

16-BIT FIRMWARE

LOC	OBJ	LINE	SOURCE
		1 +1	\$PL(68) XR SR DR EF HNGE
		2	
		3	
		4	NAME BOOT_LOADER_LEVEL0
		5	
		6	
		7	*****
		8	*****
		9	*** **
		10	*** DMS 16 BIT EXTENSION FIRMWARE **
		11	*** **
		12	*** **
		13	*** **
		14	*** VERSION: 1.0 **
		15	*** **
		16	*** **
		17	*** **
		18	*** **
		19	*** **
		20	*** COPYRIGHT MCR CORP. 1983 **
		21	*** **
		22	*****
		23	*****
		24	
		25	
		26	
		27	
		28	DATA SEGMENT WORD COMMON 'DATA'
		29	
		30	BUFFER EQU THIS WORD
0000		31	M LABEL BYTE
		32	
		33	:
F800		34	ORG 0F800H
		35	:
F800 ??		36	CURSX DB ? ;CURSOR - X - POSITION
F801 ??		37	CURSY DB ? ;CURSOR - Y - POSITION
		38	:
		39	
		40	: *** COMMAND MAIL BLOCK OF LANGUAGE AND IO BOARD ***
		41	: *****
		42	
FE01		43	ORG 0FE01H
		44	
FE01 ??		45	SR8 DB ? ;STATUS BLOCK FROM Z80
		46	
FE04		47	ORG 0FE04H
		48	
FE04 ??		49	CB16 DB ? ;COMMAND BLOCK TO 16 BIT
FE05 ??		50	SB16 DB ? ;STATUS BLOCK RETURN OF 16 BIT
FE06 ??		51	MEM16 DB ? ;AVAILABLE MEMORY SIZE
		52	
		53	
		54	DATA ENDS
		55	
		56	
		57	
		58	\$ EJECT

BOOT LOADER LEVEL0

LOC	OBJ	LINE	SOURCE
		59	
		60	PUBLIC SWITCH,EXECUTE_SOFTWARE,SWITCH_TEST
		61	
		62	EXTRN MONITOR:NEAR,STARTS:NEAR,DISPL:NEAR
		63	EXTRN KEYINP:NEAR,L_13:NEAR,DISPLAY:NEAR
		64	
		65	
		66	
----		67	BOOTER SEGMENT WORD COMMON 'CODE'
		68	
		69	
		70	ASSUME DS:DATA,CS:BOOTER,SS:DATA,ES:DATA
		71	
		72	
		73	X&DEFINE (JUMPER)(DW MONITOR DW STARTS)
		74	
		75	; *** "OUT" EQUATES ***
		76	; *****
		77	
0010		78	RAMSEL EQU 10H ;RAMS SELECT
0011		79	ROMSEL EQU 11H ;ROM SELECT
0000		80	PROCCH EQU 000H ;PROCESSOR CHANGE
		81	
		82	;*** ASCII EQUATES ***
		83	;*****
		84	
0023		85	NEWL EQU 88H ;NEW LINE COMMAND (CR+LF)
		86	
		87	
		88	; *** JUMP TABLE ***
		89	; *****
		90	
0000 E90000	E	91	JMP DISPLAY
		92	
		93	
		94	; *** COMMAND PHASE VERTEILER ***
		95	; *****
		96	
0003 88----	R	97	BEGIN: MOV AX,DATA
0006 8ED0		98	MOV SS,AX
0008 8BE0		99	MOV SP,AX
000A 8ED8		100	MOV DS,AX
000C 8EC0		101	MOV ES,AX
000E EB00		102	JMP SHORT LOOK_MAIL_BOX
		103	
		104	
0010		105	COMMAND EQU THIS WORD
		106	XJUMPER
0014 3900	E	110	DW EXECUTE_SOFTWARE ;PHASE 2
0016 4800	R	111	DW LEVEL0 ;PHASE 3
		112	
		113	
0018 E611		114	SWITCH: OUT ROMSEL,AL
001A E600		115	OUT PROCCH,AL ;RETURN TO 8 BIT
001C 90		116	NOP
		117	
0010		118	LOOK_MAIL_BOX:
001D E610		119	OUT RAMSEL,AL ;ALLWAYS RAM SELECTION
001F A004FE	R	120	MOV AL,CB16
0022 250F00		121	AND AX,0FH

BOOT_LOADER_LEVEL0

LOC	OBJ	LINE	SOURCE	
0025	3C04	122	CMP	AL,04
0027	73EF	123	JAE	SHORT_SWITCH
0029	D1E0	124	SHL	AX,1
002B	8BF0	125	MOV	SI,AX
002D	2EFA41000	126	JMP	COMMANDS13
		127		:JUMP TO COMMAND
0032		128	SWITCH_TEST:	
0032	C60605FE00	129	MOV	SB16,0
0037	EBDF	130	JMP	SWITCH
		131		:00=GOOD STATUS
0039		132	EXECUTE_SOFTWARE:	
0039	33C0	133	XOR	AX,AX
003B	8ED8	134	MOV	DS,AX
003D	8EC0	135	MOV	ES,AX
003F	8ED0	136	MOV	SS,AX
0041	8BE0	137	MOV	SP,AX
		138		
0043	EAD000----	139	JMP	FAR PTR CPMS6
		140		:JUMP IN TO CPMS6 OPERATING SOFTWARE
		141	EJECT	

