



5. MONITOR-ROM

ORG 0000

cassette	EQU 04F1
cas.init	EQU 0605
prscreen	EQU 0D00
getDOS	EQU 0E90
keybuf	EQU 6000
keytel	EQU 600C
lastkey	EQU 600D
Kytime	EQU 600E
STATKEY	EQU 600F
klok	EQU 6010
autorep,	EQU 6012
T-M	EQU 6013
i-/etype	EQU 6014
baudrate	EQU 6016
intr/INT4	EQU 6026
transfer	EQU 6030
fileleng	EQU 6032
recleng	EQU 6034
type	EQU 6041
strt-boot	EQU 6043
load	EQU 6045
sptemp	EQU 6055
memsize	EQU 605C

status key's

repeat key

type

T,D

vector CTC

tape

B,P etc.

DI		0000 F3	
JP start		0001 C3 5A 02	DI
NOP		0004 00	JP 025A
JP CPM-start		0005 C3 4F 04	NOP
JP prscreen	print scherm	0008 C3 00 0D	JP 044F
JP inpdisk		000B C3 67 04	JP 0D00
			JP 0467
WORD error	system error	000E E4 01	
JP 1013	RST debug	0010 C3 13 10	JP 1013
JP writedisk		0013 C3 7A 04	JP 047A
WORD error	system error	0016 E4 01	
JP cassette		0018 C3 F1 04	JP 04F1
JP initkey		001B C3 67 01	JP 0167

WORD getDOS	bootstrap DOS	001E 90 0E	
JP enkey	enable key	0020 C3 B4 00	JP 00B4
JP diskey	disable key	0023 C3 BC 00	JP 00BC
JP readkey	uit buffer	0026 C3 9B 04	JP 049B
JP stat.key	status key	0029 C3 89 04	JP 0489
JP clearkey	veeg buffer	002C C3 DC 04	JP 04DC
JP error		002F C3 E1 01	JP 01E1
JP piep		0032 C3 EA 01	JP 01EA
JP wisregels		0035 C3 23 04	JP 0423

; Via de interrupt wordt het keyboard
; bekeken, eventueel een toets in de
; buffer gezet (6000) en de klok
; bijgewerkt. Scanfrequentie toetsen-
; bord 20 msec, repeatdelay 1 sec,
; repeat frequentie 40 msec.

keyscan	CALL save.reg	IM 1	0038 CD A5 01	CALL 01A5
	LD HL,(klok)	update klok	003B 2A 10 60	LD HL,(6010)
	INC HL		003E 23	INC HL
	LD (klok),HL		003F 22 10 60	LD (6010),HL
	IN 00		0042 DB 00	IN 00
	CP FF	keypress?	0044 FE FF	CP FF
	LD A,00		0046 3E 00	LD A,00
	OUT 10	key-scan	0048 D3 10	OUT 10
	JR Z nokey	mode	004A 28 2D	JR Z 0079
	CALL toets	welke toets	004C CD FA 00	CALL 00FA
	LD A,B		004F 78	LD A,B
	CP FF	sh or lock?	0050 FE FF	CP FF
	JR Z bufvol		0052 28 4D	JR Z 00A1
	LD A,(lastkey)	zelfde als	0054 3A 0D 60	LD A,(600D)
	CP B	vorige?	0057 B8	CP B
	JR NZ vulbuf		0058 20 44	JR NZ 009E
	LD HL,kytime	keytime tellr	005A 21 0E 60	LD HL,600E
	DEC (HL)	timing auto-	005D 35	DEC (HL)
	JR Z klaar	repeat klaar?	005E 28 02	JR Z 0062
JF JR keynormal			0060 18 48	JR 00AA
klaar 0066 PUSH HL			0062 E5	PUSH HL
X JP skip	skip NMI		0063 C3 69 00	JP 0069

; hardware bepaald is de Non-Maskable
; Interrupt op 0066, hier zijn we nu

JP 1016	NMI int vect.	0066 C3 16 10	JP 1016
---------	---------------	---------------	---------

1016 adres in debug module ??

skip	LD HL,auto SET 0,(HL) POP HL CALL buf.key LD A,02 LD (kytime),A JR keynormal	auto repeat- flag aan. toets in buf i.v.m. CTC ?	0069 21 12 60 006C CB C6 006E E1 006F CD D0 00 0072 3E 02 0074 32 0E 60 0077 18 31	LD HL,6012 SET 0,(HL) POP HL CALL 00DO LD A,02 LD (600E),A JR 00AA
; als de repeat flag aanstaat, reset hem ; dan en veeg de toets buffer leeg, (er ; was nog geen toets ingedrukt).				
nokey	PUSH AF LD A,(auto) BIT 0,A JR Z nokl RES 0,A LD (auto),A XOR A LD (keytel),A POP AF LD A,FF LD (lastkey),A LD A,(STATKEY)	auto aan? zet auto uit veeg toets- buffer leeg.	0079 F5 007A 3A 12 60 007D CB 47 007F 28 09 0081 CB 87 0083 32 12 60 0086 AF 0087 32 0C 60 008A F1 008B 3E FF 008D 32 0D 60 0090 3A 0F 60	PUSH AF LD A,(6012) BIT 0,A JR Z 008A RES 0,A LD (6012),A XOR A LD (600C),A POP AF LD A,FF LD (600D),A LD A,(600F)
nokl	;			
; als lock niet aanstaat, dan:.....				
	BIT 2,A JR NZ nolock	lock on?	0093 CB 57 0095 20 05	BIT 2,A JR NZ 009C
; reset shift bit!!				
	AND FE LD (STATKEY),A JR keynormal		0097 E6 FE 0099 32 0F 60 009C 18 0C	AND FE LD (600F),A JR 00AA
vulbuf	CALL buf.key		009E CD D0 00	CALL 00DO
bufvol	LD A,32 LD (kytime),A LD A,B LD (lastkey),A LD A,40 OUT 10 CALL herstel EI RETI	voor CTC opbergen enable key registers vanwege IM 2	00A1 3E 32 00A3 32 0E 60 00A6 78 00A7 32 0D 60 00AA 3E 40 00AC D3 10 00AE CD AB 01 00B1 FB 00B2 ED 4D	LD A,32 LD (600E),A LD A,B LD (600D),A LD A,40 OUT 10 CALL 01AB EI RETI
; enable keyboard				
enkey	CALL save.reg CALL CTCenable JR keynormal	opbergen	00B4 CD A5 01 00B7 CD C5 00 00BA 18 EE	CALL 01A5 CALL 00C5 JR 00AA

diskey	LD A,00 OUT 10	keybord op scan mode	00BC 3E 00 00BE D3 10 00C0 3E 01 00C2 D3 8B 00C4 C9	LD A,00 OUT 10 LD A,01 OUT 8B RET
CTCoff	LD A,01 OUT 8B RET	disable CTC		
CTCenable	LD A,D5 OUT 8B	activate Z80 timer-	00C5 3E D5 00C7 D3 8B	LD A,D5 OUT 8B
; timer 20 msec				
	LD A,01 OUT 8B	controller	00C9 3E 01 00CB D3 8B 00CD FB 00CE ED 4D	LD A,01 OUT 8B EI RETI
CTC1	EI RETI	verwijder interr. CTC		
buf.key	LD A,B	toets in buf	00D0 78	LD A,B
; B bevat nieuwe toets				
	CP 58	stop?	00D1 FE 58	CP 58
	JR NZ buf1		00D3 20 07	JR NZ 00DC
; stop is ingedrukt en wordt vooraan				
; in de toets buffer gezet. Tevens wordt				
; de toetsteller op 1 gezet.				
	LD (keybuf),A	stop in buf	00D5 32 00 60	LD (6000),A
	LD A,01	Een toets	00D8 3E 01	LD A,01
	JR buf4		00DA 18 17	JR 00F3
buf1	CP 12	dubbel nul	00DC FE 12	CP 12
	JR NZ buf2		00DE 20 04	JR NZ 00E4
; zet twee maal dubbel 0 in buffer				
	LD B,A	2 maal in buf	00E0 47	LD B,A
	CALL buf2		00E1 CD E4 00	CALL 00E4
buf2	LD A,(keytel)		00E4 3A 0C 60	LD A,(600C)
	LD HL, keybuf		00E7 21 00 60	LD HL,6000
	LD L,A		00EA 6F	LD L,A
	CP 0C	buffer vol?	00EB FE 0C	CP 0C
	JR Z buf5		00ED 28 08	JR Z 00F7
buf3	LD A,B		00EF 78	LD A,B
	LD (HL),A		00F0 77	LD (HL),A
	INC HL		00F1 23	INC HL
	LD A,L	buffer teller	00F2 7D	LD A,L
buf4	LD (keytel),A	ophogen	00F3 32 0C 60	LD (600C),A
	RET		00F6 C9	RET
buf5	DEC HL	buffer vol!	00F7 2B	DEC HL
	JR buf3		00F8 18 F5	JR 00EF

toets	LD B,FF LD HL,STATKEY	00FA 06 FF 00FC 21 OF 60	LD B,FF LD HL,600F	
; STATKEY : 600F	-	-	-	
;	BIT 0 : shift	-	-	
;	BIT 2 : shift lock	-	-	
toel	LD C,00 IN A,(C) LD E,A XOR FF CALL NZ kolom INC C LD A,C CP 09 JR NZ toe1 IN A,(C) CP FF JR Z toe2 SET 0,(HL) RES 2,(HL) PUSH BC LD B,00 LD E,01 CALL scr-type POP BC RET	C: rij nummer matrix in E niets? 0106 C4 29 01 volgende rij laatste rij ? next row 010F ED 78 shift? 0113 28 0E shift aan sh-lock uit 0119 C5 011A 06 00 011C 1E 01 shift]scherm 0121 C1 0122 C9 sh lock on? ja shift off 0128 C9	00FF 0E 00 0101 ED 78 0103 5F 0104 EE FF 0109 OC 010A 79 010B FE 09 010D 20 F2 0111 FE FF 0115 CB C6 0117 CB 96 011E CD BE 01 0123 CB 56 0125 C0 0126 CB 86 0129 F5 012A 3E 03 012C B9 012D 20 19 012F CB 43 0131 20 15 0133 C5 0134 E5 0135 06 4C 0137 1E 01 0139 CD BE 01 013C E1 013D C1 013E 3A 0F 60 0141 F6 05 0143 32 0F 60 0146 F1 0147 C9	LD C,00 IN A,(C) LD E,A XOR FF CALL NZ 0129 INC C LD A,C CP 09 JR NZ 0101 IN A,(C) CP FF JR Z 0123 SET 0,(HL) RES 2,(HL) PUSH BC LD B,00 LD E,01 CALL 01BE POP BC RET BIT 2,(HL) RET NZ RES 0,(HL) RET
toe2	BIT 2,(HL) RET NZ RES 0,(HL) RET	-	-	
kolom	PUSH AF LD A,03 CP C JR NZ kol2 BIT 0,E JR NZ kol2 PUSH BC PUSH HL LD B,"L LD E,01 CALL scr-type POP HL POP BC LD A,(STATKEY) OR 05	rij met shift nee shift? 0129 F5 012A 3E 03 012C B9 012D 20 19 012F CB 43 0131 20 15 0133 C5 0134 E5 0135 06 4C 0137 1E 01 0139 CD BE 01 013C E1 013D C1 013E 3A 0F 60 0141 F6 05	PUSH AF LD A,03 CP C JR NZ 0148 BIT 0,E JR NZ 0148 PUSH BC PUSH HL LD B,4C LD E,01 CALL 01BE POP HL POP BC LD A,(600F) OR 05	
set shift en lock in STATKEY	LD (STATKEY),A POP AF RET	0143 32 0F 60 0146 F1 0147 C9	LD (600F),A POP AF RET	

kol2	POP AF	geen lock	0148 F1	POP AF
	LD E,00	kolom counter	0149 1E 00	LD E,00
kol3	RRA	toets in rij?	014B 1F	RRA
	JR C kol4	ja	014C 38 03	JR C 0151
	INC E		014E 1C	INC E
	JR kol3	volgende	014F 18 FA	JR 014B
; E bevat nu kolom positie				
kol4	LD A,C		0151 79	LD A,C
	RLCA	naar bit po-	0152 07	RLCA
	RLCA	sitie 3,4,5	0153 07	RLCA
	RLCA		0154 07	RLCA
	OR E	erbij 0,1,2	0155 B3	OR E
	LD E,A		0156 5F	LD E,A
	LD A,(STATKEY)		0157 3A 0F 60	LD A,(600F)
	BIT 0,A	shift on?	015A CB 47	BIT 0,A
	LD A,E		015C 7B	LD A,E
	JR Z kol5		015D 28 03	JR Z 0162
	LD A,48	shift erbij	015F 3E 48	LD A,48
	ADD E		0161 83	ADD E
kol5	LD B,A	real code	0162 47	LD B,A
	CALL special		0163 CD D2 01	CALL 01D2
	RET		0166 C9	RET
initkey				
	CALL CTCinit		0167 CD B2 01	CALL 01B2
	LD HL,CTC-test	vector voor	016A 21 88 01	LD HL,0188
	LD (intr),HL	CTC 6026	016D 22 26 60	LD (6026),HL
	LD A,00	test CTC	0170 3E 00	LD A,00
	OUT 10	key op scan	0172 D3 10	OUT 10
	LD A,32		0174 3E 32	LD A,32
	LD (kytime),A		0176 32 0E 60	LD (600E),A
	LD A,85	enable CTC	0179 3E 85	LD A,85
	OUT 8B	klok CTC on	017B D3 8B	OUT 8B
	LD A,01	korte tijds-	017D 3E 01	LD A,01
	OUT 8B	constante	017F D3 8B	OUT 8B
	CALL CTC1	veeg interr.	0181 CD CD 00	CALL 00CD
	IM 1	vector int 38	0184 ED 56	IM 1
	JR inil		0186 18 0D	JR 0195
; test: is er een CTC aanwezig?				
; we komen hier als er via de CTC een				
; interrupt gegenereerd is. Hiertoe				
; is hierboven de klok van de CTC aan-				
; gezet en geladen met de kortst				
; mogelijke waarde. Eerst wordt het				
; return adres van de interrupt van				
; het stack gehaald.				
CTC-test	POP HL	veeg adres	0188 E1	POP HL
	CALL CTCoFF		0189 CD C0 00	CALL 00CO

; zet de interrupt vector op de normale
; keyscan routine zoals in IM 1.

	LD HL, keyscan	018C 21 38 00	LD HL,0038
	LD (intr),HL	018F 22 26 60	LD (6026),HL
	CALL CTCenable	0192 CD C5 00	CALL 00C5
inil	LD A,40	0195 3E 40	LD A,40
	OUT 10	0197 D3 10	OUT 10
x?	LD HL, keybuf	0199 21 00 60	LD HL,6000
x	LD A,L	019C 7D	LD A,L
	(LD (keytel),A	019D 32 0C 60	LD (600C),A
	XOR A	01A0 AF	XOR A
	LD (STATKEY),A	01A1 32 0F 60	LD (600F),A
	sh, shlock	01A4 C9	RET
	off		

HL,DE,BC,AF bewaren op stack

save.reg	EX (SP),HL	ook op \$05FC	01A5 E3	EX (SP),HL
	PUSH DE		01A6 D5	PUSH DE
	PUSH BC		01A7 C5	PUSH BC
	PUSH AF		01A8 F5	PUSH AF
	DI		01A9 F3	DI
	JP (HL)		01AA E9	JP (HL)

AF,BC,DE en HL van stack halen

herstel	POP HL	registers.	01AB E1	POP HL
	POP AF	ook op \$0600	01AC F1	POP AF
	POP BC		01AD C1	POP BC
	POP DE		01AE D1	POP DE
	EI		01AF FB	EI
	EX (SP),HL		01B0 E3	EX (SP),HL
	RET		01B1 C9	RET

zet de interrupt vector voor kanaal
88 van de CTC via adres 6020

CTCinit	IM 2		01B2 ED 5E	IM 2
	LD HL,6020	interrupt	01B4 21 20 60	LD HL,6020
	LD A,L	vector	01B7 7D	LD A,L
	OUT 88	L-byte] CTC	01B8 D3 88	OUT 88
	LD A,H		01BA 7C	LD A,H
	LD I,A	H-byte]I reg	01BB ED 47	LD I,A
	RET		01BD C9	RET

