	lF	E6 7F	57	29	7C	92	57	D6 06	32 FCA4							
(MSX ROM BASIC BIC -CASET- Cassette dr	1A90 1A90 1A93	1A94 1A95	1A96 1A99	1A9C						1A9D	1A9E	lA9F	IAAI	1AA2	1AA3	1AA4	1AA5	1AA6	1AA8							
(MSX RC -CASET-	5615 5616 5617	5618 5619	5620 5621	5622	5623 5624	5625	5626	5627	5628	5629	5630	5631	5632	5633	5634	5635	5636	5637	5638	5639	5640	5641	5642	5043 7043	0044 101	C # O C

PAGE 59-4	;get [mean width]xl.75	; x2	;clear quotient				;loop till get carry	;[A]=[mean]xl.75x2/3	; compensate overhead in RDBIT routine			
3.44 01-Jan-85	A , D	A, A	B,0		ю	В	NC, SULOP	А,В	٣	(MINNID),A	A	
3.44	ΓD	ADD	ГD		SUB	INC	JR	LD	SUB	ГD	OR	RET
(MSX ROM BASIC BIOS) Macro-80 -CASET- Cassette drivers stuff	7A	87	06 00	SULOP	D6 03	04	30 FB	78	D6 03	32 FCA5	B7	C9
(MSX ROM BASIC BIOS) -CASET- Cassette drivers	1 AAB	1 AAC	laad	1 AAF	1 AAF	1 A B 1	1AB2	1 AB4	1AB5	1 AB7	1 A B A	1 ABB
(MSX RO -CASET-	5646	5647	5648	5649	5650	5651	5652	5653	5654	5655	5656	5657

PAGE 60	<pre>;[D] has lower limit for start bit ;Aborted ;Get cassette ;High state? ;No ;Aborted ;Aborted ;Aborted ;falling egde? ;No ;Initialize edge mask ;Get vidth in [C] ;Save old width ;Get new width in [C] ;Save old width ;Get new width in [C] ;Add width of 2 pulses ;Pulse too long ;Pulse too long ;Puls</pre>
01-Jan-85	from cassette A, (LOWLIM) D,A BREAKX C A, (PSG.DR) NC, DATAR NC, DATAR NC, DATAR BREAKX C A, (PSG.DR) A, (PSG.DR) C, DATAR0 E, 0 E, 0 E, 0 E, 0 CNTHLF C CNTHLF C CNTHLF C CNTHLF C CNTHLF C CNTHLF C CNTHLF C CNTHLF C CNTHLF C CNTHLF C C CNTHLF C C C C C C C C C C C C C C C C C C C
80 3.44	TAPIN: FAPIN: Read a byte LD DATAR: LD LD LD RET IN RET RET IN RET IN RET IN RET RET RET RET RET RET RET RET
BIOS) Macro-80 drivers stuff	3A FCA4 57 57 CD 046F DB A2 07 07 03 F7 03 F7 07 07 1E 00 CD 1B1F 08 1 1 81 78 81 81 81 81 81 81 81 81 81 81 81 81 81
(MSX ROM BASIC BIOS -CASET- Cassette dri	1ABC 1ABF 1AC0 1ABF 1AC0 1AC0 1AC3 1AC0 1AC3 1AC3 1AC3 1AC0 1AC2 1AC2 1AC0 1AC2 1AD2 1AD2 1AD2 1AD2 1AD2 1AD2 1AD2 1AD
(MSX RC -CASET-	5658 5659 5660 56661 56661 56664 56666 56666 5667 5672 5673 5673 5673 5673 5673 5673 5673 5673

85 PAGE 60-1	start bit has been found.	if NORMAL polarity,	REVERSE polarity.	:Initialize counter			;Legal transitions?		;Too many transitions		;Set carry if 2 or 3 transitions			assembled a bit. A check must be done to make sure	at the start of next bit field.		;Reget number of transitions	b	10 ; Wait for next transition if 0 or 2			M. ;Loop till done	return with carry set if breaked:	1				of transitions within a period specified by 'WINWID'	•	length of window = $17uSec x [WINWID] + 12.3 uSec$
01-Jan-85			if rever	L.8		RDBIT	3+1		υ	2		D		ssembled a	t the star		A,C		NC, CNTHL0	CNTHLF	Г	NZ, DATARL	BREAKX	Α, D				of transi		ndow = 17u
3.44	; Now, a valid		; =255	ΓD	DATARL:	CALL	CP	CCF	RET	CP	CCF	RR		; We've just a		••	LD	RRCA	CALL	CALL	DEC	JP	CALL	LD	RET	RDBIT:		; Count number		; length of wi
BIOS) Macro-80 drivers stuff				2E 08		CD 1B03	FE 04	3F	D8	FE 02	3F	CB 1A					79	0F	D4 1B23	CD 1B1F	2D	C2 1AE6	CD 046F	7A	C9					
(MSX ROM BASIC BIOS -CASET- Cassette driv				1AE4	1AE6	1AE6	1AE9	1AEB	1AEC	1 AED	1AEF	1 AFO					1AF2	1 AF 3	1AF4	1 AF7	lafa	1 AFB	1AFE	1801	1B02	1 B0 3				
(MSX R(-CASET-	5689	5690	5691 5603	5693	5694	5695	5696	5697	5698	5699	5700	5701	5702	5703	5704	5705	5706	5707	5708	5709	5710	5711	5712	5713	5714	5715	5716	5717	71	5719

PAGE 60-2	v ed.	for next edge		:Get width of window		;Clear # of transitions seen		;Get a bit	; Any changes?	; No	;Transition seen	Prepare for next transition		:Increment # of transitions		Get transition count				;Compensate time	-					Get transition count;	
01-Jan-85	[D],[H] and [L] are preserved	is updated to prepare	4 4	A, (WINWID)	B, A	c,0		A, (PSG.DR)	ы	P, NOTRAN	A,E		E,A	U	RDBITL	A,C								RDBITL		A,C	
3.44	[D],[H] and	[E] is updat		LD	LD	LD	RDBITL:	IN	XOR	JP	LD	CPL	LD	INC	DJNZ	LD	RET	NOTRAN:		NOP	NOP	NOP	NOP	ZNLC		LD	RET
Macro-80 stuff	•• ••		••				RD											NO	••								
ល				3A FCA5	47	0E 00		DB A2	AB	F2 1B17	7B	2F	5F	00	10 F4	79	60			00	00	00	00	10 EC		79	C9
MSX ROM BASIC BIOS) CASET- Cassette drive				1B03	1B06	1B07	1B09	1B09	1B0B	1B0C	1 B 0 F	1B10	1 B 1 1	1 B1 2	1 B13	1B15	1 B16	1B17		1B17	1 B18	1B19	1 BLA	1818		lBlD	1 B1 E
(MSX RC -CASET-	5720 5721	5722	5723	5724	5725	5726	5727	5728	5729	5730	5731	5732	5733	5734	5735	5736	5737	5738	5739	5740	5741	5742	5743	5744	5745	5746	5747

		(87 T) (6 T) (8 T)	սալովո	
PAGE 61	35.48uS	;Break? ;Yes, aborted ;Initialize counter	<pre>;# of state for this loop ;40T=11.18usec (;Pulse too long (;Read cassette (;Des ired transition? (;No (;No ())</pre>	; ; ; Load 255
01-Jan-85	.4nS x 11.18 +	BREAKX C C,0	C Z,TIMOUT A,(PSG.DR) E P,CNTHL1 A,E	E,A C cycle BREAKX
3.44	THLF: Count half cycle lT =279 period=[C]	CALL RET LD	INC JP JP LD	
-80 f	CNTHLF: ; ; ; ; ;	CNTHLO:		CPL LD FIMOUT: RET FET DEC RET CNTFUL: ; Count full ; Count full
BIOS) Macro-80 drivers stuff		CD 046F D8 0E 00	0C 28 0A DB A2 AB F2 1B25 7B	2F 5F C9 0D C9 C9 C9 C0 046F
0	lblF	1B1F 1B22 1B23 1B23	1825 1826 1828 1828 1828 1828 1828	1B2F 1B30 1B31 1B32 1B32 1B33 1B34 1B34 1B34
(MSX ROM BASIC -CASET- Cassett	5748 5749 5750 5751 5752	5755 5756 5757 5758 5758	5760 5761 5763 5763 5763 5765 5765	5767 5768 5768 5770 5771 5772 5773 5773 5775 5775 5777 5777

PAGE 61-1	; Abor ted	;Get cassette	;Low state?					
01-Jan-85	U	A, (PSG.DR)			E,0			
3.44	RET	IN	RLCA	JR	LD	CALL	JP	SUBTTL - BIO -
(MSX ROM BASIC BIOS) Macro-80 -CASET- Cassette drivers stuff	D8	DB A2	07	38 F7	1E 00	CD 1B23	C3 1B25	
OM BASIC Cassette	1B37	·1B38	1B3A	1B3B	1 B3D	1B3F	1B42	
(MSX R -CASET-	5779	5780	5781	5782	5783	5784	5785	5786

			er		outine															
62	(RST 18H) Prints char in [A], to either terminal or disk or printer depending on the flags: PRTFLG if non-zero print to printer PTRFIL if non-zero print to disk file pointed to by PTRFIL	;Save character	;Doing I/O to file? ;Nope, check for output to printer	;Restore char.	Jump with pointer to FILE OUT routine				;Output to printer?		;Nope, output to console	;Print in "RAW" mode?		;Yes, send char to printer	;restore char					
PAGE	A], to eithen ding on the if non-zero] if non-zero] to by PTRFIL	; Sav	; Nopi	; Res	um C :				;Out]		; Nop	; Pr ii		;Yes	; res					
01-Jan-85) c in { depen IFLG RFLL	AF H . OUTD	ISFLIO Z,LPTCOD	AF	IX, FILOUI	CALBAS			A, (PRTFLG)	А	Z, TTYCHR	A, (RAWPRT)	A	NZ, LPTCH1	AF			AF	•	
3.44	TDO: OUTDO (RST 18H Prints chan or printer PR	PUSH CALL	CALL JR	POP	LD	£			LD	OR	JR	LD	AND	ЯĻ	POP			HSUG		
0	OUTDO: OUTDO: OUTDO						••	LPTCOD:								••	OUTDLF:		; MTDVC7.	TOUGIN
Macro-80 e		FFF.4	45F 8		1 6C48	OlfF			416		Ь	418		6						
BASIC BIOS) OUTDO routine		F5 CD F		Fl	2	C3 0			Гц	B7	S	Ēų	A7	20 4	Fl			ЪО		
-	1B45	1B45 1B46	1B49 1B4C	1B4E	1B4F	1B53		1B56	1B56	1.B59	1B5A	1B5C	1 B5F	1B60	1B62		L B63	1B63	ואפן	T DOF
(MSX ROM - BIO -	5787 5788 5789 5791 5792 5793 5793 5795	5797 5798	5799 5800	5801	5802	5803	5804	5805	5806	5807	5808	5809	5810	5811	5812	5813	5814	5815	5816 5017	1700

PAGE 62-1	; TAB? ; No		;Print a space		, det cultent LFOS :At TAB ston?	.No. hack for more snare					:Check if CR If so load a zoro	TT is ribar LDTDAG and cond rD	CODE IS 0. OFH. Suit send	without modify Londos	See if control character	CODE IL CONCLOI CHARACLEI					;Update LPOS			:Output to MSX standard mrinter	Tonin the number was an and the	NO mapping for KATAKANA to UIBACANA	restore that to print to mind	See if graphic header
01-Jan-85	9 NZ , NOTABL		A, ' '	OUTDLP A [(T.PTDAC)]	7	NZ, MORSPL	AF				HQ0	Z, ZERLPI	C, LPTCHO		" "-13	C,LPTCH0	A. (LPTPOS)	A			(LPTPOS), A			A, (NTMSXP)	A	Z, LPTCH1	AF	CNVCHR
3.44	CP JR	, MORSPL :	LD	CALL LD	AND	JR	POP	RET		NOTABL:	SUB	JR	JR		CP	JR	LD	INC		ZERLP1:	LD		LPTCH0:	LD	AND	JR	POP	CALL
BIOS) Macro-80 routine	FE 09 20 0E		\sim	CD 1B63 3A F415	0	щ	Fl	60	••	N		28 OA	38 OB		Ч	38 07	Ēų	3C		ZE	32 F415		LF	3A F417	A7	28 1E	Fl	CD 089D
BASIC OUTDO	1B64 1B66	1 B68	1B68	1B6A 1B6D	1B70	1B72	1B74	1875		1B76	1B76	1B78	1B7A		1B7C	1B7E	1B80	1B83		1 B84	1B84		1B87	1B87	1 B8A	1B8B	1 B8D	1 B8 E
(MSX ROM - BIO -	5818 5819 5820	5821	5822	5824 5824	5825	5826	5827	5828	5829	5830	5831	5832	5833	5834	5835	5836	5837	5838	5839	5840	5841	5842	5843	5844	5845	5846	5847	5848

PAGE 62-2	;Yep ;Graphic symbol, map to space			;Graphic symbol?	;Yes, map this to space too!	;A HIRAGANA(part 1)?		; Map to KATAKANA			;HIRAGANA(part 2)?	; No	; Map to KATAKANA	;'JRC' instruction (Skip next byte)		;Restore char			;Send character out	; Sent success ful	;Direct I/O error								;Get the character
01-Jan-85	NC NZ, MAPSPC	А	P,LPTCHR	86H	C, MAPSPC	0A0H	NC , NTH I RA	A, ' '	L PTCHR		0E0H	C,LPTCHR	-	38H		AF			LPTOUT	NC	IX, DIOERR	CALBAS		A, ' '	LPTCHR			nsole	AF
3.44	RET JR	AND	JP	CP	JR	CP	ЯĻ	ADD	JR	NTHIRA:	CP	ЯŲ	SUB	DB	LPTCH1:	POP		LPTCHR:	CALL	RET	ГD	ЧĻ	MAPSPC:	ГD	JR	TTYCHR:		Output to console	POP
BIOS) Macro-80 routine	D0 20 23	A7	F2 1BAC		38 1B		30 04			EN		38 04		38	LI	Fl	••	LI	CD 085D	D0	DD 21 73B2	C3 01FF	W	3E 20	54	5	••		FI
BASIC OUTDO	1B91 1B92	1B94	1B95	1B98	1 B9A	1B9C	1 B9E	1BA0	1 BA2	1 BA4	1 BA4	1 BA6	1BA8	1 BAA	1 BAB	1 BAB		1 BAC	1 BAC	lbaf	1BB 0	1 BB4	1BB7	1 BB7	1 BB9	1 BBB			1 BBB
(MSX ROM - BIO -	5849 5850	5851	5852	5853	5854	5855	5856	5857	5858	5859	5860	5861	5862	5863	5864	5865	5866	5867	5868	5869	5870	5871	5872	5873	5874	5875	œ	5877 5878) œ

