

PENDO

MURDERED

EXCEM

OCM CMC

OC

VOORWOORD

Er zijn veel dingen waarin de P2000 zich onderscheidt van andere kleine computers. Hoewel hij, zeker buiten Nederland, nooit een doorslaand succes is geweest (volkomen ten onrechte natuurlijk), is er waarschijnlijk geen computer die zo lang heeft meegedraaid als de P2000. De machine is meer dan vijf jaar leverbaar geweest en is zelfs nu nog hier en daar te koop. Dat de P2000 zo lang in de belangstelling heeft gestaan, is aan verschillende dingen te danken. Het ontwerp was zijn tijd ver vooruit en de flexibiliteit is buitengewoon groot. In de kast is ruimte genoeg voor allerhande uitbreidingen, die braaf worden getrokken door de ingebouwde voeding. Je kunt er in één handomdraai een tekst- of een bestandsverwerker van maken.

Enfin, het is hier niet de plaats de lof te zingen over die goede ouwe P2000. Wat ik wilde benadrukken is dat de P2000 zich in nog een opzicht van de meeste andere kleine computers onderscheidt: het aanvankelijk volkomen ontbreken van goede documentatie. Wie zoals ik de machine omstreeks 1980 of 1981 heeft gekocht, moest het doen met een lijstje BASIC-instructies die je kon gebruiken. En dat was nog niet eens volledig, laat staan foutloos. En programma's waren er al helemaal niet.

Voor de gebruikers van Cassette-BASIC is er daarna veel verbeterd. Rob van der Heij en Rob van Ommering namen de BASIC-interpreter rigoreus onder handen en maakten er BASIC NL van. Dirk Kroon schreef daar een uitstekende gebruiksaanwijzing bij. Er kwamen steeds meer goed doortimmerde programma's, niet in de laatste plaats van de leden van de Nat. Lab. Computerclub. En veel leden van deze club, maar ook van de P2000 Gebruikersgroep, begonnen te graven en te spitten. Zij hebben veel informatie verzameld die voor P2000-gebruikers van belang is. Daartoe horen ook de functies van veel adressen tussen &H6000 en &H6547 waarvan u in dit boekje de neer slag vindt.

Diskridders staan nog steeds in de kou

P2000-bezitters die de eeuvele moed hebben gehad floppy disk drives aan te schaffen en met de 24K BASIC-interpreter te gaan werken, zijn nu eigenlijk nog net zo ver als vijf jaar geleden. Ik heb nog steeds niet meer documentatie dan het overzicht van BASIC-instructies dat PTIS toen verstrekte, en dat nog onvollediger is dan het lijstje dat bij Cassette-BASIC werd verstrekkt. En nog meer fouten bevat.

Om raadselachtige redenen heeft Disk-BASIC nooit dezelfde aandacht gekregen als Cassette-BASIC. Een reden zou kunnen zijn dat je, om met schijven te kunnen werken, niet alleen die Disk-BASIC-insteekmodule nodig had, maar ook een systeemschijf, een Floppy Disk Interface en natuurlijk één of twee floppy disk drives. Dat was, zeker

toen, een grap die vele duizenden guldens kostte. Daardoor ook is het aantal mensen dat met Disk-BASIC werkt tamelijk beperkt.

Toch vond ik dat ook deze mensen zo veel mogelijk aan hun trekken moesten komen. Daarom bevat dit boekje ook een overzicht van geheugenadressen die voor deze categorie P2000-bezitters misschien wat interessante aanknopingspunten biedt. Zoals ik al eerder in PTC PRINT heb geventileerd, ben ik persoonlijk erg gecharmeerd van Disk-BASIC omdat die niet alleen de mogelijkheid biedt met schijven te werken (wat toch nog aanzienlijk sneller gaat dan met de minicassette), maar ook om op een eenvoudige manier met bestanden te werken.

Wat vindt u in dit Adresboekje ?

Dit boekje bestaat uit twee delen. Het eerste geeft een overzicht van de adressen &H6000...6547 die door de Monitor en BASIC NL worden gebruikt voor allerhande huishoudelijke zaken. Bij het samenstellen ervan ben ik uitgegaan van de laatste van de overzichten die enkele keren in de Nieuwsbrieven van de P2000gg hebben gestaan, aangevuld met wat informatie uit andere bronnen. Het concept heb ik toegestuurd aan een aantal mensen met verzoek om aanvullingen en commentaar. Daaruit is nog een heleboel informatie gekomen. Zonder anderen tekort te willen doen, is een bedankje aan Dick van den Berge en Henk Boetzkes hier zeker op zijn plaats. Zij hebben honderden aanvullingen en verbeteringen aangedragen.

Het overzicht van de adressen is, voor zover het BASIC NL betreft, vrijwel volledig. Hier en daar staat "onbekend" achter een adres. Het vermoeden bestaat dat die adressen meestal niet worden gebruikt. Ik denk dat ook het aantal fouten wel meevalt. Niettemin: ontdekt u fouten of aanvullingen, dan houd ik mij aanbevolen. Misschien komt er dan te zijner tijd een verbeterde herdruk.

Anders is het gesteld met Disk-BASIC. Het overzicht van de adressen is minder volledig en te vrezen valt dat er ook wat meer fouten in staan. Dat komt doordat ik geen documentatie had waaruit ik iets te weten kon komen. Wat in dit boekje aan informatie over Disk-BASIC is opgenomen heb ik zelf moeten opgraven. Gelukkig ben ik daarbij geholpen door het feit dat BASIC NL en Disk-BASIC gedeeltelijk voor dezelfde doeleinden gebruik maken van dezelfde adressen; met name tussen &H6000 en &H61FF. Voor een ander deel blijkt dat beide interпрeters een zelfde reeks adressen gebruiken. Dit laatste betekent dat ze weliswaar voor een bepaalde functie verschillende adressen gebruiken, maar dat de functies in dezelfde volgorde staan. Zo staan bij voorbeeld de adressen voor de USR-functies bij BASIC NL van &H623E tot &H6251, en bij Disk-BASIC van &H6383 tot &H6396.

Voor dit overzicht geldt in nog sterker mate dat ik mij houd aanbevolen voor aanvullingen en verbeteringen.

Iets over de opzet van het Adresboekje

Bij het opzetten van het adresboekje heb ik moeten kiezen tussen volledigheid en overzichtelijkheid. Ik heb gekozen voor het eerste.

Dit betekent dat de omschrijvingen tamelijk uitvoerig zijn, en dat er soms maar enkele adressen op een pagina worden beschreven. De adressen staan echter op volgorde, dus als u de functie van een bepaald adres wilt weten, is dat snel te vinden. Andersom is het wat lastiger. Als u wilt weten welk adres u moet hebben om de invoerveldlengte te veranderen, is het even zoeken. Misschien is er een lezer die op zich wil nemen een "omgekeerd" adresboekje te maken: dus de functie voorop, en daarachter de adressen die zo'n functie gebruikt. Mij ontbreekt de tijd, en ik wou u niet nog langer laten wachten op dit boekje. Het heeft toch al lang genoeg geduurd.