BOOT_LOADER_LEVEL0

LOC	OBJ	LINE	SOURCE
		142	
		143	
		144	
		145	
		146	*****
		147	
		148	; POWER UP ROUTINE
		149	
		150	*****
		151	
		152	
0048		153	LEVEL0:
0048 33CD		154	XOR AX,AX ;SET AX AND START ADDRESS TO 0
004A 88FD		155	MOV ST,AX
004C 890D10		156	MOV CX,1000H ;ROM-TEST OVER 4K
004F 2EAC		157	SUM1: LODS CS:M
0051 02E0		158	ADD AH,AL
0053 E2FA		159	LOOP SUM1
0055 80C4D0		160	ADD AH,D
0058 7407		161	JZ SHORT RAMTST ;ROM-TEST O.K.
		162	
005A		163	LEVEL0_FAIL:
005A C60605FEFF	R	164	MOV SB16,OFFH
005F EBB7		165	JMP SWITCH ;LEVEL 0 FAILED
		166	
		167	
		168	; *** RAM-TEST ***
		169	; *****
		170	
0061		171	RAMTST:
0061 C60601F817	R	172	MOV CURSY,2D
0066 C60600F800	R	173	MOV CURSX,D
0068 3D1EE0D1	R	174	LEA BX,MESS1 ;MESSAGE MEMORY TEST OF BANK 0
006F E80000	E	175	CALL DISPL
0072 3D1E1102	R	176	LEA BX,MESS4
0076 E80000	E	177	CALL DISPL
		178	
		179	
0079 89A0A		180	MOV CX,2000H/3 ;WRITE A UNEVEN BYTE MUSTER IN MEMORY
007C 33FF		181	XOR DI,DI
007E		182	RAMTST1A:
007E B8AA55		183	MOV AX,55AAH
0081 AB		184	STOSW
0082 B000		185	MOV AL,D0
0084 AA		186	STOSB
0085 E2F7		187	LOOP RAMTST1A
0087 B8AA55		188	MOV AX,55AAH
008A AB		189	STOSW
		190	
008B 89A0A		191	MOV CX,2000H/3
008E 33FF		192	XOR DI,DI
0090		193	RAMTST1B:
0090 B8AA55		194	MOV AX,55AAH
0093 AF		195	SCASW
0094 7542		196	JNZ ERROR
0096 B000		197	MOV AL,D
0098 AE		198	SCASB
0099 753D		199	JNZ ERROR
009B E2F3		200	LOOP RAMTST1B
009D B8AA55		201	MOV AX,55AAH
00A0 AF		202	SCASW
00A1 7535		203	JNZ ERROR
		204	

BOOT_LOADER_LEVEL0

LOC	OBJ	LINE	SOURCE
		323	
0168	86E0	324	XCHG AH,AL
016D	FECF	325	DEC BH
016F	75EC	326	JNZ MEM_TEST
		327	
0171		328	GOOD_BANK:
0171	FEC3	329	INC BL ;GOOD BANK INC BANK INDICATOR
0173	8BC0	330	MOV CX,BP
0175	E2B1	331	LOOP NEXT_BANK ;ALL BANKS ?
0177	EB18	332	JMP SHORT MEMORY_END
		333	
0179	33C0	334	ERROR1: XOR AX,AX
017B	8ED8	335	MOV DS,AX
017D	8EC0	336	MOV ES,AX ;SET SEGMENT TO ZERO
017F	53	337	PUSH BX
0180	C60600F81E	338	MOV CURSX,30
0185	801E0502	339	LEA BX,MESS3 ;ERROR DETECTED WRITE FAILED TO SCREEN
0189	E80000	340	CALL DISPL
018C	E80000	341	ERR11: CALL KEYINP
018F	3C88	342	CMP AL,NEWL
0191	75F9	343	JNZ ERR11
0193	5B	344	POP BX
		345	
0194		346	MEMORY_END:
0194	33C0	347	XOR AX,AX
0196	8ED8	348	MOV DS,AX
0198	8EC0	349	MOV ES,AX ;SET SEGMENT TO ZERO
019A	C60601F80A	350	MOV CURSY,10
019F	881E04FE	351	MOV MEM16,BL ;SET COUNT OF MEM BANKS IN MEM16
01A3	C60605FE00	352	MOV SB16,0
01A8	E960FE	353	JMP SWITCH ;JUMP TO 8 BIT
		354	
01AB		355	BANK_DIS:
01AB	50	356	PUSH AX
01AC	33C0	357	XOR AX,AX
01AE	8ED8	358	MOV DS,AX
01B0	8EC0	359	MOV ES,AX ;SET SEGMENT TO ZERO BANK
01B2	B005	360	MOV AL,5
01B4	F6E3	361	MUL BL
01B6	53	362	PUSH BX
01B7	51	363	PUSH CX
01B8	50	364	PUSH AX
01B9	C60601F817	365	MOV CURSY,23 ;SET CURSOR FOR MESSAGES
01BE	C60600F800	366	MOV CURSX,0
01C3	801EED01	367	LEA BX,MESS1 ;WRITE MEMORY TEST ON SCREEN
01C7	E80000	368	CALL DISPL
01CA	5B	369	POP AX
		370	
01CB	8B08	371	MOV BX,AX
01CD	80061A02	372	LEA AX,BANKS
01D1	0308	373	ADD BX,AX
01D3	B509	374	MOV CH,9
01D5	E80100	375	CALL L_13+1 ;WRITE THE TESTED BANK TO SCREEN
01D8	59	376	POP CY
01D9	5B	377	POP BX
01DA	5B	378	POP AX
01DB	8EDA	379	MOV DS,DX
01DD	8EC2	380	MOV ES,DX
01DF	C3	381	RET
		382	
		383	
01ED	11	384	MESS1 DB 17,'TEST MEMORY BANK '
01E1	54455354204045		

BOOT_LOADER_LEVEL0

LOC	OBJ	LINE	SOURCE
	404F5259204241		
	4E4B20		
01F2	12	385	MESS2 DB 18, 'FATAL MEMORY ERROR'
01F3	464154414C2040		
	45404F52592045		
	52524F52		
0205	08	386	MESS3 DB 11, 'FAILED (CR)'
0206	4641494C454420		
	28435229		
0211	08	387	MESS4 DB 8, ' 0-64K'
0212	20202030203634		
	48		
021A	20363448203132	388	BANKS DB ' 64K-128K-192K-256K-320K-384K-448K-512K'
	38482031393248		
	20323536482033		
	32304820333834		
	48203434384820		
	35313248		
		389	
		390	
		391	
----		392	BOOTER ENDS
		393	
		394	
----		395	SOFTWARE SEGMENT PARA PUBLIC 'CODE'
		396	
0000		397	CPM86 EQU THIS FAR
		398	
----		399	SOFTWARE ENDS
		400	
		401	
		402	END BEGIN

