

# TABELLEN

# OVERZICHTEN

## Z-80 CODE

8H	00	10	20
0	NOP	DJNZ ..	JR NZ ..
1	LD BC,....	LD DE,....	LD HL,....
2	LD (BC),A	LD (DE),A	LD (....),HL
3	INC BC	INC DE	INC HL
4	INC B	INC D	INC H
5	DEC B	DEC D	DEC H
6	LD B,...	LD D,...	LD H,...
7	RLCA	RLA	DAA
8	EX AF	JR ..	JR Z ..
9	ADD HL,BC	ADD HL,DE	ADD HL,HL
A	LD A,(BC)	LD A,(DE)	LD HL,(....)
B	DEC BC	DEC DE	DEC HL
C	INC C	INC E	INC L
D	DEC C	DEC E	DEC L
E	LD C,...	LD E,...	LD L,...
F	RRC A	RRA	CPL

## HEXADECIMAAL - DECIMAAL

8H	000	100	200	300	400	500	600	700	800	900	A00	B00
00	00	256	512	768	1024	1280	1536	1792	2048	2304	2560	2816
10	16	272	528	784	1040	1296	1552	1808	2064	2320	2576	2832
20	32	288	544	800	1056	1312	1568	1824	2080	2336	2592	2848
30	48	304	560	816	1072	1328	1584	1840	2096	2352	2608	2864
40	64	320	576	832	1088	1344	1600	1856	2112	2368	2624	2880
50	80	336	592	848	1104	1360	1616	1872	2128	2384	2640	2896
60	96	352	608	864	1120	1376	1632	1888	2144	2400	2656	2912
70	112	368	624	880	1136	1392	1648	1904	2160	2416	2672	2928
80	128	384	640	896	1152	1408	1664	1920	2176	2432	2688	2944
90	144	400	656	912	1168	1424	1680	1936	2192	2448	2704	2960
AO	160	416	672	928	1184	1440	1696	1952	2208	2464	2720	2976
BO	176	432	688	944	1200	1456	1712	1968	2224	2480	2736	2992
CO	192	448	704	960	1216	1472	1728	1984	2240	2496	2752	3008
DO	208	464	720	976	1232	1488	1744	2000	2256	2512	2768	3024
EO	224	480	736	992	1248	1504	1760	2016	2272	2528	2784	3040
FO	240	496	752	1008	1264	1520	1776	2032	2288	2544	2800	3056

## ARRAY

TYPE	byte	
1	2	1 integer
NAAM	90	2 Z
	65	3 A
LENGTE	205	4 aantal bytes
vanaf hier	0	5 + 2 * aantal
		dimensies +1
DIMENSIE	1	6
Aantal	101	7 101 elementen
elementen	0	8
Element 0	12	9 Waarde 2A%(0)
	23	10 = 5900
Element 1	45	11 Waarde 2A%(1)
	19	12 = 4909
Element 2	127	13 Waarde 2A%(2)
	33	14 = 8575

## ASCII - CODE

37	38	39	40	41	51	96	125	11	92	42	12
53	54	55	56	57	48	45	123	8	45	43	15
5	6	7	8	9	0	-		wis	-	+	
84	85	86	73	79	80	94	91		129	7	14
116	121	117	105	111	112	64	93		55	56	57
t	u	u	i	o	p	e	, l		7	8	9
71	72	74	75	76	43	42	127		7	133	7
103	104	106	107	108	59	58	35	13	52	53	54
g	h	j	k	l	:	:		# Return	4	5	6
66	78	77	44	46	63				128	7	131
98	110	109	44	46	47				49	50	51
b	n	m	.	.	/				1	2	3

## HEXADECIMAAL - BINAIR

10	20	30	40	50
00010000	00100000	00110000	01000000	01010000
00010001	00000001	00110001	01000001	01010001
00010010	00100010	00110010	01000010	01010010
00010011	00100011	00110011	01000011	01010011
00010100	00100100	00110100	01000100	01010100
00010101	00100101	00110101	01000101	01010101
00010110	00100110	00110110	01000110	01010110
00010111	00100111	00110111	01000111	01010111
00011000	00101000	00111000	01001000	01011000
00011001	00101001	00111001	01001001	01011001
00011010	00101010	00111010	01001010	01011010
00011011	00101011	00111011	01001011	01011011
00011100	00101100	00111100	01001100	01011100
00011101	00101101	00111101	01001101	01011101
00011110	00101110	00111110	01001110	01011110
00011111	00101111	00111111	01001111	01011111

## GEHEUGENINDDELING

8h	dec.	
1-0000	- 0	1-- MONITOR in ROM ( read only memory = uit dit geheugen kan alleen worden gelezen en niets worden veranderd )
1-1000	- 4096	1-- Routines voor Cassette, Disk en beeldscherm ( Video )
1-5000	- 20480	1-- BASIC-instakmodule in slauf 1. De BASIC-INTERPRETER maakt het mogelijk om te programmeren in BASIC.
1-5000	- 20480	1-- Het vertaalt de codewoorden van de Basic-instructies in een aantal handelingen en voert ze uit. ( in ROM )
1-5000	- 20480	1-- Het VIDEOGEHEUGEN. Linker- en rechter beeldscherm.
1-5000	- 20480	1-- 24 regels en 80 kolommen, links en rechts 40 kolommen.
1-5000	- 20480	1-- Met 40/80 karakterkaart alle kolommen op het scherm.
1-5000	- 20480	1-- Het tweede Videogeheugen vanaf 5K 5800 is niet aanwezig op de P-2000 I, wel op het M-model.
1-5000	- 20480	1-- In gebruik door de Monitor voor registreruimten