Aardig voor mij is natuurlijk wel dat er al maanden geleden Adresboekjes zijn besteld en betaald. Het zal niet zo vaak gebeuren dat er van een boek vele tientallen exemplaren zijn verkocht voordat de tekst geschreven was. In elk geval gaf dit mij het prettige gevoel dat ik een aantal leden met dit boekje een plezier kan doen.

Drie kolommen

Beide overzichten van adressen bestaan uit drie kolommen. In de eerste kolom vindt u het adres in hexadecimale notatie. Werkt u liever met decimale notatie, dan zal uw P2000 de adressen graag voor u omdelen.

In de tweede kolom staat de inhoud van het adres. Dit is een omstreeden kolom; want alle adressen staan in RAM, en van alle adressen kan de inhoud dus worden veranderd. In deze kolom komen drie soorten notaties voor: hexadecimale getallen zonder haakjes, hex-getallen tussen haakjes en puntjes tussen haakjes. Voor getallen zonder haakjes geldt dat dit meestal de inhoud is van het adres, en dat die inhoud slechts bij hoge uitzondering verandert. Op adres &H6096...6097 staat bij voorbeeld als inhoud &H1814. Dat is het begin van de toetscodetabel. Alleen als u een andere toetscodetabel wilt definiëren, die uiteraard op een andere plaats in het geheugen komt te staan, moet u de inhoud van deze adressen aanpassen.

Getallen tussen haakjes zijn meestal de waarden die bij het opstarten van BASIC aan die adressen worden toegekend. Voor de USR-functies wordt bij het opstarten van BASIC NL bij voorbeeld &H289C ingevuld; dat is het beginadres van een foutafhandelingsroutine. BASIC springt daar naar toe als u een USR-functie aanroept zonder eerst die functie te hebben gedefinieerd met DEF USR. Met DEF USR0 = &Hxxxx verandert u de inhoud van het USR-adres in xxxx.

Dan zijn er nog de aanduidingen (...). Die betekenen dat de inhoud voortdurend kan veranderen. Bij voorbeeld die van de teller op &H6010 en &H6011; of van de printerkolomteller. Het zal duidelijk zijn dat de grens tussen de drie categorieën niet altijd scherp is te trekken. Schiet dus niet op de auteur als u het er niet mee eens bent.

De derde kolom bevat tamelijk uitvoerige omschrijvingen. Die spreken, denk ik, voor zichzelf. En daarmee komen we aan de vraag:

Wat hebt u eraan ?

"Geen fluit" heb ik sommigen horen zeggen. Maar dat is niet helemaal waar. De voorverkoop geeft aan dat er toch wel belangstelling is. Dit Adresboekje geeft tot op zekere hoogte aan hoe de P2000 allerlei huishoudelijke zaken regelt. Dat is interessant voor mensen die wat meer willen weten over de werking van hun computer, maar ook voor programmeurs die in hun programma's gebruik willen maken van bepaalde functies of adressen.

Toch is een waarschuwing hier op zijn plaats. Alle adressen staan in RAM. U kunt dus niet alleen met de instructie "PRINT PEEK(adres)" de inhoud van elk adres uitlezen, maar ook met de instructie "POKE adres,waarde" de inhoud van elk adres veranderen. Bij verreweg de meeste adressen heeft dat laatste echter geen zin en bij sommige adressen is het zelfs desastreus voor uw programma. Een voorbeeld van een adres waar u iets mee kunt doen is &H6016. Daar staat, zowel bij BASIC NL als bij Disk-BASIC, de snelheid waarmee de P2000 informatie naar de printer stuurt. De enige mogelijkheid om die snelheid aan te passen aan een printer van een onbekend merk, is met POKE &H6016,x. Hetzelfde verhaal geldt als u de teller op &H6010 en &H6011 op nul wilt zetten.

Van de adressen &H6020 tot &H602F, waar de adressen staan die de BASIC interpreter moet aanroepen als er een interruptie komt, kunt u beter afbliven.

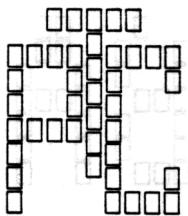
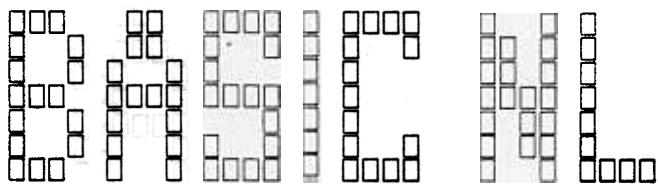
Monitor en BASIC

De Monitor is, zoals bekend, een programma dat allerhande elementaire functies van de P2000 bestuurt. De monitor is vast ingebouwd en staat op de adressen 0 tot &H0FFF. Of u nu BASIC NL, Disk-BASIC, een tekstverwerkingsmodule of wat dan ook in uw P2000 steekt, de Monitor speelt altijd een rol op de achtergrond. Omdat hij behoefte heeft om tussenresultaten en huishoudelijke informatie tijdelijk op te slaan, heeft de Monitor de adressen &H6000 tot en met &H61FF ter beschikking gekregen. Het toepassingsprogramma (bij voorbeeld een BASIC-interpreter of een tekstverwerker) mag dezelfde adressen gebruiken als dat nodig is. De toetscodebuffer zit bij voorbeeld van &H6000 tot &H600B. Toepassingsprogramma's mogen die adressen alleen gebruiken als toetscodebuffer. Met andere woorden: de functie van de adressen &H6000...61FF ligt vast en wordt bepaald door de Monitor.

Daarnaast hebben toepassingsprogramma's vaak behoefte aan een eigen stuk RAM voor huishoudelijke zaken. Daarvoor wordt een aantal RAM-adressen gereserveerd vanaf &H6200. BASIC NL gebruikt de adressen &H6200...6547 en Disk-BASIC de adressen &H6200...68FF. Samengevat: de adressen &H6000...61FF worden gebruikt door Monitor èn toepassingsprogramma; de adressen &H6200...xxxx worden alleen gebruikt door het betrokken toepassingsprogramma.

Veel plezier met dit P2000 Adresboekje.