; laat type functie op scherm zien.
 ; dit gebeurt bijv als een bootstrap-
 ; tape geladen wordt (zonder ROM module)
 ; er verschijnt dan een T (van Tape)
 ; op het scherm.
 ; B: karakter
 ; E: verplaatsing in i/o veld

scr-type	PUSH AF	01BE F5	PUSH AF
	PUSH HL	01BF E5	PUSH HL
	PUSH DE	01C0 D5	PUSH DE
	LD HL,(i/otype)	01C1 2A 14 60	LD HL,(6014)
	LD D,00	E rel i/otyp	LD D,00
	ADD HL,DE	01C4 16 00	ADD HL,DE
	LD (HL),B	naar scherm	LD (HL),B
x {	LD DE,0800	01C6 19	LD DE,0800
	ADD HL,DE	01C7 70	ADD HL,DE
	LD (HL),F5	01C8 11 00 08	LD (HL),F5
	POP DE	01CB 19	POP DE
	POP HL	01CC 36 F5	POP HL
	POP AF	01CE D1	POP AF
	RET	01CF E1	RET
		01D0 F1	
		01D1 C9	
special	LD A,(lastkey)		
	SUB 48	zelfde zonder	LD A,(600D)
	CP B	shift	SUB 48
	RET NZ		CP B
x {	LD A,B	nieuwe toets	RET NZ
	LD (lastkey),A	opbergen.	LD A,B
	RET		LD (600D),A
			RET
; system error, met een maintenance			
; module kan via een NMI achterhaald			
; worden welk type fout er was (in A).			
err0	LD A,35		
	BYTE 01		
error	LD A,31	01DE 3E <u>35</u>	LD A,35
	LD BC,303E	01E0 01 3E <u>31</u>	LD BC,313E
err1	DI	01E3 01 3E <u>30</u>	LD BC,303E
	HALT	01E6 F3	DI
	JR err1	01E7 76	HALT
		01E8 18 FC	JR 01E6

piep	CALL save.reg		01EA CD A5 01	CALL 01A5
	LD E,80	aantal konus	01ED 1E 80	LD E,80
	LD A,01	bewegingen	01EF 3E 01	LD A,01
	OUT 50	konus]	01F1 D3 50	OUT 50
	CALL 60-micro	delay	01F3 CD 05 02	CALL 0205
	XOR A		01F6 AF	XOR A
	OUT 50	konus	01F7 D3 50	OUT 50
	CALL 60-micro	delay	01F9 CD 05 02	CALL 0205
	DEC E		01FC 1D	DEC E
X[LD A,E		01FD 7B	LD A,E
	OR A	klaar ?	01FE B7	OR A
	JR NZ piel		01FF 20 EE	JR NZ 01EF
	CALL herstel.reg		0201 CD AB 01	CALL 01AB
	RET		0204 C9	RET
60-micro	LD B,80	delay	0205 06 80	LD B,80
60ml	DJNZ 60ml	sec.	0207 10 FE	DJNZ 0207
	RET		0209 C9	RET
; verplaats string, bestemming DE				
; huidig adres HL				
; lengte.in (HL)				
; ik heb geen referentie aan deze				
; routine kunnen vinden.				
\$020A	X LD A,(HL)	dummy string	020A 7E	LD A,(HL)
	LD B,00	replace	020B 06 00	LD B,00
(LD C,A (HL)		020D 4F	LD C,A
	INC HL		020E 23	INC HL
	LDIR		020F ED B0	LDIR
	RET		0211 C9	RET
ERROR	ASC CALL SERVICE			
	WORD 51EC	scherm adres	0210 B0 C9 43 41 4C 4C 20 53	OICALL S
	BYTE 0F	lengte	0218 45 52 56 49 43 45 EC 51	ERVICE1Q
	BYTE 06	cyaan	0220 0F 06 0D 50 20 48 20 49	P H I
	BYTE 0D	dubbel hoog	0228 20 4C 20 49 20 50 20 53	L I P S
	ASC P H I L I P S		0230 DC 52 0F 06 0D 4D 49 43	®R MIC
	WORD 52DC	scherm adres	0238 52 4F 43 4F 4D 50 55 54	ROCOMPUT
	BYTE 0F	lengte	0240 45 52 D0 53 07 06 0D 50	ERPS P
	BYTE 06	cyaan	0248 32 30 30 30 FF 21 12 02	2000 !
	BYTE 0D	dubbel hoog		
	ASC MICROCOMPUTER			
	WORD 53D0	scherm adres		
	BYTE 07	lengte		
	BYTE 06	cyaan		
	BYTE 0D	dubbel hoog		
	ASC P2000			
	BYTE FF	einde tekst		

ram-error	LD HL,ERROR	adres	tekst	024D 21 12 02	LD HL,0212
	LD DE,5012	video		0250 11 12 50	LD DE,5012
	LD BC,000C	lengte		0253 01 0C 00	LD BC,000C
	LDIR			0256 ED B0	LDIR
	JR err1			0258 18 8C	JR 01E6
start	LD A,01	CTC off		025A 3E 01	LD A,01
	OUT 88	flop interrupt		025C D3 88	OUT 88
	OUT 89	flop not read		025E D3 89	OUT 89
	OUT 8A			0260 D3 8A	OUT 8A
	OUT 8B	toetsenbord		0262 D3 8B	OUT 8B
	LD A,00	keybord scan		0264 3E 00	LD A,00
	OUT 10			0266 D3 10	OUT 10
	LD A,(1000)	maintenance		0268 3A 00 10	LD A,(1000)
	CP 58	module?		026B FE 58	CP 58
	JP Z 1010	direct start		026D CA 10 10	JP Z 1010
; video is er altijd, daarom stack hier					
	LD SP,57FF	in video		0270 31 FF 57	LD SP,57FF
	CALL piep			0273 CD EA 01	CALL 01EA
	LD HL,keybuf	(start ram)		0276 21 00 60	LD HL,6000
	LD BC, <u>40FF</u>	16-k 8 bit		0279 01 FF 40	LD BC,40FF
	CALL ramtest	6000-9FFF		027C CD DC 03	CALL 03DC
	OR A			027F B7	OR A
	JR NZ ram-error			0280 20 CB	JR NZ 024D
; stack mag nu in systeem RAM					
	LD SP,6200			0282 31 00 62	LD SP,6200
	LD HL,memsize			0285 21 5C 60	LD HL,605C
	INC (HL)	16-k aanwezig		0288 34	INC (HL)
; (memsize) staat nu op 1					
	LD HL,5000			0289 21 00 50	LD HL,5000
	LD BC, <u>08FF</u>	2-k 8 bit		028C 01 FF 08	LD BC,08FF
	CALL ramtest	video-1		028F CD DC 03	CALL 03DC
	OR A			0292 B7	OR A
str1	JR NZ ram-error	video error		0293 20 B8	JR NZ 024D
	LD HL,5800			0295 21 00 58	LD HL,5800
	LD BC, <u>080F</u>	2-k 4 bit		0298 01 0F 08	LD BC,080F
	CALL ramtest	video-2		029B CD DC 03	CALL 03DC
; 02: video pagina twee is wel aanwezig					
; maar toch een fout gevonden.					
; 01: geen video pagina twee (T-model)					
	CP 02			029E FE 02	CP 02
	JR Z ram-error	.		02A0 28 AB	JR Z 024D
	CP 01			02A2 FE 01	CP 01
	JR Z sta2			02A4 28 17	JR Z 02BD
	LD A,05	extra test		02A6 3E 05	LD A,05
	LD (5800),A	video-2		02A8 32 00 58	LD (5800),A
	LD A,(5800)			02AB 3A 00 58	LD A,(5800)
	AND OF			02AE E6 0F	AND OF
	CP 05	toch niet		02B0 FE 05	CP 05
	JR NZ sta2	aanwezig		02B2 20 09	JR NZ 02BD

	LD HL,T-M	type	02B4 21 13 60	LD HL,6013
	SET 0,(HL)		02B7 CB C6	SET 0,(HL)
	XOR A		02B9 AF	XOR A
sta2	LD (5800),A'		02BA 32 00 58	LD (5800),A
	LD HL,A000		02BD 21 00 A0	LD HL,A000
	LD BC, <u>40FF</u>	16-k 8 bit	02C0 01 FF 40	LD BC,40FF
	CALL ramtest	A000-DFFF	02C3 CD DC 03	CALL 03DC
sta3	CP 02	00-test	02C6 FE 02	CP 02
	JR Z ram-error	fout	02C8 28 83	JR Z 024D
	CP 01		02CA FE 01	CP 01
	JR Z sta4	not present	02CC 28 3E	JR Z 030C
; en er is dan ook niets meer aanwezig				
	? LD HL,A000		02CE 21 00 A0	LD HL,A000
	CALL 55-test		02D1 CD FE 03	CALL 03FE
	JR NZ sta4		02D4 20 36	JR NZ 030C
; toch niet aanwezig !				
	LD HL,memsize	32-k!	02D6 21 5C 60	LD HL,605C
	INC (HL)	is nu 02	02D9 34	INC (HL)
	XOR A		02DA AF	XOR A
	OUT 94	bank 0	02DB D3 94	OUT 94
	LD HL,E000		02DD 21 00 E0	LD HL,E000
	LD BC, <u>20FF</u>	8-k 8 bits	02E0 01 FF 20	LD BC,20FF
	CALL ramtest	E000-FFFF	02E3 CD DC 03	CALL 03DC
	CP 02		02E6 FE 02	CP 02
	JR Z sta3	error	02E8 28 DE	JR Z 02C8
	CP 01		02EA FE 01	CP 01
	JR Z sta4	not present	02EC 28 1E	JR Z 030C
	? LD HL,E000		02EE 21 00 E0	LD HL,E000
	CALL 55-test		02F1 CD FE 03	CALL 03FE
	JR NZ sta4	not prsent	02F4 20 16	JR NZ 030C
	LD A,01		02F6 3E 01	LD A,01
	OUT 94	bank 1	02F8 D3 94	OUT 94
	✗ LD HL,E000		02FA 21 00 E0	LD HL,E000
	LD BC, <u>20FF</u>	8-k 8 bit	02FD 01 FF 20	LD BC,20FF
	CALL ramtest	0-test	0300 CD DC 03	CALL 03DC
	OR A		0303 B7	OR A
	JR NZ str1		0304 20 8D	JR NZ 0293
	LD HL,memsize	48-k!	0306 21 5C 60	LD HL,605C
	INC (HL)	is nu 03	0309 34	INC (HL)
; A is op nul door ramtest achtergelaten				
	OUT 94	eerste bank	030A D3 94	OUT 94
sta4	LD A,(memsize)		030C 3A 5C 60	LD A,(605C)
	CP 03	48-k ?	030F FE 03	CP 03
	JR NZ sta6		0311 20 1D	JR NZ 0330
	LD A,(1000)		0313 3A 00 10	LD A,(1000)
	BIT 0,A	ROM module?	0316 CB 47	BIT 0,A
	JR NZ sta6	not present	0318 20 16	JR NZ 0330
	BIT 1,A	DOS nodig?	031A CB 4F	BIT 1,A
	JR Z sta6	niet nodig	031C 28 12	JR Z 0330

```

; Activeer-floppy controller en wacht
; even tot hij wakker is.
    LD A,04          test FDC      031E 3E 04        LD A,04
    OUT 90           0320 D3 90      0322 06 00        OUT 90
    LD B,00           0324 10 FE      0326 DB 8C        LD B,00
sta5     DJNZ sta5       delay      0328 FE 80        DJNZ 0324
    IN 8C           status FDC
; lees status floppy controller en
; kijk of hij iets zinnigs terug zegt.
    CP 80           oke?        032A CC 90 0E      IN 8C
; Main status register oke?
    CALL Z getDOS    DOS halen   032D AF          CP 80
    XOR A
    OUT 90           FDC uit     032E D3 90
sta6     LD HL,500E
    LD (i/otype),HL
    LD B,08
    LD E,00
    CALL scr-type    flash>screen 0330 21 0E 50
    LD B,09
    LD E,04
    CALL scr-type    steady>screen 0333 22 14 60
    IN 20           baudrate   0336 06 08
    BIT 0,A          printer    0338 1E 00
    LD A,07          300 baud   033A CD BE 01
    JR Z sta7
    LD A,01          /200-2400 baud 033D 06 09
sta7     LD (baudrate),A
    CALL cas.init
    CALL initkey
    CALL enkey        enable toets 033F 1E 04
    LD A,(1000)
    AND F5           bit 1,3 weg 0341 CD BE 01
    CP 54           ROM oke?    0344 DB 20
    JR NZ bootstrap  geen ROM   0346 CB 47
    LD HL,1000
    LD A,(HL)
    BIT 0,A
    JR NZ bootstrap  geen ROM   0348 3E 07
    PUSH HL
    CALL ROMtest    1000-2FFF
    POP HL
    BIT 3,(HL)
    SET 5,H          H nu 30
    CALL Z ROMtest  3000-4FFF
    LD HL,1005
    LD DE,5002
    LD BC,0008
    LDIR
    JP 1010         start module 034A 28 02
    LD A,01          034C 3E 01
    LD (baudrate),A 034E 32 16 60
    CALL cas.init
    CALL initkey
    CALL enkey        enable toets 0351 CD 05 06
    LD A,(1000)
    AND F5           0354 CD 67 01
    CP 54           0357 CD B4 00
    JR NZ 0385
    LD HL,1000
    LD A,(HL)
    BIT 0,A
    JR NZ 0385
    PUSH HL
    CALL 0407
    POP HL
    BIT 3,(HL)
    SET 5,H
    CALL Z 0407
    LD HL,1005
    LD DE,5002
    LD BC,0008
    LDIR
    JP 1010

```

; bootstrap wordt uitgevoerd als er geen
; ROM insteek module met de juiste
; kenmerken wordt gevonden. Begint met
; PHILIPS microcomputer P2000 op
; het scherm te zetten.

bootstrap	CALL prtext	\$0385	0385 CD 2E 04	CALL 042E
bool	LD B,00		0388 06 00	LD B,00
	LD E,03		038A 1E 03	LD E,03
	CALL scr-type		038C CD BE 01	CALL 01BE
; wacht tot er een tape aanwezig is.				
boo2	LD A,07	status	038F 3E 07	LD A,07
	CALL cassette	cassette	0391 CD F1 04	CALL 04F1
	JR Z boo2	niet aanwezig	0394 28 F9	JR Z 038F
	LD A,01	terugspoelen	0396 3E 01	LD A,01
	CALL cassette		0398 CD F1 04	CALL 04F1
	JR NZ bool	error! again	039B 20 EB	JR NZ 0388
	LD HL,0400	alleen header	039D 21 00 04	LD HL,0400
	LD (fileleng),HL		03A0 22 32 60	LD (6032),HL
	LD HL,0000	datalengte	03A3 21 00 00	LD HL,0000
	LD (recleng),HL		03A6 22 34 60	LD (6034),HL
	LD A,06	lees header	03A9 3E 06	LD A,06
	CALL cassette		03AB CD F1 04	CALL 04F1
	JR NZ bool	error!	03AE 20 D8	JR NZ 0388
	LD A,01	spoel terug	03B0 3E 01	LD A,01
	CALL cassette		03B2 CD F1 04	CALL 04F1
	JR NZ bool	error!	03B5 20 D1	JR NZ 0388
	LD A,(type)		03B7 3A 41 60	LD A,(6041)
	CP "P	bootstrap?	03BA FE 50	CP 50
	JR NZ bool	opnieuw	03BC 20 CA	JR NZ 0388
	LD HL,(recleng)		03BE 2A 34 60	LD HL,(6034)
	LD (fileleng),HL		03C1 22 32 60	LD (6032),HL
	LD HL,(load)		03C4 2A 45 60	LD HL,(6045)
	LD (transfer),HL		03C7 22 30 60	LD (6030),HL
	LD A,06	inlezen	03CA 3E 06	LD A,06
	CALL cassette		03CC CD F1 04	CALL 04F1
	JR NZ bool	error!	03CF 20 B7	JR NZ 0388
	LD B,00	veeg C van	03D1 06 00	LD B,00
	LD E,03	scherm	03D3 1E 03	LD E,03
	CALL scr-type		03D5 CD BE 01	CALL 01BE
	LD HL,(strt-boot)	programma	03D8 2A 43 60	LD HL,(6043)
	JP (HL)	opstarten	03DB E9	JP (HL)

```

ram test    HL= startadres
            B= high byte aantal
            C= FF :test heel byte
            = OF :test half byte
; exit:      A=0  alles oke
            =1  geen RAM aanwezig
            =2  fout bij (HL)

ramtest   PUSH HL          startadres
          POP IX           opbergen ] IX
          LD D,C
          LD C,00           BC aantal
          DEC HL
ram1      INC HL
          LD (HL),00
          LD A,(HL)
          AND D
; masker voor test staat nu in D
; als binnen het masker ook maar een
; bitje niet nul is, wordt de error
; routine aangeroepen.
          JR NZ ram2        A is nu nul
          DEC BC
          OR B               klaar?
          JR NZ ram1
          OR C               klaar?
          JR NZ ram1
          RET                alles oke
ram2      PUSH IX          start adres
          POP BC
          XOR A
          INC A               error 1
          SBC HL,BC          fout in RAM ?
          JR NZ ram3
          RET
ram3      INC A               error 2
          RET                geen RAM
; test op een byte met 55
; Z: alles in orde
; NZ: fout

55-test   LD (HL),55        test met 55
          LD A,(HL)
          CP 55
          RET NZ
          XOR A
          LD (HL),A 00        mem]00
          RET

```

03DC	E5	PUSH HL
03DD	DD E1	POP IX
03DF	51	LD D,C
03E0	0E 00	LD C,00
03E2	2B	DEC HL
03E3	23	INC HL
03E4	36 00	LD (HL),00
03E6	7E	LD A,(HL)
03E7	A2	AND D
03E8	20 08	JR NZ 03F2
03EA	OB	DEC BC
03EB	B0	OR B
03EC	20 F5	JR NZ 03E3
03EE	B1	OR C
03EF	20 F2	JR NZ 03E3
03F1	C9	RET
03F2	DD E5	PUSH IX
03F4	C1	POP BC
03F5	AF	XOR A
03F6	3C	INC A
03F7	ED 42	SBC HL,BC
03F9	20 01	JR NZ 03FC
03FB	C9	RET
03FC	3C	INC A
03FD	C9	RET
03FE	36 55	LD (HL),55
0400	7E	LD A,(HL)
0401	FE 55	CP 55
0403	C0	RET NZ
0404	AF	XOR A
0405	77	LD (HL),A
0406	C9	RET