62-3 PAGE 01-Jan-85 3.44 (MSX ROM BASIC BIOS) Macro-80 - BIO - OUTDO routine

5880 1BBC C3 08BC JP CHPUT 5881 SUBTTL -MSXCHR- MSX character set

01-Jan-85	
3.44	
Macro-80	set
ROM BASIC BIOS)	MSX character
(MSX ROM	-MSXCHR-

	00Н,00Н,00Н,00Н,00Н,00Н,00Н,00Н		00H,7EH,42H,7EH,42H,7EH,42H,7EH,42H		82H,00H,10H,92H,54H,10H,28H		44H,82H,00H,12H,14H,0F8H,14H		34H,52H,92H,00H,10H,10H,0FEH		10H,38H,54H,92H,00H,10H,28H		7CH,92H,38H,54H,0FEH,00H,10H		10H,10H,7CH,10H,10H,0FEH,00H		7EH,42H,42H,7EH,42H,42H,7EH		00H,40H,7EH,48H,3CH,28H,7EH		08H,00H,0FEH,92H,92H,0FEH,82H		82H,86H,00H,04H,0EEH,0A4H,0EFH		0A2H,0EAH,06H,00H,28H,44H,82H		3CH,14H,24H,4CH,00H,28H,0C8H		5СН,0ЕАН,6СН,0С8Н,50Н,00Н,7СН
·CCTARL.	DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB
	00		7E		92		12		00		92		54		10		7E		48		92		04		00		4C		C8
	00	00	42 7	42	10 9	28	00	14	92 (FE	54.0	28	38	10	7C]	00	42	7E	7E 4	7E	E	82	00	EF	06 (82	24 4	C8	6C (
	00	00	7E		00	10	82	F8	52	10	38		92	00	10	FЕ	42	42	40	28	00	БE	86	A4	EA	44	14	28	EA
	00	00	00	42	82	54	44	14	34	10	10	00	7C	ЕË	10	10	7E	42	00	ပ္ထ	08	92	82	ΞE	A2	28	ЗС	00	5C
1 RRF	1 BBF	1 BC 3	1BC6	1 BCA	1 BCD	lBDl	1 BD4	1 BD8	1 BDB	1 BDF	1 BE2	1 BE6	1 BE9	1 BED	1 BFO	1 BF4	1 BF7	$1 \mathrm{BFB}$	1 BFE	1C02	1C05	1C09	1000	1C10	1C13	1C17	lClA	ICLE	1C21
5882 5003	5884	5885	5886	5887	5888	5889	5890	5891	5892	5893	5894	5895	5896	5897	5898	5899	5900	5901	5902	5903	5904	5905	5906	5907	5908	5909	5910	5911	5912

PAGE

BASIC BIOS) Macro-80 3.44 01-Jan-85 PAGE 63-1 MSX character set		.8 20 7C 44 7C DB 20H,7CH,44H,7CH,44H.7CH.00H	44 7C 00	F 0C 70 10 FE DB 0CH, 70H, 10H, 0FEH, 10H, 10H, 10H	10 10 10	6 00 7E 10 1E DB 00H, 7EH, 10H, 1EH, 12H, 22H, 44H	12 22 44	D 08 00 00 7C DB 08H,00H,00H,7CH,28H,28H,28H	28 28 28	4 4Е 00 00 10 DB 4EH,00H,00H,10H,10H,10H,0FFH	10 10 FF		00 00 00	2 FF 10 10 10 DB 0FFH,10H,10H,10H,10H,10H,10H	10 10 10		10 10 10	0 10 10 1F 10 DB 10H,10H,10H,10H,10H,10H	10 10 10	10 10	10 10 10	E 10 10 10 10 DB 10H, 10H, 10H, 10H, 10H, 10H	10 10 10	5 10 10 00 00 DB 10H,10H,00H,00H,00H.0FFH.00H	00 FF 00	С 00 00 00 00 DB 00H,00H,00H,00H,00H,00H,00H,00H,00H,00H	00 00 1F	3 10 10 10 10 DB 10H,10H,10H,10H,00H,00H,00H	00 00 00	01 JJ J0
) ter	00	7C 44	7C	70 10	10	7E 10	22	00 00	28	00 00	10	00 00	00	10 10	10	F0 10	10	10 lF	10	10 10	10	10 10	10	10 00	FF	00 00	00	10 10	00	F0 10 10 10
M BASI MSX	1C25	1C28	1C2C	1C2F	1C33	1C36	1C3A	1C3D	1C41	1C44	1C48	1C4B	1C4F	1C52	1C56	1C59	IC5D	1C60	1C64	1C67	1C6B	1C6E	1C72	1C75	1C79	1C7C	1C80	1C83	1C87	1C8A
(MSX ROM -MSXCHR-	5913	5914	5915	5916	5917	5918	5919	5920	5921	5922	5923	5924	5925	5926	5927	5928	5929	5930	5931	5932	5933	5934	5935	5936	5937	5938	5939	5940	5941	5942

63-2	.10Н		H00 , H		.42Н		,44H		HJOFEH		,54H		ноо,		,20Н		,50H		, ООН		ЭН,50Н		H0F0H		ОН,4 ОН		H8A0,H		, 4 ОН		- 2 OH
PAGE	ноо, ноо, н		0H,00H,00E		Н,18Н,24Н,		Н,10Н,28Н,		H,0FEH,92F		Н,10Н,54Н,		н, оон, оон,		н,00н,20н,		н, 20н, 00н,		ноо, ноо, н		OH,0F8H,5(OH,70H,28H		C8H,10H,2(H,0A0H,40F		Н,10Н,20Н		Н,00Н,10Н
01-Jan-85	10H,1FH,00H,00H,00H,00H,00H,10H		10H,10H,0F0H,00H,00H,00H,00H		81H,42H,24H,18H,18H,24H,42H		81H,10H,7CH,10H,10H,28H,44H		82H,00H,10H,10H,0FEH,92H,0FEH		10H,10H,00H,10H,10H,54H,54H		92H,10H,30H,00H,00H,00H,00H		00H,00H,00H,00H,00H,20H		20H,20H,00H,00H,20H,00H,50H		50н, 50н, 00н, 00н, 00н, 00н, 00н		50Н, 50Н, 0 F8Н, 50Н, 0 F8Н, 50Н, 50Н		00H,20H,78H,0A0H,70H,28H,0F0H		20H,00H,0C0H,0C8H,10H,20H,40H		98H,18H,00H,40H,0A0H,40H,0A8H		90H,98H,60H,00H,10H,20H,40H		00Н,00Н,00Н,00Н,00Н,00Н,20Н
3.44 (DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB
Macro-80 set	00		00		18		10		10		10		00		00		00		00		50		A0		C8		40		00		00
Macr set	0 00	0	F0 0	00	24 1	42	C 1	44	10 1	FE	00 1	54	30 0	00	0 00	20	0 00	50	0 00	00	F8 5	50	78 A	FO	0000	40	00 4	A8	60 0	40	00
	Б.	0	E4 O	0	42 2	4	10 7	284	00 1	2 H	10 0	54 5	10 3	00	00	20 2	0	00 5	50 0	00	50 F	50 5	20 7	28 F	00	0.4	18 0	40 4	98 6	20 4	00
C BIOS) character	10 1	0 00	10 1	00 0	814	18 2	81]	10 2	82 (FE 0	10 1		92]	00	00		202			00						10 2		A0 4	606	10 2	00
BASIC BIOS MSX charact	1001	1C95	1C98	1C9C	1C9F	1CA3	1CA6	lcaa	lCAD	1CB1	1CB4	1CB8	ICBB	lcbF	1cc2	1006	1CC9	lccD	1CD0	1CD4	1CD7	1CDB	ICDE	1CE2	1CE5	1CE9	1CEC	1CF0	1CF3	1CF7	lCFA
(MSX ROM -MSXCHR-	5944	5945	5946	5947	5948	5949	5950	5951	5952	5953	5954	5955	5956	5957	5958	5959	5960	5961	5962	5963	5964	5965	5966	5967	5968	5969	5970	5971	5972	5973	5974

63-3			по +, по	лоо но	1100 / 110	лос нако		ностнос	1107/1107	ноо но	1100/110	ли 7 ви		н оон		ноон	1100/110	100 DU	HO / 111	70H 00H		ОН ОБЯН		н вон		нао ноз		ное н	110C / 11	800 0E00	
PAGE		ס שמן שמכ		10H_20H A		20H 70H	1110/11107/	20н.0ғ8н.		00 - 000 - 00	0/1100/1100	00H - 00H - 00	01 H001 H00	00 ноо ноо		604-004-00		4 OH - ROH OC		ОСЯН ВВН		.20н.20н.3		א ו א א א א א א א א א א א א א א א א א א		. АВН ОВН 3		0H.30H.5C		ООН-ОБЯН	1 110 10 1 110 0
01-Jan-85				20H.10H.10H.10H.20H A0H A0H		20H.0A8H.70H.20H.70H 0A8H.20H		00H,00H,20H,20H,0F8H,20H,20H		HOO, HOO, HOO, HOO, HOO, HOO, HOO		20H.20H.40H.00H.00H 00H 78H		HOO, HOO, HOO, HOO, HOO, HOO, HOO		ноо ноо ноо ноо ноо ноо ноо		08H.10H.20H.40H.80H.00H.70U		88H.98H.0A8H.0C8H.88H.70H		20H,60H,0A0H,20H.20H,20H,20H		00H.70H.88H.08H.10H.60H		0F8H,00H,70H,88H,08H,30H,08H		88H,70H,00H,10H,30H,50H,90H		0F8H.10H.10H.00H.00H.0F8H	
3.44		DB	1	DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB	
Macro-80 set	20	40 20	40	10 10	00	70 20	20	20 20	20	00 00	00	40 00	78	00 00	00	60 60	00	20 40	70	A8 C8	00	A0 20	F8	88 08	80	70 88	08	00 10	06	10 00	EO
	10	40	00	10	40	A8	A8	00	20	00	00	20	00	00	00	00	00	10	00	98	70 (60 2	20 I	70	60	00	30	70 0	50	10	80 E
C BIOS) character	00	40	10	20	20	20	70	00	F8	00	00	20	00	00	00	00	00	08	80	88	88	20	20	00	10	F8	08	88	30	F8	F8
BASI MSX	lcfe	1001	1D05	1D08	1D0C	1D0F	1D13	1D16	lDlA	lDlD	1D21	1D24	1D28	1D2B	1D2F	1D32	1D36	1D39	1D3D	1D40	1D44	1D47	1D4B	lD4E	1D52	1D55	1D59	1D5C	1D60	1D63	1D67
(MSX ROM -MSXCHR-	5975	5976	5977	5978	5979	5980	5981	5982	5983	5984	5985	5986	5987	5988	5989	5990	5991	5992	5993	5994	5995	5996	5997	5998	5999	6000	6001	6002	6003	6004	6005

63-4	Н,40Н	Н840'Н		ноо, і	- 10H	1,/ИН	ניון ני	1171		nuu, nuu,		,20H,00H		Н,60Н	4	H00,HC		00H,UCUH		H00, H0		Н,20Н		вн, оавн		Н,0F8Н		8Н,70Н		8H , 80H	
PAGE)ЕОН,00Н,3(, 88Н, 70Н, 00Н		, 20Н, 20Н, 20Н	100	,/UH,88H,88H	100 102 100	100, 101, 100, 100	100	100, HU2, HUU,		107, НО0, НОС		,40H,18H,30H		30H, 18H, 00H, 00H, 00H		Н,00Н,00Н,00Н,00Н,0С0Н		, 30н,60н,0С0н,00н		,10н,20н,00н		,08H,68H,0A8H		,50Н,88Н,88Н		,88н,00н,0F0н,48н,48н		,00Н,30Н,4	
01-Jan-85	10Н,08Н,10Н,0Е0Н,00Н,30Н,40Н	80H,0F0H,88H		88н,10н,20н,		/ OH , 88H , 88H , 9	1100	оон,/ UH, 88н,8		60H,UUH,UUH,U		20н,00н,00н,00н,00н		00н,20н,20н,		осон, 60н, 30н		0F8H,00H,0F8H		60н,30н,18н,		70H,88H,08H,		00н,70н,88н,		70н,00н,20н,		88н,88н,00н,		48H,48H,0F0H,00H,30H,48H,80H	
3.44 0	DB 1	DB 8		DB 8		DB		DB		DB		DB		DB (DB (DB		DB		DB		DB		DB		DB		DB	
Ma cro-80 set	0	œ		0		0		88		0		00		40		18		00		30		10		08		50		F0		00	
Macro	10 E0	40 88 88	F8	20 20		88 70				00 00		0 00	00		60			F8 0	CO		00		20		A8		F8		70	FO O	
		30 40 F0 88	00 F	10 2						000	0 00	0 00	20 0	202	30 6			00 F							A8 A	00 2		88	48 7	48 F	
s) cte		日 100 11 10 11 10 11 10 11 10 11 10 11 10 11 10 11 10 11 10 11 10 11 10 11 10 11 10 11 10 11 10 11 10 10					88 8														-					70 0	88	88	484	484	
C BIOS) character	-	0 00	7	8	7	-	œ	0	7	9	2	7	0	0	Ч	0	0	ᄪ	0	Q	9		2	0	Ψ		ω	ω	4	ч	
BASIC MSX cha	1D6A	1D6E 1D71	1D75	1D78	1D7C	1D7F	1D83	1D86	1D8A	1D8D	1091	1D94	1D98	1D9B	1D9F	1DA2	1DA6	1DA9	lDAD	1DB0	1DB4	1DB7	1 DBB	IDBE	1DC2	1DC5	1DC9	1DCC	1DD0	1DD3	
(MSX ROM -MSXCHR-	6006	6007 6008	6009	6010	6011	6012	6013	6014	6015	6016	6017	6018	6019	6020	6021	6022	6023	6024	6025	6026	6027	6028	6029	6030	6031	6032	6033	6034	6035	6036	

63-5		Н. 50Н		ЭН,ОЕ8Н		8Н,00Н	11001110	H08,H0		Н,88н		Н.88н		1,20H		,10H		H06.		00H.80H	1100/1100	н.00н		88н.88н		98н.98н		.88н		н.070н	
PAGE		H,00H,0E0		H,0E0H.00		0H.80H.0F		FOH,80H,8	•	н, 0 В8Н, 88		H,88H,0F8		Н,20Н,20Н	•	Н,38Н,10Н	•	н, оон, 88н		. 90н. 88н.		H,80H,0F8		. А8н. 88н.		C8H,0A8H,		, 88H , 88H		H, 88H, 88	
01-Jan-85		80H,80H,48H,30H,00H,0E0H,50H		48H,48H,48H,50H,0E0H,00H,00H	•	80H,80H,0F0H,80H,80H,0F8H		0F8H,80H,80H,0F0H,80H,80H		00H,70H,88H,80H,0B8H,88H		70H,00H,88H,88H,88H,88H,0F8H,88H		88H,88H,00H,70H,20H,20H		20H,20H,70H,00H,38H,10H,10H		10Н,90Н,90Н,60Н,60Н,00Н,88Н.90Н		0A0H,0C0H,0A0H,90H,80H,80H		80H,80H,80H,80H,80H,80H,0F8H,00H	•	88H,0D8H,0A8H,0A8H,0A8H,88H.88H.88H		00H,88H,0C8H,0C8H,0A8H,98H		88H,00H,70H,88H		88H,70H,00H,0F0H,88H,88H,0F0H	
3.44		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB	
Macro-80 set	80	48 30	50	48 50	F8	F0 80	00	80 FO	80	88 80	88	88 88	88	00 70	20	70 00	10	90 60	06	A0 90	80	80 80	00	A8 A8	88	C8 C8	98	70 88	88	00 FO	FO
	48	80 4	E E	48 4	100	80 I	F8 (80	80	70 8	88	00	F8 ε	88	20 2	20 7	10 1	90 9	88	CO A	00 8	80 8	F8 0	D8 A	88 8	88 C	98 9	00 7	88 8	70 0	88 F
C BIOS) character	30	80		48	EO	80	80	F8	80	00	B8	70						10	00	A0 (88	80	80 I	88 I	88	00	A8 9	88 C	88	88 7	88
BASIC MSX ch	1007	ldda	1 DDE	ldel	1DE5	1 D E 8	1 DEC	ldef	1DF3	1DF6	ldfa	ldfD	1E01	1E04	1E08	1E0B	l = 0F	1E12	1E16	1E19	leid	1E20	1E24	1E27	le2b	1E2E	1E32	1E35	1E39	1E3C	lE40
(MSX ROM -MSXCHR-	6037	6038	6039	6040	6041	6042	6043	6044	6045	6046	6047	6048	6049	6050	6051					6056		6058	6059	6060						6066	6067