Son, april 1986
Rob Geutskens



Adres	Inhoud	Omschrijving
<u>Videogeheugen</u>		
5000	(00)	1e videogeheugen, groot 1920 adressen (24 regels van elk 80 posities). Bij 40 posities per regel zijn dit twee pagina's.
----	(00)	
577F	(00)	Met de instructie POKE kan rechtstreeks informatie op het scherm worden gezet.
5780	(00)	"Scroll"-regel 1e videogeheugen. In deze regel (80 posities) wordt onder meer bij LIST de regel klaargezet die vanaf de onderkant het scherm opschuift.
57CF	(00)	
57D0	(00)	Dit laatste deel van het 1e videogeheugen kan worden gebruikt om bij voorbeeld kleine tabellen op te bergen.
----	(00)	
57FF	(00)	Schrijven naar en lezen in deze ruimte kan leiden tot strepen op het scherm. De informatie kan door de computer worden gewist bij het gebruik van cassettefuncties.
5800	(..)	2e videogeheugen, met dezelfde opbouw als het 1e videogeheugen (totaal 2048 adressen). Bij het T-model ontbreken deze adressen geheel doordat er op deze plaats geen RAM is aangebracht. Bij het M-model bevinden zich hier 4-bit adressen, die de "attributen" voor het 1e videogeheugen bevatten, dat wil zeggen aangeven hoe de tekens van het 1e videogeheugen moeten worden weergegeven (normaal, invers, knipperend, onderstreept). Adres &H5800 correspondeert met &H5700 enz. Omdat de adressen een breedte hebben van 4 bits (i.p.v. 8) zijn ze ook bij het M-model niet voor iets anders te gebruiken.
----	(..)	
5FFF	(..)	
<u>Toetsenbord</u>		
6000	(00)	Toetscodebuffer waarin de codes van maximaal 12 toetsen worden opgeslagen, die al zijn ingedrukt maar nog niet door het programma of door de BASIC zijn verwerkt. Deze adressen worden gebruikt door de monitor.
----	(00)	
600B	(00)	
600C	(00)	Toetsbufferteller. Dit adres bevat het aantal toetscodes dat is opgeborgen vanaf adres 6000 (maximaal 12).
600D	(FF)	Op dit adres staat de code van de laatst ingedrukte toets. De ingedrukte toets wordt niet automatisch verwerkt, maar de toetscode blijft staan zolang de toets ingedrukt is. Wordt de toets losgelaten, dan komt er bij de eerstvolgende toetsenbord-"scan"-routine (d.w.z. binnen 20 ms) FF (=255) te staan.
600E	32	Hier staat de herhalingstijd die geldt voor een ingedrukt gehouden toets. Normaal staat hier &H32 (=48). Verandering van de inhoud wordt meteen weer ongedaan gemaakt door de monitor (zie &H00A3 voor het T- en &H0176 voor het M-model). Als een Floppy Disk Interface is aangebracht, doet dit adres tevens dienst als CTC-teller.

Adres	Inhoud	Omschrijving
600F	(00)	Vlag voor het toetsenbord. Normaal staat er 00. 00 = geen SHIFT of SHIFT LOCK ingedrukt 01 = SHIFT ingedrukt (bit 0 = 1) 04 = SHIFT LOCK aan (bit 2 = 1) 05 = SHIFT en SHIFT LOCK ingedrukt (bit 0 = 1 en bit 2 = 1).
		<u>Teller</u>
6010	(..)	Deze twee adressen vormen samen een teller die elke 20 ms wordt opgehoogd. De teller loopt van 0 tot &HFFFF (0...65535) zolang de interrupt aan staat. Dus niet als cassette, schijf of printer werkt. Na zo'n actie loopt de teller verder.
		<u>Monitor</u>
6012		Vlagadres voor de monitor, dat voor verschillende doeleinden wordt gebruikt. Bij voorbeeld voor het automatisch herhalen van een ingedrukt gehouden toets; bit 0 = 1 betekent: "repeat" aan (zie &H006C voor vullen en &H007A...0083 voor testen); bit 1 = 1 betekent "numeric" aan.
		<u>Scherm</u>
6013	(00)	Cursorvorm. Vlagadres voor het toepassingsprogramma. Bij het opstarten van de P2000T wordt dit adres 00 gemaakt; bij de P2000M wordt het 01 (er wordt gecontroleerd of het 2e videogeheugen aanwezig is). 00 betekent dat de cursor als invers blokje zichtbaar wordt gemaakt (P2000T). Een oneven waarde op dit adres maakt bij het T-model de cursor onzichtbaar. Bij het M-model wordt de cursor zichtbaar in de vorm van een streepje. Bit 6 = 1 geeft aan dat de rustige of gemakzuchtige lijster aan staat (zie &H11E6). Bit 7 = 0 geeft LIST-vertraging bij gebruik van de gemakzuchtige lijster (zie &H111B). De inhoud kan bij gebruik van BASIC NL worden veranderd met CHR\$(1), CHR\$(2), CHR\$(14) en CHR\$(30).
6014	(00)	Pointer naar de plaats op het scherm waar de monitor info moet zetten over ondernomen "acties". Deze adressen worden echter niet gebruikt omdat er een fout zit in de desbetreffende monitorroutines. Het adres wordt wel gebruikt door de service-module.
6015	(00)	Zet u hier een schermadres neer, bij voorbeeld adres &H5000 met POKE &H6014,0:POKE &H6015,&H50, dan verschijnt op de eerste positie van het scherm (adres &H5000) de letter L als u SHIFT LOCK indrukt en de letter T als de monitor cassetteroutines uitvoert. Bij BASIC NL wordt dit adres ook gebruikt om de geheugengrootte op te slaan (01, 02 of 03; zie &H1F93).

Adres	Inhoud	Omschrijving
6016	01	<p><u>Printer</u></p> <p>Op dit adres, dat door de monitor wordt gebruikt, staat met welke snelheid de P2000 informatie naar de printer moet sturen ("baudrate"). Met onderstaande formule is dit uit te rekenen (A=PEEK(&H6016)):</p> $\text{Baudrate} = 2400 / (1+A) \quad \text{of}$ $A = (2400 / \text{baudrate}) - 1$ <p>Dit levert het volgende resultaat:</p> <ul style="list-style-type: none"> 00 = 2400 Bd 01 = 1200 Bd (waarde bij opstarten via &H034C) 03 = 600 Bd 07 = 300 Bd (waarde bij opstarten via &H0348) 0F = 150 Bd enz. <p>Normaal start de P2000 op met 1200 Bd. Zijn de pennen 2 en 6 van de printerconnector doorverbonden, dan start de P2000 op met 300 Bd. Dit wordt gecontroleerd in &H0344.</p> <p>Als bit 7 = 1 wordt niet automatisch CR + LF (Carriage Return en Line Feed) gegeven.</p>
6017	(42)	<p><u>Cassette</u></p> <p>Op dit adres staat het ASCII-karakter van de laatst opgetreden cassettefout ("tape error"). Het adres wordt gebruikt door de monitor. De karakters hebben de volgende betekenis:</p> <ul style="list-style-type: none"> 00 (&H00) = Geen fout 65 (&H41) = "A" Geen cassette 66 (&H42) = "B" Begin van de band 67 (&H43) = "C" Leesfout, record gelezen 68 (&H44) = "D" Controlegetal ("checksum") fout in startmerk 69 (&H45) = "E" Einde van de band bij schrijven 70 (&H46) = "F" Einde van de band, maar bestand is geSAVED 71 (&H47) = "G" Geen stopje 73 (&H49) = "I" Maximale terugspoeltijd verstreken; cassetteband gebroken of te lange cassetteband 74 (&H4A) = "J" Te kort datablok gelezen, maar controle-getal ("checksum") in orde, of band spoelt niet doordat deze vast zit 75 (&H4B) = "K" Verkeerde functiecode (komt niet voor in BASIC) 76 (&H4C) = "L" Einde van de band bij lezen 77 (&H4D) = "M" Geen begin-merk ("start mark") gevonden 78 (&H4E) = "N" Geen programma gevonden 87 (&H57) = "W" Verkeerd bestandstype (geen BASIC)