ROMtest	INC HL	haal test byt	0407 23	INC HL
	LD C,(HL)	uit module	0408 4E	LD C,(HL)
	INC HL	B en C en D	0409 23	INC HL
	LD B,(HL)	en E nul:	040A 46	LD B,(HL)
	INC HL] geen test	040B 23	INC HL
	LD E,(HL)		040C 5E	LD E,(HL)
	INC HL		040D 23	INC HL
	LD D,(HL)		040E 56	LD D,(HL)
; BC blok lengte DE checksum				
ROM1	LD A,B		040F 78	LD A,B
	OR C		0410 B1	OR C
	JR NZ ROM2	1 en 2 nul?	0411 20 06	JR NZ 0419
	LD A,D		0413 7A	LD A,D
	OR E		0414 B3	OR E
	RET Z	3 en 4 nul?	0415 C8	RET Z
; BC op 0000 en DE niet op 0000				
; betekent tape laden !				
	JP bootstrap			
ROM2	INC HL		0416 C3 85 03	JP 0385
	LD A,(HL)	byte 5	0419 23	INC HL
	ADD E		041A 7E	LD A,(HL)
	JR NC ROM3		041B 83	ADD E
ROM3	INC D		041C 30 01	JR NC 041F
	LD E,A		041E 14	INC D
	DEC BC		041F 5F	LD E,A
	JR ROM1		0420 0B	DEC BC
			0421 18 EC	JR 040F
; veeg een stuk scherm schoon				
;	HL: start adres			
;	A: aantal regels			
wisregel	LD B,50	80 char	0423 06 50	LD B,50
wisl	LD (HL),00		0425 36 00	LD (HL),00
	INC HL		0427 23	INC HL
	DJNZ wisl		0428 10 FB	DJNZ 0425
	DEC A	1 regel klaar	042A 3D	DEC A
	JR NZ wisregel	meer regels?	042B 20 F6	JR NZ 0423
	RET		042D C9	RET

; zet meerdere strings op scherm
; format: twee bytes schermadres
; een byte lengte
; tekst
; als afsluiting FF, anders
; weer string met schermadres etc.
; als er een M-model aanwezig is, wordt
; er 0014 bij schermadres opgeteld.

prtext	LD HL,START	adres PHILIPS	042E 21 1E 02	LD HL,021E
prtl	LD E,(HL)	text] scherm	0431 5E	LD E,(HL)
	INC HL	eerste 2 byte	0432 23	INC HL
<i>LD A,FF</i>	LD D,(HL)	bestemming	0433 56	LD D,(HL)
<i>LD B,00</i>	INC HL		0434 23	INC HL
<i>CP E</i>	x LD B,00		0435 06 00	LD B,00
<i>RET Z</i>	LD C,(HL)	lengte string	0437 4E	LD C,(HL)
	INC HL		0438 23	INC HL
	LD A,(T-M)	M-model?	0439 3A 13 60	LD A,(6013)
	BIT 0,A		043C CB 47	BIT 0,A
	JR Z prt2		043E 28 07	JR Z 0447
X	PUSH HL		0440 E5	PUSH HL
	LD HL,0014	20 pos rechts	0441 21 14 00	LD HL,0014
	ADD HL,DE		0444 19	ADD HL,DE
	EX DE,HL		0445 EB	EX DE,HL
	POP HL		0446 E1	POP HL
prt2	LD IR		0447 ED B0	LD IR
	x LD A,(HL)		0449 7E	LD A,(HL)
	x CP FF	nog meer?	044A FE FF	CP FF
	JR NZ prt1		044C 20 E3	JR NZ 0431
	x RET		044E C9	RET

; cpm dos zit in bank een vanaf E000

CPM-start	LD (sptemp),SP	.	044F ED 73 55 60	LD (6055),SP
	LD SP,6130		0453 31 30 61	LD SP,6130
	LD A,01		0456 3E 01	LD A,01
	OUT 94	bank 1	0458 D3 94	OUT 94
	CALL E000	disk entry	045A CD 00 E0	CALL E000
	PUSH AF		045D F5	PUSH AF
	XOR A		045E AF	XOR A
	OUT 94	bank 0	045F D3 94	OUT 94
	POP AF		0461 F1	POP AF
	LD SP,(sptemp)		0462 ED 7B 55 60	LD SP,(6055)
	RET		0466 C9	RET

; routine schakelt naar bank 0 om
; data te lezen vanaf disk en gaat dan
; weer terug naar bank 1: terugkomen
; in DOS system. HL bestemming, B aantal
; bytes te lezen. C: poort FDC

inpdisk	XOR A	bank 0	0467 AF	XOR A
	OUT 94		0468 D3 94	OUT 94
	IN 90	status FDC	046A DB 90	IN 90
	RRA	ready?	046C 1F	RRA
	JP NC leel		046D D2 6A 04	JP NC 046A
	INI	lees van disk	0470 ED A2	INI
	JP NZ leel	genoeg?	0472 C2 6A 04	JP NZ 046A
lee2	LD A,01	bank 1	0475 3E 01	LD A,01
	OUT 94		0477 D3 94	OUT 94
	RET		0479 C9	RET

; idem voor schrijven. HL oorsprong
; B aantal bytes, C poort nr.

writedisk	XOR A	bank 0	047A AF	XOR A
	OUT 94		047B D3 94	OUT 94
wril	IN 90	status FDC	047D DB 90	IN 90
	RRA		047F 1F	RRA
	<u>JP JR NC wril</u>		0480 30 FB	JR NC 047D
	OUTI	schrijf	0482 ED A3	OUTI
	JP NZ wril	klaar?	0484 C2 7D 04	JP NZ 047D
	<u>JP JR lee2</u>		0487 18 EC	JR 0475

; bekijk input buffer
; resultaat: Z: buffer leeg
; C: stoptoets in buffer

stat.key	LD A,(keytel)	iets in toets	0489 3A 0C 60	LD A,(600C)
	OR A	buffer?	048C B7	OR A
	RET Z	leeg	048D C8	RET Z
	LD A,(keybuf)	begin buffer	048E 3A 00 60	LD A,(6000)
	CP 58	stop?	0491 FE 58	CP 58
	JR NZ stal		0493 20 03	JR NZ 0498
	SUB 59	dus carry	0495 D6 59	SUB 59
	RET		0497 C9	RET
	SCF	niet leeg	0498 37	SCF
	CCF	geen stop	0499 3F	CCF
	RET		049A C9	RET

wacht tot er een toets in de input
buffer zit en haal hem eruit. Hij
wordt er door de interrupt ingezet.

readkey	CALL stat.key			
	JR Z readkey	keypress?	049B CD 89 04	CALL 0489
	DI		049E 28 FB	JR Z 049B
	EXX	save reg.	04A0 F3	DI
	LD HL, keytel		04A1 D9	EXX
	LD A,(HL)		04A2 21 0C 60	LD HL, 600C
	OR A	buffer leeg?	04A5 7E	LD A,(HL)
	JR Z readkey		04A6 B7	OR A
	DEC (HL)	1 minder	04A7 28 F2	JR Z 049B
	LD A,(keytel)		04A9 35	DEC (HL)
	OR A	leeg ?	04AA 3A 0C 60	LD A,(600C)
	JR NZ shiftbuf		04AD B7	OR A
	LD A,(keybuf)		04AE 20 0A	JR NZ 04BA
	LD HL, keybuf		04B0 3A 00 60	LD A,(6000)
	LD (HL),00	buffer schoon	04B3 21 00 60	LD HL, 6000
	JR real		04B6 36 00	LD (HL),00
shiftbuf	LD B,00	schuif buffer	04B8 18 19	JR 04D3
	LD C,A	op	04BA 06 00	LD B,00
	LD HL,6001		04BC 4F	LD C,A
	LD DE, keybuf		04BD 21 01 60	LD HL,6001
	LD A,(keybuf)	haal toets op	04C0 11 00 60	LD DE,6000
	LDIR	schuif	04C3 3A 00 60	LD A,(6000)
	LD HL, keybuf		04C6 ED B0	LDIR
	PUSH AF		04C8 21 00 60	LD HL,6000
	LD A,(keytel)		04CB F5	PUSH AF
	LD L,A	,	04CC 3A 0C 60	LD A,(600C)
	LD (HL),00	buffer schoon	04CF 6F	LD L,A
	POP AF		04D0 36 00	LD (HL),00
	EXX		04D2 F1	POP AF
	CP 58	stop ?	04D3 D9	EXX
	SCF		04D4 FE 58	CP 58
	JR Z rea2		04D6 37	SCF
	CCF	stop in carry	04D7 28 01	JR Z 04DA
	EI		04D9 3F	CCF
	RET		04DA FB	EI
			04DB C9	RET
clearkey	XOR A	veeg toets-	04DC AF	XOR A
	LD (keytel),A	buffer leeg	04DD 32 0C 60	LD (600C),A
	RET		04E0 C9	RET

ORG 04E1

offscan	EQU 00B4	keyboard
diskey	EQU 00BC	disable key
scr-type	EQU 01BE	C] scherm
err0	EQU 01DE	system error
ophaal	EQU 0600	
off-M	EQU 0B91	video-M
on-M	EQU 0B96	video-M
klok	EQU 6010	
caserror	EQU 6017	
newblock	EQU 6018	cassette
lengte	EQU 601A	
transfer	EQU 6030	tape
descrip	EQU 6030	header
fileleng	EQU 6032	
recleng	EQU 6034	
type	EQU 6041	B,P etc
strt-boot	EQU 6043	
load	EQU 6045	boot
recnum	EQU 604F	
motorstat	EQU 6050	
stacas	EQU 6060	
Popsave	EQU 6061	
oldblock	EQU 6064	cassette
validleng	EQU 6066	in blok tape
desl	EQU 6068	copy descrip
desleng	EQU 606A	lengte header
nrnul	EQU 606C	einde blok
telblok	EQU 606E	blokken

; betekenis inhoud caserror

;

00	operatie normaal beeindigd
41	geen cassette
42	begin van tape
43	checksum fout, record gelezen
44	checksum fout in startmerk
45	einde van de band
46	band vol bij schrijven EOF
47	geen stopje
49	rewind tijd verlopen
4A	kort record, geen checksum fout
4B	verkeerde functie keuze
4C	kort record + checksum fout
4D	geen startmerk gevonden
4E	geen record gevonden

; jumptabel voor diverse cassette
; opdrachten. entry point bij cassette

jumpcas	WORD cas.init	init pointers	04E1 05 06 B4 0B OF OC 61 OC	
	WORD rewind	spoelen cass		
	WORD]skip	1 blok		
	WORD [skip	1 blok		
	WORD EOT	last file	E4 0B 7A 05 52 05 15 05	
	WORD schrijf	file		
	WORD lees	file		
	WORD tapestat	status		
cassette	CALL opberg	save register	04F1 CD FC 05	CALL 05FC
	CP 08	verkeerd	04F4 FE 08	CP 08
	JR C cas1	argument in A	04F6 38 04	JR C 04FC
	LD A,"K	error K	04F8 3E 4B	LD A,4B
	JR cas4		04FA 18 51	JR 054D
cas1	LD HL,descrip		04FC 21 30 60	LD HL,6030
	LD (des1),HL	copieren	04FF 22 68 60	LD (6068),HL
	LD HL,0020	header 32	0502 21 20 00	LD HL,0020
	LD (desleng),HL	bytes	0505 22 6A 60	LD (606A),HL
	LD HL,jumpcas	vorm jump	0508 21 E1 04	LD HL,04E1
	SLA A	adres tabel	050B CB 27	SLA A
	LD D,00		050D 16 00	LD D,00
	LD E,A	HL pointer	050F 5F	LD E,A
	ADD HL,DE	in jumptabel	0510 19	ADD HL,DE
	OE = 2*status			
	CP OE	status?	0511 FE OE	CP OE
	JR NZ cas2		0513 20 OD	JR NZ 0522
; tapestatus: Z= no tape				
; C= write protected				
tapestat	IN 20	cass status	0515 DB 20	IN 20
	RRA	schuif 4	0517 1F	RRA
	RRA	bitjes weg	0518 1F	RRA
	RRA	write-protec	0519 1F	RRA
	RRA	in carry	051A 1F	RRA
	CPL	inverteer A	051B 2F	CPL
	BIT 0,A	notape: zero	051C CB 47	BIT 0,A
	CALL ophaal	recall reg.	051E CD 00 06	CALL 0600
	RET		0521 C9	RET

LD DE,casret	return adres	0522 11 42 05	LD DE,0542
PUSH DE	op stack	0525 D5	PUSH DE
LD A,(recnum)	nr.block	0526 3A 4F 60	LD A,(604F)
LD (telblok),A		0529 32 6E 60	LD (606E),A
XOR A		052C AF	XOR A
LD (caserror),A	alles oke	052D 32 17 60	LD (6017),A
CALL diskey	key] scan	0530 CD BC 00	CALL 00BC
PUSH HL		0533 E5	PUSH HL
LD B,"T		0534 06 54	LD B,54
LD E,03		0536 1E 03	LD E,03
CALL scr-type	in Basic een	0538 CD BE 01	CALL 01BE
POP HL	adres in ROM	053B E1	POP HL
LD E,(HL)		053C 5E	LD E,(HL)
INC HL	DE pointer	053D 23	INC HL
LD D,(HL)	jump adres	053E 56	LD D,(HL)
EX DE,HL	cass routine	053F EB	EX DE,HL
DI		0540 F3	DI
JP (HL)		0541 E9	JP (HL)

; nette beeindiging van cassette
; routines. adres van deze routine wordt
; door cas2 op stack gezet.

casret	EI	return adres	0542 FB	EI
	XOR A	flag's cass	0543 AF	XOR A
	LD (motorstat),A		0544 32 50 60	LD (6050),A
	CALL offscan		0547 CD B4 00	CALL 00B4
	LD A,(caserror)		054A 3A 17 60	LD A,(6017)
	OR A	fout in zero	054D B7	OR A
	CALL ophaal	registers	054E CD 00 06	CALL 0600
	RET		0551 C9	RET

lees	CALL blocknr	record	0552 CD A9 05	CALL 05A9
leel	CALL nextparam		0555 CD C6 05	CALL 05C6
	LD A,(telblok)	blokaantal	0558 3A 6E 60	LD A,(606E)
	LD HL,motorstat		055B 21 50 60	LD HL,6050
	CP 01	last blok?	055E FE 01	CP 01
; na laatste blok moet de motor uit-				
; gezet worden, anders niet.				
	JR NZ lee2		0560 20 04	JR NZ 0566
	RES 2,(HL)		0562 CB 96	RES 2,(HL)
; na blok motor uitzetten				
	JR lee3			
lee2	SET 2,(HL)		0564 18 02	JR 0568
; na blok motor aanzetten				
lee3	CALL blkread		0566 CB D6	SET 2,(HL)
	LD A,(caserror)			CALL 0872
	OR A	fout gemaakt?	0568 CD 72 08	LD A,(6017)
	RET NZ	terug.	056B 3A 17 60	OR A
	LD A,(telblok)	volgende blok	056E B7	RET NZ
	DEC A	gaan lezen.	056F C0	LD A,(606E)
	LD (telblok),A		0570 3A 6E 60	DEC A
	JR NZ leel	klaar?	0573 3D	LD (606E),A
	RET		0574 32 6E 60	JR NZ 0555
			0577 20 DC	RET
			0579 C9	
; begin met het aantal benodigde blokken				
; te berekenen met aanroep van blocknr				
schrijf	CALL blocknr	aantal blok's	057A CD A9 05	CALL 05A9
schl	CALL nextparam		057D CD C6 05	CALL 05C6
	LD A,(telblok)	blokaantal	0580 3A 6E 60	LD A,(606E)
	LD HL,motorstat		0583 21 50 60	LD HL,6050
	CP 01	laatste blok?	0586 FE 01	CP 01
; na laatste blok moet de motor uit				
	JR NZ sch2		0588 20 06	JR NZ 0590
; straks motor off				
	RES 2,(HL)		058A CB 96	RES 2,(HL)
; straks niet meer schrijven				
	RES 3,(HL)		058C CB 9E	RES 3,(HL)
	JR sch3		058E 18 04	JR 0594
; na blok motor aan en schrijven				
sch2	SET 2,(HL)	motor on	0590 CB D6	SET 2,(HL)
	SET 3,(HL)	schrijven	0592 CB DE	SET 3,(HL)
sch3	LD (recnum),A		0594 32 4F 60	LD (604F),A
	CALL blkwrite		0597 CD 1F 06	CALL 061F
	LD A,(caserror)		059A 3A 17 60	LD A,(6017)
	OR A	alles oke?	059D B7	OR A
	RET NZ	error	059E C0	RET NZ
	LD A,(telblok)	blokteller	059F 3A 6E 60	LD A,(606E)
	DEC A		05A2 3D	DEC A
	LD (telblok),A		05A3 32 6E 60	LD (606E),A
	JR NZ schl	meer blokken	05A6 20 D5	JR NZ 057D
	RET		05A8 C9	RET

; blocknr berekent het aantal blokken
; dat nodig is om een file weg te
; schrijven.