BOOT LOADER LEVEL0

LOC	OBJ	LINE	SOURCE	
00A3	BA0010	205	MOV	DX,1000H
00A6	B80201	206	MOV	AX,0102H
00A9		207	RAMTST1:	
00A9	33FF	208	XOR	DI,DI
00AB	88CA	209	MOV	CX,DX
00AD	F3	210	REP	STOSW
00AE	AB			
00AF	33FF	211	XOR	DI,DI
00B1	88CA	212	MOV	CX,DX
00B3	F3	213	REPZ	SCASW
00B4	AF			
00B5	7521	214	JNZ	ERROR0
00B7	D0C0	215	ROL	AL,1
00B9	D0C4	216	ROL	AH,1
00BB	73EC	217	JNC	RAMTST1
		218		
00BD	B702	219	MOV	BH,2
00BF	B855AA	220	MOV	AX,0AA55H
00C2		221	RAMTST2:	
00C2	88CA	222	MOV	CX,DX
00C4	33FF	223	XOR	DI,DI
00C6	F3	224	REP	STOSW
00C7	AB			
		225		
00C8	33FF	226	XOR	DI,DI
00CA	88CA	227	MOV	CX,DX
00CC	F3	228	REPZ	SCASW
00CD	AF			
		229		
00CE	7508	230	JNZ	ERROR0
		231		
00D0	86E0	232	XCHG	AH,AL
00D2	FECF	233	DEC	BH
00D4	75EC	234	JNZ	RAMTST2
00D6	EB0E	235	JMP	SHORT MEM_BANK_TEST
		236		
		237		
00D8	C60600F81E	238	ERROR0: MOV	CURSOR,30
00D0	8D1EF201	239	LEA	RX,MESS2
00E1	E80000	240	CALL	DISPL
00E4	EBFE	241	JMP	SHORT \$
		242		
		243		
		244	:	MEMORY EXTENSION TEST
		245	:	=====
		246		
		247		
00E6		248	MEM_BANK_TEST:	
00E6	B90300	249	MOV	CX,3
00E9	803E01FE09	250	CMP	SB8,09
00EE	7203	251	JB	UNDER
00F0	83C104	252	ADD	CX,4
00F3	BA0000	253	UNDER: MOV	DX,0
00F6	B300	254	MOV	BL,0
		255		
00F8		256	NEXT_BANK:	
00F8	8BE9	257	MOV	BP,CX
00FA	81C20010	258	ADD	DX,1000H
00FE	BEDA	259	MOV	DS,DX
0100	8EC2	260	MOV	ES,DX
		261		
0102	B855AA	262	MOV	AX,0AA55H
0105	A30000	263	MOV	BUFFER,AX

;TEST WITH A ROTATING ONE

;STORE AA55

;WRITE 'FATAL MEMORY ERROR'
;AND STOP PROCESSING;SET BANK COUNTER TO MAX BANKS=256K (512K)
;VERSION NUMBER) 10 ?

;SET BANK COUNTER TILL 512K

;SET BANK INDICATOR

;SAVE BANK COUNTER
;SET SEGMENT FOR NEXT BANK

;TEST THE FIRST TWO BYTES OF A BANK

BOOT_LOADER_LEVEL0

LOC	OBJ	LINE	SOURCE	
0108	3B060000	R 264	CMP AX,BUFFER	;FOR A PRESENT BANK
010C	7403	265	JZ L_200	
010E	E98300	266	JMP MEMORY_END	
0111	E89700	267	L_200: CALL BANK_DIS	
		268		
0114	B95555	269	MOV CX,0FFFFH/3	;WRITE AN UNEVEN BYTE PATTERN IN MEMORY
0117	33FF	270	XOR DI,DI	
		271		
0119		272	ADDTSTA:	
0119	B8AA55	273	MOV AX,55AAH	
011C	AB	274	STOSW	
011D	8000	275	MOV AL,00	
011F	AA	276	STOSB	
0120	E2F7	277	LOOP ADDTSTA	
0122	B055	278	MOV AL,55H	
0124	AA	279	STOSB	
		280		
0125	B95555	281	MOV CX,0FFFFH/3	
0128	33FF	282	XOR DI,DI	
012A		283	ADDTSTB:	
012A	B8AA55	284	MOV AX,55AAH	
012D	AF	285	SCASW	
012E	7549	286	JNZ ERROR1	
0130	8000	287	MOV AL,0	
0132	AE	288	SCASB	
0133	7544	289	JNZ ERROR1	
0135	E2F3	290	LOOP ADDTSTB	
0137	B055	291	MOV AL,55H	
0139	AE	292	SCASB	
013A	753D	293	JNZ ERROR1	
		294		
013C	33FF	295	XOR DI,DI	
013E	BE0080	296	MOV SI,8000H	;COUNTER OF A ENTIRE BANK 64 K
		297		
0141	B80201	298	MOV AX,0102H	
0144		299	ROT_ONE:	
0144	33FF	300	XOR DI,DI	
0146	8BCE	301	MOV CX,SI	
0148	F3	302	REP STOSW	
0149	AB			
014A	33FF	303	XOR DI,DI	
014C	8BCE	304	MOV CX,SI	
014E	F3	305	REPZ SCASW	
014F	AF			
0150	7527	306	JNZ ERROR1	
0152	D0C0	307	ROL AL,1	
0154	D0C4	308	ROL AH,1	
0156	73EC	309	JMC ROT_ONE	
		310		
0158	8702	311	MOV BH,2	;COUNT OF BIT PATTERN
015A	B855AA	312	MOV AX,0AA55H	
015D		313	MEM_TEST:	
015D	8BCE	314	MOV CX,SI	;SET COUNTER FOR A ENTIRE BANK = 3000 WORDS
015F	33FF	315	XOR DI,DI	
0161	F3	316	REP STOSW	;STORE AX INTO MEMORY
0162	AB			
		317		
0163	33FF	318	XOR DI,DI	
0165	8BCE	319	MOV CX,SI	
0167	F3	320	REPZ SCASW	;CHECK MEMORY WITH ACCU
0168	AF			
		321		
0169	750E	322	JNZ ERROR1	