Adres	Inhoud	Omschrijving
6018	47	&H8147 - Laadadres (buffer) voor machinetaalprogramma's, geladen van cassette (zonder ROM-pack in sleuf 1). Het adres wordt ook gebruikt bij het laden van een BASIC-programma voor het doorgeven van het laadadres.
601A	(00)	Adres waar de uitvoering van een machinetaalprogramma begint, dat is geladen zonder ROM-pack in sleuf 1.
601B	(00)	Het adres wordt ook gebruikt voor het doorgeven van de DATA-lengte bij het laden van een BASIC-programma.
601C	(00)	Deze adressen dienen als tijdelijke opslag voor de Monitor-cassetteroutines. &H601E wordt ook gebruikt als tijdelijke opslag voor het aantal af te drukken regels (zie &H0D02).
----	(00)	
601F	(00)	

Interrupties

De adressen &H6020...602F vormen een tabel van interruptievectoren voor de Monitor. Twee opeenvolgende posities bevatten het adres waar de monitor naar toe moet springen als er een interruptie komt.

6020	D6 0F	Op kanaal 0 = disk-actie voltooid of interruptie van CTC timer
6022	99 E7	Op kanaal 1 = interruptie "disk not ready"
6024	00 00	Op kanaal 2 = communicatie-interruptie
6026	38 00	Op kanaal 3 = interruptie van toetsenbord timer
6027	(00)	
----	(00)	
602F	(00)	Worden niet gebruikt

Cassette

De adressen 6030...604F zijn interne werkadressen. Ze vormen samen de "record header" of "file descriptor". Deze gaat vooraf aan elk blok van 1024 bytes dat op cassette wordt weggeschreven.

6030	(47)	De adressen &H6030 en &H6031 vormen het transferadres, dat wil zeggen het eerste lees- of schrijfadres voor cassette-files.
6031	(65)	Het adres moet worden gegeven door het toepassingsprogramma. Bij BASIC NL is dit adres gewoonlijk &H6547, als het laatste bestand dat de cassettewriter "passeeerde" een BASIC-programma was.
6032		In deze adressen staat de totale lengte van het programma of bestand. Dit in verband met het bepalen van het einde van dat programma of bestand.
6033		
6034	(00)	Aantal zinvolle bytes van het gehele programma. BASIC NL schrijft altijd hele blokken van 1024 bytes (&H0400) weg, behalve in het algemeen voor het laatste blok dat wordt weggeschreven (of ingelezen). Als het laatste blok korter is dan 1024 zinvolle bytes, wordt dit blok aangevuld met nullen tot 1024. De "zinvolle lengte" in &H6034 en 6035 is meestal korter
6035	(04)	

Adres Inhoud

Omschriiving

dan de "totale lengte" in &H 6032 en 6033. Als een programma met een totale lengte van 7 blokken wordt overschreven door een programma van bij voorbeeld 3 blokken, blijft de totale lengte 7 blokken, maar de zinvolle lengte wordt dan 3 blokken.

6036	(..)	Eerste 8 karakters van de file-naam. BASIC NL controleert alleen het eerste karakter. Bestaat de naam uit één letter, dan zijn de volgende zeven posities spaties (&H20 = 32).
----	(20)	
603D	(20)	
603E	(42)	"Extension"; de soort file. Dit om duidelijk te kunnen maken of het om een programma, een array of iets anders gaat. De "extension" heeft de volgende betekenis:
603F	(41)	
6040	(53)	

Type Omschrijving

BAS	BASIC-programma
INT	Integer array
SNG	Single precision array
DBL	Double precision array
STR	String array
FAM	Bestand, gemaakt met het Familiegeheugen
ROM	EPROM-programmer file
BIS	Plaatjes, gemaakt met de BIS-editor
PEP	Plaatjes, gemaakt met Picture Editor Program
ASS	Tekstbestand ("source file"), gemaakt met de Assembler van Ron Eindhoven (AVO Techniek)
OBJ	Machinetaalbestand ("Object file"), gemaakt met de Assembler van Ron Eindhoven (AVO Techniek)
ASM	Tekstbestand ("source file") in ASCII
SIM	Gegevensbestand logische simulator
REL	Reloceanbare (op andere adressen te plaatsen) Z80-code met symbooltabel (nog te definiëren)
PGM	Automatisch te starten Z80-machinetaalprogramma

6041	(42)	File-type; het type ROM-pack of toepassingsprogramma waaronder de file is aangemaakt. De karakters kunnen de volgende betekenis hebben:
------	------	---

64 (&H40)	= "@"	Plaatje van Peters Plaatjes Programma
65 (&H41)	= "A"	Familiegeheugen
66 (&H42)	= "B"	BASIC NL
68 (&H44)	= "D"	24K DISK BASIC
70 (&H46)	= "F"	FORTH
79 (&H4F)	= "O"	Overige
80 (&H50)	= "P"	"Stand alone"-programma (zonder ROM-pack in sleuf 1)
86 (&H56)	= "V"	BIS editor
87 (&H57)	= "W"	Tekstverwerker ("Wordprocessor")

6042	4E	Als het programmatype "P" is (machinetaalprogramma, dat zonder module in sleuf 1 van cassette wordt gelezen), Staat hier een code met de volgende betekenis:
------	----	--

D = Duits

S = Zweeds

U = Engels of Nederlands

6043	..)	Startadres voor "stand-alone"-programma's. Pointer naar het adres waar een machinetaalprogramma met identificatie "P" (op adres &H6041) in uitvoering moet worden genomen. Er hoeft geen ROM-pack in sleuf 1 aanwezig te zijn.
6044	..)	In dit adres wordt ook de lengte van een BASIC-programma gekopieerd (zie &H177D).
6045	(..)	Laadadres voor "stand-alone"-programma's. Pointer naar het adres waar een machinetaalprogramma met identificatie "P" (op adres &H6041) in het geheugen moet worden gezet. Er hoeft geen ROM-pack in sleuf 1 aanwezig te zijn.
6046	(..)	
6047	(20)	Tweede 8 karakters van de file-naam. BASIC NL controleert alleen het eerste karakter van de eerste reeks. Niet-ingevulde posities zijn gevuld met spaties (&H20 = 32).
----	(20)	
604E	(20)	
604F		Bloksteller van te lezen, te schrijven en heen of terug te spoelen blokken. Deze worden geteld aan de hand van op de band aanwezige signalen.