blocknr	LD HL,(fileleng) aantal block LD DE,0400 1 k.byte DEC HL XOR A blo1 INC A tel nr blocks SBC HL,DE klaar? JR NC blo1 LD (telblok),A nog A blokken LD HL,(transfer) installer de LD (newblock),HL nieuwe pointr LD HL,(recleng) LD (lengte),HL RET	05A9 2A 32 60 05AC 11 00 04 05AF 2B 05B0 AF 05B1 3C 05B2 ED 52 05B4 30 FB 05B6 32 6E 60 05B9 2A 30 60 05BC 22 18 60 05BF 2A 34 60 05C2 22 1A 60 05C5 C9	LD HL,(6032) LD DE,0400 DEC HL XOR A INC A SBC HL,DE JR NC 05B1 LD (606E),A LD HL,(6030) LD (6018),HL LD HL,(6034) LD (601A),HL RET
nextparam	LD DE,0400 lengte blok LD HL,(newblock) \$05C9 LD (oldblock),HL ADD HL,DE ; zet pointer voor volgende keer LD (newblock),HL begin blok XOR A LD HL,(lengte) SBC HL,DE	05C6 11 00 04 05C9 2A 18 60 05CC 22 64 60 05CF 19 05D0 22 18 60 05D3 AF 05D4 2A 1A 60 05D7 ED 52	LD DE,0400 LD HL,(6018) LD (6064),HL ADD HL,DE LD (6018),HL XOR A LD HL,(601A) SBC HL,DE
	; als Carry gezet is, is de te schrijven ; lengte kleiner dan 1 K, dus laatste ; blok moet dan geschreven worden. JR C updl LD (validlen),DE normaal LD (lengte),HL JR upd2	05D9 38 09 05DB ED 53 66 60 05DF 22 1A 60 05E2 18 OC 05E4 2A 1A 60 05E7 22 66 60 05EA 21 00 00 05ED 22 1A 60 05F0 EB	JR C 05E4 LD (6066),DE LD (601A),HL JR 05F0 LD HL,(601A) LD (6066),HL LD HL,0000 LD (601A),HL EX DE,HL
upd1	LD HL,(lengte) LD (validlen),HL restje LD HL,0000 LD (lengte),HL klaar!	05F1 ED 5B 66 60 05F5 AF	LD (6066) XOR A
upd2	EX DE,HL LD DE,(validleng) XOR A SBC HL,DE bepaal aantal LD (nrnul),HL nullen RET	05F6 ED 52 05F8 22 6C 60 05FB C9	SBC HL,DE LD (606C),HL RET

opberg	EX (SP),HL	ook op 01A5	05FC E3 05FD D5 05FE C5 05FF E9	EX (SP),HL PUSH DE PUSH BC JP (HL)
haal	POP HL POP BC POP DE EX (SP),HL RET	ook op 01AB	0600 E1 0601 C1 0602 D1 0603 E3 0604 C9	POP HL POP BC POP DE EX (SP),HL RET
cas.init	XOR A LD (stacas),A LD (motorstat),A IN 20 AND 18 CP 10 RET NZ LD HL,stacas SET 2,(HL) LD E,40 LD H,"1 JP put	cass aanw.? cass aanwezig cassette off	0605 AF 0606 32 60 60 0609 32 50 60 060C DB 20 060E E6 18 0610 FE 10 0612 C0 0613 21 60 60 0616 CB D6 0618 1E 40 061A 26 31 061C C3 54 0B	XOR A LD (6060),A LD (6050),A IN 20 AND 18 CP 10 RET NZ LD HL,6060 SET 2,(HL) LD E,40 LD H,31 JP OB54
blokwrite	CALL castest RET NZ LD A,(stacas) BIT 0,A	geen cas,stop EOF ?	061F CD 83 0B 0622 C0 0623 3A 60 60 0626 CB 47	CALL OB83 RET NZ LD A,(6060) BIT 0,A
; indien EOF moet teruggespoeld worden	JR Z blwl BIT 1,A JR Z terug	writemasker? 1 blok terug	0628 28 0B 062A CB 4F 062C 28 38	JR Z 0635 BIT 1,A JR Z 0666
; reset write bit in stacas	RES 1,A LD (stacas),A JR wrblock	tapestatus write data	062E CB 8F 0630 32 60 60 0633 18 4C	RES 1,A LD (6060),A
blwl	CALL wrdata LD A,(caserror) CP 4D JR Z blo3 CP 00 RET NZ LD E,4A LD H,60 CALL 157msec RET NZ LD A,(motorstat) BIT 2,A LD H,60 JR Z offmotor LD A,48 OUT 10 RET	r mark w data no startmark oke ? fout forward write data] tape gap end data error motor off? key + forward	0635 CD C5 06 0638 3A 17 60 063B FE 4D 063D 28 22 063F FE 00 0641 C0 0642 1E 4A 0644 26 60 0646 CD B0 0A 0649 C0 064A 3A 50 60 064D CB 57 064F 26 60 0651 28 05 0653 3E 48 0655 D3 10 0657 C9	JR 0681 CALL 06C5 LD A,(6017) CP 4D JR Z 0661 CP 00 RET NZ LD E,4A LD H,60 CALL OAB0 RET NZ LD A,(6050) BIT 2,A LD H,60 JR Z 0658 LD A,48 OUT 10 RET

offmotor	LD A,40 OUT 10 LD E,A CALL 120-msec RET	motor uit + key enable cassette uit	0658 3E 40 065A D3 10 065C 5F 065D CD D5 0A 0660 C9	LD A,40 OUT 10 LD E,A CALL 0AD5 RET
	LD HL,stacas SET 0,(HL)	EOF mark	0661 21 60 60 0664 CB C6	LD HL,6060 SET 0,(HL)
terug	CALL shortskip LD A,(caserror) CP "B JR Z wrblock CP 00 RET NZ LD HL,stacas SET 5,(HL) CALL #0C13 LD A,(caserror) CP 00 RET NZ	backwards begin tape alles oke	0666 CD 5D 0C 0669 3A 17 60 066C FE 42 066E 28 11 0670 FE 00 0672 C0 0673 21 60 60 0676 CB EE 0678 CD 13 0C 067B 3A 17 60 067E FE 00 0680 C0	CALL 0C5D LD A,(6017) CP 42 JR Z 0681 CP 00 RET NZ LD HL,6060 SET 5,(HL) CALL 0C13 LD A,(6017) CP 00 RET NZ
; schrijf	startmerk + datablok			
wrblock	CALL wr-cass LD E,4A LD H,61 CALL 157msec LD A,(caserror) CP 00 RET NZ LD A,(motorstat) BIT 3,A JR NZ wrbl CALL EOT LD A,(caserror) CP "E RET NZ LD A,"F LD (caserror),A RET	forward write gap end data fout ? schrijven? last file band vol? vol bij EOF	0681 CD F0 06 0684 1E 4A 0686 26 61 0688 CD B0 0A 068B 3A 17 60 068E FE 00 0690 C0 0691 3A 50 60 0694 CB 5F 0696 20 0F 0698 CD E4 0B 069B 3A 17 60 069E FE 45 06A0 C0 06A1 3E 46 06A3 32 17 60 06A6 C9	CALL 06F0 LD E,4A LD H,61 CALL 0AB0 LD A,(6017) CP 00 RET NZ LD A,(6050) BIT 3,A JR NZ 06A7 CALL OBE4 LD A,(6017) CP 45 RET NZ LD A,46 LD (6017),A RET
	RES 3,A LD (motorstat),A LD A,(stacas) SET 1,A LD (stacas),A LD A,(motorstat) BIT 2,A RET NZ LD A,42 OUT 10 CALL 120msec LD H,61 JR offmotor	no write writemask na w motor of key + write	06A7 CB 9F 06A9 32 50 60 06AC 3A 60 60 06AF CB CF 06B1 32 60 60 06B4 3A 50 60 06B7 CB 57 06B9 C0 06BA 3E 42 06BC D3 10 06BE CD D0 0A 06C1 26 61 06C3 18 93	RES 3,A LD (6050),A LD A,(6060) SET 1,A LD (6060),A LD A,(6050) BIT 2,A RET NZ LD A,42 OUT 10 CALL 0ADO LD H,61 JR 0658

;lees start mark, write data

wrdata	CALL off-M CALL readmark LD A,(caserror) CP 00 JR Z wrd2 CALL on-M RET LD A,4A OUT 10 CALL writegap JR NZ wrd1 CALL wr-record CALL on-M LD A,(caserror) CP 00 RET Z LD H,60 JP offmotor	strepen M typ forward write voor M model	06C5 CD 91 0B 06C8 CD C2 08 06CB 3A 17 60 06CE FE 00 06D0 28 04 06D2 CD 96 0B 06D5 C9 06D6 3E 4A 06D8 D3 10 06DA CD B6 0A 06DD 20 F3 06DF CD 35 07 06E2 CD 96 0B 06E5 3A 17 60 06E8 FE 00 06EA C8 06EB 26 60 06ED C3 58 06	CALL 0B91 CALL 08C2 LD A,(6017) CP 00 JR Z 06D6 CALL 0B96 RET LD A,4A OUT 10 CALL 0AB6 JR NZ 06D2 CALL 0735 CALL 0B96 LD A,(6017) CP 00 RET Z LD H,60 JP 0658
wr-cass	LD A,4A OUT 10 LD A,(stacas) BIT 4,A JR Z wrcl CALL \$0AE9 JR wrc2 CALL 516msec RET NZ	forward write	06F0 3E 4A 06F2 D3 10 06F4 3A 60 60 06F7 CB 67 06F9 28 05 06FB CD E9 0A 06FE 18 03 0700 CD BB 0A 0703 C0	LD A,4A OUT 10 LD A,(6060) BIT 4,A JR Z 0700 CALL 0AE9 JR 0703 CALL 0ABB RET NZ
wrc1	CALL off-M LD (spsave),SP	start gap	0704 CD 91 0B 0707 ED 73 61 60 070B E5 070C E5	CALL 0B91 LD (6061),SP PUSH HL
wrc2	PUSH HL CALL \$072A CALL #084E JR NZ wrc4 CALL writegap JR NZ wrc4 CALL wr-record JR NZ wrc4 CALL M-on RET	M-video off	070D CD 2A 07 0710 CD 4E 08 0713 20 0E 0715 CD B6 0A 0718 20 09 071A CD 35 07 071D 20 04 071F CD 96 0B 0722 C9	PUSH HL CALL 072A CALL 084E JR NZ 0723 CALL 0AB6 JR NZ 0723 CALL 0735 JR NZ 0723 CALL 0B96 RET
wrc3	LD H,61 CALL offmotor JR wrc3	error ?	0723 26 61 0725 CD 58 06 0728 18 F5	LD H,61 CALL 0658 JR 071F
wrc4				

\$072A	LD HL,0003 PUSH HL PUSH HL LD HL,#0813 PUSH HL JR #0775	dummy	072A 21 03 00 072D E5 072E E5 072F 21 13 08 0732 E5 0733 18 40	LD HL,0003 PUSH HL PUSH HL LD HL,0813 PUSH HL JR 0775
wr-record	LD (spsave),SP PUSH HL PUSH HL LD HL,#084E PUSH HL LD HL,0003 PUSH HL PUSH HL LD HL,#0813 PUSH HL LD HL,(nrnul) LD A,L OR H JR Z #0755 PUSH HL PUSH HL LD HL,\$0807 PUSH HL	ret adres	0735 ED 73 61 60 0739 E5 073A E5 073B 21 4E 08 073E E5 073F 21 03 00 0742 E5 0743 E5 0744 21 13 08 0747 E5 0748 2A 6C 60 074B 7D 074C B4 074D 28 06 074F E5 0750 E5 0751 21 07 08 0754 E5	LD (6061),SP PUSH HL PUSH HL LD HL,084E PUSH HL LD HL,0003 PUSH HL PUSH HL LD HL,0813 PUSH HL LD HL,(606C) LD A,L OR H JR Z 0755 PUSH HL PUSH HL LD HL,0807 PUSH HL
#0755	LD HL,(validleng) LD A,L OR H JR Z #0765 PUSH HL LD HL,(oldblock) PUSH HL LD HL,#07FC PUSH HL		0755 2A 66 60 0758 7D 0759 B4 075A 28 09 075C E5 075D 2A 64 60 0760 E5 0761 21 FC 07 0764 E5	LD HL,(6066) LD A,L OR H JR Z 0765 PUSH HL LD HL,(6064) PUSH HL LD HL,07FC PUSH HL
#0765	LD HL,(desleng) LD A,L OR H JR Z #0775 PUSH HL LD HL,(desl) PUSH HL LD HL,#07FC PUSH HL		0765 2A 6A 60 0768 7D 0769 B4 076A 28 09 076C E5 076D 2A 68 60 0770 E5 0771 21 FC 07 0774 E5	LD HL,(606A) LD A,L OR H JR Z 0775 PUSH HL LD HL,(6068) PUSH HL LD HL,07FC PUSH HL
#0775	LD IY,bitout LD C,00 SUB A SCF LD B,08 LD A,AA		0775 FD 21 95 07 0779 0E 00 077B 97 077C 37 077D 06 08 077F 3E AA	LD IY,0795 LD C,00 SUB A SCF LD B,08 LD A,AA

	EXX	0781 D9	EXX
#0785	LD DE,0000	0782 11 00 00	LD DE,0000
	EXX	0785 D9	EXX
	LD D,A	0786 57	LD D,A
!!	EX AF	0787 08	EX AF
	IN 20	0788 DB 20	IN 20
	AND 30	078A E6 30	AND 30
	; bit 4: cassette aanwezig		
	; bit 5: begin - einde signaal		
	CP 20 begin-eind ?	078C FE 20	CP 20
	JP NZ wr-error	078E C2 3A 08	JP NZ 083A
	LD A,00	0791 3E 00	LD A,00
	RR A	0793 CB 1F	RR A
bitout	LD A,D	0795 7A	LD A,D
	NOP	0796 00	NOP
	AND 01	0797 E6 01	AND 01
	OR 4A	0799 F6 4A	OR 4A
	EX (SP),HL dummy	079B E3	EX (SP),HL
	EX (SP),HL dummy	079C E3	EX (SP),HL
	NOP dummy	079D 00	NOP
	OUT 10	079E D3 10	OUT 10
	EX (SP),HL dummy	07A0 E3	EX (SP),HL
	EX (SP),HL dummy	07A1 E3	EX (SP),HL
	NOP dummy	07A2 00	NOP
	JR #07A5	07A3 18 00	JR 07A5
#07A5	NOP	07A5 00	NOP
	EX AF haal flags op	07A5 .08	EX AF
	JR C \$07D8	07A7 38 2F	JR C 07D8
	EX AF	07A9 08	EX AF
	LD A,D	07AA 7A	LD A,D
	EXX	07AB D9	EXX
	AND 01	07AC E6 01	AND 01
	XOR E	07AE AB	XOR E
	LD E,A	07AF 5F	LD E,A
	AND 01	07B0 E6 01	AND 01
	JR Z #07D3	07B2 28 1F	JR Z 07D3
	LD A,02	07B4 3E 02	LD A,02
	XOR E	07B6 AB	XOR E
	LD E,A	07B7 5F	LD E,A
	LD A,40	07B8 3E 40	LD A,40
#07BA	XOR D	07BA AA	XOR D
	RRA	07BB 1F	RRA
	RR E	07BC CB 1B	RR E
	JR C #07C2	07BE 38 02	JR C 07C2
	JR \$07C4	07C0 18 02	JR 07C4
#07C2	OR 80	07C2 F6 80	OR 80
\$07C4	LD D,A	07C4 57	LD D,A
	EXX	07C5 D9	EXX
%07C6	XOR A	07C6 AF	XOR A
	RR D	07C7 CB 1A	RR D
	CCF	07C9 3F	CCF
	ADC 4A	07CA CE 4A	ADC 4A
	OUT 10	07CC D3 10	OUT 10

DJNZ #07E3		07CE 10 13	DJNZ 07E3
LD B,08		07D0 06 08	LD B,08
JP (HL)		07D2 E9	JP (HL)
#07D3	JR NZ #07BA	07D3 20 E5	JR NZ 07BA
	JP #07BA	07D5 C3 BA 07	JP 07BA
\$07D8	EX AF	07D8 08	EX AF
	LD A,(IX+00) dummy	07D9 DD 7E 00	LD A,(IX+00)
	LD A,04	07DC 3E 04	LD A,04
%07DE	DEC A	07DE 3D	DEC A
	JR NZ %07DE	07DF 20 FD	JR NZ 07DE
	JR %07C6	07E1 18 E3	JR 07C6
#07E3	EX AF	07E3 08	EX AF
	JP PE \$081F	07E4 EA 1F 08	JP PE 081F
	LD A,80	07E7 3E 80	LD A,80
	DEC A	07E9 3D	DEC A
	EX AF	07EA 08	EX AF
	POP HL	07EB E1	POP HL
	EXX	07EC D9	EXX
	LD A,2D	07ED 3E 2D	LD A,2D
	LD C,L	07EF 4D	LD C,L
	LD B,L	07F0 45	LD B,L
	LD C,B	07F1 48	LD C,B
	LD C,A	07F2 4F	LD C,A
	LD B,(HL)	07F3 46	LD B,(HL)
	LD B,L	07F4 45	LD B,L
	LD D,D	07F5 52	LD D,D
	DEC L	07F6 2D	DEC L
	POP HL	07F7 E1	POP HL
	POP BC	07F8 C1	POP BC
	EXX	07F9 D9	EXX
	JR bitout	07FA 18 99	JR 0795
#07FC	EX AF	07FC 08	EX AF
	EXX	07FD D9	EXX
	RR A	07FE CB 1F	RR A
	OR A	0800 B7	OR A
	LD A,(HL)	0801 7E	LD A,(HL)
	CPI	0802 ED A1	CPI
	JP #0785	0804 C3 85 07	JP 0785
\$0807	EX AF	0807 08	EX AF
	EXX	0808 D9	EXX
	RR A	0809 CB 1F	RR A
	OR A	080B B7	OR A
	LD A,00	080C 3E 00	LD A,00
	CPI	080E ED A1	CPI
	JP #0785	0810 C3 85 07	JP 0785
	EX AF	0813 08	EX AF
	EXX	0814 D9	EXX
	LD A,E	0815 7B	LD A,E
	LD E,D	0816 5A	LD E,D
	LD D,AA	0817 16 AA	LD D,AA
	SCF	0819 37	SCF
	CPI	081A ED A1	CPI
	JP #0785	081C C3 85 07	JP 0785

\$081F	EX AF		081F 08	EX AF
	RET Z		0820 C8	RET Z
	OR 00		0821 F6 00	OR 00
	OR 00		0823 F6 00	OR 00
	OR 00		0825 F6 00	OR 00
	NOP		0827 00	NOP
	INC C		0828 0C	INC C
	DI		0829 F3	DI
	DEC C		082A 0D	DEC C
	LD IY,bitout	dummy	082B FD 21 95 07	LD IY,0795
	NOP	dummy	082F 00	NOP
	LD IY,bitout	dummy	0830 FD 21 95 07	LD IY,0795
	LD A,(IY+00)	dummy	0834 FD 7E 00	LD A,(IY+00)
	JP bitout		0837 C3 95 07	JP 0795

; einde band tijdens schrijven
; haal een adres van het stack, zonder
; het stack of de stackpointer te
; verstoren, en jump daarheen.