63-6	, 88н		0Н,88Н		0Н,70Н		,00H		Н,20Н		, 88н		,50H		,0A8H		,88Н,50Н		, 88н		I,0F8H		ноо,н		1,70H		I,70H		(,10Н		1,00H
PAGE	0Н,70Н,88Н		68H,00H,0F		(,90Н,88Н,0		8H,88H,70H		20H,20H,20		,88H,88H,88H,88H		, 88Н, 88Н, 88Н		, 88H , 88H , 88H ,		,88H,00H,88H,8		, 88Н,00Н,88Н		, 20Н, 20Н, 00Н		,40H,80H,0F8H		,40H,40H,40H		,20H,70H,20H		.0Н,10Н,10Н		20H,50H,88H
01-Jan-85	80H,80H,80H,00H,70H,88H,88H		88H,0A8H,90H,68H,00H,0F0H,88H		88H,0F0H,0A0H,90H,88H,00H,70H		88H,80H,70H,08H,88H,70H,00H		0F8H, 20H, 20H, 20H, 20H, 20H, 20H		00Н,88Н,88Н,8		70Н,00Н,88Н,8		50Н,20Н,00Н,8		0A8H,0D8H,88F		20Н,50Н,88Н,8		88H,70H,20H,2		08н,10н,20н,4		70H,40H,40H,40H,4		00H,88H,50H,2		20H,00H,70H,10H,10H,10H,10H,10H		10н,70н,00н,20н,50н,88н,00н
3.44 0																															
• m	DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB
Ma <i>c</i> ro-80 set	00		68		06		08		20		88		88		88		00		88		20		40		40		20		10		20
Macr set	80 (88	906	88	A0			00		20	88	88	88	50	00	A8	88	50		88	20	F8	20	00	40	70	50	70	70	10	00
	80 8	88		FO	F0 7		80	70 (20 2	20 2	88	88	00	88	20 (D8				70		10	F8	40	40	88	20	00		70
C BIOS) character	808	70 8				88			F8				70 (20	40			20		10
BASIC MSX ch	1E43	1E47	1E4A	1E4E	1E51	1E55	1E58	1E5C	1E5F	1E63	1E66	1E6A	1E6D	1E71	1E74	1E78	1E7B	1E7F	1E82	1E86	1E89	1E8D	1E90	1E94	1E97	1 E9 B	1E9E	1EA2	1EA5	1EA9	leac
(MSX ROM -MSXCHR-	6068	6069	6070	6071	6072	6073	6074	6075	6076	6077	6078	6079	6080	6081	6082	6083	6084	6085	6086	6087	6088	6089	6090	6091	6092	6093	6094	6095	6096	6097	6098

01-Jan-85 PAGE 63-7		00H.00H.00H 00H 00H 00H 00H		00H.00H.00H.00H.0F8H 00H A0H 20H		10H,00H,00H,00H,00H,00H,00H		004,704,084,784,884,004,004		80H,80H,0B0H,0C8H,88H,0C8H,0B0H		00H,00H,00H.70H,88H 80H 88H		70Н,00Н,08Н,08Н,68Н,68Н,98Н,88Н		98Н.68Н.00Н.00Н.00Н.70Н 88н		0Р8н.80н.70н.00н 10н эви эон		0F8H.20H.20H.20H.20H		68Н,98Н,98Н,68Н,08Н,70Н,80Н		80H.0F0H.88H.88H.88H.88H		20H,00H,60H,20H,20H,20H,20H		00H.10H.00H.30H.10H 10H 10H		90H,60H,40H,40H,40H,48H 50H 60H	
3.44		DB	1	DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB	
OM BASIC BIOS) Macro-80 - MSX character set	1EB0 50 88 00	1EB3 00 00 00 00	1EB7 00 00 00	1EBA 00 00 00 F8	1EBE 00 40 20	1EC1 10 00 00 00	1EC5 00 00 00	1EC8 00 70 08 78	1ECC 88 78 00	1ECF 80 80 B0 C8	1ED3 88 C8 B0	1ED6 00 00 00 70	1EDA 88 80 88	1EDD 70 00 08 08	IEE1 68 98 88	1EE4 98 68 00 00	1EE8 00 70 88	1EEB F8 80 70 00	1EEF 10 28 20	1EF2 F8 20 20 20	1EF6 00 00 00	1EF9 68 98 98 68	1EFD 08 70 80	1F00 80 F0 88 88	1F04 88 88 00	20		1F0E 00 10 00 30	1F12 10 10 10	lFl5 90 60 40 40	1F19 48 50 60
(MSX ROM -MSXCHR-	6609	6100	6101	6102	6103	6104	6105	6106	6107	6108	6109	6110	6111	6112	6113	6114	6115	6116	6117	6118	6119	6120	6121	6122	6123	6124	6125	6126	6127	6128	6129

01-Jan-85 PAGE 63-8	50Н,48Н,00Н,60Н,20Н,20Н,20Н,20Н		20н, 20н, 70н, 00н, 00н, 00н, 00н, 0D0н		0A8H, 0A8H, 0A8H, 0A8H, 0A8H, 00H, 00H, 00H		0B0H, 0C8H, 88H, 88H, 88H, 00H, 00H		00Н,70Н,88Н,88Н,88Н,70Н,00Н		00H,00H,0B0H,0C8H,0C8H,0B0H,80H		80Н,00Н,00Н,68Н,98Н,98Н,68Н		08H,08H,00H,00H,00H,0B0H,0C8H,80H		80H,80H,00H,00H,00H,78H,80H		0F0H,08H,0F0H,00H,40H,40H,0F0H		40H,40H,48H,30H,00H,00H,00H		90Н,90Н,90Н,90Н,68Н,00Н,00Н,00Н		00Н, 88Н, 88Н, 88Н, 50Н, 20Н, 00Н		00H , 00H , 88H , 0A8H , 0A8H , 0A8H , 50H		00Н,00Н,00Н,88Н,50Н,20Н,50Н		
	50		20		0A		0E		00		00		80		90		80		0E		40		96		00		00		00		
3.44	DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		6
Ma <i>c</i> ro-80 set	60		00		A8		88		88		C8		68		00		00		00		30		06		88		A8		88		0
Mac set	00	20	70	DO	A8	00	88	00	88	00	BO	80	00	68	00	80	00	80	F0	FO	48	00	90	00	88	00	88	50	00	50	2
()	48	20	20	00	A8	00	C8	00	70	70	00	BO	00	98	08	C8	80	78	08	40	40		90		88	20	00	A8	00	20	0
C BIOS) character	50	20	20	00	A8	00	BO	88	00	88	00	C8	80	98	08	BO	80	00	FO	40	40	00	90	68	00	50	00	A8	00	50	00
BASIC MSX ch	lFlC	1F20	1F23	1F27	lF2A	1F2E	lf3l	1F35	lF38	1F3C	lF3F	lF4 3	lF46	lF4A	lF4D	1F51	lF54	1F58	LF5B	lf5F	1F62	1F66	1F69	lF6D	1F70	1F74	1F77	lf7b	lF7E	1F82	
(MSX ROM -MSXCHR-	6130	6131	6132	6133	6134	6135	6136	6137	6138	6139	6140	6141	6142	6143	6144	6145	6146	6147	6148	6149	6150	6151	6152	6153	6154	6155	6156	6157	6158	6159	

1-85 PAGE 63-9		68н,08н,70н,00н,00н,0г8н,10н		20H,40H,0F8H,00H,18H,20H,20H		40H,20H,20H,18H,00H,20H,20H		20H,00H,20H,20H,20H,20H,00H,0C0H		20H,20H,10H,20H,20H,0C0H,00H		40H,0A8H,10H,00H,00H,00H,00H,00H		00Н,00Н,00Н,00Н,00Н,00Н,00Н,00Н		00H,00H,10H,38H,7CH,0FEH,0FEH		38H,7CH,00H,6CH,0FEH,0FEH,0FEH		7CH, 38H, 10H,00H, 38H, 38H, 0FEH		0FEH,0D6H,10H,7CH,00H,10H,38H		7СН,0FEH,7СН,38Н,10Н,00Н,00Н		78H,84H,84H,84H,84H,84H,78H,00H		00H,78H,0FCH,0FCH,0FCH,0FCH,78H		00Н,40Н,0FEH,48Н,70Н,48Н,82Н	
01-Jan-85		68н,08		20H,40		40H,20		20H,00		20H,20		40H,0A		00 ' H00		00'H00		38H,7C		7CH,38		0FEH,0		7CH,0F		78H , 8 4		00H , 78		00H ,4 0	
3.44		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB	
C BIOS) Macro-80 character set	86 88	08	F8 10	40	18 20 20	40 20 20 18	00 20 20	20 00 20 20	20 00 CO	20 20 10 20	20 C0 00	40 A8 10 00	00 00 00	00 00 00 00	00 00 00	00 00 10 38	7C FE FE	38 7C 00 6C	FE FE	7C 38 10 00	38 38 FE	FE D6 10 7C	00 10 38	7C FE 7C 38	10 00 00	78 84 84 84	84 78 00	00 78 FC FC	FC FC 78	40	70 48 82
BASI MSX	1F89	1F8C	1F90	1F93	1F97	1F9A	lf9E	lfal	lFA5	lfA8	1 FAC	lfAF	1FB 3	lFB6	l FBA	lfbd	lFCl	1FC4	1FC8	l FCB	1FCF	lFD2	1 FD6	1FD9	lfDD	1FE0	1FE4	1FE7	lfeb	1 FEE	lff2
(MSX ROM -MSXCHR-	6161	6162	6163	6164	6165	6166	6167	6168	6169	6170	6171	6172	6173	6174	6175	6176	6177	6178	6179	6180	6181	6182	6183	6184	6185	6186	6187	6188	6189	6190	6191

63-10			H		-				_		_		_		_		_						Н		θH	1					Н
63-	н, зсі		H,42F		H,1CF		H,00F		H,00E		H,00E		H,24H	•	H, OCH		H3 8 H	•	H, 3CH	•	H 00H		I,OFE		0.HO		H00.1		1.18н)H,50
PAGE	7СН,00Н,00Н,00Н,00Н,10Н,7ЕН,3СН		5AH, 34H, 00H, 00H, 00H, 40H, 42H		42H,52H,20H,00H,00H,00H,00H,1CH		1CH, 22H, 02H, 0CH, 00H, 00H, 00H		18H,7EH,18H,30H,6EH,00H,00H		00H,12H,7EH,3CH,52H,34H,00H		00H,00H,28H,7CH,2AH,22H,24H		00H,00H,00H,08H,5CH,6AH,0CH		30H,00H,00H,00H,00H,08H,0EH,38H		4СН, ЗАН, 00Н, 00Н, 00Н, 00Н, 00Н, 3СН	•	02H,02H,1CH,00H,00H,00H,00H		00H,00H,00H,00H,00H,00H,20H,0FEH		20H,7CH,0AAH,0B2H,64H,00H,00H		80H,82H,82H,82H,90H,60H,00H	- - -	1CH,00H,7CH,02H,02H,04H,18H		00H, 38H, 00H, 0FEH, 08H, 30H, 50H
Ρi	0Н,1С		0H,0C		0H,0C		сн, ос		0H,6E		сн,52		CH, 2A		8H, 5C		0H,08		О 0 , НС	•	00'нс		00'HC)B2H,		2H,90	-	2Н, 02	•	РЕН , О
10	0 , HOC		0, HOC		20H,0		0.1H,0		.8H,3		/ EH, 3		28H,7		0H,0		0H,0		0H,00		CH,0(OH, 01		AAH, (•	2H, 8,	•	CH, 01	•	0H,01
01-Jan-85	00Н,(34H,(52H,		22H,(7 EH,]		12H,7		00H,2		00Н,С		0 O H , C		3AH,0		02H,1		00H,0		7 CH,0	•	82H,8	•	00H,7	• .	38H , 0
01-0	7CH ,		5AH,		42H,		1CH,		18н,		00H,		00Н,		0 OH		30Н,		4CH,		02Н,		0 0 H		20H,	•	80H,		1CH,		, ноо
14																															
3.44	DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB
Macro-80 set	-		_		_				_																						
Macro set	00 0	•	00 0	•••	00 (•	20	_	30	_	30	_	7C		08		00		00		00		00		B2		82		02		FΕ
		30	00	42	20	1C	02	00	18	00	7E	00	28	24	00	00	00	38	00	ЗС	IC	00	00	FE	AA	00	82	00	7C	18	00
ter (00	7 E	34	40	52	00	22	00	7E	00	12	34	00	22	00	6A	00	OE	3A	00	02	00	00	20	7C	00	82	60	00	04	38
C BIOS) character	70	10	5A	00	42	00	lC	00	18	9 Е	00	52	00	2A	00	50	30	08	4C	00	02	00	00	00	20	64	80	90	ΓC	02	00
BASIC BIOS MSX charac	lff5	l FF9	1 FFC	2000	2003	2007	200A	200E	2011	2015	2018	201C	201F	2023	2026	202A)2D	2031	2034	2038	203B	203F	142	2046	2049	204D	2050	2054	2057	205B	05E
	F	A	Ξ	5	5	5	5	5	2	5	2(20	5	5	20	5	2(20	5	20	20	20	20	20	20	20	20	20	20	20	20
ROI HR-																															
(MSX ROM -MSXCHR-	6192	6193	6194	6195	6196	6197	6198	6199	6200	6201	62 02	6203	6204	6205	6206	6207	6208	6209	6210	6211	6212	6213	6214	6215	6216	6217	6218	6219	6220	6221	6222
) – V –	Q	é	é	9	9	9	و	9	9	6	6	6	9	6	62	62	6	62	62	6	62	62	62	629	62	62	62	62	62	62	62

63-11	,0A2H		н , 4АН	,08н		,18н		,04н		,00H		, 7EH		, 80Н		84H		, 38н		OFEH		,28Н		ноо'		Н00'		5 OH	
	, 7CH		12.10	OFCH		18н,		00Н,		,48H		,80н		,7CH		80H,		,08н		44H,(64H,		ноо,) (НОС		90Н,(
PAGE	АН, 22н,)Н , 44Н,	I,10H,0		I,00H,I		I,18H,(H,84H		H, 4 OH		3H,08H		1,80H,8		I,OFEH,		I,04H,4		1,00H,6)H, 3CH		I,00H,C		I,60H,9	
01-Jan-85	9 EH,00H,20H,0FAH,22H,7CH,0A2H		UAZH,4CH,UUH,4UH,44H,UFZH,4AH	48H,88H,30H,00H,10H,0FCH,08H		3EH,04H,80H,7CH,00H,18H		304,604,604,304,184,004		84H,0BEH,84H,84H,84H		00H,0FCH,02H,00H,40H,80H		00H,10H,16H,0F8H,08H,7CH,80H		78H,00H,80H,80H,80H,80H,80H,84H		88H,70H,00H,08H,0FEH,08H,38H		48H, 38H, 08H, 00H, 04H, 44H, 0FEH		44H,44H,40H,3EH,00H,64H		30H, 0FEH, 20H, 40H, 3CH, 00H, 00H		ноо, ноо, ноо, ноо, ноо, ноо		00H,00H,00H,00H,60H,90H,60H	
4																													
3.44	DB	t t	20	DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB	
Macro-80 set	FA	c	4 U	00		7C		30		84		00		F8		80		08		00		3E		40		00		00	
Macr set	50 20 F		400 44		08	80 7	18	60 3	04	84 8	00	02 0	7E	16 F	80	80 8	84	0 00	38	080	ΕE		28	20 4	00	0 00	00		60
	30 5 00 2	7C A			F C O	04 8	18 1	60 6	00	BE 8	48 C	FC	80 7	10 1	7C 8	8 00	80 8	70 0	08 3		44 F	44 4	64 2	FE 2	00	00	00	0 00	906
icte	08 9E 0		A2 44 F		10 F	3E C	00	30 6	18 0	84 E	84 4	00 F		00 1	08 7	78 0	80 8	88 7	FE 0	48 3	04 4	44 4	00	30 F	30.0	00	00	0 00	0
C BIOS) character	0 0	0	4 9	Ф	Η	(°)	0	m	Г	ω	ω	0	4	0	0		ω	ω	щ	4	0	Ф	0	m	m	0	0	0	9
BASIC MSX ch	2062 2065	2069	2070 2070	2073	2077	207A	207E	2081	2085	2088	208C	208F	2093	2096	209A	209D	2 0 A 1	20A4	20A8	20AB	20AF	20B2	20B6	2 0 B 9	2 0 B D	20C0	20C4	20C7	20CB
(MSX ROM -MSXCHR-	6223 6224	6225	6227 6227	6228	6229	6230	6231	6232	6233	6234	6235	6236	6237	6238	6239	6240	6241	6242	6243	6244	6245	6246	6247	6248	6249	6250	6251	6252	6253