CHARLES VAN DER LINDEN



# HEXADECIMAL - B I N A I R

T.00

&H	00	10	20	30	40	50	60	70	&H
0	00000000	00010000	00100000	00110000	01000000	01010000	01100000	01110000	0
1	00000001	00010001	00000001	00110001	01000001	01010001	01100001	01110001	1
2	00000010	00010010	00100010	00110010	01000010	01010010	01100010	01110010	2
3	00000011	00010011	00100011	00110011	01000011	01010011	01100011	01110011	3
4	00000100	00010100	00100100	00110100	01000100	01010100	01100100	01110100	4
5	00000101	00010101	00100101	00110101	01000101	01010101	01100101	01110101	5
6	00000110	00010110	00100110	00110110	01000110	01010110	01100110	01110110	6
7	00000111	00010111	00100111	00110111	01000111	01010111	01100111	01110111	7
8	00001000	00011000	00101000	00111000	01001000	01011000	01101000	01111000	8
9	00001001	00011001	00101001	00111001	01001001	01011001	01101001	01111001	9
A	00001010	00011010	00101010	00111010	01001010	01011010	01101010	01111010	A
B	00001011	00011011	00101011	00111011	01001011	01011011	01101011	01111011	B
C	00001100	00011100	00101100	00111100	01001100	01011100	01101100	01111100	C
D	00001101	00011101	00101101	00111101	01001101	01011101	01101101	01111101	D
E	00001110	00011110	00101110	00111110	01001110	01011110	01101110	01111110	E
F	00001111	00011111	00101111	00111111	01001111	01011111	01101111	01111111	F

&H	80	90	A0	B0	C0	D0	E0	F0	&H
0	10000000	10010000	10100000	10110000	11000000	11010000	11100000	11110000	0
1	10000001	10010001	10100001	10110001	11000001	11010001	11100001	11110001	1
2	10000010	10010010	10100010	10110010	11000010	11010010	11100010	11110010	2
3	10000011	10010011	10100011	10110011	11000011	11010011	11100011	11110011	3
4	10000100	10010100	10100100	10110100	11000100	11010100	11100100	11110100	4
5	10000101	10010101	10100101	10110101	11000101	11010101	11100101	11110101	5
6	10000110	10010110	10100110	10110110	11000110	11010110	11100110	11110110	6
7	10000111	10010111	10100111	10110111	11000111	11010111	11100111	11110111	7
8	10001000	10011000	10101000	10111000	11001000	11011000	11101000	11111000	8
9	10001001	10011001	10101001	10111001	11001001	11011001	11101001	11111001	9
A	10001010	10011010	10101010	10111010	11001010	11011010	11101010	11111010	A
B	10001011	10011011	10101011	10111011	11001011	11011011	11101011	11111011	B
C	10001100	10011100	10101100	10111100	11001100	11011100	11101100	11111100	C
D	10001101	10011101	10101101	10111101	11001101	11011101	11101101	11111101	D
E	10001110	10011110	10101110	10111110	11001110	11011110	11101110	11111110	E
F	10001111	10011111	10101111	10111111	11001111	11011111	11101111	11111111	F

# HEXADECIMAL - D E C I M A A L

&H	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0	&H
0	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240	0
1	1	17	33	49	65	81	97	113	129	145	161	177	193	209	225	241	1
2	2	18	34	50	66	82	98	114	130	146	162	178	194	210	226	242	2
3	3	19	35	51	67	83	99	115	131	147	163	179	195	211	227	243	3
4	4	20	36	52	68	84	100	116	132	148	164	180	196	212	228	244	4
5	5	21	37	53	69	85	101	117	133	149	165	181	197	213	229	245	5
6	6	22	38	54	70	86	102	118	134	150	166	182	198	214	230	246	6
7	7	23	39	55	71	87	103	119	135	151	167	183	199	215	231	247	7
8	8	24	40	56	72	88	104	120	136	152	168	184	200	216	232	248	8
9	9	25	41	57	73	89	105	121	137	153	169	185	201	217	233	249	9
A	10	26	42	58	74	90	106	122	138	154	170	186	202	218	234	250	A
B	11	27	43	59	75	91	107	123	139	155	171	187	203	219	235	251	B
C	12	28	44	60	76	92	108	124	140	156	172	188	204	220	236	252	C
D	13	29	45	61	77	93	109	125	141	157	173	189	205	221	237	253	D
E	14	30	46	62	78	94	110	126	142	158	174	190	206	222	238	254	E
F	15	31	47	63	79	95	111	127	143	159	175	191	207	223	239	255	F

T A B E L L E N  
11-11-86

Charles van der Linden & Zn.  
Broederhof 11, 5504 JC Veldhoven.

( c )



# H E X A D E C I M A A L - D E C I M A A L

T.002

&H	000	100	200	300	400	500	600	700	800	900	A00	B00	C00	D00	E00	F00
00	00	256	512	768	1024	1280	1536	1792	2048	2304	2560	2816	3072	3328	3584	3840
10	16	272	528	784	1040	1296	1552	1808	2064	2320	2576	2832	3088	3344	3600	3856
20	32	288	544	800	1056	1312	1568	1824	2080	2336	2592	2848	3104	3360	3616	3872
30	48	304	560	816	1072	1328	1584	1840	2096	2352	2608	2864	3120	3376	3632	3888
40	64	320	576	832	1088	1344	1600	1856	2112	2368	2624	2880	3136	3392	3648	3904
50	80	336	592	848	1104	1360	1616	1872	2128	2384	2640	2896	3152	3408	3664	3920
60	96	352	608	864	1120	1376	1632	1888	2144	2400	2656	2912	3168	3424	3680	3936
70	112	368	624	880	1136	1392	1648	1904	2160	2416	2672	2928	3184	3440	3696	3952
80	128	384	640	896	1152	1408	1664	1920	2176	2432	2688	2944	3200	3456	3712	3968
90	144	400	656	912	1168	1424	1680	1936	2192	2448	2704	2960	3216	3472	3728	3984
A0	160	416	672	928	1184	1440	1696	1952	2208	2464	2720	2976	3232	3488	3744	4000
B0	176	432	688	944	1200	1456	1712	1968	2224	2480	2736	2992	3248	3504	3760	4016
C0	192	448	704	960	1216	1472	1728	1984	2240	2496	2752	3008	3264	3520	3776	4032
D0	208	464	720	976	1232	1488	1744	2000	2256	2512	2768	3024	3280	3536	3792	4048
E0	224	480	736	992	1248	1504	1760	2016	2272	2528	2784	3040	3296	3552	3808	4064
F0	240	496	752	1008	1264	1520	1776	2032	2288	2544	2800	3056	3312	3568	3824	4080
																4096