BOOT_LOADER_LEVEL0

XREF SYMBOL TABLE LISTING

NAME	TYPE	VALUE	ATTRIBUTES, XREFS
??SEG	SEGMENT		SIZE=0000H PARA PUBLIC
ADDTSTA	L NEAR	0119H	BOOTER 272# 277
ADDTSTB	L NEAR	012AH	BOOTER 283# 290
BANK_DIS	L NEAR	01ABH	BOOTER 267 355#
BANKS	V BYTE	021AH	BOOTER 372 388#
BEGIN	L NEAR	0003H	BOOTER 97# 402 402
BOOTER	SEGMENT		SIZE=0241H WORD COMMON 'CODE' 67# 70 392
BUFFER	V WORD	0000H	DATA 30# 263 264
CB16	V BYTE	FED4H	DATA 49# 120
COMMAND	V WORD	0010H	BOOTER 105# 126
CPMB6	L FAR	0000H	SOFTWARE 139 397#
CURSX	V BYTE	F800H	DATA 36# 173 238 338 366
CURSY	V BYTE	F801H	DATA 37# 172 350 365
DATA	SEGMENT		SIZE=FED7H WORD COMMON 'DATA' 28# 54 70 70 97
DISPL	L NEAR	0000H	EXTRN 62# 175 177 240 340 368
DISPLAY	L NEAR	0000H	EXTRN 63# 91
ERR11	L NEAR	018CH	BOOTER 341# 343
ERROR0	L NEAR	0008H	BOOTER 196 199 203 214 230 238#
ERROR1	L NEAR	0179H	BOOTER 286 289 293 306 322 334#
EXECUTE_SOFTWARE	L NEAR	0039H	BOOTER PUBLIC 60 110 132#
GOOD_BANK	L NEAR	0171H	BOOTER 320#
KEYIMP	L NEAR	0000H	EXTRN 63# 341
L_13	L NEAR	0000H	EXTRN 63# 375
L_200	L NEAR	0111H	BOOTER 265 267#
LEVEL0	L NEAR	0048H	BOOTER 111 153#
LEVEL0_FAIL	L NEAR	005AH	BOOTER 163#
LOOK_MAIL_BOX	L NEAR	0010H	BOOTER 102 118#
M	V BYTE	0000H	DATA 31# 157
MEM_BANK_TEST	L NEAR	00E6H	BOOTER 235 248#
MEM_TEST	L NEAR	0150H	BOOTER 313# 326
MEM16	V BYTE	FED6H	DATA 51# 351
MEMORY_END	L NEAR	0194H	BOOTER 266 332 346#
MESS1	V BYTE	01E0H	BOOTER 174 367 384#
MESS2	V BYTE	01F2H	BOOTER 239 385#
MESS3	V BYTE	0205H	BOOTER 339 386#
MESS4	V BYTE	0211H	BOOTER 176 387#
MONITOR	L NEAR	0000H	EXTRN 62# 107
NEUL	NUMBER	0088H	85# 342
NEXT_BANK	L NEAR	00F8H	BOOTER 256# 331
PROCCH	NUMBER	0000H	80# 115
RANSEL	NUMBER	0010H	78# 119
RANTST	L NEAR	0061H	BOOTER 161 171#
RANTST1	L NEAR	00A9H	BOOTER 207# 217
RANTST1A	L NEAR	007EH	BOOTER 182# 187
RANTST1B	L NEAR	0090H	BOOTER 193# 200
RANTST2	L NEAR	00C2H	BOOTER 221# 234
RANSEL	NUMBER	0011H	79# 114
ROT_ONE	L NEAR	0144H	BOOTER 299# 309
SB16	V BYTE	FED5H	DATA 50# 129 164 352
SB8	V BYTE	FED1H	DATA 45# 250
SOFTWARE	SEGMENT		SIZE=0000H PARA PUBLIC 'CODE' 395# 399
STARTS	L NEAR	0000H	EXTRN 62# 108
SUM1	L NEAR	004FH	BOOTER 157# 159
SWITCH	L NEAR	0018H	BOOTER PUBLIC 60 114# 123 130 165 353
SWITCH_TEST	L NEAR	0032H	BOOTER PUBLIC 60 128#
UNDER	L NEAR	00F3H	BOOTER 251 253#

END OF SYMBOL TABLE LISTING

BOOTER_IO_DRIVER

SERIES-III 8086/8087/8088 MACRO ASSEMBLER V1.1 ASSEMBLY OF MODULE BOOTER_IO_DRIVER
 OBJECT MODULE PLACED IN :F1:DM1610.OBJ
 ASSEMBLER INVOKED BY: A86.86 :F1:DM1610.SRC

LOC	OBJ	LINE	SOURCE
		1 +1	\$PAGELENGTH(68) XREF SYMBOLS DEBUG EP NOMR
		2	
		3	
		4	NAME BOOTER_IO_DRIVER
		5	
		6	
		7	*****
		8	*****
		9	***
		10	*** DMS 16 BIT EXTENSION FIRMWARE **
		11	***
		12	*** IDENTIFIER: AMF-DW16FI 0101-00 **
		13	***
		14	*** VERSION: 1.0 **
		15	***
		16	*** DATE: 24.02.83 **
		17	***
		18	*** ORIG. PLANT: MCR AUGSBURG **
		19	***
		20	*** COPYRIGHT MCR CORP. 1983 **
		21	***
		22	*****
		23	*****
		24	
		25	
		26	
		27	
		28	DATA SEGMENT WORD COMMON 'DATA'
		29	
0000		30	BUFFER EQU THIS WORD
0000		31	M LABEL BYTE
		32	
F800		33	ORG 0F800H
		34	
F800 ??		35	CURSX DB ? ;CURSOR - X - POSITION
F801 ??		36	CURSY DB ? ;CURSOR - Y - POSITION
		37	
F830		38	ORG 0F830H
		39	
F830 ??		40	BUFSTA DB ? ;CRT-BUFFER START ADR.
		41	
F900		42	ORG 0F900H
		43	
F900 ??		44	CEADL DB ? ;LOW BYTE OF CURSOR
F901 ??		45	CEADH DB ? ;HIGH BYTE OF CURSOR
F902 ??		46	CDAD DB ? ;POS OF PIXEL
F903 ????		47	SP1 DW ? ;START OF PAGE 1
F905 ??		48	LP11 DB ? ;LENGTH OF PAGE 1 LOW
F906 ??		49	LP12 DB ? ; " " " 1 HIGH
F907 ????		50	SP2 DW ? ;START OF PAGE 2
F909 ??		51	LP21 DB ? ;LENGTH OF PAGE 2 LOW
F90A ??		52	LP22 DB ? ; " " " 2 HIGH
F90B ??		53	INWFL6 DB ?
		54	
----		55	DATA ENDS
		56	
		57 +1	\$ EJECT