wr-error	LD D,A		083A 57	LD D,A
	LD HL,(spsave)		083B 2A 61 60	LD HL,(6061)
	DEC HL	1 adres terug	083E 2B	DEC HL
	DEC HL		083F 2B	DEC HL
	DEC HL	2 adres terug	0840 2B	DEC HL
	DEC HL		0841 2B	DEC HL
	DEC HL	3 adres terug	0842 2B	DEC HL
	LD A,(HL)	ophalen in HL	0843 7E	LD A,(HL)
	DEC HL		0844 2B	DEC HL
	LD L,(HL)	vorm adres	0845 6E	LD L,(HL)
	LD H,A		0846 67	LD H,A
	LD SP,(spsave)	herstel SP	0847 ED 7B 61 60	LD SP,(6061)
	LD B,00		084B 06 00	LD B,00
	JP (HL)	en ga er heen	084D E9	JP (HL)

; Zero: no error. Non zero: error

#084E	LD A,4A	write forward	084E 3E 4A	LD A,4A
	OUT 10		0850 D3 10	OUT 10
	LD A,00		0852 3E 00	LD A,00
	LD (caserror),A	alles oke	0854 32 17 60	LD (6017),A
	LD A,B		0857 78	LD A,B
	LD B,00		0858 06 00	LD B,00
	LD HL,(klok)		085A 2A 10 60	LD HL,(6010)
	ADD HL,BC		085D 09	ADD HL,BC
	LD (klok),HL		085E 22 10 60	LD (6010),HL
	CP 08		0861 FE 08	CP 08
	RET Z		0863 C8	RET Z
	BIT 4,D		0864 CB 62	BIT 4,D
	LD A,"A		0866 3E 41	LD A,41
	JR NZ #086E		0868 20 04	JR NZ 086E
	SUB A		086A 97	SUB A
	INC A	voor Z-flag	086B 3C	INC A
	LD A,"E		086C 3E 45	LD A,45
#086E	LD (caserror),A		086E 32 17 60	LD (6017),A
	RET		0871 C9	RET

plkread	CALL %0B9A RET NZ BIT 0,(HL) JR Z #0880 .LD A,"M .LD (caserror),A RET	0872 CD 9A 0B 0875 C0 0876 CB 46 0878 28 06 087A 3E 4D 087C 32 17 60 087F C9	CALL 0B9A RET NZ BIT 0,(HL) JR Z 0880 LD A,4D LD (6017),A RET
#0881	CALL M-off video CALL readmark LD HL,(6051) LD (6053),HL LD A,(caserror) CP 00 JR Z #0897 CALL M-on video RET	0880 CD 91 0B 0883 CD C2 08 0886 2A 51 60 0889 22 53 60 088C 3A 17 60 088F FE 00 0891 28 04 0893 CD 96 0B 0896 C9	CALL OB91 CALL 08C2 LD HL,(6051) LD (6053),HL LD A,(6017) CP 00 JR Z 0897 CALL OB96 RET
#0897	CALL 69msec JR NZ #08A4 CALL #091C LD A,(caserror) CP 00	0897 CD DA 0A 089A 20 08 089C CD 1C 09 089F 3A 17 60 08A2 FE 00	CALL OADA JR NZ 08A4 CALL 091C LD A,(6017) CP 00
#08A4	LD H,62 JR Z \$08AF CALL offmotor CALL M-on video RET	08A4 26 62 08A6 28 07 08A8 CD 58 06 08AB CD 96 0B 08AE C9	LD H,62 JR Z 08AF CALL 0658 CALL OB96 RET
\$08AF	LD E,48 key + forward CALL 157msec gap end data CALL M-on video LD A,(motorstat) BIT 2,A na w moto off RET NZ LD H,62 JP offmotor	08AF 1E 48 08B1 CD B0 0A 08B4 CD 96 0B 08B7 3A 50 60 08BA CB 57 08BC C0 08BD 26 62 08BF C3 58 06	LD E,48 CALL OAB0 CALL OB96 LD A,(6050) BIT 2,A RET NZ LD H,62 JP 0658
readmark	LD A,48 key + forward OUT 10 CALL #0AC0 RET NZ	08C2 3E 48 08C4 D3 10 08C6 CD C0 0A 08C9 C0	LD A,48 OUT 10 CALL OAC0 RET NZ
\$08CA	CALL \$0906 LD DE,(nrnul) CALL #0A64 LD HL,(nrnul) LD (nrnul),DE LD A,(caserror) CP 00 JR NZ #08E7 LD A,H OR L JR NZ \$08CA RET	08CA CD 06 09 08CD ED 5B 6C 60 08D1 CD 64 0A 08D4 2A 6C 60 08D7 ED 53 6C 60 08DB 3A 17 60 08DE FE 00 08E0 20 05 08E2 7C 08E3 B5 08E4 20 E4 08E6 C9	CALL 0906 LD DE,(606C) CALL 0A64 LD HL,(606C) LD (606C),DE LD A,(6017) CP 00 JR NZ 08E7 LD A,H OR L JR NZ 08CA RET

#08E7	CP 4A	08E7 FE 4A	CP 4A
	JR Z \$08CA	08E9 28 DF	JR Z 08CA
	CP 4C	08EB FE 4C	CP 4C
	JR Z \$08CA	08ED 28 DB	JR Z 08CA
	CP 43	08EF FE 43	CP 43
	JR Z \$08CA	08F1 28 D7	JR Z 08CA
	CP 4E	08F3 FE 4E	CP 4E
	JR NZ #0901	08F5 20 OA	JR NZ 0901
	LD HL, stacas	08F7 21 60 60	LD HL, 6060
	SET 0,(HL)	08FA CB C6	SET 0,(HL)
	LD A,"M	08FC 3E 4D	LD A,4D
	LD (caserror),A	08FE 32 17 60	LD (6017),A
#0901	LD H,63	0901 26 63	LD H,63
	JP offmotor	0903 C3 58 06	JP 0658
\$0906	LD (spsave),SP	0906 ED 73 61 60	LD (6061),SP
	LD HL,0000	090A 21 00 00	LD HL,0000
	PUSH HL	090D E5	PUSH HL
	PUSH HL	090E E5	PUSH HL
	LD HL,#0A25	090F 21 25 OA	LD HL,0A25
	PUSH HL	0912 E5	PUSH HL
	PUSH HL	0913 E5	PUSH HL
	PUSH HL	0914 E5	PUSH HL
	LD HL,#0AOB	0915 21 0B OA	LD HL,0AOB
	PUSH HL	0918 E5	PUSH HL
	JP #0953	0919 C3 53 09	JP 0953
#091C	LD HL,#0A64	091C 21 64 OA	LD HL,0A64
	PUSH HL	091F E5	PUSH HL
	LD (spsave),SP	0920 ED 73 61 60	LD (6061),SP
	LD HL,0000	0924 21 00 00	LD HL,0000
	PUSH HL	0927 E5	PUSH HL
	PUSH HL	0928 E5	PUSH HL
	LD HL,#0A25	0929 21 25 OA	LD HL,0A25
	PUSH HL	092C E5	PUSH HL
	LD HL,(validleng)	092D 2A 66 60	LD HL,(6066)
	LD A,L	0930 7D	LD A,L
	OR H	0931 B4	OR H
	JR Z #093D	0932 28 09	JR Z 093D
	PUSH HL	0934 E5	PUSH HL
	LD HL,(oldblock)	0935 2A 64 60	LD HL,(6064)
	PUSH HL	0938 E5	PUSH HL
	LD HL,#0A10	0939 21 10 OA	LD HL,0A10
	PUSH HL	093C E5	PUSH HL
#093D	LD HL,(desleng)	093D 2A 6A 60	LD HL,(606A)
	LD A,H	0940 7C	LD A,H
	OR L	0941 B5	OR L
	JR Z #094D	0942 28 09	JR Z 094D
	PUSH HL	0944 E5	PUSH HL
	LD HL,(des1)	0945 2A 68 60	LD HL,(6068)
	PUSH HL	0948 E5	PUSH HL
	LD HL,#0A10	0949 21 10 OA	LD HL,0A10
	PUSH HL	094C E5	PUSH HL

#094D	PUSH HL PUSH HL LD HL,#0AOB PUSH HL	094D E5 094E E5 094F 21 0B 0A 0952 E5	PUSH HL PUSH HL LD HL,0AOB PUSH HL
#0953	LD A,40 key OUT 10 LD A,48 key + forward OUT 10 CALL 300-micro XOR 40 LD H,A LD IY,#096A LD C,00 LD DE,4CE5 IN 20 AND 70 CP H JR Z #099B AND 30 CP 20 begin-eind? JR NZ #098A DEC DE LD A,D OR E JR Z \$098D NOP INC C DI DEC C LD A,48 key + forward OUT 10 LD IY,#096A JP (IY) JP 1-mil.sec LD A,"N LD (caserror),A LD SP,(spsave) LD L,00 LD B,08 RET	0953 3E 40 0955 D3 10 0957 3E 48 0959 D3 10 095B CD 78 0B 095E EE 40 0960 67 0961 FD 21 6A 09 0965 0E 00 0967 11 E5 4C 096A DB 20 096C E6 70 096E BC 096F 28 2A 0971 E6 30 0973 FE 20 0975 20 13 0977 1B 0978 7A 0979 B3 097A 28 11 097C 00 097D 0C 097E F3 097F 0D 0980 3E 48 0982 D3 10 0984 FD 21 6A 09 0988 FD E9 098A C3 39 0A 098D 3E 4E 098F 32 17 60 0992 ED 7B 61 60 0996 2E 00 0998 06 08 099A C9	0953 3E 40 0955 D3 10 0957 3E 48 0959 D3 10 095B CD 78 0B 095E EE 40 0960 67 0961 FD 21 6A 09 0965 0E 00 0967 11 E5 4C 096A DB 20 096C E6 70 096E BC 096F 28 2A 0971 E6 30 0973 FE 20 0975 20 13 0977 1B 0978 7A 0979 B3 097A 28 11 097C 00 097D 0C 097E F3 097F 0D 0980 3E 48 0982 D3 10 0984 FD 21 6A 09 0988 FD E9 098A C3 39 0A 098D 3E 4E 098F 32 17 60 0992 ED 7B 61 60 0996 2E 00 0998 06 08 099A C9
#096A			
#098A			
\$098D			
#099B	LD (6051),DE LD IY,\$09AE EXX LD E,07 EXX EX AF SUB A SCF LD DE,0000	099B ED 53 51 60 099F FD 21 AE 09 09A3 D9 09A4 1E 07 09A6 D9 09A7 08 09A8 97 09A9 37 09AA 11 00 00	LD (6051),DE LD IY,09AE EXX LD E,07 EXX EX AF SUB A SCF LD DE,0000

#09AD	EX AF	save AF	09AD 08	EX AF
\$09AE	LD A,H		09AE 7C	LD A,H
	XOR 40		09AF EE 40	XOR 40
	LD H,A		09B1 67	LD H,A
	LD B,00		09B2 06 00	LD B,00
#09B4	IN 20		09B4 DB 20	IN 20
	LD L,A	save data bit	09B6 6F	LD L,A
	AND 70		09B7 E6 70	AND 70
	CP H		09B9 BC	CP H
	JR Z rd-klok		09BA 28 04	JR Z 09C0
	DJNZ #09B4		09BC 10 F6	DJNZ 09B4
	JR 1-mil.sec		09BE 18 79	JR 0A39
rd-klok	EX AF		09C0 08	EX AF
	JR C #09E9		09C1 38 26	JR C 09E9
	EX AF		09C3 08	EX AF
	XOR A		09C4 AF	XOR A
	RLC L		09C5 CB 05	RLC L
	RLA		09C7 17	RLA
	XOR E		09C8 AB	XOR E
	LD E,A		09C9 5F	LD E,A
	AND 01		09CA E6 01	AND 01
	JR Z #09D4		09CC 28 06	JR Z 09D4
	LD A,02		09CE 3E 02	LD A,02
	XOR E		09D0 AB	XOR E
	LD E,A		09D1 5F	LD E,A
	LD A,40		09D2 3E 40	LD A,40
#09D4	XOR D		09D4 AA	XOR D
	RRA		09D5 1F	RRA
	RR E		09D6 CB 1B	RR E
	JR NC \$09DC		09D8 30 02	JR NC 09DC
	OR 80		09DA F6 80	OR 80
\$09DC	LD D,A		09DC 57	LD D,A
%09DD	RRC L		09DD CB 0D	RRC L
	EXX		09DF D9	EXX
	RR D		09E0 CB 1A	RR D
	DEC E		09E2 1D	DEC E
	JR NZ \$09EC		09E3 20 07	JR NZ 09EC
	LD E,08		09E5 1E 08	LD E,08
	JP (IX)		09E7 DD E9	JP (IX
#09E9	EX AF		09E9 08	EX AF
	JR %09DD		09EA 18 F1	JR 09DD
\$09EC	EX AF	herstel F	09EC 08	EX AF
	JP PE #09FB		09ED EA FB 09	JP PE 09FB
	LD B,80		09F0 06 80	LD B,80
	DEC B		09F2 05	DEC B
	EX AF		09F3 08	EX AF
	POP IX		09F4 DD E1	POP IX
	POP HL		09F6 E1	POP HL
	POP BC		09F7 C1	POP BC
	EXX		09F8 D9	EXX
	JR \$09AE		09F9 18 B3	JR 09AE

#09FB	EX AF EXX NOP INC C DI DEC C LD A,48 OUT 10 LD IY,\$09AE JP (IY)	key + forward	09FB 08 09FC D9 09FD 00 09FE 0C 09FF F3 0A00 0D 0A01 3E 48 0A03 D3 10 0A05 FD 21 AE 09 0A09 FD E9	EX AF EXX NOP INC C DI DEC C LD A,48 OUT 10 LD IY,09AE JP (IY)
#0AOB	EX AF EXX SUB A JR #09AD		0A0B 08 0A0C D9 0A0D 97 0A0E 18 9D	EX AF EXX SUB A JR 09AD
#0A10	EXX LD A,D OR E JR NZ \$0A1A EX AF XOR A JP %0A1E		0A10 D9 0A11 7A 0A12 B3 0A13 20 05 0A15 08 0A16 AF 0A17 C3 1E 0A	EXX LD A,D OR E JR NZ 0A1A EX AF XOR A JP 0A1E
\$0A1A	EX AF SCF RLA OR A		0A1A 08 0A1B 37 0A1C 17 0A1D B7	EX AF SCF RLA OR A
%0A1E	EXX LD (HL),D CPI EXX JR #09AD		0A1E D9 0A1F 72 0A20 ED A1 0A22 D9 0A23 18 88	EXX LD (HL),D CPI EXX JR 09AD
#0A25	INC BC EXX LD A,D OR E JR NZ \$0A2F EX AF XOR A JR #0A33		0A25 03 0A26 D9 0A27 7A 0A28 B3 0A29 20 04 0A2B 08 0A2C AF 0A2D 18 04	INC BC EXX LD A,D OR E JR NZ 0A2F EX AF XOR A JR 0A33
\$0A2F	EX AF SCF RLA OR A		0A2F 08 0A30 37 0A31 17 0A32 B7 0A33 06 80 0A35 05	EX AF SCF RLA OR A
#0A33	LD B,80 DEC B JP #09AD		0A36 C3 AD 09	LD B,80 DEC B JP 09AD

```

; delay 1 millisec na stoppen tape

l-mil.sec LD B,1E
l-m      DJNZ l-m          delay
        IN 20
        AND 30
        LD HL,(spsave)
        SBC HL,SP
        LD SP,(spsave)
        CP 20          begin-eind?
        RET NZ
        LD A,00
        LD (caserror),A
        LD B,08
        LD A,D
        OR E
        RET Z
        EX AF
        BIT 0,A
        RET Z
        BIT 1,A
        RET Z
        EX AF
        LD A,20
        LD B,00
        RET

```

0A39 06 1E	LD B,1E
0A3B 10 FE	DJNZ 0A3B
0A3D DB 20	IN 20
0A3F E6 30	AND 30
0A41 2A 61 60	LD HL,(6061)
0A44 ED 72	SBC HL,SP
0A46 ED 7B 61 60	LD SP,(6061)
0A4A FE 20	CP 20
0A4C C0	RET NZ
0A4D 3E 00	LD A,00
0A4F 32 17 60	LD (6017),A
0A52 06 08	LD B,08
0A54 7A	LD A,D
0A55 B3	OR E
0A56 C8	RET Z
0A57 08	EX AF
0A58 CB 47	BIT 0,A
0A5A C8	RET Z
0A5B CB 4F	BIT 1,A
0A5D C8	RET Z
0A5E 08	EX AF
0A5F 3E 20	LD A,20
0A61 06 00	LD B,00
0A63 C9	RET