&H	0000	1000	2000	3000	4000	5000	6000	7000	&H
000	000	4096	8192	12288	16384	20480	24576	28672	000
100	256	4352	8448	12544	16640	20736	24832	28928	100
200	512	4608	8708	12800	16896	20992	25088	29184	200
300	768	4864	8960	13056	17152	21248	25344	29440	300
400	1024	5120	9216	13312	17408	21504	25600	29696	400
500	1280	5376	9472	13568	17664	21760	25856	29952	500
600	1536	5632	9728	13824	17920	22016	26112	30208	600
700	1792	5888	9984	14080	18176	22272	26368	30464	700
800	2048	6144	10240	14336	18432	22528	26624	30720	800
900	2304	6400	10496	14592	18688	22784	26880	30976	900
A00	2560	6656	10752	14848	18944	23040	27136	31232	A00
B00	2816	6912	11008	15104	19200	23296	27392	31488	B00
C00	3072	7168	11264	15360	19456	23552	27648	31744	C00
D00	3328	7424	11520	15616	19712	23808	27904	32000	D00
E00	3584	7680	11776	15872	19968	24064	28160	32256	E00
F00	3840	7936	12032	16128	20224	24320	28416	32512	F00

&H	8000	9000	A000	B000	C000	D000	E000	F000	&H
000	32768	36864	40960	45056	49152	53248	57344	61440	000
100	33024	37120	41216	45312	49408	53504	57600	61696	100
200	33280	37376	41472	45568	49664	53760	57856	61952	200
300	33536	37632	41728	45824	49920	54016	58112	62208	300
400	33792	37888	41984	46080	50176	54272	58368	62464	400
500	34048	38144	42240	46336	50432	54528	58624	62720	500
600	34304	38400	42496	46592	50688	54784	58880	62976	600
700	34560	38656	42752	46848	50944	55040	59136	63232	700
800	34816	38912	43008	47104	51200	55296	59392	63488	800
900	35072	39168	43264	47360	51456	55552	59648	63744	900
A00	35328	39424	43520	47616	51712	55808	59904	64000	A00
B00	35584	39680	43776	47872	51968	56064	60160	64256	B00
C00	35840	39936	44032	48128	52224	56320	60416	64512	C00
D00	36096	40192	44288	48384	52480	56576	60672	64768	D00
E00	36352	40448	44544	48640	52736	56832	60928	65024	E00
F00	36608	40704	44800	48896	52992	57088	61184	65280	F00
								65536	

 T A B E L L E N  
 11-11-86

 Charles van der Linden & Zn.  
 Broederhof 11, 5504 JC Veldhoven.

( c )

&H	00	10	20	30	40	&H
0	NOP	DJNZ ..	JR NZ ..	JR NC .....	LD B,B	0
1	LD BC,....	LD DE,....	LD HL,....	LD SP,....	LD B,C	1
2	LD (BC),A	LD (DE),A	LD (....),HL	LD (....),A	LD B,D	2
3	INC BC	INC DE	INC HL	INC SP	LD B,E	3
4	INC B	INC D	INC H	INC (HL)	LD B,H	4
5	DEC B	DEC D	DEC H	DEC (HL)	LD B,L	5
6	LD B,..	LD D,..	LD H,..	LD (HL),..	LD B,(HL)	6
7	RLCA	RLA	DAA	SCF	LD B,A	7
8	EX AF	JR ..	JR Z ..	JR C ..	LD C,B	8
9	ADD HL,BC	ADD HL,DE	ADD HL,HL	ADD HL,SP	LD C,C	9
A	LD A,(BC)	LD A,(DE)	LD HL,(....)	LD A,(....)	LD C,D	A
B	DEC BC	DEC DE	DEC HL	DEC SP	LD C,E	B
C	INC C	INC E	INC L	INC A	LD C,H	C
D	DEC C	DEC E	DEC L	DEC A	LD C,L	D
E	LD C,..	LD E,..	LD L,..	LD A,..	LD C,(HL)	E
F	RRCA	RRA	CPL	CCF	LD C,A	F

&H	50	60	70	80	90	A0	&H
0	LD D,B	LD H,B	LD (HL),B	ADD B	SUB B	AND B	0
1	LD D,C	LD H,C	LD (HL),C	ADD C	SUB C	AND C	1
2	LD D,D	LD H,D	LD (HL),D	ADD D	SUB D	AND D	2
3	LD D,E	LD H,E	LD (HL),E	ADD E	SUB E	AND E	3
4	LD D,H	LD H,H	LD (HL),H	ADD H	SUB H	AND H	4
5	LD D,L	LD H,L	LD (HL),L	ADD L	SUB L	AND L	5
6	LD D,(HL)	LD H,(HL)	HALT	ADD (HL)	SUB (HL)	AND (HL)	6
7	LD D,A	LD H,A	LD (HL),A	ADD A	SUB A	AND A	7
8	LD E,B	LD L,B	LD A,B	ADC B	SBC B	XOR B	8
9	LD E,C	LD L,C	LD A,C	ADC C	SBC C	XOR C	9
A	LD E,D	LD L,D	LD A,D	ADC D	SBC D	XOR D	A
B	LD E,E	LD L,E	LD A,E	ADC E	SBC E	XOR E	B
C	LD E,H	LD L,H	LD A,H	ADC H	SBC H	XOR H	C
D	LD E,L	LD L,L	LD A,L	ADC L	SBC L	XOR L	D
E	LD E,(HL)	LD L,(HL)	LD A,(HL)	ADC (HL)	SBC (HL)	XOR (HL)	E
F	LD E,A	LD L,A	LD A,A	ADC A	SBC A	XOR A	F