BOOTER_IO_DRIVER

LOC	OBJ	LINE	SOURCE
		58	
		59	
		60	
		61	PUBLIC KEYINP,KYLOOK,NODCRT,SETCU,CALCHR,DISPL,L_13,DISPL1
		62	PUBLIC DISPLAY
----		63	
		64	BOOTER SEGMENT BYTE COMMON 'CODE'
		65	
		66	
		67	ASSUME DS:DATA,CS:BOOTER,SS:DATA,ES:DATA
0250		68	
		69	ORG 250H
		70	
		71	
		72	; *** "OUT" EQUATES ***
		73	; *****
		74	
0010		75	RANSEL EQU 10H ;RAMS SELECT
0011		76	ROMSEL EQU 11H ;ROM SELECT
		77	
		78	
		79	
		80	;*****
		81	;
		82	; DRIVER FOR KEYBOARD-CONTROLLER 8741
		83	;
		84	;*****
		85	
		86	
		87	; *** "IN/OUT" EQUATES ***
		88	; *****
		89	
0040		90	KEYDAT EQU 40H ;READ KEYBOARD DATA
0041		91	KEYSTA EQU 41H ;READ " STATUS
		92	
		93	
0250 E441		94	KEYINP: IN AL,KEYSTA
0252 D0D8		95	RCR AL,1
0254 73FA		96	JNC KEYINP
0256 E440		97	IN AL,KEYDAT ;READ KEYBOARD-CHARACTER
0258 C3		98	RET
		99	
0259 E441		100	KYLOOK: IN AL,KEYSTA
025B 2401		101	AND AL,1
025D 7501		102	JNZ SHORT L_1
025F C3		103	RET
0260 E440		104	L_1: IN AL,KEYDAT
0262 C3		105	RET
		106	
		107	
		108	;*****
		109	;
		110	; CRT-ROUTINES
		111	;
		112	;*****
		113	
		114	
		115	
		116	
		117	; *** CONTROL CHARACTER EQUATES ***
		118	; *****
		119	
0000		120	NOMOV EQU 00H ;NO CURSOR MOVE

BOOTER_IO_DRIVER

LOC	OBJ	LINE	SOURCE
002A		121	PROMPT EQU 2AH
0020		122	SPACE EQU 20H
0088		123	MEWL EQU 88H
0001		124	HOME EQU 01H
000C		125	SCRER EQU 0CH
0088		126	RUBOUT EQU 8BH
		127	
		128	
		129	
		130	;*****
		131	;
		132	; GDC-EQUATES
		133	;
		134	;*****
		135	
		136	
0050		137	CHARA EQU 80 ;80 CHARACTER/LINE
		138	
		139	
		140	; *** GDC PORT EQUATES ***
		141	; *****
		142	
00A1		143	GDCOM EQU 0A1H ;WRITE COMMAND
00A0		144	GDCSTA EQU 0A0H ;READ STATUS
00A0		145	GDCPAR EQU 0A0H
		146	
		147	; *** GDC STATUS EQUATES ***
		148	; *****
		149	
0002		150	FIFULL EQU 02H ;FIFO IS FULL
		151	
		152	; *** GDC COMMAND EQUATES ***
		153	; *****
		154	
00A8		155	START EQU 68H ;START DISPLAY/END IDLE-MODE
0049		156	CURS EQU 49H ;SPECIFY CURSOR-POSITION
004A		157	MASK EQU 4AH ;LOAD MASK-REGISTER
004C		158	FIGS EQU 4CH ;SPECIFY FIGURE DRAWING-PARAMETER
0070		159	PRAM EQU 70H ;LOAD PARAMETER-RAM
0020		160	WDAT EQU 20H ;WRITE DATA INTO DISPLAY MEMORY
		161	; (COMMAND+TYPE+MODE)
0000		162	TYWORD EQU 00H ;DATA TRANSFER BYTES
		163	;MODE OF R/W MEMORY CYCLE
0000		164	MOREPL EQU 00H ;REPLACE WITH PATTERN
		165	
		166	
		167	
		168	
J 55		169	MODCRT: PUSH BP
4 56		170	PUSH SI
5 57		171	PUSH DI
6 53		172	PUSH BX
7 51		173	PUSH CX
8 52		174	PUSH DX
9 9C		175	PUSHF
A 50		176	PUSH AX ;SAVE ASCII-CHARACTER
		177	
		178	; *** CHECK IF ANY CONTROL CHARACTER ***
		179	; *****
		180	
3 A8F0		181	TEST AL,0F0H
1 7438		182	JZ SHORT SETCHA
1 3C00		183	CMP AL,MOV

BOOTER_IO_DRIVER

LOC	OBJ	LINE	SOURCE	
0271	7457	184	JE	GETRE ;NO CURSOR MOVE
0273	3C88	185	CHP	AL,NEUL
0275	740A	186	JE	SHORT CARRET ;CARRIDGE RETURN
0277	3C01	187	CHP	AL,HOME
0279	7420	188	JE	SHORT CHP ;HOME POSITION
027B	3C0C	189	CHP	AL,SCRER
027D	7423	190	JE	SHORT SCERA ;SCREEN ERASE
027F	EB26	191	JMP	SHORT SETCHA ;SET CHARACTER INTO CRT-BUFFER DEPENDING
		192		
0281	C60600F800	R 193	CARRET: MOV	CURSX,0
0284	A001F8	R 194	MOV	AL,CURSY
0289	FEC0	195	INC	AL
028B	3C19	196	CHP	AL,25
028D	7507	197	JNE	CAR1
028F	50	198	PUSH	AX
0290	E88700	199	CALL	SCROLL
0293	58	200	POP	AX
0294	FEC8	201	DEC	AL
0296	A201F8	R 202	CAR1: MOV	CURSY,AL
0299	EB2F	203	JMP	SHORT GETRE
		204		
		205	; *** CURSOR HOME POSITION ***	
		206	; *****	
		207		
029B	33C0	208	CHP: XOR	AX,AX
029D	A300F8	R 209	MOV	WORD PTR CURSX,AX
02A0	EB28	210	JMP	SHORT GETRE
		211		
		212	; *** SCREEN ERASE ***	
		213	; *	
02A2	E88800	214	SCERA: CALL	HCLSCR
02A5	EBF4	215	JMP	CHP
		216		
		217	; *** SET CHARACTER ***	
		218	; *	
02A7	A100F8	R 219	SETCHA: MOV	AX,WORD PTR CURSX
02AA	3C50	220	CHP	AL,80
02AC	7504	221	JNZ	SHORT CHA1
02AE	32C0	222	XOR	AL,AL
02B0	FEC4	223	INC	AH
02B2	80FC19	224	CHA1: CHP	AH,25
02B5	7507	225	JNZ	SHORT CHA2
02B7	50	226	PUSH	AX
02B8	E88F00	227	CALL	SCROLL
02BB	58	228	POP	AX
02BC	FECC	229	DEC	AH
02BE	80C8	230	CHA2: MOV	CX,AX
02C0	FEC0	231	INC	AL
02C2	A300F8	R 232	MOV	WORD PTR CURSX,AX
02C5	58	233	POP	AX
02C6	50	234	PUSH	AX
02C7	E80900	235	CALL	WRNCHR
		236		
02CA	58	237	GETRE: POP	AX
02CB	90	238	POPF	
02CC	5A	239	POP	DX
02CD	59	240	POP	CX
02CE	58	241	POP	BX
02CF	5F	242	POP	DI
02D0	5E	243	POP	SI
02D1	50	244	POP	BP
02D2	C3	245	RET	
		246		