#0A04	LD H,A LD A,B CP 08 JR Z #0A98 error info in bitvorm in H register BIT 4,H LD A,"A JR NZ vulerror BIT 5,H LD A,"E JR Z vulerror LD A,"L INC L DEC L JR NZ vulerror EXX DEC BC DEC BC DEC BC LD (nrnul),BC EXX LD A,"C LD (caserror),A \$0A8A vulerror	0A64 67 0A65 78 0A66 FE 08 0A68 28 2E 0A6A CB 64 0A6C 3E 41 0A6E 20 17 0A70 CB 6C 0A72 3E 45 0A74 28 11 0A76 3E 4C 0A78 2C 0A79 2D 0A7A 20 0B 0A7C D9 0A7D 0B 0A7E 0B 0A7F 0B 0A80 ED 43 6C 60 0A84 D9 0A85 3E 43 0A87 32 17 60 0A8A 7D 0A8B 32 63 60 0A8E 06 00 0A90 2A 10 60 0A93 09 0A94 22 10 60 0A97 C9	LD H,A LD A,B CP 08 JR Z 0A98 BIT 4,H LD A,41 JR NZ 0A87 BIT 5,H LD A,45 JR Z 0A87 LD A,4C INC L DEC L JR NZ 0A87 EXX DEC BC DEC BC DEC BC LD (606C),BC EXX LD A,43 LD (6017),A LD A,L LD (6063),A LD B,00 LD HL,(klok) ADD HL,BC LD (klok),HL RET
#0A98	XOR A CP L LD A,"J JR NZ vulerror LD A,(caserror) CP 00 JR NZ \$0A8A EXX DEC BC DEC BC DEC BC LD (nrnul),BC EXX JR \$0A8A	0A98 AF 0A99 BD 0A9A 3E 4A 0A9C 20 E9 0A9E 3A 17 60 0AA1 FE 00 0AA3 20 E5 0AA5 D9 0AA6 0B 0AA7 0B 0AA8 0B 0AA9 ED 43 6C 60 0AAD D9 0AAE 18 DA	XOR A CP L LD A,4A JR NZ 0A87 LD A,(6017) CP 00 JR NZ 0A8A EXX DEC BC DEC BC DEC BC LD (606C),BC EXX JR 0A8A
157msec	PUSH BC LD C,32 JP \$0B3C	0AB0 C5 0AB1 0E 32 0AB3 C3 3C 0B	PUSH BC LD C,32 JP 0B3C

writegap	PUSH BC LD C,1B JR delay	85 msec	OAB6 C5 OAB7 0E 1B OAB9 18 3A	PUSH BC LD C,1B JR OAF5
516msec	PUSH BC LD C,A4 JR delay	start gap	OABB C5 OABC 0E A4 OABE 18 35	PUSH BC LD C,A4 JR OAF5
#0AC0	LD A,(stacas) BIT 4,A JR Z 120msec RES 4,A LD (stacas),A CALL 500msec RET NZ		OAC0 3A 60 60 OAC3 CB 67 OAC5 28 09 OAC7 CB A7 OAC9 32 60 60 OACC CD F2 0A OACF CO	LD A,(6060) BIT 4,A JR Z OADO RES 4,A LD (6060),A CALL OAF2 RET NZ
120msec	PUSH BC LD C,26 JR delay		OADO C5 OAD1 0E 26 OAD3 18 20	PUSH BC LD C,26 JR OAF5
120-msec	PUSH BC LD C,26 JR \$OB3C		OAD5 C5 OAD6 0E 26 OAD8 18 62	PUSH BC LD C,26 JR OB3C
69msec	PUSH BC LD C,16 JR delay	69 msec	OADA C5 OADB 0E 16 OADD 18 16	PUSH BC LD C,16 JR OAF5
101msec	PUSH BC LD C,20 JR \$OB3C	101 ms delay	OADF C5 OAE0 0E 20 OAE2 18 58	PUSH BC LD C,20 JR OB3C
261msec	PUSH BC LD C,53 JR delay	261 ms delay	OAE4 C5 OAE5 0E 53 OAE7 18 0C	PUSH BC LD C,53 JR OAF5
\$0AE9	RES 4,A LD (stacas),A 1-sec		OAE9 CB A7 OAEB 32 60 60 OAAE CD F2 0A	RES 4,A LD (6060),A CALL OAF2
500msec	CALL 500msec RET NZ		OAF1 CO OAF2 C5 OAF3 0E 9F	RET NZ
;	500msc	ga verder met 500 ms delay		PUSH BC LD C,9F
bij entry staat BC al op stack			OAF5 06 AF	LD B,AF
delay	LD B,AF	C: delay	OAF7 DB 20	IN 20
dell	IN 20		OAF9 E6 30	AND 30
	AND 30		OAFB FE 20	CP 20
	CP 20	begin-eind?	OAFD 20 07	JR NZ OB06
	JR NZ #OB06	iets anders	0AFF 10 F6	DJNZ OAF7
	DJNZ dell	not found	OB01 0D	DEC C
	DEC C			
;	lengte test afhankelijk van C		OB02 20 F1	JR NZ OAF5
	JR NZ delay	alweer end?	OB04 C1	POP BC
	POP BC	herstel	OB05 C9	RET
	RET			

#OB06	OR A	OB06 B7	OR A
	LD A,"A	OB07 3E 41	LD A,41
	JR NZ \$OB0D	OB09 20 02	JR NZ OBOD
	LD A,"E	OB0B 3E 45	LD A,45
- \$OB0D	LD (caserror),A	OB0D 32 17 60	LD (6017),A
	LD H,A	OB10 67	LD H,A
	LD A,40	OB11 3E 40	LD A,40
	OUT 10	OB13 D3 10	OUT 10
	LD E,A	OB15 5F	LD E,A
	LD A,(stacas)	OB16 3A 60 60	LD A,(6060)
	BIT 2,A	OB19 CB 57	BIT 2,A
	JR Z #OB2A	OB1B 28 0D	JR Z OB2A
	CALL put	OB1D CD 54 0B	CALL OB54
	LD A,C	OB20 79	LD A,C
	SBC 02	OB21 DE 02	SBC 02
	LD C,A	OB23 4F	LD C,A
	JR C #OB39	OB24 38 13	JR C OB39
	LD B,8F	OB26 06 8F	LD B,8F
	JR \$OB2C	OB28 18 02	JR OB2C
#OB2A	LD B,AF	OB2A 06 AF	LD B,AF
\$OB2C	IN 20	OB2C DB 20	IN 20
	AND 30	OB2E E6 30	AND 30
	XOR 20	OB30 EE 20	XOR 20
	JR C #OB2A	OB32 38 F6	JR C OB2A
	DJNZ \$OB2C	OB34 10 F6	DJNZ OB2C
	DEC C	OB36 0D	DEC C
- #OB39	JR NZ #OB2A	OB37 20 F1	JR NZ OB2A
	INC C	OB39 0C	INC C
	POP BC	OB3A C1	POP BC
	RET	OB3B C9	RET
\$OB3C	LD A,(stacas)	OB3C 3A 60 60	LD A,(6060)
	BIT 2,A	OB3F CB 57	BIT 2,A
	JR Z delay	OB41 28 B2	JR Z OAF5
	DEC C	OB43 0D	DEC C
	DEC C	OB44 0D	DEC C
	DEC C	OB45 0D	DEC C
	CALL put	OB46 CD 54 0B	CALL OB54
	LD A,(caserror)	OB49 3A 17 60	LD A,(6017)
	LD H,A	OB4C 67	LD H,A
	CALL put	OB4D CD 54 0B	CALL OB54
	LD B,70	OB50 06 70	LD B,70
	JR dell	OB52 18 A3	JR OAF7

```

; schrijf een byte naar tape
; start bit=1, stop bit=0
; data in H register
; E bevat de volgende opdrachten:
;   40 = cassette off
;   42 = write
;   44 = reverse
;   48 = forward
;   4A = forward + write
; dit alles met enable keyboard

```

put	LD B,09 LD A,E AND 7F LD E,A RLC E	9300 T-states masker start bit = 1	OB54 06 09 OB56 7B OB57 E6 7F OB59 5F OB5A CB 03	LD B,09 LD A,E AND 7F LD E,A RLC E
put1	XOR 80 OUT 10 CALL DELAY SRL H CCF LD A,E ADC 00	invert bit 7 bit 7] tape timing data] carry inverteer carry: info	OB5C EE 80 OB5E D3 10 OB60 CD 72 0B OB63 CB 3C OB65 3F OB66 7B OB67 CE 00	XOR 80 OUT 10 CALL OB72 SRL H CCF LD A,E ADC 00
; data nu in bit 0 van A	RRCA DJNZ put1 AND 7F OUT 10 RRC E	bit 0] bit 7 9 bits B reg stop bit=0 naar tape draai E terug	OB69 0F OB6A 10 F0 OB6C E6 7F OB6E D3 10 OB70 CB 0B	RRCA DJNZ OB5C AND 7F OUT 10 RRC E
DELAY	LD D,3D	980 T-states	OB72 16 3D	LD D,3D
DELL	DEC D JR NZ DELL RET	delay	OB74 15 OB75 20 FD OB77 C9	DEC D JR NZ OB74 RET
300-micro	LD B,2E	300 microsec	OB78 06 2E	LD B,2E
300delay	DJNZ 300delay IN 20 AND 40 SET 5,A RET	klok van tape	OB7A 10 FE OB7C DB 20 OB7E E6 40 OB80 CB EF OB82 C9	DJNZ OB7A IN 20 AND 40 SET 5,A RET
castest	IN 20 AND 18 RET Z CP 18 RET Z LD A,"G LD (caserror),A RET	write protec no cassette	OB83 DB 20 OB85 E6 18 OB87 C8 OB88 FE 18 OB8A C8 OB8B 3E 47 OB8D 32 17 60 OB90 C9	IN 20 AND 18 RET Z CP 18 RET Z LD A,47 LD (6017),A RET

M-off	LD A,FF OUT 70 RET	disable wait line M-model "	0B91 3E FF 0B93 D3 70 0B95 C9	LD A,FF OUT 70 RET
	XOR A OUT 70 RET	enable wait line M-model	0B96 AF 0B97 D3 70 0B99 C9	XOR A OUT 70 RET
%0B9A	LD HL,stacas BIT 1,(HL) RET Z RES 1,(HL) JP err0 CALL EOT LD A,(caserror) CP 00 RET NZ LD HL,stacas SET 0,(HL) RET	write masker last file	0B9A 21 60 60 0B9D CB 4E 0B9F C8 0BA0 CB 8E 0BA2 C3 DE 01 0BA5 CD E4 0B 0BA8 3A 17 60 0BAB FE 00 0BAD C0 0BAE 21 60 60 0BB1 CB C6 0BB3 C9	LD HL,6060 BIT 1,(HL) RET Z RES 1,(HL) JP 01DE CALL OBE4 LD A,(6017) CP 00 RET NZ LD HL,6060 SET 0,(HL) RET
rewind	CALL %0B9A RET NZ LD A,44 OUT 10 LD B,67 CALL 1-sec JR NZ #OBCC DJNZ #OBBE LD A,"I te lang, tap LD (caserror),A breuk ? LD C,0F LD A,(caserror) CP 45 JR NZ #OBDF LD A,00 alles oke LD (caserror),A LD HL,stacas SET 4,(HL) RES 0,(HL) LD H,67 JP offmotor	spoelen cass key+reverse max 103 sec te lang, tap breuk ? alles oke	0BB4 CD 9A 0B 0BB7 C0 0BB8 3E 44 0BBA D3 10 0BBC 06 67 0BBE CD EE 0A 0BC1 20 09 0BC3 10 F9 0BC5 3E 49 0BC7 32 17 60 0BCA 0E 0F 0BCC 3A 17 60 0BCF FE 45 0BD1 20 0C 0BD3 3E 00 0BD5 32 17 60 0BD8 21 60 60 0BDB CB E6 0BDD CB 86 0BDF 26 67 0BE1 C3 58 06	CALL 0B9A RET NZ LD A,44 OUT 10 LD B,67 CALL 0AEE JR NZ OBCC DJNZ OBBE LD A,49 LD (6017),A LD C,0F LD A,(6017) CP 45 JR NZ OBDF LD A,00 LD (6017),A LD HL,6060 SET 4,(HL) RES 0,(HL) LD H,67 JP 0658

EOT	CALL castest	last file	OBE4 CD 83 0B	CALL OB83
	RET NZ		OBE7 C0	RET NZ
	LD A,4A	write forward	OBE8 3E 4A	LD A,4A
	OUT 10		OBEA D3 10	OUT 10
	LD B,0F		OBEC 06 0F	LD B,0F
\$OBEE	CALL 120msec		OBEE CD D0 0A	CALL OADO
	JR NZ #OC03		OBF1 20 10	JR NZ OC03
	DJNZ \$OBEE		OBF3 10 F9	DJNZ OBEE
	LD A,00		OBF5 3E 00	LD A,00
	LD (caserror),A		OBF7 32 17 60	LD (6017),A
	LD HL,stacas		OBFA 21 60 60	LD HL,6060
	RES 1,(HL)		OBFD CB 8E	RES 1,(HL)
	SET 0,(HL)		OBFF CB C6	SET 0,(HL)
	RES 4,(HL)		OC01 CB A6	RES 4,(HL)
#OC03	LD A,42	key+write	OC03 3E 42	LD A,42
	OUT 10		OC05 D3 10	OUT 10
	CALL 120msec		OC07 CD D0 0A	CALL OADO
	LD H,68		OC0A 26 68	LD H,68
	JP offmotor		OC0C C3 58 06	JP 0658
]skip	CALL %OB9A		OC0F CD 9A 0B	CALL OB9A
	RET NZ		OC12 C0	RET NZ
#OC13	LD A,48	key + forward	OC13 3E 48	LD A,48
	OUT 10		OC15 D3 10	OUT 10
	CALL #OAC0		OC17 CD C0 0A	CALL OAC0
	RET NZ		OC1A C0	RET NZ
	CALL 69msec		OC1B CD DA 0A	CALL OADA
	RET NZ		OC1E C0	RET NZ
	LD A,"N		OC1F 3E 4E	LD A,4E
	LD (caserror),A		OC21 32 17 60	LD (6017),A
	CALL %OCEE		OC24 CD EE 0C	CALL OCEE
	LD D,EA		OC27 16 EA	LD D,EA
#OC29	CALL \$OCBF		OC29 CD BF 0C	CALL OCBF
	JR C #OC51		OC2C 38 23	JR C OC51
	JR NZ #OC3A		OC2E 20 0A	JR NZ OC3A
	LD D,23		OC30 16 23	LD D,23
	LD A,00		OC32 3E 00	LD A,00
	LD (caserror),A		OC34 32 17 60	LD (6017),A
	LD A,H		OC37 7C	LD A,H
	JR #OC29		OC38 18 EF	JR OC29
#OC3A	LD A,(caserror)		OC3A 3A 17 60	LD A,(6017)
	CP 00		OC3D FE 00	CP 00
	JR NZ \$OC56		OC3F 20 15	JR NZ OC56
	LD HL,stacas		OC41 21 60 60	LD HL,6060
	BIT 5,(HL)		OC44 CB 6E	BIT 5,(HL)
	JR Z #OC4B		OC46 28 03	JR Z OC4B
	RES 5,(HL)		OC48 CB AE	RES 5,(HL)
	RET		OC4A C9	RET
#OC4B	LD HL,telblok		OC4B 21 6E 60	LD HL,606E
	DEC (HL)		OC4E 35	DEC (HL)
	JR NZ #OC13		OC4F 20 C2	JR NZ OC13
#OC51	LD H,66		OC51 26 66	LD H,66
	JP offmotor		OC53 C3 58 06	JP 0658
\$OC56	LD HL,stacas		OC56 21 60 60	LD HL,6060
	SET 0,(HL)		OC59 CB C6	SET 0,(HL)
	JR #OC51		OC5B 18 F4	JR OC51

shortskip	LD L,01 JR [sk1	backwards	0C5D 2E 01 0C5F 18 0B	LD L,01 JR OC6C
[skip	CALL %0B9A RET NZ LD HL,stacas RES 0,(HL) LD L,00	1 blok	0C61 CD 9A 0B 0C64 C0 0C65 21 60 60 0C68 CB 86 0C6A 2E 00	CALL 0B9A RET NZ LD HL,6060 RES 0,(HL) LD L,00
	skip blok terug	\$0C6C		
	L = 01 short skip			
	L = 00 geen short skip			
	LD A,44 OUT 10 CALL %0CEE LD E,E1 LD D,01 CALL \$OCBF JR C [sk6 JR NZ [sk2 DEC E JR NZ [sk3 BIT 0,L JR NZ #OCB8	key+reverse	0C6C 3E 44 0C6E D3 10 0C70 CD EE 0C 0C73 1E E1 0C75 16 01 0C77 CD BF 0C 0C7A 38 26 0C7C 20 F5 0C7E 1D 0C7F 20 F6 0C81 CB 45 0C83 20 33 0C85 16 27 0C87 CD BF 0C 0C8A 38 16 0C8C 28 F7 0C8E 26 64 0C90 1E 44 0C92 CD DF 0A 0C95 20 10 0C97 21 6E 60 0C9A 35 0C9B 20 CF 0C9D 3E 00 0C9F 32 17 60 0CA2 26 65 0CA4 CD 58 06 0CA7 3A 17 60 0CAA FE 45 0CAC C0 0CAD 3E 42 0CAF 32 17 60 0CB2 21 60 60 0CB5 CB E6 0CB7 C9	LD A,44 OUT 10 CALL OCEE LD E,E1 LD D,01 CALL OCBF JR C OCA2 JR NZ OC73 DEC E JR NZ OC77 BIT 0,L JR NZ OCB8 LD D,27 CALL OCBF JR C OCA2 JR Z OC85 LD H,64 LD E,44 CALL OADF JR NZ OCA7 LD HL,606E DEC (HL) JR NZ OC6C LD A,00 LD (caserror),A
[sk5	LD A,00			
[sk6	LD H,65 CALL offmotor			
[sk7	LD A,(caserror) CP "E RET NZ LD A,"B LD (caserror),A LD HL,stacas SET 4,(HL) RET	einde tape? begin tape		

#OCB8	CALL 261msec JR NZ [sk7 JR [sk5	OCB8 CD E4 0A OCBB 20 EA OCBD 18 DE	CALL OAE4 JR NZ OCA7 JR OC9D
\$OCBF	LD C,08	OCBF OE 08	LD C,08
#OCC1	XOR 80	OCC1 EE 80	XOR 80
	LD H,A	OCC3 67	LD H,A
\$OCC4	LD B,00	OCC4 06 00	LD B,00
%OCC6	IN 20	OCC6 DB 20	IN 20
	AND B0	OCC8 E6 B0	AND B0
	CP H	OCCA BC	CP H
	JR Z #OCDD	OCCB 28 10	JR Z OCDD
	DJNZ %OCC6	OCCD 10 F7	DJNZ OCC6
	AND 30	OCCF E6 30	AND 30
	CP 20	OCD1 FE 20	CP 20
	JR NZ #OCE2	OCD3 20 0D	JR NZ OCE2
	DEC D	OCD5 15	DEC D
	JR NZ \$OCC4	OCD6 20 EC	JR NZ OCC4
	LD A,H	OCD8 7C	LD A,H
	XOR 80	OCD9 EE 80	XOR 80
	INC D	OCDB 14	INC D
	RET	OCDC C9	RET
	reset Carry		
	D]1, Non Zero		
#OCDD	LD A,H	OCDD 7C	LD A,H
	DEC C	OCDE 0D	DEC C
	JR NZ #OCC1	OCDF 20 E0	JR NZ OCC1
	RET	OCE1 C9	RET
#OCE2	OR A	OCE2 B7	OR A
	LD A,"E	OCE3 3E 45	LD A,45
	JR Z \$OCE9	OCE5 28 02	JR Z OCE9
	LD A,"A	OCE7 3E 41	LD A,41
\$OCE9	LD (caserror),A	OCE9 32 17 60	LD (6017),A
	SCF	OCEC 37	SCF
	RET	OCED C9	RET
%OCEE	IN 20	OCEE DB 20	IN 20
	AND 80	OCFO E6 80	AND 80
	SET 5,A	OCF2 CB EF	SET 5,A
	RET	OCF4 C9	RET

ORG OCF5

scr-type EQU 01BE P] scherm
stat.key EQU 0489 carry if stop
readkey EQU 049B key] A reg.