Cassette

6050	(Commandovlag voor de cassettemotor; er worden 2 bits gebruikt door de Monitor: bit 2 = 0 zet de motor uit bit 2 = 1 zet de motor aan bit 3 = 0 niet (meer) schrijven bit 3 = 1 (blijf) schrijven
6051	(..)	Tijdelijke opslagadressen (4) voor casetteroutines. Wordt gebruikt door de monitor.
----	(..)	
6054	(..)	

Stack pointer

6055	(..)	Op dit adres wordt de oude waarde van de stack pointer bewaard, als de tweede geheugen-"bank" wordt gebruikt.
6056	(..)	

NOP

6057	(00)	
----	(00)	
605B	(00)	Deze vijf posities worden niet gebruikt.

Geheugen

605C	(01)	Bij het opstarten controleert de monitor hoeveel geheugen aanwezig is en zet het resultaat hier neer. De getallen hebben de volgende betekenis:
------	------	---

- 01 16 Kbyte
- 02 32 Kbyte
- 03 48 Kbyte of meer

Disk

605D (01)

Als bij het opstarten een Disk Controller en een diskette-station met schijf en gesloten klepje aanwezig zijn, leest de monitor de eerste twee sporen van de schijf in. Daarna controleert de monitor of dat een systeemschijf was. Op adres &HE000 in de tweede geheugenbank moet dan &HF3 staan (zie &H0EC6 e.v.). Is dat niet het geval, dan zet de monitor hier 00 neer, in alle andere gevallen 01.

Er hoeft bij BASIC NL niet per se een systeemschijf aanwezig te zijn. De eerste twee sporen worden ingelezen, onverschillig wat erop staat, en weggeschreven op de adressen &HE000 tot &FFFFF in de tweede geheugen-"bank".

Bij gebruik van JWS-DOS (voor de monitor is dat geen systeemschijf) worden eveneens de eerste twee sporen ingelezen en weggeschreven van &HE000...FFFF in de tweede geheugen-"bank". Op spoor 1 staat het besturingssysteem; op spoor 2 de index. Op adres &H605D komt 00 te staan.

00 Ingelezen schijf is geen PDOS-systeemschijf.

01 Geen Disk Controller, geen Disk Drive, geen schijf in Disk Drive A, Disk Drive niet ingeschakeld, klepje open of PDOS-systeemschijf ingelezen.

Printer

605E 1D

Adres van de eerste printer-vertaal tabel (= &H1F1D). BASIC NL heeft geen eerste printer-vertaal tabel (de inhoud van &H1F1D = 00).

605F 1F

Cassette

6060 (00)

Vlagadres voor de cassettestatus, gebruikt door de monitor. De inhoud van dit adres mag niet door de gebruiker worden veranderd.

Er zijn 5 bits in gebruik. Als alle bits = 0 zijn, is er geen fout opgetreden. Anders geldt:

bit 0 = 1 geen startmerk

bit 1 = 1 niet (meer) schrijven

bit 2 = 1 zend cassette-ERROR-code naar de printer

bit 4 = 1 er is een Begin Of Tape-error (BOT-error)

bit 5 = 1 zet de motor uit

6061 (...)

Dit adres wordt door de casetteroutines van de monitor gebruikt om de stack pointer te bewaren.

6062 (...)

Tijdelijke opslag voor de casetteroutines van de monitor.

6063 (...)

Laadadres. Pointer naar het adres van waar af data van cassette in het geheugen moet worden geladen.

6064 (...)

6065 (...)

Adres	Inhoud	Omschrijving
6066	(..)	Bloklengte. Dit adres bevat de lengte van het laatst ingelezen of weggeschreven blok.
6067	(..)	
6068	(30)	Beginadres van de cassette-header of "file descriptor". Deze begint meestal op &H6030.
6069	(60)	
606A	(20)	Lengte van de cassette-header. Deze is altijd 32 bytes lang (&H0020).
606B	(00)	
606C	(..)	Deze adressen bevatten het aantal nullen waarmee een onvolledig blok moet worden aangevuld tot 1024.
606D	(..)	
606E	(..)	Hier staat de lengte in blokken van de cassette file.
606F	(..)	
 <u>Disk</u>		
De adressen &H6070...6086 worden bij het opstarten gevuld door de monitor. De inhoud komt van &H0FE8...0FFE.		
6070	(00)	Transfer-adres voor PDOS (&HE000 in de tweede geheugen- "bank").
6071	(E0)	
6072	09	Lengte opdracht (is altijd &H09).
6073	(42)	Functie-opdracht. 42 = lezen van schijf 45 = schrijven naar schijf
6074	(01)	Nummer van de Disk Drive
6075	(01)	Spoornummer
6076	(00)	Schijfkant
6077	(..)	Sectornummer (0...&H0F)
6078	01	Transmissiesnelheid ("baudrate") (is altijd 01)
6079	10	Aantal sectoren per spoor (is altijd &H10 = 16)
607A	0E	Ruimte tussen de sectoren ("gap space") (is altijd &H0E)
607B	(00)	Data lengte
607C	03	Zoeken: aantal bytes van de opdracht (altijd &H03)
607D	0F	code opdracht (&H0F)
607E	(01)	nummer van de disk drive
607F	(..)	sectornummer
6080	02	Recall: aantal bytes van de opdracht (altijd &H02)
6081	07	code opdracht (&H07)
6082	(01)	nummer van de disk drive
6083	03	Specificatie: aantal bytes van de opdracht (altijd &H03)
6084	03	code opdracht (&H03)
6085	60	parameter 1,2 (&H60)
6086	34	parameter 3,4 (&H34)

Adres	Inhoud	Omschrijving
6087	(..)	
----	(..)	Werkadressen voor de monitor
608D	(..)	
608E	(..)	Tijdelijke positie van de stack pointer bij het laden van
608F	(..)	de disk-constanten van &H0FE8 naar &H6070 (zie &H0E91 en &H0EDA)
		De adressen &H6090...60EA vormen een tabel van 91 constanten die bij het opstarten van BASIC NL in zijn geheel wordt geladen. De tabel staat van &H18A4 tot &H18FF in de eerste ROM van BASIC NL. Het laden gebeurt met de routine die begint op &H1F5A.
		<u>Scherm</u>
6090	21	Dit adres verwijst naar een vertaaltabel voor het scherm
6091	1A	die begint op &H1A21. De tabel verwisselt de ASCII-codes voor pondtekken en hekje ((&H23 en &H5F) en voor dakje en deelteken (&H5C en &H7E); dit in verband met de compatibiliteit.
		De tabel heeft de volgende vorm:
		1A21 aantal paren van de tabel (04)
		1A22 &H23
		1A23 wordt vervangen door &H5F
		1A24 &H5F
		1A25 wordt vervangen door &H23
		enz.
		<u>Printer</u>
6092	21	Adres van de printer-vertaaltabel. Hiervoor wordt dezelfde
6093	1A	tabel gebruikt als voor het scherm (zie &H6090...6091). De tekens die naar de printer gaan worden weer "terugvertaald".
		<u>Toetsenbord</u>
6094	1D	Dit adres verwijst naar een correctietabel die het mogelijk maakt om toetscodes in andere ASCII-codes om te zetten. De vorm is zoals is uitgelegd bij &H6090...6091. Met behulp van deze tabel (waarvan het startadres in &H1F1D moet staan) is het bij voorbeeld mogelijk om de WIS-SCHERM- en de TAB-toets zo te programmeren dat ze een piepje geven. Dit om binnen een programma een "zekere" lay-out te beschermen van INPUT of LINE INPUT.
6095	1F	Bij BASIC NL is deze tabel leeg (eerste positie = 00).
6096	14	Beginadres van de toetscodetabel (&H1814).
6097	18	De tabel zet de code van de toets om in een ASCII-code. Voor toets n kijkt BASIC NL naar de n-de positie van de tabel.