BOOTER_IO_DRIVER

LOC	OBJ	LINE	SOURCE
		247	
		248	*****
		249	;
		250	; G D C DRIVER ROUTINES
		251	;
		252	*****
		253	
		254	
		255	;ENTRY: <AL> = ASCII CHARACTER
		256	; <CH> = Y-POSITION
		257	; <CL> = X-POSITION
		258	
0203	53	259	WRMCHR: PUSH BX
0204	52	260	PUSH DX
0205	51	261	PUSH CX
0206	50	262	PUSH AX
		263	
0207	E82C00	264	CALL CALCHR ;CALCULATE CHARACTER-POS.
020A	E8FC00	265	CALL SETCU
020D	E81300	266	CALL SETCCA ;SEND CHARACTER
02E0	B020	267	MOV AL,WDAT+TYWORD+MOREPL
02E2	E82E01	268	CALL OUTCHD
02E3	58	269	POP AX ;<A>=ASCII-CHARACTER
02E6	E83001	270	CALL OUTPAR
02E9	A00BF9	271	MOV AL,INWFLG ;FETCH ATTRIBUT
02EC	E82A01	272	CALL OUTPAR
02EF	59	273	POP CX
02F0	5A	274	POP DX
02F1	5B	275	POP BX
02F2	C3	276	RET
		277	
		278	
02F3	E8C400	279	SETCCA: CALL SETMSK
02F6	B04C	280	MOV AL,FIGS
02F8	E81801	281	CALL OUTCHD
02F8	B002	282	MOV AL,2
02FD	E81901	283	CALL OUTPAR
0300	32C0	284	XOR AL,AL
0302	E81401	285	CALL OUTPAR
0305	C3	286	RET
		287	
0306		288	CALCHR:
0306	F70603F9FFFF	289	TEST SP1,0FFFFH
030C	750C	290	JNZ SHORT CALCH1
030E	B050	291	MOV AL,B0 ;Y*B0 + X = <CEADL>
0310	F6E5	292	MUL CH ;<CH>= Y-POS
0312	B500	293	CALCH2: MOV CH,0
0314	03C1	294	ADD AX,CX
0316	A300F9	295	MOV WORD PTR CEADL,AX
0319	C3	296	RET
		297	
031A		298	CALCH1:
031A	B050	299	MOV AL,B0 ;<CH>=Y-POS
031C	F6E5	300	MUL CH ;Y * B0
031E	030603F9	301	ADD AX,SP1 ;<SP1> + <Y*B0>=AX
0322	BA0007	302	MOV DX,25*B0
0325	3BC2	303	CMR AX,DX
0327	72E9	304	JB CALCH2
0329	2BC2	305	SUB AX,DX
032B	EBE5	306	JMP CALCH2
		307	
		308	
		309	

BOOTER_IO_DRIVER

LOC	OBJ	LINE	SOURCE
		310	; **** CLEAR SCREEN ****
		311	; *****
		312	
		313	
0320	E86F00	314	NCLSCR: CALL SETCR ;SET CURSOR IN HOME-POS.
0330	E80700	315	CALL CLRL ;CLEAR SCREEN
0333	E85800	316	CALL INIT1A
0336	E88800	317	CALL SCROL1 ;SET PARTITION
0339	C3	318	RET
		319	
		320	
033A	B020	321	CLRL: MOV AL,WDAT+TYWORD+MOREPL
033C	E8D400	322	CALL OUTCMD
033F	B020	323	MOV AL,' ' ;SPACE-WRITE
0341	E8D500	324	CALL OUTPAR
0344	32C0	325	XOR AL,AL
0346	E8D000	326	CALL OUTPAR ;CLEAR ATTRIB.
0349	C3	327	RET
		328	
		329	
		330	
		331	; **** SCROLL SCREEN ****
		332	; *****
		333	
		334	
034A	881E03F9	R 335	SCROLL: MOV BX,SP1
034E	891E0DF9	R 336	MOV WORD PTR CEADL,BX
0352	E88400	337	CALL SETCU ;SET CURSOR
0355	E86200	338	CALL SETMSK ;SET WORD-TRANSFER
		339	
0358	B04C	340	MOV AL,FIGS ;SET LENGTH (80-CHAR)
035A	E88600	341	CALL OUTCMD ;FOR DELETE 1 LINE
035D	B002	342	MOV AL,2
035F	E88700	343	CALL OUTPAR
0362	B04F	344	MOV AL,80-1
0364	E88200	345	CALL OUTPAR
0367	32C0	346	XOR AL,AL
0369	E8AD00	347	CALL OUTPAR
		348	
036C	E8C8FF	349	CALL CLRL ;CLEAR LAST LINE
036F	881E0DF9	R 350	MOV BX,WORD PTR CEADL ;SET CURSOR
0373	E86300	351	CALL SETCU
		352	
0376	830603F950	R 353	ADD SP1,80 ;POINTER + 80-CHAR.
		354	
0378	A006F9	R 355	MOV AL,LP12
037E	FEC8	356	DEC AL
0380	7503	357	JNZ SHORT L_31 ;LENGTH1 - 1
0382	EB7D90	358	JMP INIT10
0385		359	L_31:
0385	A206F9	R 360	MOV LP12,AL
		361	
0388	FED60AF9	R 362	INC LP22 ;LENGTH2 + 1
038C	EB63	363	JMP SHORT SCROL1
		364	
		365	
		366	
		367	
		368	
038E	33C0	369	INIT1A: XOR AX,AX
0390	A303F9	R 370	MOV SP1,AX
0393	A307F9	R 371	MOV SP2,AX
0396	A20AF9	R 372	MOV LP22,AL