&H	BO	CO	DO	EO	FO	&H
0	OR B	RET NZ	RET NC	RET PO	RET P	0
1	OR C	POP BC	POP DE	POP HL	POP AF	1
2	OR D	JP NZ .....	JP NC .....	JP PO .....	JP P .....	2
3	OR E	JP .....	OUT ..	EX (SP),HL	DI	3
4	OR H	CALL NZ .....	CALL NC .....	CALL PO .....	CALL P .....	4
5	OR L	PUSH BC	PUSH DE	PUSH HL	PUSH AF	5
6	OR (HL)	ADD ..	SUB ..	AND ..	OR ..	6
7	OR A	RST 00	RST 10	RST 20	RST 30	7
8	CP B	RET Z	RET C	RET PE	RET M	8
9	CP C	RET	EXX	JP (HL)	LD SP,HL	9
A	CP D	JP Z .....	JP C .....	JP PE .....	JP M .....	A
B	CP E	diversen	IN	EX DE,HL	EI	B
C	CP H	CALL Z .....	CALL C .....	CALL PE .....	CALL M .....	C
D	CP L	CALL .....	diversen	diversen	diversen	D
E	CP (HL)	ADC ..	SBC ..	XOR ..	CP ..	E
F	CP A	RST 08	RST 18	RST 28	RST 38	F

ED ..	nn = ....					
&H	40	50	60	70	A0	B0
0	IN B,(C)	IN D,(C)	IN H,(C)	---	LDI	LDIR
1	OUT (C),B	OUT (C),D	OUT (C),H	---	CPI	CPIR
2	SBC HL,BC	SBC HL,DE	SBC HL,HL	SBC HL,SP	INI	INIR
3	LD (nn),BC	LD (nn),DE	---	LD (nn),SP	OUTI	OUTIR
4	NEG	---	---	---	---	---
5	RETN	---	---	---	---	---
6	IM 0	IM 1	---	---	---	---
7	LD I,A	LD A,I	RRD	---	---	---
8	IN C,(C)	IN E,(C)	IN L,(C)	IN A,(C)	LDD	LDDR
9	OUT (C),C	OUT (C),E	OUT (C),L	OUT (C),A	CPD	CPDR
A	ADC HL,BC	ADC HL,DE	ADC HL,HL	ADC HL,SP	IND	INDR
B	LD BC,(nn)	LD DE,(nn)	---	LD SP,(nn)	OUTD	OUTDR
C	---	---	---	---	---	---
D	RETI	---	---	---	---	---
E	---	IM 2	---	---	---	---
F	LD R,A	LD A,R	RLD	---	---	---

DD .. d	(d : - 128 tot + 127)				
&H	FD .. d	IX = IX	FD i.p.v. DD : IX = IX		
09	ADD IX,BC	71 LD (IX+d),C	DD CB d ..		
19	ADD IX,DE	72 LD (IX+d),D	FD CB d ..		
21	LD IX,....	73 LD (IX+d),E			
22	LD (....),IX	74 LD (IX+d),H	06 RLC (IX+d)	86 RES 0,(IX+d)	
23	INC IX	75 LD (IX+d),L	0E RRC (IX+d)	8E RES 1,(IX+d)	
29	ADD IX,IX	77 LD (IX+d),A	16 RL (IX+d)	96 RES 2,(IX+d)	
2A	LD IX,(....)	7E LD A,(IX+d)	1E RR (IX+d)	9E RES 3,(IX+d)	
2B	DEC IX	86 ADD (IX+d)	26 SLA (IX+d)	A6 RES 4,(IX+d)	
34	INC (IX+d)	8E ADC (IX+d)	2E SRA (IX+d)	AE RES 5,(IX+d)	
35	DEC (IX+d)	96 SUB (IX+d)	36 ---	B6 RES 6,(IX+d)	
36	LD (IX+d),..	9E SBC (IX+d)	3E SRL (IX+d)	BE RES 7,(IX+d)	
39	ADD IX,SP	A6 AND (IX+d)	46 BIT 0,(IX+d)	C6 SET 0,(IX+d)	
46	LD B,(IX+d)	AE XOR (IX+d)	4E BIT 1,(IX+d)	CE SET 1,(IX+d)	
4E	LD C,(IX+d)	B6 OR (IX+d)	56 BIT 2,(IX+d)	D6 SET 2,(IX+d)	
56	LD D,(IX+d)	BE CP (IX+d)	5E BIT 3,(IX+d)	DE SET 3,(IX+d)	
5E	LD E,(IX+d)	E1 POP IX	66 BIT 4,(IX+d)	E6 SET 4,(IX+d)	
66	LD H,(IX+d)	E5 PUSH IX	6E BIT 5,(IX+d)	EE SET 5,(IX+d)	
6E	LD L,(IX+d)	E9 JP (IX)	76 BIT 6,(IX+d)	F6 SET 6,(IX+d)	
70	LD (IX+d),B	F9 LD SP,IX	7E BIT 7,(IX+d)	FE SET 7,(IX+d)	

CB ..	registers : B,C,D,E,H,L,(HL),A		
&H	register	register	register
00-07	RLC	00-07	RES 0,
08-0F	RRC	08-0F	RES 1,
10-17	RL	09-0F	RES 2,
18-1F	RR	08-0F	RES 3,
20-27	SLA	A0-A7	RES 4,
28-2F	SRA	A8-AF	RES 5,
30-37	?	B0-B7	RES 6,
38-3F	SRL	B8-BF	RES 7,
40-47	BIT 0,	C0-C7	SET 0,
48-4F	BIT 1,	C8-CF	SET 1,
50-57	BIT 2,	D0-D7	SET 2,
58-5F	BIT 3,	D8-DF	SET 3,
60-67	BIT 4,	E0-E7	SET 4,
68-6F	BIT 5,	E8-EF	SET 5,
70-77	BIT 6,	F0-F7	SET 6,
78-7F	BIT 7,	FB-FF	SET 7,