Omschrijving

Het is mogelijk in RAM een eigen toetscodetabel te definiëren en het beginadres (dat de toetscode 00 bevat, overeenkomend met de pijl naar links) op adres &H6096 en 6097 te zetten.

Scherm

6098 21
6099 1A Dit adres bevat het eerste adres van de scherm-vertaaltabel, die begint op &H1A21. Dit is weer dezelfde tabel als bij &H6090 en 6091. Hij is nodig om het afdrukken van het scherm door de printer mogelijk te maken.

LIST-regelnummer

609A (00)
609B (00) Regelnummer vanaf waar wordt geLIST bij gebruik van de gemakzuchtige of "rustige" lijster (SHIFT 5 op het kleine toetsenbord).

Printer

609C 11
609D 1F Op dit adres staat het eerste adres van de tweede printervertaaltabel (&H1F11), bedoeld voor printers die geen blokjes kunnen afdrukken. De tabel heeft dezelfde vorm als die in &H1A21 (zie &H6090 en 6091). Het aantal paren is 3 (eerste positie = 03). Het eerste en tweede paar zetten het pondteken om in een hekje en omgekeerd. Het derde paar zet &H7F (blokje) om in &H20 (spatie).

NOP

609E 00
609F 00 Deze adressen worden niet gebruikt.

Invoerveldlengte

60A0 FE Zowel in een programma als in de directe stand kan in dit adres de maximale hoeveelheid karakters per invoerregel worden gezet (bij voorbeeld bij INPUT en LINE INPUT. Normaal is dit &HFE+1 = 255. Verandert u dit in 00, dan kan ten hoogste één karakter per keer worden ingevoerd.

Printer

60A1 (00) In dit adres wordt het aantal afgedrukte regels van de printer bijgehouden. Zodra dit aantal gelijk is aan de inhoud van &H60AA (standaard &H42 = 66) wordt een aantal regels overgeslagen. Dit aantal staat in &H60A9 (standaard 6). Met LPRINT CHR\$(12) wordt het papier het resterende aantal regels getransporteerd en wordt deze teller op 01 gezet.

60A2 (03) Printertype. De inhoud heeft de volgende betekenis:
00 Halftransparant; karakters groter dan 128 worden doorgegeven, maar kunnen via de tabellen worden vertaald.
01 Als 00.

Adres	Inhoud	Omschrijving
		02 Viewdata zonder blokjes. 03 Viewdata met blokjes. De opgegeven "offset" wordt uitgevoerd (zie &H60A3 en 60A4). 04 Transparant. Alles wordt onveranderd doorgegeven.
60A3	80	"Offset" eerste 32 Viewdata-blokjes. Deze waarde wordt voor printertype 03 opgeteld bij de waarden die naar de printer worden gestuurd. Dit geldt voor de blokjes gedefinieerd door: grafisch ASCII 32...63.
60A4	60	"Offset" tweede 32 Viewdata-blokjes. Deze waarde wordt voor printertype 03 opgeteld bij de waarden die naar de printer worden gestuurd. Dit geldt voor de blokjes gedefinieerd door: grafisch ASCII 96...127.
		<u>Scherm</u>
60A5	(20)	Snelheid waarmee op het scherm wordt gePRINT met de SHIFT-toets ingedrukt. De inhoud kan elke waarde tussen 00 en 255 hebben. 255 (&HFF) laagste snelheid 0 (&H00) hoogste snelheid
60A6	(00)	Snelheid waarmee op het scherm wordt gePRINT zonder dat de SHIFT-toets is ingedrukt. De inhoud kan elke waarde tussen 00 en 255 hebben. 255 (&HFF) laagste snelheid 0 (&H00) hoogste snelheid
		<u>Zoek-karakter</u>
60A7	(3A)	Karakter dat in de EDIT-stand met SHIFT-pijl-naar-rechts wordt gebruikt om een karakter te zoeken of te verwijderen. Bij "S" ("search") of "K" wordt dit adres gevuld met de ASCII-code van deze letters. Anders wordt het eerste deelteken gezocht (&H3A = :).
		<u>NOP</u>
60A8	00	Dit adres wordt niet gebruikt.
		<u>Printer</u>
60A9	(06)	Aantal regels dat de printer overslaat aan het einde van een pagina ("skip over perforation"). 00 op dit adres betekent dat geen regels worden overgeslagen. Zie ook &H60A1.
60AA	(42)	Aantal regels dat op een pagina moet worden afgedrukt. Het totale aantal regels per pagina is te berekenen door de inhouden van &H60A9 en &H60AA op te tellen . Bij het opstarten is dat $6 + 66 (=&H42) = 72$. Dit is het normale aantal regels voor een pagina A4. Er is nogal wat papier in de handel dat iets korter is dan A4 en dat in totaal 66 regels per pagina heeft. Vul dan op dit adres 60 (&H3C) in.