; (baudrate)] baudrate:
; 00 - 80 2400
; 01 - 81 1200
; 03 - 83 600
; 07 - 87 300
; 0F - 8F 150
; 1F - 9F 75
; als bit 7 van (baudrate) is gezet
; wordt elk karakter ongewijzigd geprint

baudrate EQU 6016
actual EQU 601E actual key
prtabel EQU 605E printer tabel

; voer string aan printer routine
; HL op begin string, einde string: 00
; bij entry staat A op 00

prstring	LD C,(HL)	karakter] C	OCF5 4E	LD C,(HL)
	SUB A	A was nul	OCF6 97	SUB A
	OR C	C ook nul?	OCF7 B1	OR C
	RET Z	klaar !	OCF8 C8	RET Z
	INC HL	volgende kar	OCF9 23	INC HL
	CALL printer		OCFA CD 5D 0E	CALL 0E5D
	RET C	printer error	OCFD D8	RET C
	JR prstring	volgend kar.	OCFE 18 F5	JR OCF5

; printscherm

; HL : startadres eerste regel
; A : aantal te printen regels
; C : maximale lengte regel
; Als C 00 is, wordt alleen carriage-
return linefeed geprint

Algemeen: D register bevat vlaggen:
bit 7 : grafische printer
bit 4 : grafisch karakter
bit 0 : stop toets in buffer

Bit 7 van karakter betekent:
onderstrepen.

prscreen	EX AF		EX AF
SUB A	wordt nul	OD00 08	SUB A
LD (actual),A	op nul gezet	OD01 97	LD (601E),A
CALL prready?	klaar?	OD02 32 1E 60	CALL ODCC
RET NZ	printer error	OD05 CD CC 0D	RET NZ
EX AF		OD08 C0	EX AF
AND A		OD09 08	AND A
JR Z prstring	nul regels	OD0A A7	JR Z 0CF5
CALL P-screen	"P" op scherm	OD0B 28 E8	CALL ODE0
PUSH HL	alle register	OD0D CD EO 0D	PUSH HL
PUSH DE	naar stack	OD10 E5	PUSH DE
LD D,00	geen STOP	OD11 D5	LD D,00
nextline	PUSH BC	OD12 16 00	PUSH BC
	PUSH AF	OD14 C5	PUSH AF
		OD15 F5	

; bit 0 in D gezet betekent stop-toets

	BIT 0,D	stop ?	OD16 CB 42	BIT 0,D
	JR Z nostop	dus niet	OD18 28 05	JR Z 0D1F
	CALL inv.cursor	poets cursor	OD1A CD DA 0D	CALL ODDA
	JR STOP!		OD1D 18 08	JR 0D27
*nostop	CALL stat.key		OD1F CD 89 04	CALL 0489
	JR NC NOSTOP	STOP ?	OD22 30 0D	JR NC 0D31
	CALL haaltoets	verwyder STOP	OD24 CD D7 0D	CALL ODD7
STOP!	CALL haaltoets	wacht op key	OD27 CD D7 0D	CALL ODD7
	JR NC NOSTOP	geen stop	OD2A 30 05	JR NC 0D31
	LD (actual),A		OD2C 32 1E 60	LD (601E),A
	JR endline	2 maal STOP	OD2F 18 76	JR ODA7
NOSTOP	PUSH HL		OD31 E5	PUSH HL
	LD A,(baudrate)		OD32 3A 16 60	LD A,(6016)
	AND 80	isoleer bit 7	OD35 E6 80	AND 80
	LD D,A		OD37 57	LD D,A

; hoogste bit geeft aan of er grafisch
; geprint moet worden. Dan wordt de
; waarde onveranderd aan de printer
; doorgegeven (bit 7 in D). Als dit zo
; is wordt de komplette regel doorgegeven
; Is dit niet zo, dan wordt er eerst uit
; gezocht hoeveel tekens aan het einde
; niet hoeven worden geprint.
; (spatie of 00).

	JR NZ graf	grafisch	0D38 20 12	JR NZ 0D4C
	LD B,00	C lengte	0D3A 06 00	LD B,00
	ADD HL,BC	einde regel	0D3C 09	ADD HL,BC
	INC C		0D3D 0C	INC C
space?	DEC C	C: max lengte	0D3E 0D	DEC C
	JR Z CR-LF	begin regel ?	0D3F 28 50	JR Z 0D91
	DEC HL		0D41 2B	DEC HL
	LD A,(HL)	kar! ophalen	0D42 7E	LD A,(HL)
	AND A	nul ?	0D43 A7	AND A
	JR Z space?		0D44 28 F8	JR Z 0D3E
	CP 20	spatie ?	0D46 FE 20	CP 20
	JR Z space?		0D48 28 F4	JR Z 0D3E
	POP HL	herstel begin	0D4A E1	POP HL
	PUSH HL		0D4B E5	PUSH HL

; C bevat nu aantal te printen karakters

graf	LD B,C	naar B	0D4C 41	LD B,C
prgraf	LD C,(HL)	kar halen	0D4D 4E	LD C,(HL)
	CALL inv.cursor	cursor]scherm	0D4E CD DA 0D	CALL ODDA
	LD A,C		0D51 79	LD A,C
	CP 1D	nu grafisch ?	0D52 FE 1D	CP 1D
	JR NZ nograf	nee	0D54 20 02	JR NZ 0D58
	SET 0,D	grafisch bit	0D56 CB C2	SET 0,D
nograf	BIT 7,D	graf printer?	0D58 CB 7A	BIT 7,D
	JR Z normal		0D5A 28 05	JR Z 0D61

; teletext grafische printer

	CALL printer		0D5C CD 5D 0E	CALL 0E5D
	JR prigraf		0D5F 18 27	JR 0D88

; normale printer

normal	CP 98	conceal disp.	0D61 FE 98	CP 98
	JR Z CR-LF		0D63 28 2C	JR Z 0D91
	AND 7F	reset bit 8	0D65 E6 7F	AND 7F
	JR Z kar]pri	nul ?	0D67 28 1C	JR Z 0D85
	CP 09	TAB	0D69 FE 09	CP 09
	JR NC norl	dus kleur	0D6B 30 02	JR NC 0D6F
	RES 4,D	no grafics	0D6D CB A2	RES 4,D
norl	CP 11	start graphic	0D6F FE 11	CP 11
	JR C kar]pri		0D71 38 12	JR C 0D85

; grafics loopt van 10 tot 17

	CP 18	inside range?	0D73 FE 18	CP 18
	JR NC nor2		0D75 30 02	JR NC 0D79
	SET 4,D	grafics :	0D77 CB E2	SET 4,D
nor2	BIT 4,D	? grafics	0D79 CB 62	BIT 4,D
	JR Z kar]pri		0D7B 28 08	JR Z 0D85
	SUB 40		0D7D D6 40	SUB 40
	CP 20	non printable	0D7F FE 20	CP 20
	JR C kar]pri	wordt spatie	0D81 38 02	JR C 0D85
	LD C,20		0D83 OE 20	LD C,20

; alle karakters via tabel printer

	kar]pri	CALL prkar	naar printer	0D85 CD F2 0D	CALL ODF2
	prigraf	CALL inv.cursor		0D88 CD DA 0D	CALL ODDA
		JR C abnormal	Carry = error	0D8B 38 19	JR C ODA6
		INC HL		0D8D 23	INC HL
		DJNZ prgraf	volgend kar.	0D8E 10 BD	DJNZ OD4D
		DEC HL		0D90 2B	DEC HL
CR-LF		BIT 7,D	graf. printen	0D91 CB 7A	BIT 7,D
		JR NZ abnormal	nee	0D93 20 11	JR NZ ODA6
		LD C,0D	carriage retn	0D95 0E 0D	LD C,0D
		CALL printer		0D97 CD 5D 0E	CALL OE5D
		LD C,0A	linefeed	0D9A 0E 0A	LD C,0A
		LD A,(HL)		0D9C 7E	LD A,(HL)

; karakter 98 is cursor op 18 "conceal
; display". Dit resulteert in formfeed

	CP 98	vervang 98	0D9D FE 98	CP 98
	JR NZ noconceal	door 0C:	0D9F 20 02	JR NZ ODA3
	LD C,0C	formfeed	0DA1 0E 0C	LD C,0C
noconceal	CALL printer		0DA3 CD 5D 0E	CALL OE5D
abnormal	POP HL	iets fout ?	0DA6 E1	POP HL
endline	POP BC	oorspr. AF	0DA7 C1	POP BC
	LD A,B	aantal regels	0DA8 78	LD A,B
	POP BC		0DA9 C1	POP BC
	JR C klaar	printer error	0DAA 38 0A	JR C ODB6
	PUSH DE		0DAC D5	PUSH DE
	LD DE,0050	bereken begin	0DAD 11 50 00	LD DE,0050
	ADD HL,DE	next line	0DB0 19	ADD HL,DE
	POP DE		0DB1 D1	POP DE
	DEC A	regel teller	0DB2 3D	DEC A
	JP NZ nextline	nog meer?	0DB3 C2 14 0D	JP NZ OD14
klaar	POP DE	herstel DE	0DB6 D1	POP DE
	CALL wisP	wis P, scherm	0DB7 CD EC 0D	CALL ODEC
	POP HL	herstel HL	0DBA E1	POP HL
	RET		0DBB C9	RET

prwait	LD B,06	counter wacht	ODBC 06 06	LD B,06
prwl	CALL prready?	op printer	ODBE CD CC 0D	CALL ODCC
	JR Z prw2	hij is er!	ODC1 28 07	JR Z ODCA
	INC DE		ODC3 13	INC DE
	LD A,D	blijf tellen	ODC4 7A	LD A,D
	OR E		ODC5 B3	OR E
	JR NZ prwl	niet klaar	ODC6 20 F6	JR NZ ODBE
	DJNZ prwl	en opnieuw	ODC8 10 F4	DJNZ ODBE
rw2	CCF	fout in carry	ODCA 3F	CCF
	RET	flag	ODCB C9	RET

prready?	SCF		ODCC 37	SCF
	IN 20	is de printer	ODCD DB 20	IN 20
	BIT 1,A	beschikbaar?	ODCF CB 4F	BIT 1,A
	RET NZ	not ready	ODD1 C0	RET NZ
	IN 20	waarom nu nog	ODD2 DB 20	IN 20
	BIT 1,A	eens?	ODD4 CB 4F	BIT 1,A
	RET		ODD6 C9	RET

aaltoets CALL readkey

✓ High bit (HL) komt invers in carry
 en wordt weer terug geschoven. Als
 ; HL een scherm adres is wordt op deze
 ; manier de cursor gezet, resp. gewist
 afhankelijk of hij er al stond.

inv.curs	RL (HL)		ODD7 CD 9B 04	CALL 049B
	CCF		ODDA CB 16	RL (HL)
	RR (HL)		ODDC 3F	CCF
	RET		ODDD CB 1E	RR (HL)
			ODDF C9	RET

scr	PUSH DE		ODEO D5	PUSH DE
	PUSH BC		ODE1 C5	PUSH BC
	LD B,"P	zet "P	ODE2 06 50	LD B,50
:reen	LD E,03	met offset op	ODE4 1E 03	LD E,03
	CALL scr-type	scherm	ODE6 CD BE 01	CALL 01BE
	POP BC		ODE9 C1	POP BC
	POP DE		ODEA D1	POP DE
	RET		ODEB C9	RET

sp	PUSH DE	haal de "P	ODEC D5	PUSH DE
	PUSH BC	van scherm	ODED C5	PUSH BC
	LD B,00		ODEE 06 00	LD B,00
	JR screen		ODFO 18 F2	JR ODE4

karakter in C via vertaal tabel naar
 ; printer. Registers bewaard.

kar	PUSH HL	save register	ODF2 E5	PUSH HL
	PUSH DE		ODF3 D5	PUSH DE
	PUSH BC		ODF4 C5	PUSH BC
	CALL pr-kar		ODF5 CD FC 0D	CALL 0DFC
	POP BC	restore reg.	ODF8 C1	POP BC
	POP DE		ODF9 D1	POP DE
	POP HL		ODFA E1	POP HL
	RET		ODFB C9	RET

; stuur karakter naar printer. Kijk
 ; eerst welk type printer er is en stuur
 ; afhankelijk daarvan het karakter door
 ; een vertaal tabel voor daisy of matrix
 ; daisy kan b.v. backspace + underline
 ; matrix kan grafisch.
 ; Carry na afloop betekent printer error

pr-kar	LD HL,(prtabel)	vertaal tabel
	LD D,00	
	LD E,(HL)	lengte in E
	IN 20	
	AND 04	daisy wheel?
	LD A,C	
	RES 7,C	del underline
	JR Z matrix	geen daisy
	BIT 7,A	underline ?
	JR Z pr-1	nee
	PUSH BC	
	LD C,5F	underline!
	CALL printer	
	LD C,08	backspace
	CALL NC printer	
	POP BC	
	RET C	printer error

ODFC 2A 5E 60	LD HL,(605E)
ODFF 16 00	LD D,00
OE01 5E	LD E,(HL)
OE02 DB 20	IN 20
OE04 E6 04	AND 04
OE06 79	LD A,C
OE07 CB B9	RES 7,C
OE09 28 32	JR Z OE3D
OE0B CB 7F	BIT 7,A
OE0D 28 0D	JR Z OE1C
OE0F C5	PUSH BC
OE10 0E 5F	LD C,5F
OE12 CD 5D 0E	CALL OE5D
OE15 0E 08	LD C,08
OE17 D4 5D 0E	CALL NC OE5D
OE1A C1	POP BC
OE1B D8	RET C

; daisy vertaal tabel is dubbel. Voor
 ; elk karakter in de tabel zijn er twee
 ; in de vertaling. Bijv om onderstrepen
 ; mogelijk te maken.

pr-1	ADD HL,DE	start daisy
	ADD HL,DE	vertaal tabel
	INC HL	
	LD E,(HL)	lengte tabel
	INC E	
pr-2	INC HL	vergelijk met
	DEC E	vertaal tabel
	JR Z mat0	einde tabel?
	LD A,(HL)	kar in tabel
	CP C	= te printen?
	INC HL	
	LD B,(HL)	eerste kar
	INC HL	
	JR NZ pr-2	try again
	PUSH BC	
	LD C,(HL)	tweede kar
	CALL escape?	
	LD C,08	backspace
	CALL NC printer	naar printer
	POP BC	
	LD C,B	eerste kar
	CALL NC escape?	
	RET	

OE1C 19	ADD HL,DE
OE1D 19	ADD HL,DE
OE1E 23	INC HL
OE1F 5E	LD E,(HL)
OE20 1C	INC E
OE21 23	INC HL
OE22 1D	DEC E
OE23 28 17	JR Z OE3C
OE25 7E	LD A,(HL)
OE26 B9	CP C
OE27 23	INC HL
OE28 46	LD B,(HL)
OE29 23	INC HL
OE2A 20 F5	JR NZ OE21
OE2C C5	PUSH BC
OE2D 4E	LD C,(HL)
OE2E CD 48 0E	CALL OE48
OE31 0E 08	LD C,08
OE33 D4 5D 0E	CALL NC OE5D
OE36 C1	POP BC
OE37 48	LD C,B
OE38 D4 48 0E	CALL NC OE48
OE3B C9	RET 50

mat0	LD E,(HL)	vertaal tabel	0E3C 5E	LD E,(HL)
matrix	INC E	lengte in E	0E3D 1C	INC E
mat1	INC HL		0E3E 23	INC HL
	DEC E		0E3F 1D	DEC E
	JR Z prnormal	einde tabel	0E40 28 13	JR Z 0E55
	LD A,(HL)	kar in tabel	0E42 7E	LD A,(HL)
	INC HL		0E43 23	INC HL
	CP C	= te printen?	0E44 B9	CP C
	JR NZ mat1	niet gelijk	0E45 20 F7	JR NZ 0E3E
	LD C,(HL)		0E47 4E	LD C,(HL)
; als bit 7 in de printer tabel is gezet				
; wordt er eerst een ESCAPE naar de				
; printer gestuurd en dan het karakter				
escape?	BIT 7,C	escape nodig?	0E48 CB 79	BIT 7,C
	RES 7,C		0E4A CB B9	RES 7,C
	PUSH BC		0E4C C5	PUSH BC
	LD C,1B	escape	0E4D OE 1B	LD C,1B
	CALL NZ printer	NZ] printer	0E4F C4 5D 0E	CALL NZ 0E5D
	POP BC	karakter	0E52 C1	POP BC
	JR printer	naar printer	0E53 18 08	JR 0E5D
; alleen printable karakters printen				
prnormal	RES 7,C	maskeer high	0E55 CB B9	RES 7,C
	LD A,20	spatie	0E57 3E 20	LD A,20
	CP C	printable ?	0E59 B9	CP C
	JR C printer	ja !	0E5A 38 01	JR C 0E5D
	LD C,A	spatie in C	0E5C 4F	LD C,A
; stuur karakter in C register naar				
; printer, registers blijven intakt				
; accumulator wordt gewijzigd				
printer	PUSH BC		0E5D C5	PUSH BC
	PUSH DE		0E5E D5	PUSH DE
	CALL prwait	wacht printer	0E5F CD BC OD	CALL ODBC
	JR C exitpr	exit + error	0E62 38 29	JR C 0E8D
	LD D,0A	0A bits	0E64 16 0A	LD D,0A
	DI		0E66 F3	DI
pri1	LD A,C0		0E67 3E C0	LD A,C0

bit 7, poort 10 is printer data
 prwait verlaat routine met NC
 en zorgt zo voor startbit