# VIDEO - GEHEUGEN

T.007

## linkerscherm

kolom		0	1	2	3		regel
&H		0123456789	0123456789	0123456789	0123456789		
1	5000 20480	.....	.....	.....	.....	20519 5027	1
2	5050 20560	.....	.....	.....	.....	20599 5077	2
3	50A0 20640	.....	.....	.....	.....	20679 50C7	3
4	50F0 20720	.....	.....	.....	.....	20759 5117	4
5	5140 20800	.....	.....	.....	.....	20839 5167	5
6	5190 20880	.....	.....	.....	.....	20919 51B7	6
7	51E0 20960	.....	.....	.....	.....	20999 5207	7
8	5230 21040	.....	.....	.....	.....	21079 5257	8
9	5280 21120	.....	.....	.....	.....	21159 52A7	9
10	52D0 21200	.....	.....	.....	.....	21239 52F7	10
11	5320 21280	.....	.....	.....	.....	21319 5347	11
12	5370 21360	.....	.....	.....	.....	21399 5397	12
13	53C0 21440	.....	.....	.....	.....	21479 53E7	13
	5410 21520	.....	.....	.....	.....	21559 5437	14
15	5460 21600	.....	.....	.....	.....	21639 5487	15
16	54B0 21680	.....	.....	.....	.....	21719 54D7	16
17	5500 21760	.....	.....	.....	.....	21799 5527	17
18	5550 21840	.....	.....	.....	.....	21879 5577	18
19	55A0 21920	.....	.....	.....	.....	21959 55C7	19
20	55F0 22000	.....	.....	.....	.....	22039 5617	20
21	5640 22080	.....	.....	.....	.....	22119 5667	21
22	5690 22160	.....	.....	.....	.....	22199 56B7	22
23	56E0 22240	.....	.....	.....	.....	22279 5707	23
24	5730 22320	.....	.....	.....	.....	22359 5757	24

## rechterscherm

	4	5	6	7		
	0123456789	0123456789	0123456789	0123456789		
1	5028 20520	.....	.....	.....	20559 504F	1
2	5078 20600	.....	.....	.....	20639 509F	2
3	50C8 20680	.....	.....	.....	20719 50EF	3
	5118 20760	.....	.....	.....	20799 513F	4
5	5168 20840	.....	.....	.....	20879 518F	5
6	51B8 20920	.....	.....	.....	20959 51DF	6
7	5208 21000	.....	.....	.....	21039 522F	7
8	5258 21080	.....	.....	.....	21119 527F	8
9	52A8 21160	.....	.....	.....	21199 52CF	9
10	52F8 21240	.....	.....	.....	21279 531F	10
11	5348 21320	.....	.....	.....	21359 536F	11
12	5398 21400	.....	.....	.....	21439 53BF	12
13	53E8 21480	.....	.....	.....	21519 540F	13
14	5438 21560	.....	.....	.....	21599 545F	14
15	5488 21640	.....	.....	.....	21679 54AF	15
16	54D8 21720	.....	.....	.....	21759 54FF	16
17	5528 21800	.....	.....	.....	21839 554F	17
18	5578 21880	.....	.....	.....	21919 559F	18
19	55C8 21960	.....	.....	.....	21999 55EF	19
20	5618 22040	.....	.....	.....	22079 563F	20
21	5668 22120	.....	.....	.....	22159 568F	21
22	56B8 22200	.....	.....	.....	22239 56DF	22
23	5708 22280	.....	.....	.....	22319 572F	23
24	5758 22360	.....	.....	.....	22399 577F	24
25	57B0 22400	r 57A8 22440	26.57D0 22480	r 57F8 22520-22527	(57FF)	

T A B E L L E N  
11-09-86

Charles van der Linden & Zn.  
Broederhof 11, 5504 JC Veldhoven.

# ASCII - CODE

T.008

27	33	34	95	36	37	38	39	40	41	61	96	125	11	92	42	12
CODE	1	2	3	4	5	6	7	8	9	0	-	123	8	45	43	15
													wis	-	+	
25	81	87	69	82	84	89	85	73	79	80	94	91		129	7	14
9	113	119	101	114	116	121	117	105	111	112	64	93		55	56	57
TAB	q	w	e	r	t	y	u	i	o	p	@	J		7	8	9
	65	83	68	70	71	72	74	75	76	43	42	127		7	133	7
	97	115	100	102	103	104	106	107	108	59	58	35	13	52	53	54
	a	s	d	f	g	h	j	k	l	;	:	#	Return	4	5	6
62	90	88	67	86	66	78	77	44	46	63				128	7	131
60	122	120	99	118	98	110	109	44	46	47				49	50	51
<	z	x	c	v	b	n	m	,	.	/				1	2	3
29	31			spatie				26	30					130	5	3
16	17			32				18	19					48	48	46
-----cursortoetsen-----														0	00	,

CHR\$ ( ... )	15	wis tot cursorpunt	131	geel	147	gr.geel
0 einde toonstring	16	cursor naar links	132	blauw	148	gr.blauw
1 cursor aan	17	cursor omhoog	133	paars	149	gr.paars
2 cursor uit	18	cursor omlaag	134	cyaan	150	gr.cyaan
3 OUT 48,0 + CHR\$(29)	19	cursor naar rechts	135	wit	151	graf.wit
4 pos.cursor+chr\$ R+K	20	cursor op kolom k	136	knipperen		
5 scherm naar printer	21	wis tot einde regel	137	niet knipperen		
6 cursorpunt huidige	22	wis - einde venster	140	normale hoogte		
7 piep	23	begin toonstring	141	dubbele hoogte		
8 cursor naar links	24	cursor naar rechts	152	verbergen		
9 horizontale TAB	28	venster 24 r. 40 k.	153	grafisch aaneen		
10 cursor omlaag	29	CHR\$(13) + CHR\$(10)	154	gescheiden grafisch		
11 wis karakter links	30	deel-PRINT uit	156	zwarte achtergrond		
12 wis venster	31	cursor op cursorpunt	157	gekleurde achtergr.		
13 cursor linker kant	129	rood	145	graf.rood	158	herhaal graf.kar.
14 deel-PRINT	130	groen	146	gr.groen	159	zet uit CHR\$(158)

## T O E T S - C O D E

104	118	135	76	79	77	73	78	126	112	117	119	140	116	115	114	112
32	46	63	4	7	5	1	6	54	41	45	47	68	44	43	42	40
CODE	1	2	3	4	5	6	7	8	9	0	-		wis	-	+	
80	75	107	108	111	109	105	110	145	121	125	127	132		123	122	120
8	3	35	36	39	37	33	38	70	49	53	55	60		51	50	48
TAB	q	w	e	r	t	y	u	i	o	p	@			7	8	9
	106	83	84	87	85	81	86	134	137	141	143	92	124	139	138	136
	34	11	12	15	13	9	14	62	65	69	71	20	52	67	66	64
	a	s	d	f	g	h	j	k	l	;	:	#	Return	4	5	6
98	82	99	100	103	101	97	102	94	129	133				131	130	128
26	10	27	28	31	29	25	30	22	57	61				59	58	56
<	z	x	c	v	b	n	m	,	.	/				1	2	3
72	74			89	spatie			93	95					91	90	88
0	2			17				21	23					19	18	16
-----cursortoetsen-----														0	00	,