Omschriiving

60AB (50)	Aantal karakters per regel dat de printer afdrukt. In tegenstelling tot de vensterbreedte (zie &H60B0) staat 80 (&H50) hier ook voor 80 posities (dus niet 79).
	<u>Cassette</u>
60AC (00)	Dit adres maakt het mogelijk de cassettereorder bepaalde dingen te laten doen. Een POKE naar dit adres werkt maar eenmaal.
1 (&H01)	Houdt bij de eerstvolgende CLOAD-opdracht het terugspoelen tegen.
80 (&H50)	Zorgt ervoor dat tijdens het uitvoeren van een programma zo nodig de vraag "Hier overheen ?" wordt gesteld. De klemtoon valt op <u>tijdens een programma</u> , want in de directe stand vraagt CSAVE wel altijd toestemming om te mogen overschrijven.
128 (&H80)	Er wordt niets gevraagd bij CSAVE.
	<u>Scherm</u>
60AD (00)	Venster-kantlijn. Het aantal posities links van het venster.
60AE (00)	Het aantal regels boven het venster.
60AF (17)	Het aantal regels van het venster + 1. De waarde 00 geeft 1 regel; de waarde 23 (&H17) geeft 24 regels.
60B0 (27)	Breedte van het venster + 1. De waarde 00 geeft een breedte van 1 positie; de waarde 39 (&H27) geeft een breedte van 40 posities; 79 (&H4F) geeft 80 posities. Dit is ook de breedte waarmee LISTings op het scherm verschijnen.
60B1 (...)	Geheugenadres van de cursor (bij voorbeeld &H5000). Het cursor-adres kan als volgt worden berekend:
60B2 (...)	adres = &H5000 + PEEK(&H60B3) + 80*PEEK(&H60B4).
60B3 (...)	Positie van de cursor op de regel. Deze positie is op te vragen met PRINT POS(X), waarin "X" een dummy-argument is (elke andere letter werkt ook).
60B4 (...)	Regelnummer waarop de cursor zich bevindt.
60B5 00	Dit adres wordt niet meer gebruikt. Bij BASIC UK bevatten &H60B4 en 60B5 de positie van de cursor in het tweede video-geheugen (&H5700...5FFF). Meestal was de inhoud van dit adres gelijk aan die van &H60B1 en 60B2.
60B6 (01)	Op dit adres wordt aangegeven of kleine letters automatisch moeten worden omgezet in grote. Omschakelen gebeurt met SHIFT TAB. Bij elke SHIFT TAB wordt de inhoud met 1 verhoogd. De inhoud heeft de volgende betekenis: even kleine letters worden omgezet in grote oneven geen omzetting van kleine letters in grote

<u>Adres</u>	<u>Inhoud</u>	<u>Omschrijving</u>
		<u>Sprongadres</u>
60B7 C3 A2 12		JP 12A2 Een sprong naar dit adres geeft een sprong naar de "scroll"-routine die begint op &H12A2.
		<u>NOP</u>
60BA (00)		Dit adres wordt niet gebruikt.
60BB (00)		Dit adres wordt niet gebruikt.
		<u>Scherm en toonstring</u>
60BC (00)		Op dit adres wordt een vlag gezet als CHR\$(4) (om iets op een bepaalde plaats op het scherm te zetten) of als CHR\$(23) moet worden uitgevoerd (toonstring). De inhoud is &H0D bij CHR\$(4) (deze wordt geladen in &H10F3: LD A,0D) en &H1D bij CHR\$(23) (geladen in &H10F7: LD A,1D).
60BD (00)		In dit adres komt de Y-waarde bij het uitvoeren van CHR\$(4) CHR\$(Y)CHR\$(X) en de T-waarde bij uitvoeren van CHR\$(23) CHR\$(T)CHR\$(N1)..... Deze waarden worden ingevuld in &H10AA.
		<u>Tabulator</u>
60BE (09)		TAB-disable. Dit adres maakt het mogelijk om de moeilijk te controleren TABs af te vangen. 09 TAB toegestaan 00 TAB niet toegestaan
		<u>NOP</u>
60BF (00)		Dit adres wordt niet gebruikt.
		<u>Sprongadressen</u>
60C0 C3 86 13		Deze tabel bevat een aantal spronginstructies naar machine-taalroutines. Elk spronginstructie beslaat drie opeenvolgende adressen. Op de eerste positie staat C3 (= JP). We vermelden steeds alleen de eerste positie van de spronginstructies. De bedoeling van deze spronginstructies is dat u het adres waar naartoe wordt gesprongen (het adres achter C3) kunt vervangen door het beginadres van een eigen machinetaal-routine die hetzelfde op een andere manier doet.
60C3 C3 9A 13		JP 1386. Sprong naar een <u>schermroutine</u> op &H1386 die een karakter, waarvan de ASCII-code in register A van de Z80 staat, op het scherm zet. LIST, PRINT en EDIT springen naar dit adres.
		JP 139A. Sprong naar een <u>printerroutine</u> op &H139A. Deze routine biedt een karakter, na verwerking, aan de printer-driver aan. LPRINT en LLIST springen naar dit adres. In register A staat de ASCII-code van het af te drukken karakter.

<u>Adres</u>	<u>Inhoud</u>	<u>Omschrijving</u>
60C6	C3 12 1A	JP 1A12. Sprong naar een <u>printerroutine</u> die het af te drukken karakter naar de printer stuurt.
60C9	C3 58 17	JP 1758. Sprong naar een <u>cassetteroutine</u> , gebruikt bij <u>CLOAD</u> . Er kan ook worden gesprongen naar bij voorbeeld een <u>schijfroutine</u> . De parameters staan op &H6130...614F.
60CC	C3 F0 16	JP 16F0. Sprong naar een <u>cassetteroutine</u> , gebruikt bij <u>CSAVE</u> . Er kan ook worden gesprongen naar bij voorbeeld een <u>schijfroutine</u> . De parameters staan op &H6130...614F.
		<u>Vlag</u>
60CF	(00)	De waarde &H84 op dit adres is een vlag voor "zet de volgende BASIC-regel in de EDIT-stand". Wordt gebruikt als in de EDIT-stand de toets "N" ("next") is ingedrukt.
		<u>Vervolg sprongadressen</u>
60D0	00 00 00	<p>Ruimte voor een sprong naar een eigen machinetaalroutine. Dit schijnbaar lege adres is hoogst nuttig, want bij uitvoering van <u>elke</u> BASIC-instructie komt de interpreter langs dit adres via de aanroep in &H27D3: CALL 1050. Van adres &H1050 springt BASIC naar dit adres, &H60D0. Staan op &H60D0 tot 60D5 uitsluitend nullen, dan loopt BASIC door tot &H60D6: de toetsenbord-statusroutine, en gaat de uitvoering van de desbetreffende BASIC-instructie door.</p> <p>Zet u op &H60D0 een onderschep-opdracht, bij voorbeeld CALL 0026 (CD 26 00 = wacht op het indrukken van een toets), dan wacht BASIC voor het uitvoeren van elke instructie op het indrukken van een toets. Er is dan een "single step BASIC" ontstaan.</p> <p>Als JWS-DOS is geladen staat hier: CD 88 EA = CALL EA88.</p> <p>Bij het uitvoeren van elke BASIC-instructie (óók in de directe stand !) springt BASIC even naar adres &HEA88 om te kijken of één van de functies van JWS-DOS is aangeroepen.</p>
60D3	00 00 00	<p>Ruimte voor een sprong naar een eigen machinetaalroutine. Voor dit adres geldt hetzelfde als voor &H60D0.</p> <p>Beide adressen mogen alleen worden gevuld met een sprong (JP xx xx) naar eigen machinetaalroutines die eindigen met de terugsprong JP 60D6, óf met aanroep routines (CALL xx xx) omdat anders de executielus van de BASIC-interpreter wordt onderbroken.</p>
60D6	C3 29 00	<p>JP 0029. Sprong naar de <u>toetsenbord-statusroutine</u> op &H0029, die controleert of een toets ingedrukt is geweest. Dit sprongadres kan worden gebruikt om "toetsen af te vangen". De code van de laatst ingedrukte toets staat op &H600D.</p> <p>Carry-vlag gezet: de STOP-toets is ingedrukt geweest Zero-vlag gezet: er is geen toetscode in &H600D</p>