(MSX ROM -MSXCHR-	BAS I MSX	C BIOS) character		Ma cı set	Macro-80 set	3.44	01-Jan-85	PAGE 63	63-12
6254	2 OCE	00	38	20	20	DB	00H, 38H, 20H, 20H, 20H, 00H, 00H	н,20н,00н,00	Н
6255	2 0 D 2	20	00	00					
6256	20D5	00	00	00	00	DB	00H,00H,00H,00H,00H,20H,20H	н,00н,20н,20	Н
6257	20D9	00	20	20					
6258	2 0DC	20	- ЕО	00	00	DB	20H,0E0H,00H,00H,00H,00H,00H,00H	04,004,004,0	HO
6259	2 0E0	00	00	00					
6260	20E3	80	40	20	00	DB	80H,40H,20H,00H,00H,00H,00H	н,00н,00н,00	Н
6261	2 0E7	00	00	00					
6262	20EA	30	30	00	00	DB	30H, 30H, 00H, 00H, 00H, 0F8H, 08H	Н,ООН,ОF8Н,О	8H
6263	2 0 E E	00	н 18	08					
6264	20F1	F8	08	10	20	DB	0F8H,08H,10H,20H,40H,00H,00H	OH,40H,00H,0	НО
6265	20F5	40	00	00					
6266	20F8	00	FО	10	60	DB	00H,0F0H,10H,60H,40H,80H,00H	0H,40H,80H,0	HO
6267	20FC	40	80	00					
6268	2,0FF	00	10	20	60	DB	00H,10H,20H,60H,0A0H,20H,20H	н, одон, 20н, 2	HO
6269	2103	AO	20	20					
6270	2106	00	00	20	FO	DB	00H,00H,20H,0F0H,90H,10H,20H	он,90н,10н,2	HО
6271	210A	90	10	20					
6272	210D	40	00	00	00	DB	40H,00H,00H,00H,0F0H,20H,20H	Н,ОFOH,20Н,2	HO
6273	2111	FO	20	20					
6274	2114	20	FΟ	00	00	DB	20H,0F0H,00H,00H,20H,20H,0F0H,60H	0H,20H,0F0H,	60Н
6275	2118	20	FO	60					
6276	211B	AO	AO	20	00	DB	0A0H,0A0H,20H,00H,00H,40H,0F8H	00H,00H,40H,	0F8H
6277	211F	00	40	F1 8					
6278	2122	48	50	40	40	DB	48H,50H,40H,40H,00H,00H,00H	Н,00Н,00Н,00	Н
6279	2126	00	00	00					
6280	2129	70	10	10	10	DB	70H,10H,10H,10H,0F8H,00H,00H	Н, ОF8Н, ООН, О	HO
6281	212D	F8	00	00					
6282	2130	00	FO	10	F0	DB	00H,0F0H,10H,0F0H,10H,0F0H,00H	FOH,10H,0F0H	H00 ' 1
6283	2134	10	FO	00					
6284	2137	00	00	A8	A8	DB	00H,00H,0A8H,0A8H,08H,08H,10H,20H	А8Н,08Н,10Н,	20H

63-13		ноо, но		ОН, 20Н		H,60H		78H,88H		H,0F8H		ЭН,10Н		ноо,но		3H , 88H		20H,20H		H80, H		Н06'І		3H , 0 8H)H,0F8H		H00, H		ЭЕОН, ООН	
PAGE		OH, OF8H, O(08H,28H,3(8H,10H,20H		00H,20H,01		ОН, ООН, ООН		OH, OF8H, O(50H,90H,1(28H,28H,48		20H,0F8H,2		,48Н,88Н,08Н,08Н		0H,78H,50H		0H,00H,0F8		F8H,00H,5(0H,20H,00F		,08H,10H,0	
01-Jan-85		00H,00H,00H,00H,00H,0F8H,00H,00H		00H,00H,0F8H,08H,28H,30H,20H		20H,40H,00H,08H,10H,20H,60H		0A0H,20H,20H,00H,20H,0F8H		88H,08H,10H,20H,00H,00H,0F8H		20H, 20H, 20H, 20H, 20H, 0F8H, 00H, 10H		0F8H,10H,30H,50H,90H,10H,00H		20H,0F8H,28H,28H,28H,28H,48H		00H,20H,0F8H,20H,0F8H,20H		20H,00H,78H,4		10H,20H,00H,40H,78H,50H,90H		10H,10H,20H,00H,00H,0F8H,08H		08H,08H,08H,08H,0F8H,00H,50H,0F8H		50H, 50H, 10H, 10H, 20H, 00H, 00H		0C0H,08H,0C8H,08H,10H,0E0H,00H	
3.44		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB	
Macro-80 set		00		08		08		00		20		20		50		28		20		48		40		00		F8		10		08	
Macı set	20	00	00	F8	20	00	60	20	88	10	F8	20	10	30	00	28	88	F8		78	08	00	90	20	08	08	F8		00	C8	00
) ter	10	00	00	00	30	40	20	20	F8	08	00	20	00	10	10	F8	48	20	20	00	08	20	50	10	F8	08		50	00	08	Е0
C BIOS) character	08	00	F8	00	28	20	10	AO	20	88	00	20	F8	F8	90	20	28	00	F8	20	88	10	78	10	00	08	00	50	20	S	10
BASIC F MSX cha	213B	213E	2142	2145	2149	214C	2150	2153	2157	215A	215E	2161	2165	2168	216C	216F	2173	2176	217A	217D	2181	2184	2188	218B	218F	2192	2196	2199	2 19D	21A0	21A4
(MSX ROM -MSXCHR-	6285	6286	6287	6288	6289	6290	6291	6292	6293	6294	6295	6296	6297	6298	6299	6300	6301	6302	6303	6304	6305	6306	6307	6308	6309	6310	6311	6312	6313	6314	6315

63-14	Н		HO	m		п		HO		JA8H		HC		HC		F		HC		н		HOUC		3H		3H		F		Ŧ
63.	00Н,0F8Н,08Н,10Н,20Н,50Н,88Н		00H,40H,0F8H,48H,50H,40H,40H	,48H,08H,10H	•	20H,40H,00H,78H,48H,78H,88H		08H,10H,20H,00H,10H,0E0H,20H		0F8H,20H,20H,40H,40H,00H,0A8H,0A8H		0A8H,08H,08H,10H,20H,00H,70H		00H,0F8H,20H,20H,20H,20H,40H,00H		40H,40H,60H,50H,48H,40H,40H		00H,20H,0F8H,20H,20H,20H,20H		40H,00H,00H,70H,00H,00H,00H		00H,0F8H,00H,00H,0F8H,08H,0D0H		20H, 50H, 88H, 00H, 20H, 0F8H, 08H		30H,0E8H,20H,20H,00H,08H,08H		08H,10H,20H,40H,80H,00H,20H		10H,48H,48H,48H,48H,48H,88H,00H
PAGE	20H,5	1	20H,4	8H,08		8H,78		OH, 0I		00H,C		20H,C		20H,4		8H,40		20H,2		0H,0C		0F8H,		OH, 0E		00H,C		OH, OC		8H , 88
ц	,10Н,		,48H,	88H.4		78H,4		00Н,1		,40H,		,10H,		,20H,		50H,4		,20Н,		7 OH, 0		, ООН,		00H,2		,20Н,		40H,8		4 8H , 4
85	н,08н		, 0F8H	38H,00H,88H,88H		, ноо,		,20Н,		Н,20Н		Н,08Н		Н,20Н		, 60н,		,0F8H		, ООН,		н, оон		,88н,		Н,20Н		,20H,		,48H,
01-Jan-85	H,0F8	•	н,40Н	Н00, Н	•	H , 4 OH		Н,10Н		8H,20		8H,08		H,0F8		H,40H		Н,20Н		ноо, е		H,0F8		H02,E		H, 0E8		H01,E		Н 48Н
01	00		00	38]		201		081		0F:		0 A 8		100		401		00		401		100		201		301		081		101
3.44	DB	ļ	DB	DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB
(*)			H					Г				Ц		ц		Ц		Ц		I		ы		Ц		Ц		Ц		П
0																														
Macro-80 set	10	•	48	88		78		00		40		10		20		50		20		70		00		00		20		40		48
Mac set	08	88	н с И К	88	10	00	88	20	20	20	A8	08	70	20	00	60	40	F.8	20	00	00	00	DO	88	08	20	08	20	20	48
(er	F 18	50	40	00	08	40	78	10	ЕO	20	A8	08	00	F8	40	40	40	20	20	00	00	F8	08	50	F8	E8	08	10	00	48
C BIOS) character	00	20	00 2	8 8 8 8	48	20	48	08	10	F.8	00	A8	20	00	20	40	48	00	20	40	00	00	F8	20	20	30	00	08	80	10
BASIC MSX ch	21A7	21AB	2 LAE	21B5	21B9	21BC	21C0	21C3	21C7	21CA	21CE	21D1	21D5	21D8	21DC	21DF	21E3	21E6	21EA	21ED	21F1	21F4	21F8	21FB	21FF	2202	2206	2209	220D	210
Σ	2	2	2 0	1 0	2	7	7	2	2	7	7	7	2	7	7	7	2	2	0	2	2	2	2	2	2	2	N	2	2	2
MSX ROM ISXCHR-		~		. ~		~ .					~	~	~	~		<u>.</u>						~	~	~		<u>.</u> .				
(MSX RO -MSXCHR-	6316	6317	6318	6320	6321	6322	6323	6324	6325	6326	6327	6328	6329	6330	6331	6332	6333	6334	6335	6336	6337	6338	6339	6340	6341	6342	6343	6344	6345	6346

63-15		0Н,78Н		Н,20Н	•	Н,08Н	•	Н,20Н		F8H,08H	•	Н00.Н	•	Н,10Н		H00'E		, 80Н		Н,20Н	•	Н,50Н	-	,10H		3H , 0 8H		H00, H0		I , 4 8H	
PAGE		, 80Н, 80Н, 8	•)8H,08H,10		OH,0A0H,10	•	DH,0F8H,20		О, НОО, НОО,	•	H,00H,0F0	•	OH, 08H, 001		H,88H,0F8		H,20H,50H	•	'8H,20H,201		'8H,48H,48I		H,70H,10H		OH,00H,0F		F8H,00H,7(0H,20H,00I	
01-Jan-85		80H,80H,0F8H,80H,80H,80H,80H,78H		00H,0F8H,08H,08H,08H,08H,10H,20H	•	40H,00H,00H,40H,0A0H,10H,08H		08H,00H,00H,20H,0F8H,20H,20H		0A8H,0A8H,20H,00H,00H,0F8H,08H	•	08H,50H,20H,10H,00H,0F0H,00H		60H,00H,00H,0F0H,08H,00H,10H		20H, 40H, 80H, 90H, 88H, 0F8H, 00H		08H,08H,08H,50H,20H,50H,50H,80H	•	00H,78H,20H,0F8H,20H,20H,20H	•	18H,00H,40H,0F8H,48H,48H,50H	•	40H,40H,00H,00H,70H,10H,10H		10H,10H,0F8H,00H,00H,0F8H,08H	•	0F8H,08H,08H,08H,0F8H,00H,70H,00H	•	0F8H,08H,08H,10H,20H,00H,48H	
3.44		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB	
Macro-80 set	00	F8 80	78	08 08	20	00 40	08	00 20	20	20 00	08	20 10	00	00 FO	10	80 90	00	08 50	80	20 F8	20	40 F8	50	00 00	10	F8 00	08	08 F8	00	08 10	8
	88	80]	08	F8	10	00	10	00	20	A8	F8 (50	FO	00	00	40 8	F8 (08 (50	78	20	7 00	48	40 (10	10 1	F8 (080	70 0	08 0	00 4
C BIOS) character	48	80	80	00	08	40	A0	08	F8	A8	00	08	00			20	88		20			18	48	40	70	10	00	F8	00	F8	20 (
BASIC F MSX cha	2214	2217	221B	221E	2222	2225	2229	222C	2230	2233	2237	223A	223E	2241	2245	2248	224C	224F	2253	2256	225A	225D	2261	2264	2268	226B	226F	2272	2276	2279	227D
(MSX ROM -MSXCHR-	6347	6348	6349	6350	6351	6352	6353	6354	6355	6356	6357	6358	6359	6360	6361	6362	6363	6364	6365	6366	6367	6368	6369	6370	6371	6372	6373	6374	6375	6376	6377

63-16	Н00'Н	H06,H		Н,50Н		8H , 88H		НВИ, ИВН	лоо по	1100/110	0H,48H		Н,60Н		н,00н		АОН,9ЕН		,04н,04н		2H , 02H		СН, 30Н		н,1сн		H,0F2H		но8,но	
PAGE	8H,10H,20	OH,50H,58		OH,48H,48		F8H,88H,8		UF8H,88H,			ЕОН,00Н,9		OH, OOH, OO		00, ноо, но		5EH,80H,0		40H,0F8H,		0H,0FCH,0		0H,0FEH,0		38H,00H,10H,12H,1CH		,40H,3EH,00H,24H		AAH,10H,0	
01-Jan-85	48H,48H,48H,48H,48H,10H,20H,00H	104,504,504,504,504,504,584,904		00H,40H,40H,40H,40H,48H,48H,50H		60H,00H,00H,0F8H,88H,88H,88H		88H, UF8H, U0H, UF8H, 88H, 88H, U8H	08H 10H 30H 00H 00H 0CH01 H00		08H,08H,10H,0E0H,00H,90H,48H		ноэ, ноо, ноо, ноо, ноо, ноо, ноо		90H,60H,00H,00H,00H,00H,00H		40H,0FEH,40H,5EH,80H,0A0H,9EH		00H,20H,0FEH,40H,0F8H		78H,00H,00H,00H,0FCH,02H,02H		04H, 38H, 00H, 00H, 0FEH, 0CH, 30H		40H,40H,38H,0	•	30H,40H,40H,3		48H,48H,9CH,0AAH,10H,00H,80H	
3.44	DB	DB		DB	l	DB	l	DB	מע	2	DB		DB		DB		DB		DB		DB		DB		DB		DB		DB	
Macro-80 set	48 48	50 50	06	40 40		00 F8		00 F8	08 20 00		10 E0		00 00	60	00 00	00	40 5E	9E	FE 40	04	00 00	02	00 00	30	38 00	1C	40 3E	F2	9C AA	
	48	2 O 2 O	58	40	48	00	88	н 19 19 19 19 19 19 19 19 19 19 19 19 19	20 0		80		00	00	60	00	ЪE	AO	20	04	00	02	38	Я	40	12	40	24	48	
C BIOS) character	48	10	50	00	48	60	88	800		88	080	00	00	00	06	00	40	80	00	F8	78	F D	04	ΕE	40	10	30	00	48	
BASIC MSX ch	2 12	2287 2287	228B	228E	2292	2295	2299	229C	22AU	LACC	22AA	22AE	22Bl	22B5	22B8	22BC	22BF	22C3	22C6	22CA	22CD	22D1	22D4	22D8	22DB	22DF	22E2	22E6	22E9	
(MSX ROM -MSXCHR-	6378	63 80 63 80	6381	6382	6383	6384	6385	6386	638/ 6200	0200 6389	6390	6391	6392	6393	6394	6395	6396	6397	6398	6399	6400	6401	6402	6403	6404	6405	6406	6407	6408	