	JR NC pri2	data in carry	OE69 30 02	; JR NC OE6D
	RES 7,A		OE6B CB BF	RES 7,A
pri2	OUT 10		OE6D D3 10	OUT 10
	LD B,49	eerste delay	OE6F 06 49	LD B,49
	LD A,(baudrate)		OE71 3A 16 60	LD A,(6016)
	RES 7,A	graphics flag	OE74 CB BF	RES 7,A
	INC A		OE76 3C	INC A
pri3	DJNZ pri3	init delay	OE77 10 FE	DJNZ OE77
	LD B,4E	fast delay	OE79 06 4E	LD B,4E
	DEC A	slow delay	OE7B 3D	DEC A
	ADD 00	dummy	OE7C C6 00	ADD 00
	JR NZ pri3	lengte: A	OE7E 20 F7	JR NZ OE77
	SCF		OE80 37	SCF

; schuif laagste bit van C in carry

RR C	schuif bit	OE81 CB 19	RR C
------	------------	------------	------

; automatisch schuiven de stop bits in C
; veroorzaakt door SCF 2 regels hoger

	DEC D	aantal bits	OE83 15	; DEC D
	JR NZ pril	0A bits klaar	OE84 20 E1	JR NZ OE67
	LD B,A		OE86 47	LD B,A
pri4	DJNZ pri4	delay	OE87 10 FE	DJNZ OE87
	CALL prwait	wacht printer	OE89 CD BC OD	CALL ODBC
	EI	tot alles	OE8C FB	EI
exitpr	POP DE	klaar is.	OE8D D1	POP DE
	POP BC		OE8E C1	POP BC
	RET		OE8F C9	RET

ORG 0E90

CPM/sect EQU 046A lees sector

interruptvectoren van CTC

INT1 EQU 6020 interrupt
INT2 EQU 6022 vectoren
disk! EQU 605D disk aanwezig

; default waardes flop in RAM

descrip EQU 6070 default
RDTRACK EQU 6072 lees track
TRACKNR EQU 6075 in readtrack
STEP EQU 607C zoek track
TRA-1 EQU 607F tracknr - 1
CALIBRATE EQU 6080 kop > track 1
SPECIFY EQU 6083 timing FDC
status EQU 6087 ~~T/M 8D~~ controller
spsave EQU 608E

~~stepper~~ 6073 command 6074 flopkop ~~sector~~
6085 stepHUT 6076 head ~~echo~~
; FDC staat voor Floppy Disk Controller
6086 HLT 607A gap leng ~~me~~
6081 calcom 6082 flop~~cal~~ 607E flop~~soek~~ ~~sector~~
; poorten voor de floppy handling:

; poort 8C: input status FDC
;
; poort 8D: bit 0 op poort 90 selecteert
; 0: data I/O disk-geheugen
; 1: input: lees status registers
; FDC
; output: geef FDC opdracht

; poort 90 is een output poort voor de
; controle signalen van de floppy
; bit 0: FDC enable
; 0: data transport
; 1: registers schrijven-lezen
; bit 1: terminal count FDC
; bit 2: FDC reset
; bit 3: motor on
; bit 4: disable select FDC
; 0: normaal
; 1: noselect
; N.B. bit 4 wordt op de normale 48-k
; plank niet gebruikt, wel de in
; ontwikkeling zijnde plank van P2C2

entry	DI LD (spsave),SP LD A,01 LD (disk!),A LD HL,default LD DE,descrip LD BC,0017 LDIR CALL init CALL on-motor	return adres opbergen disk aanwezig disk descrip van ROM naar RAM copieren kop-0,timing	OE90 F3 OE91 ED 73 8E 60 OE95 3E 01 OE97 32 5D 60 OE9A 21 E8 0F OE9D 11 70 60 OEAO 01 17 00 OEAB CD 19 OF OEAE 2A 70 60 OEB1 01 00 10 OEB4 09 OEB5 22 70 60 OEB8 21 75 60 OEBB 3E 02 OEBD BE OEBE 28 06 OEC0 34 OEC1 CD 7D OF OEC4 18 E5 OEC6 21 00 E0 OEC9 3E F3 OECB BE OECCE 28 04 OECE AF OECF 32 5D 60 OED2 F3 OED3 3E 03 OED5 D3 88 OED7 AF OED8 D3 90 OEDA ED 7B 8E 60 OEDE AF OEDF D3 94 OEE1 C9	DI LD (608E),S LD A,01 LD (605D),A LD HL,0FE8 LD DE,6070 LD BC,0017 LDIR CALL OEE2 CALL OF88 CALL OF19 LD HL,(6070) LD BC,1000 ADD HL,BC LD (6070),H LD HL,6075 LD A,02 CP (HL) JR Z OEC6 INC (HL) CALL OF7D JR OEAB LD HL,E000 LD A,F3 CP (HL) JR Z OED2 XOR A LD (605D),A DI LD A,03 OUT 88 XOR A OUT 90 LD SP,(spsave) XOR A OUT 94 RET
ent1	CALL leestrack LD HL,(descrip) LD BC,1000 ADD HL,BC LD (descrip),HL LD HL,TRACKNR LD A,02 CP (HL) JR Z ent2 INC (HL) CALL gotrack JR ent1 LD HL,E000 LD A,F3 CP (HL) JR Z klaar XOR A . LD (disk!),A	default E000 tweede track naar F000 en invullen. track 2 al gehad ? max 2 track's inc tracknr naar track en inlezen default lees check byte boot disk? boot mislukt wel CTC, geen system flop		
ent2	LD HL,E000 LD A,F3 CP (HL) JR Z klaar XOR A . LD (disk!),A			
klaar	DI LD A,03 OUT 88 XOR A OUT 90 LD SP,(spsave) XOR A OUT 94 RET	motor uit herstel stack selecteer mem bank 0		

; initialiseer de FDC, zet kop op
; eerste track, zet de interrupt vectors
; wacht 350 msec en verwijder interrupt

init IM 2
DI
LD A,04 floppy reset
OUT 90
CALL delay 350 msec
CALL reti verwijder int
CALL 2-result 2-byte result
CALL zet-int. vectoren
LD HL,SPECIFY zet de timing
CALL voerfdc voor de FDC
CALL Calibrate kop > track 1
RET

OEE2 ED 5E IM 2
OEE4 F3 DI
OEE5 3E 04 LD A,04
OEE7 D3 90 OUT 90
OEE9 CD FF 0E CALL OEFF
OEEC CD E6 0F CALL OFE6
OEEF CD D9 0F CALL OFD9
OEF2 CD B7 0F CALL OFB7
OEF5 21 83 60 LD HL,6083
OEF8 CD A5 0F CALL OFA5
OEFB CD 08 0F CALL OF08
OFEF C9 RET

; wachtlus 350 millisec

delay^{6ns}
dell ↓
f8c LD BC,0000
DJNZ dell
DEC C
JR NZ dell klaar ?
RET

OF0F 01 00 00 LD BC,0000
OF02 10 FE DJNZ OF02
OF04 0D DEC C
OF05 20 FB JR NZ OF02
OF07 C9 RET

; zet kop op eerste track, met resul-
; taten status register wordt niets
; gedaan.

Calibrate LD HL,dummy
LD (INT1),HL
LD HL,CALIBRATE naar track 0
CALL voerfdc
HALT wacht

OF08 21 D6 0F LD HL,OFD6
OF0B 22 20 60 LD (6020),HL
OF0E 21 80 60 LD HL,6080
OF11 CD A5 0F CALL OFA5
OF14 76 HALT

; zodra de kop op zijn bestemming is
; wordt een interrupt gegenereerd
HALT wacht hierop
CALL 2-result 2-status reg.
RET
DEC tec 2x!
BIT 4(tec)
JR NZ
RET

OF15 CD D9 0F CALL OFD9
OF18 C9 RET

leestrack	LD IY,RDTRACK		OF19 FD 21 72 60	LD IY,6072
	LD HL,vervang	int routine	OF1D 21 5A 0F	LD HL,0F5A
	LD (INT1),HL	naar vector	OF20 22 20 60	LD (6020),HL
	XOR A		OF23 AF	XOR A
	INC A	A > 01, zero	OF24 3C	INC A
	LD (IY+05),A	sector nummer	OF25 FD 77 05	LD (IY+05),A
	LD HL,RDTRACK		OF28 21 72 60	LD HL,6072
	CALL voerfdc		OF2B CD A5 0F	CALL OFA5
	LD A,C5		OF2E 3E C5	LD A,C5
	OUT 89		OF30 D3 89	OUT 89
	LD A,01		OF32 3E 01	LD A,01
	OUT 89		OF34 D3 89	OUT 89
	LD HL,(descrip)	default E000	OF36 2A 70 60	LD HL,(6070)
	LD C,8D	poort nummer	OF39 0E 8D	LD C,8D
; register	I/O FDC, floppy, motor on		OF3B 3E 0D	LD A,0D
	LD A,0D		OF3D D3 90	OUT 90
	OUT 90		OF3F 1E 10	LD E,10
	LD E,10		OF41 3E 01	LD A,01
	LD A,01	selecteer mem	OF43 D3 94	OUT 94
	OUT 94	bank 1	OF45 DB 90	IN 90
leel	IN 90 DMA-Ra?	FDC ready?	OF47 1F	RRA
	RRA	schuift in	OF48 D2 45 0F	JP NC OF45
	JP NC leel	carry	OF4B ED A2	INI
	INI	zet byte in	OF4D C3 45 0F	JP OF45
	JP leel	memory		
; na 1 track lezen via de interrupt				
; afgeleid naar de aanroep routine				
wat gebeurt hier?			OF50 1D	DEC E
	DEC E		OF51 C2 6A 04	JP NZ 046A
	JP NZ CPM/sec		OF54 3E 0E	LD A,0E
	LD A,0E		OF56 D3 90	OUT 90
	OUT 90			
; wacht op interrupt ?? ik weet het niet				
wacht	JR wacht	eindeloze lus.	OF58 18 FE	JK OF58

; een zeer eigenzinnige interrupt
; routine, hij verwijdert het oorspron-
; kelijke return adres van het stack en
; zet er iets anders voor terug

vervang POP HL
LD HL,7-result installeer
PUSH HL routine op
dummy EI stack en
 RETI ga er heen

OF5A E1
OF5B 21 62 OF
OF5E E5
OF5F FB
OF60 ED 4D

POP HL
LD HL,OF62
PUSH HL
EI
RETI

; lees 7 status bytes uit van FDC

x 7-result LD A,03
OUT 89
LD B,07 lees 7 status
CALL result registers
RET

OF62 3E 03
OF64 D3 89
OF66 06 07
OF68 CD 90 OF
OF6B C9

LD A,03
OUT 89
LD B,07
CALL OF90
RET

; geef kop opdracht naar track te gaan
; gespecificeerd in TRA-1 en wacht
; tot hij er is, geen reactie op inhoud
; status registers

→
track LD HL,dummy geen aktie na
LD (INT1),HL interrupt.
LD HL,STEP opdracht
CALL voerfdc voor FDC.
HALT wacht op
x | CALL 2-result interrupt.
RET
result

OF6C 21 D6 OF
OF6F 22 20 60
OF72 21 7C 60
OF75 CD A5 OF
OF78 76
OF79 CD D9 OF
OF7C C9

LD HL,OFD6
LD (6020),HL
LD HL,607C
CALL OFA5
HALT
CALL OFD9
RET

; breng de twee bytes voor bepaling
; van track op orde en ga erheen.

gotrack LD A,(TRACKNR)
DEC A
LD (TRA-1),A
CALL track naar track
RET

OF7D 3A 75 60
OF80 3D
OF81 32 7F 60
OF84 CD 6C OF
OF87 C9

LD A,(6075)
DEC A
LD (607F),A
CALL OF6C
RET

; aktiveer de motor en wacht 350 msec
; tot hij op toeren is

on-motor LD A,0C
; floppy, motor on
OUT 90 motor on
delay **x** CALL delay
RET

OF88 3E 0C
OF8A D3 90
OF8C CD FF 0E
OF8F C9

LD A,0C
OUT 90
CALL OEFF
RET

; lees de status registers van de FDC
; aantal bytes in B register
; bestemming naar (HL)

Result-7 LD B,7
result LD A,03
OUT 89
LD HL,status
; floppy, motor on
LD A,0C
OUT 90
res-1 CALL ready FDC klaar?
IN 8D status reg.
LD (HL),A opbergen
INC HL
DJNZ res-1 nog meer?
RET

OF90 3E 03
OF92 D3 89
OF94 21 87 60
OF97 3E 0C
OF99 D3 90
OF9B CD B0 OF
OF9E DB 8D
OFA0 77
OFA1 23
OFA2 10 F7
OFA4 C9

LD A,03
OUT 89
LD HL,6087
LD A,0C
OUT 90
CALL OFB0
IN 8D
LD (HL),A
INC HL
DJNZ OF9B
RET

; geef FDC instrukties

voerfdc LD B,(HL) aantal bytes
voe-1 INC HL
CALL ready
LD A,(HL) next byte
OUT 8D naar FDC
DJNZ voe-1 volgende
RET

OFA5 46
OFA6 23
OFA7 CD B0 OF
OFAA 7E
OFAB D3 8D
OFAD 10 F7
OFAF C9

LD B,(HL)
INC HL
CALL OFB0
LD A,(HL)
OUT 8D
DJNZ OFA6
RET

; is de FDC klaar, of in staat om
; opdracht te accepteren.

ready IN 8C lees status
BIT 7,A klaar ?
JR Z ready nog niet
RET

OFB0 DB 8C
OFB2 CB 7F
OFB4 28 FA
OFB6 C9

IN 8C
BIT 7,A
JR Z OFB0
RET

; zet de interrupt adressen en voer
; de CTC en I register met vector

zet-int. LD HL,INT1
LD A,H HL:=6020
LD I,A 60 in I reg.
LD A,L 20 naar CTC
OUT 88
LD A,D5
OUT 88
LD A,01
OUT 88
(LD HL,dummy zet interrupt
(LD (INT1),HL adressen neer
(LD HL,klaar
(LD (INT2),HL
EI
RET

OFB7 21 20 60
OFBA 7C
OFBB ED 47
OFBD 7D
OFBE D3 88
OFC0 3E D5
OFC2 D3 88
OFC4 3E 01
OFC6 D3 88
OFC8 21 D6 0F
OFCB 22 20 60
OFCE 21 D2 0E
OFD1 22 22 60
OFD4 FB
OFD5 C9

LD HL,6020
LD A,H
LD I,A
LD A,L
OUT 88
LD A,D5
OUT 88
LD A,01
OUT 88
LD HL,0FD6
LD (6020),HL
LD HL,0ED2
LD (6022),HL
EI
RET

; dummy interrupt vector, als er een
; interrupt komt wordt er niets gedaan
; alleen de interrupt van de CTC wordt
; uitgezet m.b.v. RETI

Xdummy EI
RETI

OFD6 FB
OFD7 ED 4D

EI
RETI

lees twee status bytes van FDC
bestemming naar (HL)

X2-result LD A,08
; sense status FDC
OUT 8D
X? CALL ready
LD B,02 lees 2 status
CALL result registers uit
| RET

OFD9 3E 08
OFDB D3 8D
OFDD CD B0 OF
OFE0 06 02
OFE2 CD 90 OF
OFE5 C9

LD A,08
OUT 8D
CALL OFB0
LD B,02
CALL OF90
RET

; reti is nodig om interrupt van CTC
; weg te halen

Xreti RETI

OFE6 ED 4D

RETI

```

; Flop descriptor
; gebruikt door boot van PDOS
; default waardes voor disk controller

default WORD E000      bestemming
rdtrack  BYTE 09        aantal bytes
          BYTE 42        read track
          BYTE 01 H       kop 0 drive 1
nrtrack   BYTE 01 = 11   track nummer
          BYTE 00 - HEN  kop nummer
sectornr  BYTE 01       sector nummer
          BYTE 01       bytes/sector
          BYTE 10 Fc     sector/track
          BYTE 0E Gpl    gaplengte
          BYTE 00 D1     data lengte

step      BYTE 03        aantal bytes
          BYTE 0F        seek track
          BYTE 01       kop 0 drive 1
tra-1     BYTE 01       track nummer

calibrate BYTE 02      aantal bytes
          BYTE 07        kop > track 0
          BYTE 01       drive 1

specify  BYTE 03      aantal bytes
          BYTE 03        timing FDC

; nb, voor de drives die via de P2C2
; geleverd zijn kan de volgende steprate
; aanzienlijk hoger zijn. In dat geval
; mag hier een E0 staan. Het gevolg is
; dat de drive veel rustiger loopt.

          BYTE 60        step rate
          BYTE 34        head load

          BYTE FF

```