&H	dec.	
I-----I		MONITOR in ROM ( read only memory = uit dit geheugen
I-0000 - 0 -I--		kan alleen worden gelezen en niets worden veranderd )
I		Routines voor Cassette, Disk en beeldscherm ( Video )
I-1000 - 4096-I--		BASIC-insteekmodule in sleuf 1. De BASIC-INTERPRETER
I		maakt het mogelijk om te programmeren in BASIC.
I		Het vertaalt de codewoorden van de Basic-instructies
I		in een aantal handelingen en voert ze uit. ( in ROM ).
I-5000 - 20480-I--		Het VIDEOGEHEUGEN. Linker- en rechter beeldscherm.
I		24 regels en 80 kolommen, links en rechts 40 kolommen.
I		Met 40/80 karakterkaart alle kolommen op het scherm.
I		Het tweede Videogeheugen vanaf &H 5800 is niet aanwezig
I		op de P-2000 T, wel op het M-model.
I-6000 - 24576-I--		In gebruik door de Monitor voor registerruimten
I		en bufferblokken.
I-6200 - 25088-I--		Administratie- en bufferruimte voor
I		Philips Cassette-Basic.
I-6547 - 25927-I--		Basic-ruimte : begin te lezen uit &H 625C en &H 625D.
I		Gevuld met regels., regelaanwijzers, BASIC-tokens enz.
I		I-- Variabelenruimte : begin te lezen uit &H 6405 en &H 6406.
I		integer-, string variabele, enkele en dubbele precisie.
I		I-- Array-ruimte : begin te lezen uit &H 6407 en &H 6408.
I		Ruimte voor arrays reserveren d.m.v. DIMmen.
I		I-- Vrije ruimte : begin te lezen uit &H 6409 en &H 640A.
I		Ruimte afhankelijk van grootte andere ruimten.
I		I-- STACK = stapel : einde te lezen uit &H 6258 en &H 6259.
I		Bewaring van o.a.terugspringadressen van GOSUB, FOR-NEXT.
I		I-- String-ruimte : normaal 50 bytes, in te stellen met b.v.
I		CLEAR 4000
I		I-- Gereserveerd : begin -1 te lezen uit &H 63B8 en &H 63B9.
I		in te stellen met b.v.: CLEAR 50, &H DFFF.
I-FFFF - 65535-I--		Einde geheugen

GEHEUGENBANKEN

&H	dec.	I-----I																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	</
----	------	---------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	----

BANKSWITCHING

Charles van der Linden & Zn.

11-11-86

Broederhof 11, 5504 JC Veldhoven.



TYPE :

```

A! = VARPIR(AA%(0))
de geheugenplaats van de
eerste byte van element 0
-----
Wegschrijven naar cassette :
CSAVE * ZA% : onder naam ZA%
CSAVE * ZA% @ " Array " :
    onder naam " Array "

```

## STRING - VARIABLE

Voorbeeld AZ\$ = "plaats"

```

S T R I N G R U I M T E
I-----I
5525 I    118 I    v    string
    26 I    97 I    a    2
    27 I   110 I    n
I-----I-
5528 I   112 I    p
    29 I  108 I    l
    30 I   97 I    a
    31 I   97 I    a    1
    32 I  116 I    t
    33 I  115 I    s
I-----I
5534 I  100 I    d    0
    35 I  101 I    e

```

De computer kent slechts twee getallen n.l. 0 en 1.  
 Nul voor geen stroom, is er wel stroom dan betekent dat "1".  
 Op deze wijze worden cijfers omgezet in elektrische stroomstootjes.  
 Bij nul wordt de stroom gestopt, bij 1 laat men de stroom doorgaan.  
 Volgens het binaire stelsel kunnen nu getallen geformeerd worden.  
 Het tweetallige stelsel gebruikt alleen de cijfers 0 en 1.

Om een reeks binaire getallen te maken, slaat men alle andere decimale getallen, die dan dus niet bestaan, doodgewoon over :

bin.	0-1-10-11-100-101-110-111-1000-1001-1010-1011-1100-1101-1110-1111-10000
dec.	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
hex.	0 1 2 3 4 5 6 7 8 9 A B C D E F 10

Hexadecimaal is volgens het zestientallig stelsel.

De getallen tien tot vijftien worden voorgesteld door de letters A-F.

tot de macht	0	1	2	3	4	5	6	7	8	
decimaal	10	1	10	100	1000	10000	100000	1000000	10000000	100000000
binair	2	1	2	4	8	16	32	64	128	256
hexadec.	16	1	16	256	4096	65536	1048576	enz.		

Een byte bestaat uit acht bits, het grootste getal onder te brengen in een byte is 255, n.l. binair 11111111.

De Z-80 microprocessor binnenin de P-2000 heeft een bereik van 64 Kilobytes.  
 D.w.z. er zijn  $64 \times 1024 = 65536$  ( 0-65535 ) geheugenplaatsen mogelijk.

65535 komt overeen met het 16 bits-getal 1111111111111111.

Elke plaats ( byte ) kan een getal van max. 255 bevatten.  
 Wanneer we een getal groter dan 255 willen vastleggen, dan zijn er meer plaatsen nodig.

Een getal A%, b.v. 25927 wordt opgeslagen in twee bytes.  
 De tweede ( hoge ) byte wordt gevuld met het getal dat ontstaat door met 256 te delen en naar beneden af te ronden ( integer-deling ).  
 ( ) lage ( eerste ) byte gaat het restant van de deling bevatten ( modulo ).

HB = A% ~ 256	of	25927 ~ 256	= 101	of	HB = INT ( 25927/256 )	= 101
LB = A% MOD 256	of	25927 MOD 256	= 71		LB = 25927 - ( 256 * H )	= 71

Werken met hexadecimale getallen maakt alles eenvoudiger :  
 Het getal 25927 is hexadecimaal 6547 n.l. HEX\$ ( 25927 ) = &H 6547

&H 65 is de hoge byte, &H 47 de lage byte :  
 LB = HEX\$ ( 71 ) = &H 47  
 HB = HEX\$ ( 101 ) = &H 65

Een hexadecimaal getal wordt dus omgekeerd gesplitst :  
 &H 625C wordt &H 5C en &H 62.