63-17	ноо'но	ли бон		Н,4АН		0A2H		Н,84Н		, осбн		18H		ноо,		ноо,		Н,18Н		4CH		,0A4H		Н,62Н		, 7CH		Н,20Н	
PAGE)Н,0ВЕН,0С	AU-DAGH OA		2H,62H,0CE		Н,54Н,92Н,		H, OBEH, 84)H,08H,4CH		Н,00Н,20Н,		АН, 18Н, 00Н		сн, 06н, 02н		IH, 9CH, 0A6		I,7EH,38H,		IH,24H,7EH)H,0FCH,24		H,04H,44H		4H,00H,20	
01-Jan-85	9 EH , 8 0H , 8 0H , 0 A 0H , 0 B EH , 0 C 0H , 0 0 H	44Н - 4CH - 7АН - 0ААН - 0А6Н - 0ААН - 6CH		00H,40H,0ECH,52H,62H,0CEH,4AH		4CH,00H,00H,38H,54H,92H,0A2H		0A2H,4CH,00H,04H,0BEH,84H,84H		9EH,0A4H,5CH,00H,08H,4CH,0C6H		46H,44H,44H,38H,00H,20H,18H		20H,16H,8AH,0CAH,18H,00H,00H		20H,70H,0D8H,8CH,06H,02H,00H		3EH,84H,0BEH,84H,9CH,0A6H,18H		00H,08H,7EH,08H,7EH,38H,4CH		3AH,00H,0E0H,24H,24H,7EH,0A4H		0A4H,68H,00H,20H,0FCH,24H,62H		0A0H,62H,3CH,00H,04H,44H,7CH		0С6Н,0ААН,92Н,64Н,00Н,20Н,20Н	
-10	1 <u>3</u> 6	441	1	100		4CI		0A3		ЭEI		4 6 F		20F		20F		3EI		00E		3AI		0A4		0A(006	
3.44	DB	DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB		DB	
80																													
Macro-80 set	AO	AA		52		38		04		00		38		CA		8C		84		08		24		20		00		64	
Mac set		00 7A		ы С					84	50	C6	44	18	8A	00	D8	00	BE	18	7 E	4C	Е0 Е	A4	00	62	ЗС	7C	92	20
) ter		00 40		40					84	A4	4C	44	20	16	00	70	02	84	A6	08	38	00	7E	68	24	62	44	AA	20
C BIOS) character	10 9E	BE 44	A6	00	62	4C	54	A2	BE	ЭE	08	46	00	20	18	20	90	3E	90	00	7E	3A	24	A4	ក្ដ	AO	04	C6	00
BASIC MSX ch	22ED 22F0	22F4 22F7	22FB	22FE	2302	2305	2309	230C	2310	2313	2317	231A	231E	2321	2325	2328	232C	232F	2333	2336	233A	233D	2341	2344	2348	234B	234F	2352	2356
(MSX ROM -MSXCHR-	6409 6410	6411 6412	6413	6414	6415	6416	6417	6418	6419	6420	6421	6422	6423	6424	6425	6426	6427	6428	6429	6430	6431	6432	6433	6434	6435	6436	6437	6438	6439

01-Jan-85 PAGE 63-18	78H,20H,78H,22H,1CH,00H,00H		48H,0FCH,4AH,42H,4CH,40H,00H		08H,0BCH,0CAH,8AH,0BCH,08H,30H		00H,08H,08H,0EH,08H,78H,8CH		72H,00H,38H,84H,80H,0FCH,0C2H	02H, 38H,00H,00H,42H,42H,42H		62H,04H,18H,00H,7CH,08H,30H	0DCH,62H,92H,7CH,00H,20H,2CH	0F4H,24H,64H,0E4H,26H,00H,7CH		18H,20H,5CH,82H,02H,7CH,00H	40H,60H,0DCH,62H,42H,0C2H,5CH		00H,10H,30H,20H,70H,48H,0CEH		84H,00H,00H,00H,00H,00H,00H,00H		00Н,00Н,00Н,00Н,00Н,00Н,00Н,00Н		00Н,00Н,00Н,00Н	L, Screen editor - Line input and function character
3.44	DB		DB		DB		DB		DB	DB		DB	DB	DB		DB	DB		DB		DB		DB		DB	SUBT'TL - MSXINL,
BASIC BIOS) Macro-80 MSX character set	59 78 20	35D IC	48	4 C	2367 08 BC CA 8A	Я				02	42	2383 62 04 18 00			26	18	40	42		3 AA 70	3AD 84 00	3 B 1 00		3 B8	3BB	
(MSX ROM -MSXCHR-				6443	6444	6445		6447	6448	6450		6452					6460		6462	6463			6466		6468	6469

PAGE 64		;During AUTO mode?	;Yes, then fake INLIN to prevent 0 from ;deleting line number	get input		· · · · · · · · · · · · · · · · · · ·	rerminate previous line;	;Mark first position
01-Jan-85 tion character	oint	H.PINL A, (AUTFLG) A	NZ,INLIN L,O	INLINI ion mark then	H.QINL A,'?' 18H A,'' 18H	H.INLI HL,(CSRY) L	NZ, TERMIN L	(FSTPOS), HL A (INTFLG), A
Macro-80 3.44 01-Jan-85 - Line input and function character	PINLIN: ; ; Main entry point	CALL LD AND	ម្តី ភ្នំ	JR INLINI QINLIN: ; ; Output question mark then get input ;	CALL LD RST RST	INLIN: CALL LD DEC	CALL INC INLINL:	LD XOR LD
BASIC BIOS) Macro-80 Screen editor - Line i		CD FDDB 3A F6AA A7		Ω T	CD FDE0 3E 3F DF 3E 20 DF		C4 0C29 2C	22 FBCA AF 32 FC9B
(MSX ROM BASI(- MSXINL, SCT e	6470 6471 23BF 6472 6473 6474	6475 23BF 6476 23C2 6477 23C5		6481 23CA 6482 23CC 6483 6484 6485			000	6498 23E0 6499 23E3 6500 23E4

PAGE 64-1				;SCI Max	;Do functions		;Output a character		;Not a terminator					;Cnt-C, return with carry set	;No, return carry clear		
01-Jan-85 tion character		CHGET	HL, SCITBL-2	С,0ВН	INDJMP	AF	NZ, INLOUT	AF	NC, INLIN2		return to BASIC (break or CR)		HL, BUFMIN	Z			
IOS) Macro-80 3.44 01-Jan-85 editor - Line input and function character	INLIN2:	CALL	ΓD	LD	CALL	HSUI	CALL	POP	JR	••	; return to BA	••	ГD	RET	CCF	RETURN:	RET
BIOS) Macro-80 . editor - Line ir		CD 10CB	21 2437	0E 0B	CD 0919	F5	C4 23FF	Fl	30 EE				21 F55D	C8	3F		C9
	23E7	23E7	23EA	2 3 E D	2 3EF	23F2	23F3	23F6	23F7				23F9	23FC	23FD	2 3 F E	23FE
(MSX ROM BASIC - MSXINL, Screen	6501	6502	6503	6504	6505	6506	6507	6508	6209	6510	6511	6512	6513	6514	6515	6516	6517

PAGE 65		:Save character to output	TAB?	: No pe	;Discard stack		;Map to space	4		:Make it zero based.	:Reached TAB ston?	Not vet, continue				:Restore character	:points insert mode flag	:Graphic header byte?	:Yes, send as is	control char?	;branch if so Reset insert mode	save char to output	get insert mode flag	thest	if insert mode, make room to incort	restore char to output		coutput char	
01-Jan-85 ion character		AF	6	NZ, OUTNTB	AF		A, ' '	INLOUT	A, (CSRX)	A	7	NZ, OUTTAB				AF	HL, INSFLG	I	Z, INLOTO	-	C, INLOTI	AF	A, (HL)	A	NZ, INSERT	AF		18H	
Macro-80 3.44 01-Jan-85 - Line input and function character	I NLOUT:	HSUQ	CB	JR	POP	OUTTAB:	LD	CALL	LD	DEC	AND	JR	RET	OUTNTB:	•••	POP	LD	CP	JR	CP	JR	PUSH	LD	AND	CALL	POP	INLOT0:	RST	RET
BIOS) Macro-80 editor - Line i		F5	FE 09		Fl			7	Бц	3D		20 F3	C9			Fl				FE 20		F5	7E	A7	C4 24F2	Fl		DF	C9
BASIC Screen	23FF	23FF	2400	2402	2404	2405	2405	2407	240A	240D	240E	2410	2412	2413		J.	4	J.	4	4	4	4	4	4	4	4	4	2426	<#
(MSX ROM - MSXINL,	6518 6519 6520	5	5	ы.	ີດ	ы́.	10	5	10	Б.	8	8	5	5	5	3	<u> </u>	<u>.</u>	3	<u>.</u>	4	54	54	54	6544	54	54	6547	54

PAGE 65-1		;reset insert mode	;send this control char			;Set insert mode and exit		;Set overwrite mode					
01-Jan-85 ion character		(HT) , 0	18H	3EH		3EH		А	AF	CKERCS	AF	(CSTYLE), A	CKDPCS
BASIC BIOS) Macro-80 3.44 01-Jan-85 Screen editor - Line input and function character	I NLOTI :	ΓD	RST	DB	SETINS:	DB	SETOVW:	XOR	HSUG	CALL	POP	LD	JP
MSX ROM BASIC BIOS) Macro-80 MSXINL, Screen editor - Line in		36 00	DF	3E		3E		AF	F5	CD 0A2E	Fl	32 FCAA	C3 09E1
BASIC Screen	2428	2428	242A	242B	242C	242C	242D	242D	242E	242F	2432	2433	2436
(MSX ROM - MSXINL,	6549 6550	6551	6552	6553	6554	6555	6556	6557	6558	6559	6560	6561	6562

66		
PAGE		
3.44 01-Jan-85	and function character	
(MSX ROM BASIC BIOS) Macro-80	- MSXINL, Screen editor - Line input and function character	6563

	;Delete previous char		;Toggle insert flag	: Es ca pe		; Back word		;Next word				; Erase to end of line		; Abor t		;Carriage return	1	;Delete whole line		:Delete character at cursor)	- Process special characters
BL: ole of function characters	-		DB 12H DW THETTING			DB 02H		DB 06H				DB 05H			DW LBREAK		DW LCRRET	DB 15H	DW LERASE	DB 7FH	DW LDELNX	- MSXINL, Screen editor
SCITBL: ; Table ;	08	2561	12 24F5	lB	2 3 FE	02	260E	06	25F8	0E	25D7	05	25B9	03	24C5	0D	245A	15	25AE	7F	2550	SUBTTL
2439	43	43	243C 243D	43	2440	2442	44	44	4	2448	4	244B	4	244E	5	2451	10	-	2455		-	
6563 6564 6565 6565 6566	6568	5 C	6571 6571	2		~	57	~	57	~	57	58	\sim	\sim	58	58	58	58	58	\sim	~	6590

240	number of first visual AUTO mode? get from top of line during AUTO mode iffer pointer iffer pointer iffer cont iffer count iffer count iffer count iffer count iffer count iffer count iffer count iffer count iffer count iffer count
PAGE 67	L=line During No Always BUF BUF Max cou Max cou Max cou save bu Save Save bu Save bu S
01-Jan-85 racters	LCRRET: :::::::::::::::::::::::::::::::::::
ro-80 3.44 0l-Ja Process special characters	LCRRET: :::::::::::::::::::::::::::::::::::
r Mao	CD 266C 3A F6AA A7 A7 A7 26 01 26 01 26 01 26 01 11 F55E 20 20 20 20 20 20 20 20 20 20 20 20 20
M BASIC BIOS) , Screen edito	2477 2477 24476 24477 24476 24476 24477 24470 24470 24470 24470 24470 24470 24470 24470 24470 24470 24470 24470 24470 24470 24470 24470 24470
(MSX ROM - MSXINL,	6591 6593 6593 6594 6594 6595 6599 6601 6601 6603 6603 6603 6603 6603 6603

PAGE 67-1 24	;Yes, ignore this	Special graphic character?	No, proceed normally	Decrement BUF size counter before storing	At end of BUF, so ignore this		;Store header byte for graphic symbol						;Store byte in buffer	;Bump buffer pointer	Decrement BUF size counter	At end of BUF		;Next column	;Max column reached?		Not yet	Save buffer pointer	Is this line terminated?	Restore buffer pointer;	Assume not, start from top of next line	; No			last+l		Back up buffer pointer
01-Jan-85 racters	Z, LCRNUL		NC, LCRNRM	В	Z,LBLKSP	C, A	A,1	(DE),A	DE	A,C	A, 'e'		(DE),A	DE	В	Z,LBLKSP		Н	A, (LINLEN)	Н	NC, LCR2	DE	GETTRM	DE	Н,1	Z,LCR1			Suppress trailing blanks, [DE]=last+l		DE
ro-80 3.44 01-Ja Process special characters	JR	CP	JR	DEC	JR	ΓD	LD	ΓD	INC	LD	ADD	LCRNRM:	LD	INC	DEC	JR	LCRNUL:	INC	LD	CP	JR	PUSH	CALL	POP	LD	JR	LBLKSP:		; Suppress trai	••	DEC
BIOS) Mac editor -	28 14	FE 20	30 0B	05	28 1D	4F	3E 01	12		79	C6 40		12	13	05	28 10		24	3A F3B0	BC	30 DB	D5	CD 0C1D	DI	26 01	28 D1					1B
BASIC Screen	2479	247B	247D	247F	2480	2482	2483	2485	2486	2487	2488	248A	248A	248B	248C	248D	248F	248F	4	2493	4	49	2497	249A	249B	249D	249F				249F
(MSX ROM - MSXINL,	6622	6623	6624	6625	6626	6627	6628	6629	6630	6631	6632	6633	6634	6635	6636	6637	6638	6639	6640	6641	6642	6643	6644	6645	6646	6647	6648	6649	6650	6651	6652

PAGE 67-2	;Get stored character	;Is it space?	;Yes, ignore this									; Point past last valid character	;Load terminator	;Put it in BUF		;Load character to echo to console	;Reset Z-flag, (say not break)		; Save this flag		; Save current cursor position		; Move cursor to start of next line	;Clear possible INSFLG		;Restore flags	;Set carry indicating end of input	;Discard return address (XRA A;RET)	;If break, Z flag is set		
01-Jan-85 racters	A, (DE)		Z,LBLKSP	HL	DE	CKDPCS	DE	HL				DE	A	(DE),A		A ,0DH	A		AF	TERMIN	POSIT	A , 0AH	18H	А	(INSFLG), A	AF		ΗL			
rro-80 3.44 01-Ja Process special characters	LD	CP	JR	HSUG	PUSH	CALL	POP	POP	••	; Terminate	•••	INC	XOR	LD	FAKECR:	ΓD	AND	LNXTLN:	PUSH	CALL	CALL	LD	RST	XOR	LD	POP	SCF	POP	RET	LBREK0:	••
) Mac tor -		FE 20	28 FA	E5	D5	CD 09E1	DI	El				13	AF	12		3E 0D	A7		F5	0		0	DF	AF	32 FCA8	Fl	37	El	C9		
BASIC BIOS Screen edi	24A0	24A1	24A3	24A5	24A6	24A7	24AA	24AB				24AC	24AD	24AE	24AF	24AF	24Bl	24B2	24B2	24B3	24B6	24B9	24BB	24BC	24BD	24C0	24C1	24C2	24C3	24C4	
(MSX ROM - MSXINL,	6653	6654	6655	6656	6657	6658	6659	6660	6661	6662	6663	6664	6665	6666	6667	6668	6669	6670	6671	6672	6673	6674	6675	6676	6677	6678	6679	6680	8	6682	6683