BOOTER_ID_DRIVER

LOC	OBJ	LINE	SOURCE	
0399	C60606F919	R 373	MOV	LP12,25
039E	C3	374	RET	
		375		
039F	E82600	376	SETCR:	CALL CUHOM1 ;WHOLE GRAPHIC HOME
03A2	E81500	377	CALL	SETMSK
03A5	B04C	378	MOV	AL,FIGS ;SET DIRECTION AND NUMBER AND WORDS
03A7	E86900	379	CALL	OUTCMD
03AA	B002	380	MOV	AL,2 ;DIR = 2
03AC	E86A00	381	CALL	OUTPAR
03AF	B0FF	382	MOV	AL,OFFH ;LENGTH 16K WORD
03B1	E86500	383	CALL	OUTPAR
03B4	B03F	384	MOV	AL,3FH ;HIGH BYTE
03B6	E86D00	385	CALL	OUTPAR
03B9	C3	386	RET	
		387		
03BA	B04A	388	SETMSK:	MOV AL,MASK_ ;SET MASK TO STORED VALUES
03BC	E85400	389	CALL	OUTCMD
03BF	B0FF	390	MOV	AL,OFFH ;LOAD LOW BYTE
03C1	E85500	391	CALL	OUTPAR
03C4	E85200	392	CALL	OUTPAR
03C7	C3	393	RET	
		394		
03C8	B049	395	CUHOM1:	MOV AL,CURS
03CA	E84600	396	CALL	OUTCMD ;CURSOR HOME
03CD	32CD	397	XOR	AL,AL
03CF	E84700	398	CALL	OUTPAR
03D2	E84400	399	CALL	OUTPAR
03D5	E84100	400	CALL	OUTPAR
03D8	C3	401	RET	
		402		
03D9	B049	403	SETCU:	MOV AL,CURS
03DB	E83500	404	CALL	OUTCMD ;SET CURSOR
03DE	A00DF9	R 405	MOV	AL,CEADL ;LOW BYTE OF CURSOR
03E1	E83500	406	CALL	OUTPAR
03E4	A001F9	R 407	MOV	AL,CEADH ;HIGH BYTE OF CURSOR
03E7	E82F00	408	CALL	OUTPAR
03EA	A002F9	R 409	MOV	AL,CDAD ;POSITION OF PIXEL
03ED	E82900	410	CALL	OUTPAR
03F0	C3	411	RET	
		412		
03F1	B070	413	SCROL1:	MOV AL,PRAM+0
03F3	E81D00	414	CALL	OUTCMD ;SET START AND LENGTH OF
03F6	B90800	415	MOV	CX,8 ;PAGE 1 AND PAGE 2
03F9	B01E03F9	R 416	LEA	BX,SP1
03FD	E82A00	417	CALL	SEWPAR
0400	C3	418	RET	
		419		
0401	33CD	420	INI10:	XOR AX,AX
0403	A303F9	R 421	MOV	SP1,AX
0406	A307F9	R 422	MOV	SP2,AX ;START OF PAGE 1 = 0
0409	A20AF9	R 423	MOV	LP22,AL ;LENGTH OF PAGE 2 = 0
040C	C60606F919	R 424	MOV	LP12,25 ;LENGTH OF PAGE 1 = 25
0411	EBDE	425	JMP	SCROL1
		426		
		427		
		428	; *** SUBROUTINES OF GDC ***	
		429	; *****	
		430		
0413	E80900	431	OUTCMD:	CALL FIFRDY ;PUT A COMMAND IN FIFO
0416	E6A1	432	OUT	GDCCON,AL
0418	C3	433	RET	
		434		
0419	E80300	435	OUTPAR:	CALL FIFRDY ;PUT A PARAMETER TO FIFO

BOOTER_IO_DRIVER

LOC	OBJ	LINE	SOURCE
041C	E6A0	436	OUT GDCPAR.AL
041E	C3	437	RET
		438	
041F	9C	439	FIFRDY: PUSHF
0420	50	440	PUSH AX
0421	E4A0	441	FIFO10: IN AL,GDCSTA ;READ GDC-STATUS
0423	24D2	442	AND AL,FIFULL
0425	75FA	443	JNZ FIFO10 ;JUMP IF FIFO FULL
0427	58	444	POP AX
0428	90	445	POPF
0429	C3	446	RET
		447	
		448	
042A		449	SEMPAR:
042A	8A07	450	MOV AL,[BX] ;LOAD PARAMETER
042C	E8EAF	451	CALL OUTPAR ;OUTPUT PARAMETER
042F	43	452	INC BX
0430	E2F8	453	LOOP SEMPAR ;LOOP UNTIL END OF PARAMETER-LIST
0432	C3	454	RET
		455	
		456	
		457	*****
		458	;
		459	;
		460	;
		461	*****
		462	
		463	
		464	;
		465	*** DISPLAY ROUTINE ***
		466	*****
0433	2E8A2F	467	DISPL: MOV CH,CS:[BX]
0436	43	468	L_13: INC BX
0437	2E8A07	469	MOV AL,CS:[BX]
043A	E826FE	470	CALL MODCRT
043D	FEC0	471	DEC CH
043F	75F5	472	JNZ L_13
0441	43	473	INC BX
0442	C3	474	RET
		475	
0443	8A07	476	DISPL1: MOV AL,[BX]
0445	E81BFE	477	CALL MODCRT
0448	43	478	INC BX
0449	E2F8	479	LOOP DISPL1
044B	C3	480	RET
		481	
044C		482	DISPLAY PROC FAR
		483	
044C	8A0F	484	MOV CL,[BX]
044E	32ED	485	XOR CH,CH
0450	43	486	DISP_N: INC BX
0451	8A07	487	MOV AL,[BX]
0453	E800FE	488	CALL MODCRT
0456	E2F8	489	LOOP DISP_N
0458	C8	490	RET
		491	
		492	DISPLAY ENDP
		493	