Een groot getal weer terug vormen uit twee bytes gaat als volgt :  
 A = inhoud lage byte + 256 x inhoud hoge byte of  
 A = PEEK ( LB ) + 256 \* PEEK ( HB )

Bijvoorbeeld :  
 A = PEEK ( &H 6405 ) + 256 \* PEEK ( &H 6406 )



De Z-80 processor werkt met geheugenplaatsen die registers worden genoemd. Van belang zijn de registers:

A ( accumulator ), F ( vlagregister ), B, C, D, E, H, L ( secundaire registers )  
 SP ( stackpointer ), PC ( program counter ), IX, IY ( index-registers )

De secundaire registers worden ook in paren gebruikt, om getallen groter dan 255 te kunnen opslaan : B-C, D-E, H-L.

De processor voert operaties uit aan de hand van de Z-80 Code.  
 Een machinetaalprogramma is een reeks van instructies in binaire getallen.  
 In de praktijk wordt gewerkt met hexadecimale getallen.

Het programmeren gaat gemakkelijker door te werken met Assembleertaal.  
 Deze taal hanteert afkortingen ( Mnemonics ) van de uiteindelijke operatie die van de Z-80 verwacht wordt, b.v.

LD A,B	laad in register A het getal uit register B
LD HL,5000	laad in registerpaar HL het getal 5000, d.w.z. 00 in L, 50 in H
LD C,(HL)	laad in C de inhoud van de plaats die het getal in HL aanwijst
LD (E000),HL	laad in geheugenplaats &H E000 + E001 het getal uit HL
LD A,(650D)	laad in register A het getal uit geheugenplaats &H 650D
LD (HL),L	laad in de geheugenplaats die HL aanwijst het getal uit register L
INC DE	increase = verhoog het getal in register DE met 1
ADD HL,BC	addition = tel de inhoud van BC op bij de inhoud van HL
SBC HL,DE	subtract = trek de inhoud van DE af van de inhoud van HL
JP 9000	jump = spring naar de instructies vanaf plaats 9000
CALL 1059	call = roep de subroutine vanaf 1059 aan en keer terug
PUSH BC	zet de inhoud van BC even weg op de STACK
POP DE	vul DE weer met de inhoud die op de stapel is gezet
CP 01	compare = vergelijk het getal in register A met het getal 01
RET	return = terug ( naar BASIC )

Deze instructies worden d.m.v. het ASSEMBLER-programma vertaald in hexadecimale getallen. Deze getallen worden uiteraard aan de processor binair aangeboden. Bijvoorbeeld :

LD BC,0800	01 00 08	OUT 94	D3 94
LD (DE),A	12	SBC HL,SP	ED 72
DEC H	25	LD IX,7000	DD 21 00 70

Omgekeerd, vertalen van Z-80 Operation Code naar Assembleertaal heet Disassemble.

Blockmove is het d.m.v. een machinetaal-routine verplaatsen van een blok gegevens vanaf de ene geheugenplaats naar een andere.  
 In feite wordt er niets verplaatst, maar gekopieerd, op de oude plaats zijn na afloop nog alle gegevens te vinden.  
 Het HL register moet worden gevuld met het startadres.  
 In register DE het bestemmingsadres.  
 Register BC dient het aantal te "verplaatsen" bytes te bevatten.  
 Deze waarden kunnen worden opgeslagen in de ongebruikte ruimte vanaf &H 6070, om later via de routine te worden uitgelezen.

LD HL , ( 6071 )	2A 71 60	beginadres in HL uit plaatsen &H 6071 en &H 6072
LD DE , ( 6073 )	ED 5B 73 60	bestemming in DE uit plaatsen &H 6073 en &H 6074
LD BC , ( 6075 )	ED 4B 75 60	aantal bytes in BC uit &H 6075 en &H 6076
LDIR	ED B0	blockmove
RET	CS	terug naar Basic

BANKSWITCHING  
 11-11-86

Charles van der Linden & Zn.  
 Broederhof 11, 5504 JC Veldhoven.

In de P2000 INTERPRETER kan de schrijfroutine aangeroepen worden op adres &H 105C, de leesroutine op &H 1059.

De registers moeten als volgt worden gevuld :

In register A de beginletter van de naam ( in ASCII-code ) waaronder het programma op cassette moet worden opgeslagen.

In registerpaar HL het startadres. In registerpaar DE het einde + 1. Deze waarden kunnen naar behoefte verschillen en moeten tijdelijk ergens worden opgeslagen om ze daar te laten uitlezen.

Een bruikbare ruimte bevindt zich op de plaatsen &H 6070 - &H 6086, eigenlijk bestemd voor de DISK-ROUTINE.

De Cassette-routine komt er als volgt uit te zien :

```
LD A , ( 6070 )    3A 70 60    beginletter in register A uit &H 6070
LD HL, ( 6071 )    2A 71 60    beginadres in HL uit &H 6071 en &H 6072
LD DE, ( 6073 )    ED 5B 73 60 eind +1 in paar DE uit &H 6073 en &H 6074
CALL 1059          CD 59 10    leesroutine
RET               C9          terug naar Basic
Om te schrijven wordt CALL 105C gebruikt : CD 5C 10.
```

```
10 DATA 3A,70,60,2A,71,60,ED,5B,73,60,
    CD,59,10,C9
20 FOR I = 0 TO 13 : READ A$ :
    POKE &H 6075 + I , VAL ( "&H" + A$ )
30 NEXT : DEFUSR 1 = &H 6075
40 DEF FN B ( A ) = INT ( A / 256 ) :
    DEF FN A ( A ) = A - 256 * FN B ( A )
50 PRINT CHR$ ( 12 ) "1 = lezen
    2 = schrijven " ; : INPUT A
60 ON A GOTO 70 , 80
70 POKE &H 6080 , &H 59 : GOTO 90
80 POKE &H 6080 , &H 5C
90 PRINT "Naam" ; : INPUT N$
100 POKE &H 6070 , ASC ( N$ )
110 PRINT "Startadres " ; : INPUT A$
120 PRINT "Eindadres " ; : INPUT B$
130 POKE &H 6071 , FN A ( A ) :
    POKE &H 6072 , FN B ( A )
140 POKE &H 6073 , FN A ( B ) :
    POKE &H 6074 , FN B ( B )
150 A$ = USR 1 ( N$ )
```

Vanaf plaats &H 6075 kan de machinetaal worden opgeslagen.