	;Bump line counter		;Line terminated?	:No, check next line	;Set to overwrite mode	;Load 0 in Acc, and set Z flag	Say no character in BUF	;Set to first column	;Save cursor position	;Initialize sound chip and queue	Check if STOP trap is active or not		;Yes, fake CR	Executing BASIC program in ROM?		:Yes. fake CR	
rlocess special characters ; Control-C input ;	Г		GETTRM	Z,LBREKO	SETOVW	A	(BUF),A	Н,1	НL	GICINI	CKSTTP	HL	C, FAKECR	A, (BASROM)	A	NZ, FAKECR	LNXTLN
; Control-C input ;	INC	LBREAK:	CALL	JR	CALL	XOR	LD	LD	HSUG	CALL	CALL	POP	JR	LD	AND	ЯĊ	JR
	2C			28 FA		AF	32 F55E	26 01	E5	CD 04BD		EI	38 D2		A7		18 CD
	24C4	24C5	24C5	24C8	24CA	24CD	24CE	24D1	24D3	24D4	24D7	24DA	24DB	24DD	24E0	24E1	24E3
6684 6685	6686	6687	6688	6689	6690	6691	6692	6693	6694	6695	6696	6697	6698	6699	6700	6701	6702

PAGE 68	;Get current insert flag	;Toggle insert status and affect Z flag ;Set to overwrite mode	;Set to insert mode	Frase cursor before operation Load raw code for space	;Save current cursor position ;Save previous character ;Get current character in C	Restore previous character in [E] Save current character C=previous character	<pre>;Put it on screen ;Restore current character in C ;Check if end of line ;Bump column counter ;End of line?</pre>
01-Jan-85 racters	t mode flag HL,INSFLG A,(HL)	0FFH (HL),A Z,SETOVW	SETINS nk	CKERCS HL, (CSRY) C, '''	HL BC GETVRM	DE BC C,E	PUTVRM BC A,(LINLEN) H H
ro-80 3.44 01-Ja Process special characters	TGLINS: ; ; Toggle insert mode ; LD HL,IN LD A,(HL	XOR LD	JP INSERT: ; ; Insert a blank ;	CALL LD LD INSI:	PUSH INS2: PUSH CALL	POP PUSH LD	CALL POP LD CP CP
BIOS) Macro-80 . editor - Proce	21 FCA8 7E	E N	C3 242C	CD 0A2E 2A F3DC 0E 20	E5 C5 CD (BD8		CD 0BE6 Cl 3A F3B0 24 BC
BASIC Screen	24E5 24E5 24E5 24E8	24E9 24EB 24EC	24EF 24F2	24F2 24F5 24F8 24FA	24FA 24FB 24FB 24FC	24FF 2500 2501	2502 2505 2506 2509 250A
(MSX ROM - MSXINL,	6703 6704 6705 6706 6707 6708 6709	6710 6711 6712	6713 6714 6715 6716 6717	6718 6719 6720 6721	6722 6723 6724 6725	6726 6727 6728 6728	6729 6730 6731 6732 6733

PAGE 68-1	Get current attribute in Acc If not, continue till end of line	line, code of character wrapped to next	Restore current cursor position: Is this line terminated?	;Line not terminated on this visual	The current line is terminated. A check must be made to	determine if a wrapped character is a space, or we're inserting	so, we have to open a next line to		; Move last character to A for comparison		; Save the condition	; No, open next line	;Are we trying to insert at the EOL?	••	;Yes, open next line	;Discard stack	;Display cursor again		:Unterminate this line	;Go to next row	;Save character code	; Save position of character in operation	;Bottom of screen?
4 01-Jan-85 characters	A, D NC, INS2	finished a l in [C].	HL GETTRM	Z, INS6	line is termi	a wrapped ch	f-line. If sc		A,C	-	AF	NZ, INS3	A, (LINLEN)	Н	Z, INS3	AF	CKDPCS		UNTERM	L	BC	用	GETLEN
3.4 special	LD JR	Now we just finished line is held in [C].	POP CALL	JR	The current	determine if	at the end-of-line. If insert.		LD	CP	PUSH	ĥ	LD	CP	JR	POP	JP	INS3:	CALL	INC	PUSH	HSUG	CALL
Macro-80 - Process	•						•~ ••											Ϊ.	-				
<u>ب</u> ا	7A 30 ED		El CD OCID	28 37					79	FE 20	F5	20 0A	3A F3B0		28 04	Fl	C3 09El		CD 0C2A	2C	C5		CD 0C32
BASIC BIOS) Screen editor	250B 250C		250E 250F	2512					2514	2515	2517	2518	251A	251D	251E	2520	2521	2524	2524	2527	2528	2529	252A
(MSX ROM BASIC BIOS) - MSXINL, SCreen edito	6734 6735 6736		6740 6741	6742	4 4	4	6746 6747	· •	4	ŝ	ŝ	ŝ	S	ŝ	ഹ	in	<u>م</u>	ເດີເຕ	i o	6	6	76	6764

6769 6771 6772 6772 6773 6773 6773 6774 6775 6775 6775 6775 6776 25335 6778 6779 2533 6781 2533 6781 2533 6782 2535 6781 2535 6783 2535 6783 2535 6783 2535 6783 2544 6783 2544 6783 2544 6783 2544 6783 2544 6783 2544 6783 2544 6783 2544 6783 2544 6783 2544 6783 2544 6783 2544 6783 2544 6783 2544 6783 2544 6783 25545 6783 25545 6783 25545 6783 25545 6783 25545 6783 25545 6783 25545 6783 25545 6783 25545 6783 25545 6783 25545 6783 25545 6783 25545 6783 25545 6783 25545 25545 6783 25545 6783 25545 6783 25545 6783 25545 6783 25545 6783 25545 6783 25545 6783 25545 6773 6773 6773 6773 6773 6773 6773 677
--

PAGE 68-3	<pre>Bump row counter ;Start from first column ;Pass character to next line</pre>
BASIC BIOS) Macro-80 3.44 01-Jan-85 Screen editor - Process special characters	; INC L LD H,l JR INSI
SIC BIOS) Macro-80 reen editor - Proces	254B 2C 254C 26 01 254E 18 AA
(MSX ROM BASIC - MSXINL, Screer	6796 6797 25, 6798 25, 6799 25,

						;At rightmost position?	; Nope	;Is this a terminated line?	;Yes, place a space there.		;Move cursor right	n L	;Fall into 'delete prev. character'								;Are we at top of line?	; No	;Yes	;Save current cursor position	Look a line above	;At top of screen			;Is previous line terminated?	; Yes
			nt character		A, (LINLEN)	Н	NZ, LDELXI	GETTRM	NZ, DELET5		A,1CH	18H	HL, (CSRY)			ous character		甩	CKERCS	HL	Н	NZ,DELET2	Н	HL	Г	Z,DELET1	A, (LINLEN)	Н,А	GETTRM	NZ,DELETI
	LDELNX:	••	; Delete current character	••	LD	CP	JR	CALL	JR	LDELX1:	LD	RST	ΓD	DELETE:		; Delete previous character	•	HSUG	CALL	POP	DEC	JP	INC	HSUT	DEC	JR	LD	LD	CALL	JR
					3A F3B0	BC	20 05	CD 0C1D	20 3A		3E 1C	DF	2A F3DC					E5	CD 0A2E	El	25	C2 257A	24	E5	2D		3A F3B0	67		20 01
	2550				ഹ	ഹ	2554	ഹ	ഹ	ഹ	ഹ	ഹ	ഹ	S				ഹ	വ	ഹ	ന	ഹ	ഹ	ഹ	ഹ	ഹ	ഹ	2572	ഹ	ഹ
6800	6801	6802	6803	6804	6805	6806	6807	6808	6809	6810	6811	6812	6813	6814	6815	6816	6817	6818	6819	6820	6821	6822	6823	6824	6825	6826	6827	6828	6829	6830

PAGE 69-1	;No, substitue by current HL	Get saved cursor position	·Cot access access access	ACC NEW CATSOF FOSTCION			;Just over strike with blank			;Get current character and attribute		;Output it to left of current position						;Do next till end of visual			;Load raw code for space	4		; End of line, all done				;Get first character next visual	
01-Jan-85 racters	(SP), HL	HL	(CCDV) HI		A, (LINLEN)	Н	Z,DELET5	Н		GETVRM	Н	PUTVRM	Н	Н	A, (LINLEN)	A	Н	NZ, DELET4	Н		с, -	PUTVRM	GETTRM	NZ, CKDPCS	HL	Г	н,1	GETVRM	(SP),HL
cro-80 3.44 01-Ja Process special characters	EX DELET1:	POP	DELET2: LD	DELET3:	LD	CP	JR	INC	DELET4:	CALL	DEC	CALL	INC	INC	ILD	INC	CP	JR	DEC	DELET5:	LD	CALL	CALL	JP	HSUG	INC	ΓD	CALL	EX
BIOS) Mac editor -	E3	El	22 F3DC	1	3A F3B0	BC	28 12	24		CD 0BD8	25	CD 0BE6	24	24	3A F3B0	3C	BC	20 F0	25		0E 20	CD 0BE6	CD 0C1D	C2 09E1	ES	2C	26 01	CD 0BD8	Е3
BASIC Screen	2578 2579	2579	257A 257A	257D	257D	2580	2581	2583	2584	2584	2587	2588	258B	258C	258D	2590	2591	2592	2594	2595	2595	2597	259A	259D	25A0	25A1	25A2	25A4	25A7
(MSX ROM - MSXINL,	6831 6832	83	6834 6835	83	6837	83	6839	84	6841	6842	6843	6844	6845	6846	6847	6848	6849	6850	6851	6852	6853	6854	6855	ഹ	ഹ	ഹ	ഹ	6860	9

PAGE	
01-Jan-85	racters
3.44	Process special characters
Macro-80	I
MSX ROM BASIC BIOS)	Screen editor
(MSX ROM	- MSXINL,

PUTVRM	HL	DELET3
CALL	POP	JR
CD 0BE6	El	18 CF
2 5 A 8	25AB	25AC
6862	6863	6864

;Put at last position last line

69-2

PAGE 70	;Set L=first visual this logical ;Is this line terminated? ;Save the condition ;Erase to end-of-line ;Restore condition ;Yes ;Go to next line ;Bump row counter ;And continue
3.44 01-Jan-85 ial characters	ERASE: Frase logical line CALL CKERCS CALL CKERCS CALL CKERCS CALL CKERCS CALL CKERCS CALL CKERCS CALL CKERCS CALL CKERCS PUSH HL CCALL CKERCS POP HL CALL CKERCS POP HL CALL CKERCS POP HL CALL CKERCS POP HL CALL CKERCS POP HL CALL CKERCS POP HL CALL CKERCS CALL CKERCS
Macro-80 3.44 01-Ja - Process special characters	LERASE: ; Erase loo ; Erase loo ; Trunc: UR ; Truncate ; Truncate ; Put Put Put Put Put Put Put Put
BIOS) Ma editor -	CD 0A2E CD 266C 22 F3DC 18 05 E1 E1 E1 E1 E1 CD 0AEE F1 CD 0AEE F1 CD 0AEE F1 CD 05 F3 CD 05 F1 S2 CD 05 F1 S2 CD 05 S2 CD 05 S2 CD 24EE F1 S3 CD 24 CD 266C CD 265C CD 265C C
M BASIC , Screen	25AE 25AE 25AE 25BJ 25BJ 25BJ 25BJ 25BJ 25CJ 25CB 25CB 25CB 25CC 25CC 25CC 25CC 25CC
(MSX ROM - MSXINL,	6865 6866 6866 6866 6879 6877 6877 68779 6887 6888 6888

l ine

PAGE 70-1					fildse cutsor . Ant mirront mirsor nosition	der carter careor fosteron		· · · · · · · · · · · · · · · · · · ·	nina ted?	;No, look at next line					;Reached start of line?		;Get a character at the cursor	; Spa ce?	;Yes, skip this		;Advance cursor to point to end of line	;Re-display cursor						;Still in word?
01-Jan-85 racters			; Append to current line		UKERCS	LL		Г	GETTRM	Z,LAPl	A, (LINLEN)	Н, А	Н		Н	Z,LAP3	GETVRM	-	Z,LAP2		ADVCUR	DPCSOW			: word	CKERCS	PRVCHK	NXTCHK
3.44 01-Ja special characters			id to cu		CALL	DEC		INC	CALL	JR	LD	LD	INC		DEC	JR	CALL	СР	JR		CALL	JR			to next word	CALL	CALL	CALL
cro-80 Process spec	LAPPND:	••	; Apper	••			LAP1:							LAP2:						LAP3:			L NXTWD:	••	; Move	•		: LWN I
r Hao						2A F3DC 2D		2C	CD 0C1D	28 FA	3A F3B0	67	24		25	28 07	CD 0BD8	FE 20	28 F6		CD 0A5B	18 D5				CD 0A2E		CD 2624
1 BASIC BIOS) Screen editor	25D7				25D7	25DA 25DD	2 5 D E	25DE	25DF	25E2	2 5 E 4	25E7	25E8	2 5E9	2 5E9	2 5 E A	25EC	2 5EF	2 5F1	25F3	25F3	25F6	25F8			25F8	25FB	25FE 25FE
(MSX ROM - MSXINL,	δ	6	6898	6899	6900	6901 6902	6903	6904	6905	6906	6907	6908	6069	6910	6911	6912	σ	σ	σ	6916	σ	6918	16	92	6921 6923	92	92	6925 6926

PAGE 70-2	;Reached screen bott ;Yes	;Reached word? ;Reached screen bott ;Not yet		;Still in separator? ;Reached screen top, ;Yes	;Reached separator? ;Reached screen top, ;Not yet	;Get current cursor ;Advance cursor ;Get an actual heigh ;[D],[E] hold the de
01-Jan-85 Iracters	Z,DPCSOW C,LNW1	NXTCHK Z, DPCSOW NC, LNW2	JR DPCSOW to previous word	CKERCS PRVCHK Z, DPCSOW NC, LBW1	PRVCHK Z,DPCSOW C,LBW2 ADVCUR DPCSOW and check	HL, (CSRY) ADVCUR GETLEN E, A A, (LINLEN)
ro-80 3.44 01-Ja Process special characters	JR JR LNW2:		JR LBCKWD: ; Move to prev ;	CALL LBW1: CALL CALL JR LRW2:	CALL JR JR CALL JR JR K: K:	LD CALL CALL CALL LD LD
BASIC BIOS) Macro-80 Screen editor - Proce	2601 28 CA 2603 38 F9 2605	CD 28 30	18	260E CD 0A2E 2611 CD 2634 2614 28 B7 2616 30 F9 2618	2618 CD 2634 2618 28 B0 261D 38 F9 261F CD 0A5B 2622 18 A9 2624 2624	2624 2A F3DC 2627 CD 0A5B 262A CD 0C32 262D 5F 262E 3A F3B0
(MSX ROM - MSXINL,	6927 6928 6929					

tom, abort:

ctom, abort

p, abort <u>(</u>•

, abort

position

pht of screen dead end position

PAGE 70-3					;Get current cursor position	;Regress cursor	;[D],[E] hold the dead end position				is regarded as separator	•	;Get updated cursor position	;Reached dead end?	;Yes, return with Z flag	;Jump to RESZRO when done		;Get ASCII code of character at [H],[L]						;Set carry if "A""Z"	ı				;Set carry if "a""z"
01-Jan-85 racters	D, A PRVCK1				HL, (CSRY)	BS	DE,0101H			t character	set if the character		HL, (CSRY)	2 0H	2	DE, RESZRO	DE	GETVRM	.0.		NC		υ	A		NC	1+1Z'	υ	'a'
0 3.44 0l-Ja ess special characters	LD JR	PRVCHK:	: • Mrvalaft and chock		LD	CALL	LD	PRVCK1:		; Check current character	; Carry set if		LD	RST	RET	LD	FUSH	CALL	CP	CCF	RET	CP	RET	CP	CCF	RET	CP	RET	CP
BIOS) Macro-80 editor - Process	57 18 09				2A F3DC	CD 0A4C	1010 11						2A F3DC	E7	C8	11 2668	D5	CD 0BD8	FE 30	3F	D0	FE 3A	D8	FE 41	3F	D0	FE 5B	D8	FE 61
BASIC BIOS) Screen edito	2631 2632	2634			2634	2637	263A	263D					263D	2640	2641	2642	2645	2646	2649	264B	264C	264D	264F	2650	2652	2653	2654	2656	2657
(MSX ROM - MSXINL,	6958 6959	6960	6961 6962	6963	6964	6965	6966	6967	6968	6969	6970	6971	6972	6973	6974	6975	6976	6977	6978	6979	6980	6981	6982	98	98	6985	6986	6987	6988