BOOTER_IO_DRIVER

XREF SYMBOL TABLE LISTING

NAME	TYPE	VALUE	ATTRIBUTES, XREFS
??SEG	SEGMENT		SIZE=0000H PARA PUBLIC
BOOTER.	SEGMENT		SIZE=0459H BYTE COMMON 'CODE' 64# 67 494
BUFFER.	V WORD	0000H	DATA 30#
BUFFSTA.	V BYTE	F830H	DATA 40#
CALCH1.	L NEAR	031AH	BOOTER 290 298#
CALCH2.	L NEAR	0312H	BOOTER 293# 304 306
CALCHR.	L NEAR	0306H	BOOTER PUBLIC 61 264 288#
CAR1.	L NEAR	0296H	BOOTER 197 202#
CARRET.	L NEAR	0281H	BOOTER 186 193#
CDAD.	V BYTE	F902H	DATA 46# 409
CEADH.	V BYTE	F901H	DATA 45# 407
CEADL.	V BYTE	F900H	DATA 44# 295 336 350 405
CHA1.	L NEAR	0282H	BOOTER 221 224#
CHA2.	L NEAR	028EH	BOOTER 225 230#
CHARA.	NUMBER	0050H	137#
CHP.	L NEAR	029BH	BOOTER 188 208# 215
CLRL.	L NEAR	033AH	BOOTER 315 321# 349
CUHOM1.	L NEAR	03C8H	BOOTER 376 395#
CURS.	NUMBER	0049H	156# 395 403
CURXS.	V BYTE	F800H	DATA 35# 193 209 219 232
CURSY.	V BYTE	F801H	DATA 36# 194 202
DATA.	SEGMENT		SIZE=F90CH WORD COMMON 'DATA' 28# 55 67 67 67
DISP_M.	L NEAR	0450H	BOOTER 486# 489
DISPL.	L NEAR	0433H	BOOTER PUBLIC 61 467#
DISPL1.	L NEAR	0443H	BOOTER PUBLIC 61 476# 479
DISPLAY.	L FAR	044CH	BOOTER PUBLIC 62 482# 492
FIFO10.	L NEAR	0421H	BOOTER 441# 443
FIFRDY.	L NEAR	041FH	BOOTER 431 435 439#
FIFULL.	NUMBER	0002H	150# 442
FIGS.	NUMBER	004CH	158# 280 340 378
GDCCOM.	NUMBER	00A1H	143# 432
GDCPAR.	NUMBER	00A0H	145# 436
GDCSTA.	NUMBER	00A0H	144# 441
GETRE.	L NEAR	02CAH	BOOTER 184 203 210 237#
HONE.	NUMBER	0001H	124# 187
INIT10.	L NEAR	0401H	BOOTER 358 420#
INIT1A.	L NEAR	038EH	BOOTER 316 369#
INVLFG.	V BYTE	F90BH	DATA 53# 271
KEYDAT.	NUMBER	0040H	90# 97 104
KEYIMP.	L NEAR	0250H	BOOTER PUBLIC 61 94# 96
KEYSTA.	NUMBER	0041H	91# 94 100
KYLOOK.	L NEAR	0259H	BOOTER PUBLIC 61 100#
L_1.	L NEAR	0240H	BOOTER 102 104#
L_13.	L NEAR	0436H	BOOTER PUBLIC 61 468# 472
L_31.	L NEAR	0385H	BOOTER 357 359#
LP11.	V BYTE	F905H	DATA 48#
LP12.	V BYTE	F906H	DATA 49# 355 360 373 424
LP21.	V BYTE	F909H	DATA 51#
LP22.	V BYTE	F90AH	DATA 52# 362 372 423
M.	V BYTE	0000H	DATA 31#
MASK.	NUMBER	004AH	157# 383
MCLSCR.	L NEAR	032DH	BOOTER 214 314#
MODCRT.	L NEAR	0263H	BOOTER PUBLIC 61 169# 470 477 488
MOREPL.	NUMBER	0000H	164# 267 321
NEWL.	NUMBER	0088H	123# 185
MONOV.	NUMBER	0000H	120# 183
OUTCM9.	L NEAR	0413H	BOOTER 268 281 322 341 379 389 396 404 414 431#
OUTPAR.	L NEAR	0419H	BOOTER 270 272 283 285 324 326 343 345 347 381 383 385 391 392 398 399 400 406 408 410 435#

BOOTER_IO_DRIVER

NAME	TYPE	VALUE	ATTRIBUTES, XREFS
PRAH. .	NUMBER	0070H	159# 413
PROMPT. .	NUMBER	002AH	121#
RANSEL. .	NUMBER	0010H	75#
RONSEL. .	NUMBER	0011H	76#
RUBOUT. .	NUMBER	0080H	126#
SCERA. .	L HEAR	02A2H	BOOTER 190 214#
SCREER. .	NUMBER	000CH	125# 189
SCROL1. .	L HEAR	03F1H	BOOTER 317 363 413# 425
SCROLN. .	L HEAR	034AH	BOOTER 199 227 335#
SEMPAR. .	L HEAR	042AH	BOOTER 417 449# 453
SETCCA. .	L HEAR	02F3H	BOOTER 266 279#
SETCHA. .	L HEAR	02A7H	BOOTER 182 191 219#
SETCQ. .	L HEAR	039FH	BOOTER 314 376#
SETCU. .	L HEAR	0309H	BOOTER PUBLIC 61 265 337 351 403#
SETNSK. .	L HEAR	03BAH	BOOTER 279 338 377 388#
SP1. .	V WORD	F903H	DATA 47# 289 301 335 353 370 416 421
SP2. .	V WORD	F907H	DATA 50# 371 422
SPACE. .	NUMBER	0020H	122#
START. .	NUMBER	0060H	155#
TYWORD. .	NUMBER	0000H	162# 267 321
WDAT. .	NUMBER	0020H	160# 267 321
WRNCHR. .	L HEAR	0203H	BOOTER 235 259#

END OF SYMBOL TABLE LISTING