Om een adres in twee bytes op te slaan wordt gebruik gemaakt van nevenstaande functies.

Keuze : lezen of schrijven ?

Lezen.

Schrijven.

Op plaats &H 6070 de naam van het programma.

Alvorens de cassette-routine aan te roepen moeten de uitleesadressen worden opgevuld.

Aanroep van de cassette-routine.

Op deze wijze kan elk willekeurig deel van het geheugen naar cassette worden geschreven of van cassette in een willekeurig stuk geheugen worden geladen.

#### ACHTERWAARDS WISSEN : ( naar P2C2 6.5.1 )

Het kan voorkomen dat een cassette " rommel " bevat en daardoor onbruikbaar lijkt. De cassette kan gewist worden terwijl de recorder achteruit loopt.

```
FOR I = 1 TO 40000 : OUT 16 , 70 : NEXT
```

Gewist wordt van einde tot begin band. Het begin van de band wordt niet gedetecteerd, daarom vlug uitnemen als de cassette stopt.

#### WISSEN VAN PROGRAMMA'S ( naar P2C2 6.5.2. )

Het kan zijn dat er halverwege een cassette de rest van de programma's verwijderd moeten worden. Dit kan als volgt :

1. CLOAD het programma dat als laatste moet blijven en CSAVE het weer.

```
2. Typ in : FOR I = 0 TO 1000 : OUT 16 , 74 : NEXT
```

Het bandje loopt nu ca. 3 seconden en wist daarmee de rest.

B A N K S W I T C H I N G

11-11-86

Charles van der Linden & Zn.

Broederhof 11, 5504 JC Veldhoven.



Het besturingssysteem ( OPERATING SYSTEM ) van de mini-cassetterecorder.

Alle functies beginnen op startadres &H 0018 in de MONITOR.

Via slechts vier bytes machinetaal zijn ze allen bereikbaar voor de programmeur.

LD A,(HL)        7E        haal het funktienummer op

RST 18            DF        aanroep cassette-besturing

LD (HL),A        77        geef foutcode terug

RET              C9        terug naar Basic

De machinetaal kan in het geheugen worden gezet b.v. op de plaatsen die bestemd zijn voor de DISK-routine n.l. van &H 6070 - &H 6086. Stel vanaf &H 6083.

Het Basicprogramma luidt dan :

10 DATA 7E,DF,77,C9

20 FOR I = 0 TO 3 : READ A\$ : POKE &H 6083 + I , VAL ( " &H " + A\$ ) : NEXT

30 DEFUSR = &H 6083

Na RUNnen van dit programma kunnen de cassettefuncties worden aangeroepen via : PRINT USR ( getal ). Het getal moet korresponderen met de gevraagde opdracht.

Opdracht Betekenis

- 0     Initialiseer cassetterecorder ( dit gebeurt steeds bij het aanzetten van de P-2000 ).
- 1     Spoel terug naar het begin van de band.
- 2     Spoel N blokken vooruit. Het getal N moet eerst op plaats &H 604F worden gePOKEt.  
B.v. POKE &H604F , 10 : PRINT USR ( 2 )    ( spoel 10 blokken vooruit ).
- 3     Spoel N blokken terug .  
B.v. POKE &H604F , 19 : PRINT USR ( 3 )    ( spoel 19 blokken terug ).
- 4     Schrijf " einde band ". De P-2000 weet dat vanaf dit punt de rest van de cassette leeg is.
- 5     Opnemen : Een stuk geheugen naar de cassette schrijven.  
Het startadres moet op de plaatsen &H 6030 en &H 6031 worden gePOKEt.  
De lengte in bytes op de adressen &H 6032 en &H 6033, alsmede op &H 6034 en &H 6035.  
De eerste 8 letters van de naam vanaf plaats &H 6036.  
De tweede 8 tekens van &H 6047 - &H 604E.  
Voorbeeld : Een deel geheugen vanaf 34968 ter lengte van 2634 bytes naar cassette onder naam "CASSETTE-ROUTINE"  
Eerst getallen naar hexadecimaal vertalen :  
HEX \$ ( 34968 ) = &H 8898    ;    HEX \$ ( 2634 ) = &H 0A4A  
Daarna de TAPE-HEADER adressen vullen :  
POKE &H 6030 , &H 98 : POKE &H 6031 , &H 88  
POKE &H 6032 , &H 4A : POKE &H 6033 , &H 0A  
POKE &H 6034 , &H 4A : POKE &H 6035 , &H 0A  
A\$ = "CASSETTE-ROUTINE"  
FOR I = 1 TO 8 : POKE &H 6036 + I - 1 , ASC ( MID\$ ( A\$ , I ) ) : NEXT  
FOR I = 1 TO 8 : POKE &H 6047 + I - 1 , ASC ( MID\$ ( A\$ , I+8 ) ) : NEXT  
De plaatsen &H 603E - &H 6040 duiden de soort file aan. BAS = BASIC  
Op plaats &H 6041 kan het type van de file worden gezet.  
B.v. B = Basic. Na dit alles PRINT USR ( 5 ) en er wordt geschreven.
- 6     Lees een stuk geheugen van de band. Als bij 5.  
De plaatsen &H 6030 en &H 6031 worden gevuld met het laadadres waarnaartoe de gegevens van cassette moeten worden getransporteerd.
- 7     Status van de cassetterecorder. Werkt niet met bewuste machinetaal.

De TAPEHEADER is het blok van 32 bytes die elk programma op cassette voorafgaat. Bij schrijven worden de adressen &H 6030-&H 604F uitgelezen naar cassette. Bij laden worden deze adressen vanaf cassette gevuld.

B A N K S W I T C H I N G

11-11-86

Charles van der Linden & Zn.

Broederhof 11, 5504 JC Veldhoven.