PAGE 70-4	;Check for Hiragana (86H)	;Reset Z flag without affecting C flag	<pre>first visual line in logical line L L C,GTFSTl C,GTFSTl C,GTFSTl C,Get terminator C,Get terminator C,Get first line L C,FSTPOS) C,Get first line L A,(FSTPOS) C,Get first line L N K H,l C,FSTPOS) C,Get first line C H,l K K C,FSTPOS) C,Get first line C H,l K K K K K K K K K K K K K K K K K K K</pre>
01-Jan-85 iracters	NC 'z'+1 C 86H NC 0A0H 0A0H 0A6H	A,0 A	irst visual lir L Z,GTFST GETTRM Z,GTFRST L A,(FSTPOS) L A,(FSTPOS) L H,1 NZ HL,(FSTPOS)
cro-80 3.44 01-Ja Process special characters	CCF RET CP CP CP CCF CP CCF CCF CCF CCF CCF CCF	RESZRO: LD INC RET	; Set H,L to f ; GTFRST: GTFRST: DEC JR JR CALL JR GTFST1: INC LD CP LD RET RET RET
BIOS) Mac 1 editor -	3F D0 FE 7B D8 FE 86 D0 FE A0 FE A0 3F	3E 00 3C C9	2D 28 05 28 05 28 F8 28 F8 28 F8 3A FBCA 3A FBCA BD 26 01 26 01 20 20 23 FBCA
(MSX ROM BASIC - MSXINL, Screen	6989 2659 6990 265A 6991 265B 6992 265B 6994 2660 6995 2661 6995 2661 6995 2661 6998 2665		7004 7005 7006 7007 7008 7010 7010 7010 7011 2667 7011 2667 7013 2674 7013 2674 7013 2674 7013 2674 7015 7015 2679 7016 2679 7016 2679 7017 2679 7018 2677

PAGE (MSX ROM BASIC BIOS) Macro-80 3.44 01-Jan-85 - MSXINL, Screen editor - Process special characters

7020

UN

70-5

MSX BIOS CROSS REFERENCE

		5431			5755	
EF - 1		5404			5712	
PAGE XREF		5256			5672	
		5215			5666	
	6954	1437 5168			5521 6965	5871
	6947	1387 5113 1684		5533# 5544# 5427	5500 2297 2497	5803 437 2083
_ '	6917	1296 4756 1660	6699 3485#	5520 5542 5377 5377	1733 2144# 2478	2768 404# 3275 1471
Macro-80 LISTING -	3518# 2166# 3041	1255 4725 6599 1584	2571 1914 5522#	5523# 5519 5535 5530 5282	1008# 1932 6692 2391 1430#	363# 419# 366# 1413# 365 3193 3193 3193 1159
MSX BASIC ROM BIOS) - BIOS CROSS REFERENCE	2664 1930 2892# 5236 5238	1163 4407 6476 1574	923 1690 170	5516 5494 5511 5511 5259	167 1916 6610 2087 6513 136	252 412 90 135 3055 3055 3201 1140
C ROM I	* * * * * 	* * * * 	ж 1 Ц Ц Ц 20 Ц Ц Ц Ц 8	* * * * * 	# # # # # # 	*******
(MSX BASIC ROM BIOS) - BIOS CROSS REFEREN	ACTION ADVCUR ALPJMP ASCPCT1 ASCPCT2	ATRBAS ATRBYT AUTFLG BAKCLR	BASROM BDRCLR BEEP BEGIN	BITU BITI BITIOT BITOUT BRDATR	BREAKX BS BUF BUFEND BUFEND CALATR	CALBAS CALESL CALLF CALPAT CALSLT CALSLT CAPST CGCAP1 CGCAP1 CGPBAS

4787	6892
4736 5370 2813	6856
4683 5268 2451#	6790
4589 5247 2126	6757
1525 1688 1688 1644 44 5401 4533 4533 2071	3415 6658
1521 1652 1652 6502 6502 6502 6502 4925 1820 4925 1820 5880 5880	3411 3413 6562
182 183 183 194 195 195 195 195 195 195 195 195	1850 # 2807 # 2822 # 3431 # 2051 # 2059 #
117324 17324 17324 17570 1570 1570 1199 17324 17	1842 51 156 2814 169 943 1826
**************************************	*******
CGPNT CGFNT CGSND1 CGSND1 CGCABL CHCLTX CHCBD1 CHCBD1 CHGBD1 CHGBD1 CHGBD1 CHGBD1 CHGBD1 CHGBD1 CHGBD1 CHGBD1 CHGBD1 CHGBD1 CHGBD1 CHCBD1 CHKAM CHRLP1 CHVLP1 CHPLP2 CHPLP2 CHPUT	CH PUT3 CHRGTR CHSNS1 CHSNS1 CHSNS1 CHSNS1 CKCNTC CKCNTC CKCDPC0 CKDPCS

4833

4818

4803

PAGE XREF - 2

	6938	4 989
	6923	4936
	6 9 0 0	4924
EF - 3	6879	4897
PAGE XREF	6870	4882
	6819	4858 1943
	6718	4843 1941
	6608	4672 1924
_ 1	3417 6559 737# 750 827	6696 3238 4663 1547# 1568# 1581#
Ma cr o-80 LISTING	2106# 2114# 778 778 778 751# 856# 856# 812# 812# 812#	843 851 851 851 853 853 853 853 853 853 855 1557 1557 1557 1557 1555 1555 1555
IIOS) FERENCE	953 1822 717# 732# 732# 739# 758 801# 810# 814# 821	825 835 835 846 835 846 835 846 418 418 1555 1126 1555 1126 1126 11314 1389 1403
(MSX BASIC ROM BIOS) - BIOS CROSS REFERENCE	***************************************	**************************************
(MSX BAS - BIOS	CKERCO CKERCS CKRM05 CKRM10 CKRM15 CKRM20 CKRM30 CKRM30 CKRM50 CKRM50 CKRM60 CKRM65	CLREARY OCCRRM75 CCRSTTP CLICKW CLICKW CLIKFL CLIKFL CLIKFL CLOC CLOC CLRTX1 CLRTX1 CLRTX2 CLRTX2 CLRTX2 CLRTX2 CLRTX2 CLRTX2 CLRTX3 CLRTX3 CLRTX3 CLSR2 CLSR3

PAGE XREF - 4

																						6835									
																						6813) 1)								
																						6776	•								
																						6119									
																						6493									
	4992											5848										2375									
	4846											4397								2118		2183									
	4671	2818	5782			5749#						2421		2257	2596				2201#	2110	6528	2128	6972								
	4662	2594	5774#	5784	5785	5709						1839		2221	2439	5353	5354	3515	1965	2063	2196	2073	6964	6561					5687		
1711	4634	2450	5616	5757#	5765	5681	1869#		1807	1800#	1803#	1781#	2129	2206#	2376	5321	5322	3506#	1922	2055	2004	1901	6953	2089	5459	3116	5670	5676	5685	5711	
1542# 4500	4098	2363	5585	5708	5759#	5678	1849	1912#	1797#	1795	1791	161	2076	1926	2355	5303	5304	3504	1560	2049	1827	1851	6901	2041	5455#	995	5665#	5671#	5679#	5694#	
ŧ ŧ 	⋕ = ┥┍	# : ⊶i,	1#	1#	1#	1#	1#	1#	1#	1#	1#		1#	1#	1#	1#		1#	1#	1#	1#	1#	6872	1#	1#	1#	1#	1#	1#	1#	5502#
CLSSUB	CMASN	CNSDFG	CNTFUL	CNTHL0	CNTHL1	CNTHLF	CNTPUT	CNTTBL	CNVCH1	CNVCH2	CNVCH 3	CNVCHR	CODSAV	CR	CRTCNT	CSAVEA	CSAVEM	CSDLY1	CSHOME	CSRSW	CSRX	CSRY		CSTYLE	CTWOFL	CURLIN	DATAR	DATAR0	DATARI	DATARL	DATAW

S
Т
XREF
PAGE

Ma cr o-80	- SNILSIT	5518
BIOS)	REFERENCE	5514#
BASIC ROM	CROSS	#
(MSX BA	- BIOS	DATAWI.

	6941	2356
1290	6933	2301#
1249	6931	2289
1176# 2173#	6927	2263
6832# 6852# 6782 1150 1959	6918 2098 2430 2438 4885 #	2250
5518 4146# 6830 6834 6834 6839 6814 6814 1131 1131 1131 1131 1131 1131	2185 4876 6890 6890 23998 2433 2408 24328 2428 2411 24128 2411 2411 2413 2413 2413 2413 2413 2413	2226
5514 6826 6826 68226 68226 68891 18689 1175 1175 1108 122158 108 108 108 108 108 108 108 108 108 10	2180 216 6886 6886 23958 2091 2091 2413# 2403 2403 2403 2403 2403 2403 2403 2937#	1949
**************************************	* * * * * * * * * * * * * * * * * * * *	1#
DATAWL DCOMPR DELET1 DELET2 DELET3 DELET4 DELET6 DELEN0 DELEN1 DELEN1 DELEN1 DELEN1 DELEN1 DELEN1 DELEN1 DELEN1 DISSC1 DLN DLN	DOWN1 DOWNC DPCSOW DSFKCL DSPCS1 DSPFK1 DSPFK4 DSPFK4 DSPFK6 DSPFK6 DSPFK6 DSPFK6 DSPFK6 DSPFK6 DSPFNK DWNC10 EASYTB	ELN

PAGE XREF - 6

																			4957												
																			4946												
																			4835												
																			4820												
																			4805	•											
																			4789			1664#									
	1317																		4737	5352		1580									
	1268	976					6884				1384#								4681	5318		1575								4070	
	1169#	498					2334	2342			1315						974	6701	4656#	5302		1559							2359#	3104	
	1145	476#	484#		1982#	4014	2308#	2327#	2121#	2346#	1266	2325	1984	1992	971#	960#	880	6698	4418	5073		1383		4052#	1674			3114#	1567	2387	2815
	109	61	478	2761	1928	4011#	1945	1947	2113	174	1167	2319#	1843	1939#	964	936	364	6667#	220	4979	5802	115	4071	4043	1667#		3093	3097	173	2384	23.90
1124#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	4969	1#	#	1#	1#	1#	4075#	1#	1#	1#	1#	1#
EMSITB	ENASCR	ENASLT	ENESLT	ENSTOP	ENTESC	LOCCHK	EOL	EOP	ERACSR	ERAFNK	ERASPR	EREOL1	ESCCNT	ESCTBL	EXABOL	EXCABO	EXPTBL	FAKECR	FETCHC		FILOUL	F IL VRM	FKTABL	FL PMOT	FL VRM1	FNKDEF	FNKFLG	FNKINT	FNKSB	FNKSTR	FNK SWI

														6855																		
														6829																		
EF - 7									6955					6808							6977											
PAGE XREF									6764					6741							6913											
									2590#		3428			6688							6860											
									2336		3425			6644							6842											
	4406		7018						2259		3226	4364#		2586							6725			6695								
I	1679		7014			2465#			2223		2803	4351		2562#		4176#			4161#		6618			3505								
Macro-80 LISTING	1655	4201#	6498	3230#	2476#	2282	2458#	4321#	2177	4405	1027	4291	4287#	2269		4168	4169#	4193	3547	4194#	2501#		1109	1056#	2031#	2020#	4404#	4440	4431	4432#	4441#	4448#
MSX BASIC ROM BIOS) Macro-80 - BIOS CROSS REFERENCE LISTING	1385	246	2236	3218	2464	2243	2086	4293	2007	1506#	1005	4263	3678	2231	7010	1096	250	4190#	249	4188	2075	63	1083#	146	2000	1998	4399	4416#	4422#	4428	4437	4444
LC ROM E CROSS RI	1#	1#	1#	1#	1#	1#	1#	1#]#	1#	1#	1#	1#	1#	6905	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#]#
(MSX BASIC ROM - BIOS CROSS	FORCLR	FORMAT	FSTPOS	GENCLK	GETILI	GETLLN	GET8B	GETBAK	GETLEN	GETPAT	GETPNT	GETPTR	GETQ	GETTRM		GETVC1	GETVC2	GETVCL	GETVCP	GETVCX	GETVRM	GETYPR	GICINI	GICINI	GORSET	GOSET	GPRT05	GPRT10	GPRT20	GPRT30	GPRT40	GPRT50

GPRT60 GPRT70 GPRT80	# # # H H H	4447 4462 4468	45 46 47		
	# # _	4410 4408	4443 4461	4453 4471	4459
		5.0	2 r	- r	
	1#	57	4612	4862	4901
	1#	1573			
	1#	4401	4446	4451	4456#
	1#	8	5111		
	1#	1787			
	1#	1256	1283		
	1#				
	1#	138	4389#		
] #	3365	3377#		
	1#	1422	1425#		
	1#	137	1420	1440#	
	1#	2	5232#		
	1#	6598	6871	7007	7011
	1#	7009	7012#		
	1#	186	86		
	1#	8	89		
	1#	1526#	54		
	1#	187	3807#		
	1#	3888	3891#		
	1#	2689	3698	3726#	3804
	1#	184	3683#		
	1#	185	3783#		
	1#				
	1#	81			
	1#	07			
	1#	2370			

5202

4845# 3987 3915 4837 6505 3723 1264 1722 4822 1994 1263 1245# 3476# 2251# 2009# 4065# 4526# 4841# 1987# 1995# 4807 1897 1260# 1889# 1846 1989 2005 1051 99 129 4061 5798 4199 6475 6486 2625 1703 5485 5551 4521 4811 4791 1951 1730 1759 4139 3160 2125 2350 4203 1470 6492 2993 2621 非非非非非非非非非非非非非非非非非非非非非非非非非不见了它们也们也能 ******* INIFNK H. NMI H. OUTD H. PHYD H. PINL H.QINL H.TIMI H.TOTE INDJMP INIGR1 INIGRP H.ISFL H.KEYC H.LPTO H.LPTS HRSSCL HR ZMOV HR ZMV1 INESCI INESC2 Н.КЕҮІ H.INLI HEADER H.ERAC H.ERAF H. FORM H.INIP H.KYEA INESC HIGH INGI ILN

4012

						6894	3419 2723
507 L	1466#	1720 1719	6	527	6758#	6708	3217 2720
1313 1310 1308	2 0 0 0 2 0 0 0 0 2 0	4 0 0 0	6478 6497# 6509 6546# 6549#	519 799 735	ഗ്ര്യറ്റ്	6677 6770 2288 2647	944 2672
1300# 1302# 1305#	ンキアー	128 98 1088 127	164 6481 6501# 6538 6540	6723#	77 77 54 54	6536 2258# 2279# 2638	927 2624 2645
# # # # 	* * * * ~ ~ ~		* * * * * 		1# 6766 1# 1# 1# 1#	# # # # h h h	# # # F F F
INIML1 INIML2 INIML3 TNIML3	INIPAT INIPAT INIPTI INIT	INIT32 INIT10 INITQ INITXT	INLIN INLINI INLOTO INLOTO	INLOUT INSI INSI	INS3 INS4 INS45 INS5 INS6 INSERT	INSFLG INSLN0 INSLN1 INSLN1	I NTFLG I NTRET I NTVAL

6500 2731#

	4454#			
3 4 37 5799	1905# 5218# 4412 3060		2785#	1002#
922# 4135# 1593 2653	0 4 0 0 0 3 0 3 0 0	3265# 3256 3290# 3307# 3173	0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2	3050# 3802# 962 3765#
168 247 1579# 2651	1887 5191 4398 3025	3260 1053 3267 3269 3002	3261 3263 55560 55555 2795 2719 2773 2738 2738 2738 2738 2738 2738 2738	2895 3786 251 3703
# # # # - T T T		* * * * * 	* * * * * * * * * * * * * * * * * * *	# # # #
I SCNTC I SFL I O JFLVRM	JMPBC JMPWRT JPPPAL JPUTCH	KAIUEO KANAMD KANANO KANASF KANASF	KANJNO KANJSF KEEPH KEEPL KEYANY KEYCHK KEYCHK KEYCK1 KEYCK1 KEYCK3 KEYCK3 KEYCK3 KEYCK3 KEYCK3 KEYCK3 KEYCK3	KEYSFT KEYTRG KILBUF KSTKTB

PAGE XREF - 11

3270# 2883 6655 6648# 3017 2879 3268 MSX BASIC ROM BIOS) Macro-80 - BIOS CROSS REFERENCE LISTING -3034# 3007# 3061# 2875 3090# 3360# 2992 3113 3080# 3189# 3018# 3005# 3252# 3169# 3696# 3216# 6934# 6637 3206# 6916# 6896# 3063 3120 2852 3264 6906 6915 (MSX BASIC ROM BIOS) 2914# 2924# 2908# 2867 3086 3098# 2844# 2873 3001 2859# 2903# 2898# 3107# 3150# 6903# 6910# 2999 2865 2995 2863 3262 3004 2871 2869 2861 3686 2877 3214 6912 6279 6575 6626 1# 1 1# 1# # 2899 2901 1 # 1 T т# 1 1# 1# 1# 1# 1# 1# 1# т# # 1# 1# 1# 1# 1# ### 2900 2902 2881 KY1 SFC KYl SFT KYEASY KYFNC1 KYFUNC KY JT AB KYKLOK KYLOCK KYNUM KYSTOP KYANYl KYCODI KYSTCK KYLCNT KY1 NOM KYCLTB KYCLAS KYGRAP **KYCLA0 KYFNC3** KYKANA LAPPND KYALP **KYFNC2** KYKANI KYSTPl LBCKWD LBLKSP KYCLS LAP3 LAPI LAP2

3156#

PAGE XREF - 12

		2492
EF - 13		2474
PAGE XREF		2401 6957
		2323 6907
		2150 6847
		2139 6837
I	2479 2498 2153#	2003 6827
Macro-80 LISTING -	6689 6689 6689 6684 6689 6689 6689 6689	4050 6866 1920 4347 6928 6928 6928 6928 6928 6928 6919 1979 1979
MSX BASIC ROM BIOS) - BIOS CROSS REFERENCE	6583 6682 6682 6682 6613 1457 1116 1116 1117 1116 1117 1116 1117 11117 1117	4 020 6587 205 1154 1135 6753 6753 6753 6753 6577 1955 6577 1955 5506
IC ROM I CROSS RI	***************************************	**************************************
(MSX BASIC ROM BIOS) - BIOS CROSS REFERENC	LBREAK LBREAK LBRI LBW2 LBW2 LCR1 LCR1 LCRNRM LCRNRM LCRNRM LCRNUL LCRRET LDELNX LDELNX LDIRVM LDIRVM LDIRVM LDIRVM	LERASE LF LF LFTQ LFTQ LINL32 LINL40 LINT40 LNN1 LNN2 LNN2 LNN2 LNXTLN LOW LOC LOW

			5874								4091#															
0 A A A	F / T		5867#		5841	•			5016#		4982				5025						3631					
747	r	5843# 5864#	5861	Č	م م	75	872	4540#	5010		4972				5001					23	3619					
5663 1740 1055	74	5836 5846	858	5805#	1/26# 5824	1735	5854	4413	5006	4987#	4960	4977#	4983#		65	1333		4633#	9	5221#	ഹ	8	0	5483	4706#	4955#
5638 623# 624#	25 25 34	5833 5811	5852	800	451 1751	160		219		4966	4949	4834	4975	1295	1591	1297	1293	4631	4590	5071	57	5821#	4045	5479#	4684	4804
# # # 	## 557	# # F F	1#	# = 	# # T	1# 1	1#	1#] #	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#		1#		1#	1#	1#
LOWLIM LPT.DW LPT.SB	LPT.ST LPTABO	L PTCH0 L PTCH1	LPTCHR	L PTCOD	LPTPOS	LPTSTT	MAPSPC	MAPXYC	MDNC	MHCMOV	IVMZHM	MLFTC	MLFTC1	MLTATR	MLTCGP	ML TNAM	MLTPAT	MMPXY1	MMPXYC	MNSTCX	MORACT	MORSPL	MOTRON	MOTRWT	MREADC	MRGTC

5421# 5038# 5395 3657 2557 2769# 5328 5033 3642 1553 2788 5420 5347 5417 5055# 5084 5411# 3946# 2766 6603# 5738# 5336# 4764# 4996# 4 96 7 # 4 6 4 1 # 5022# 1114# 5043# 5047# 4057# 3938# 5830# 4961# 5350# 4745# 5360 5030 1076 2657 2669 1157 2771 3027 5361 4788 5340 4899 4926 1073 1070 1070 5045 1138 2752 2752 5356# 4738 4759 124 2885# 3187# 5080# 6601 5730 227 3935 3943 2763 5819 5269 4860 4819 5341 **4**953 5371 # # # # # - - - - - -1# # # # # # h l l l l 1# 1 1 1 **1**# MUSINT MUSITB NMSFTB NOSTOP MSCNR1 MSCNR2 MTSBRD NAMBAS NONEGI NONEG2 NSETCX MUSICF NEWKEY NOTAUT NOTRAN MSCANL MSCANR MUSCLL **NOMITVM** NOTABL NSTC10 **MRGTC1** MSETC MTDNC MTLFT MTRGT MTUPC IVMTVM MSETCI MUPC NOKEY IWN

3670

PAGE XREF - 15

	2726 4954	010	1380 1380 3679
	6949# 2725 1702 4914#		
58 3394 291 3394	64 93 91 91	147 147 147 147 147 147 147 147 147 147	0010400000 0010400000000000000000000000
8 M 8 M 1 M 1 M 1 M 1 M 1 M 1 M 1 M 1 M	2642 5844 6926 1029 4797 4874	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	3920 3897 5181 5181 1523 4740 1832 3856 3856 3859 3859
	# # # # # # # 		**********
NSTC20 NSTC30 NSTC40 NSTC50 NSTC50 NTBKS2 NTBKS2 NTH1RA	NTINTT NTMSXP NXTCHK OLDKEY OLDSCR ONBRD1 ONBRDR	ONGSBF OUTDLP OUTDO OUTGI OUTTTB PADX PADX	FADA1 PADY PATWR1 PATWR7 PBDHRT PDL1 PDL2 PDL3

4976

2787

4017#

1427

-

PDLPI 1# 3840 3843#
4197#
4 / T #
5253#
5242#
-34=
1821 1830#
1.00 00/3 292 334
4118
1016 1022
696 1015
6940 6944
6967#
3178 3469
3710 3844
2494#
2284 2440
2894 3033
1026 2804
259#

COE A	7989 7089 1													5155										2713 3127#) 				1400 1444		
7783	0044													5151										2710					1376		
0229	6710													4750	5425									2707					1327		
0510#	# 7107										3909	1531		4708	5382						3975#			2704				1322	1278		
0050	1										3846	1482		4690	5310						3969			2701				1273	1237		
2131	1	# L L E V				872			5744		3712	304		4685	5285		4711		3959#		3964	2784	166	2644				1232	1219		
2105	64 R2#	4368 4368	0 0 1	3674		871		5715#	5735	299#	3481#	289#	4112#	1606#	4674#	4712	4701#		3930	3996	3925	2721	988	2634	1000#	5531#	6573	1214	1180	4808#	5126#
1854	166	1333	4324	3560	4384	296	5809	5695	5727#	291	148	49	241	111	225	4696#	4694	2767	3927	3983#	3924	1033	965	967	6975	5495	6516#	1205	1173	4796	5122
1#	: #	= =# + ~~	= = , ,	= # / /	1#] #	1#	1#	1#] #	1#	1#	1#]#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1 #	1 #
PUTVRM	OTNLIN	OSTART	OUEBAK	QUEUEN	QUEUES	RAMLOW	RAWPRT	RDBIT	RDBITL	RDESLT	RDPSG	RDSLT	RDVDP	RDVRM	READC	READC0	READC1	READYR	REDCOD	REDLOP	REDPAD	REPCNT	REQSTP	REQTRP	RESZRO	RETRET	RETURN	RG0SAV	RG1 SAV	RGHTC1	RGTEXT

PAGE XREF - 19

(MSX BASIC ROM BIOS) Ma cro-80 - BIOS CROSS REFERENCE LISTING -

																										2455					
																										1648					
	5418																									1551	544#	500#		5435	
2170	5390																									1292	486	477		5226	
2135#	5227						5279		5198#		4475#										5315#					1251	420	411		4727#	2446#
1961	4198#	2043#	4116#	3647	1976#	2293#	4023	5192#	5185		4411	5364#	5387	5388#	5391#	5397#	5261#	5293	5296#	5314	5312	6564#	4518#	4502#		1152	342	331	4714#	4425	2372
1855	212	2038	239	3645#	1969	1853	3902	5170	5176	979	218	231	5379#	5384	5381	5334	230	5284#	5287	5306#	5309	6503	4513	4497	2670	1133	301	290	224	226	2352
1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#]#	1#	1#	1#	1#	1#	1#	1#	1#	1#
RIGHT	RIGHTC	RSET10	RSLREG	RSTFLI	RSTMOD	RUBOUT	RUNFLG	SAMEBG	SAMEFG	SAVSTK	SCALXY	SCANL	SCANL1	SCANL2	SCANL3	SCANL4	SCANR	SCANRI	SCANR2	SCANR3	SCANR4	SCITBL	SCLXOK	SCLYOK	SCNCNT	SCRMOD	SELEXP	SELPRM	SETATR	SETC	SETCHK

4537

2540

			2516	3210		
			לוגנ	3151		
3F - 20			1666 1	3084		
PAGE XREF			ן ה ה ראר	3064 3064		
			1600	3036		2211
		2505	1496 1	3022	3789	2189
	1336	6895 1630#	1335# 1472	2996	3705# 917	2182#
<u>م</u> ا	1269# 1318#	6712 1610 1347# 1349#	1285 1228# 1210# 1298	2816	3687 915	2165
IOS) Macro-80 FERENCE LISTING	1267 6713 1316 1970#	6690 1454 1345 1343 1360	1244 1168 2584# 1144 1257	2764 566 513 533 3717# 3721#	2682 492 4124 887 887 2048 3704 3704	3/4/# 4056 4041# 2143
BIOS) EFERENCE	133 6554# 134 1967	6556# 113 1337 1340 1357#	1227 132 1858 131 131	2382 563# 509# 531# 3713 3716	2678 430 242 881# 2631 2639 3689#	3688 4044# 199 1866 981
X BASIC ROM B BIOS CROSS RE	# # # # h h h	* * * * * *	* * * * * *	*******	**********	* * * * *
(MSX BASIC ROM BIOS) - BIOS CROSS REFEREN	SETGRP SETINS SETMLT SETMOD	SETOVW SETRD SETREG SETRG1 SETRG2	SETTSCM SETT32 SETTRM SETTXT SETWRT	SFTKEY SLEXP1 SLPRM1 SLPRM2 SLPRM2 SLSTC1 SLSTC2	SLSTCK SLTTBL SNSMAT SSLTLP STATFL STCSSW STICKI	STATEL STMOTI STMOTIR STOCSR STOP

PAGE XREF - 21

5331

	5323				5602																					6672						
I	4665#				5591													2199							4850#	6495			5415	1696#		
DNITICIT	43	3651#	04	65	28	60	5600#		5620		5489#	5499				1242		2190#	5659#	5462#	5568#	5450#	5469#	5501#	4436	2579#	6704#	5770#	5380	973		3796#
NET ENENCE	222	4	2027	649	5577#	5583#	5597	5608#	5615#	46	5486	5493#	1162	1158	1662	1156	1160	1918	194	195	193	198	196	197	217	2314	6571	5762	81	7	2694	79
	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#
COTA	STOREC	STRTMS	STSTYL	SULOP	SVN05	0 IN XS	1 INXS	SYN20	SYN30	SYNCHR	SYNCWL	SYNLP1	T32ATR	T32CGP	T32COL	T3 2NAM	T32PAT	TAB	TAPIN	TAPIOF	TAPION	TAPOFF	TAPOON	TAPOUT	TDOWNC	TERMI N	TGL INS	TIMOUT	TLEFT	TOTEXT	TRGFLG	TRIGI

(MSX BASIC ROM B - BIOS MSX CROS	X BASIC ROM B BIOS MSX CROS	S IO	S) Macro-80 REFERENCE LISTING	- SNI				PAGE XREF	3F - 22		
TRIG2	; T	3798#	3806								
трртвг	⋕ ⋕ ⊣ ┍	4420 7633	4/81# 7643	767C	802C	046C 0706	9709 9709	0170	1015		
TRUNC	## 4 ⊢ F	6581	6874#	0074	1017	0011	1014	77 / 7	T 7 T C		
TRUNCI	1#	6873	6881#	6889							
TRYAGN	1#	3914	3921#	3940	3948						
TTYCHR	1#	5808	5875#								
TTYPOS	1#	1829									
TUPC	1#	215	4890#								
TWOPWR	1#	4595	4617#								
TXTCGP	1#	1139									
TXTNAM	1#	1137	1225								
UNTERM	# T	2582#	6760								
UP	1#	1934	1957	2159#							
UPC	1#	214	4918#								
UPC10	1#	4908	4911	4927#							
UPDATE	1#	3225	3393#	3427							
V.COLR	1#	600#									
VADDR	1#	2316	2473	2491	2504	2515	2521#				
VADDR1	1#	2543	2546#								
VADDR2	1#	2545	2550#								
VCBA	1#	3666	4184								
VCBB	1 #	3667									
VCBC	1#	3668									
VDP.CW	1#	597#	1197	1200	1623	1627	1638	1641			
VDP.DRW	1#	296	1261	1306	1458	1485	1499	1604	1613	1669	2321
	2518										
VDP.SR	1#	598#	2622	4114							
VOICAQ	1#	1080									
VOICEN	1#	4175									
VOICOF	1#	3563	3632#								
VRTMOV	1#	4889	4931#								

PAGE XREF - 23		1092 1112 3443# 3493 3496 3499 3502 3628 3640 1236 1241 1277 1282 1326 1332 1365	1595 # 4766 5110 5114 5219 3620 3622 3672 #
	5409	1050 1 3625 3 1224 1	1409 1 3587 3
) TING -	2398	346 1047 3612 1218	1397 3580
X BASIC ROM BIOS) Macro-80 BIOS MSX CROSS REFERENCE LISTING	4935# 951 5724 5393 347# 339#	329# 1044 3608 1186#	1393 4120# 5212 3613# 4516# 4509# 4509# 4500#
BIOS) SS REFER	4933 9493 53326 33256 33256 33256 3325 5281 5281 5332 33281 5281 5281 5332 5332 5332 5332 5332 5332 5332 533	53 147 3594 110 1694	1004 112 240 240 3501 4508 4508 4492 4492
(MSX BASIC ROM BIOS - BIOS MSX CROSS RE	**********	1# 3591 1379	, # # # # # # # # # #
(MSX BAS - BIOS	VRTMV1 WATINT WINWID WORK1 WORK2 WORK3 WRESED WRESET WRPRIM	WRSLT WRTPSG WRTVDP	WRTVRM WSLREG WTPTAB XEPER XGETQ XREGTV XPOSTV XVOL YNEGTV YPOSTV