Adres	Inhoud	Omschrijving
		Als in &H60D0 en &H60D3 geen eigen machinetaalroutines worden aangeroepen, springt BASIC bij elke instructie van &H60D0 via zes NOPs naar dit adres.
60D9	C3 56 19	<u>JP 1956.</u> Sprong naar de BASIC-invoerroutine op &H1956. Deze begint met CALL 0026: wacht op een toetsindruk.
60DC	C3 7B 24	<u>JP 247B.</u> Sprong naar string-SAVE-routine.
60DF	00	Dit adres wordt niet gebruikt.
60E0	C3 9B 1A	<u>JP 1A9B.</u> Sprong naar de routine die de ingetikte regels verwerkt (bij EDIT en het invoeren van BASIC-regels).
60E3	C9 00 00	<u>RET.</u> Dit sprongadres is in gebruik als bij EDIT de toets "N" (next) wordt ingedrukt. De instructie RET betekent dat BASIC terug弹簧t naar het adres volgend op dat waar hij vandaan kwam.
60E6	C9 00 00	<u>RET.</u> Dit adres wordt door de monitor gebruikt bij de opdracht RST 10 (ReStart 10). RST 10 (zie &H0010) leidt tot JP 1013, en dat leidt tot JP 60E6 = RET. RST 10 is de enige nog vrije RST-instructie. Bij gebruik van de service module wordt dit adres gebruikt als "debug breakpoint" om de Z80-registers te bekijken.
60E9	C9 00 00	<u>RET.</u> Dit adres biedt een uitwijkmöglichheid voordat de stuurcodes CHR\$(1)...CHR\$(31) worden verwerkt. Als JWS-DOS is geladen, staat hier JP EE5F. Vanaf &HEE5F wordt gekeken of één van de functietoetsen is ingedrukt.
		Dit adres wordt ook gebruikt voor de interpretatie van een eigen functietoetsentabel.
60EC	00 00 00	Vrij voor eigen sprongadres
60EF	00 00 00	Vrij voor eigen sprongadres
		<u>Cursor</u>
60F2	(00)	Horizontale positie van het cursorpunt ten opzichte van het venster.
60F3	(00)	Verticale positie van het cursorpunt ten opzichte van het venster. <u>Opmerking:</u> cursor en cursorpunt hebben niet dezelfde positie. Als de cursor ergens in een regel staat, is het cursorpunt meestal de positie aan de linkerkantlijn van die regel. Zie ook gebruiksaanwijzing blz. 24.
60F4	(00)	Verticale positie van de cursor ten opzichte van de regel waarop de deelLIST (gemakzuchtige lijster) begint. De computer weet hierdoor hoe vaak de regels op het scherm moeten "scrollen" voordat "meer" op de onderste regel moet worden gezet (zie &H119D).

Adres	Inhoud	Omschrijving
60F5	(00)	
60F6	(00)	Deze adressen bevatten het regelnummer van de volgende regel als tijdens EDIT de "N" (next line) is ingedrukt. Als er geen volgende regel meer is, wordt de inhoud 0000.
		<u>NOP</u>
60F7	(00)	
----	(00)	
60FB	(00)	Deze vijf adressen worden niet gebruikt.
		<u>Cursor</u>
60FC	(00)	
60FD	(00)	Hier wordt tijdelijk de cursorpositie opgeslagen als tijdens EDIT de regel naar het scherm wordt overgebracht (zie &H1DE8).
		<u>NOP</u>
60FE	(00)	Deze adressen worden niet gebruikt.
60FF	(00)	
		<u>Cassette</u>
6100	(..)	Hier wordt de eerste (significante) letter van de naam gezet bij CLOAD* (zie &H1DDF en &H17C8).
6101	00	Wordt niet gebruikt.
6102	00	Wordt niet gebruikt.
6103	(20)	Deze 8 adressen vormen het eerste deel van de hulp-hulp-"header". In deze hulp-hulp-"header" worden de eerste 8 letters verzameld van de naam bij CSAVE, voordat die wordt overgebracht naar de hulp-"header" (&H6130...614F). Niet-gebruikte adressen hebben de inhoud &H20 (= 32 = spatie).
----	(20)	
610A	(20)	
610B	(20)	Tweede deel van de hulp-hulp-"header", waarin de tweede 8 letters van de naam bij CSAVE tijdelijk worden bewaard.
----	(20)	
6112	(20)	
6113	(00)	
----	(00)	Vrij voor eigen routines (29 adressen).
612F	(00)	
		De adressen &H6130...614F vormen de hulp-"header". Hierin worden door de monitor alle gegevens verzameld voordat die bij CSAVE worden overgebracht naar de echte "header" in de adressen &H6030 ...604F. De functie van de adressen in de hulp-"header" correspondeert steeds met die van de echte "header". &H6130 correspondeert dus met &H6030 enzovoort.

Adres Inhoud

Omschrijving

Bij CSAVE en CLOAD wordt altijd met de echter "header" gewerkt. U hoeft zelf nooit de inhoud van &H6130...614F te vullen, want dit gebeurt door de monitor. Als u BASIC gebruikt, en ook bij "stand-alone"-programma's (zonder BASIC-module) kunt u het voorbereidende werk in &H6130...614F vermijden door de gewenste parameters direct in &H6030...604F te zetten.

6130	(..)	In deze adressen &H6130 en &H6131 zet de monitor het transferadres. Bij BASIC NL is dit adres gewoonlijk &H6547.
6132	...	In deze adressen zet de monitor de totale lengte van het programma of bestand.
6133	(..)	
6134	(00)	In deze adressen zet de monitor het zinvolle aantal bytes van het programma of bestand.
6135	(04)	
6136	...	Eerste 8 karakters van de file-naam.
----	(20)	
613D	(20)	
613E	(42)	
613F	(41)	"Extension"; de soort file.
6140	(53)	
6141	(42)	File-type; het type ROM-pack of toepassingsprogramma waaronder de file is aangemaakt.
6142	4E	Als het programmatype "P" is, zet de monitor hier een code voor de taal neer.
6143	(..)	Startadres voor "stand-alone"-programma's.
6144	(..)	
6145	(..)	Laadadres voor "stand-alone"-programma's.
6146	(..)	
6147	(20)	
----	(20)	Tweede 8 karakters van de file-naam.
614E	(20)	
614F	(00)	Record nummer
		<u>Kladblokruimte</u>
6150	(00)	Kladblokruimte van 176 bytes.
----	(00)	Vrij voor het opslaan van bij
61FF	(00)	voorbeeld tussenresultaten.

Adres Inhoud

Omschrijving

De adressen &H6200...6260 vormen een tabel met 97 constanten die bij het opstarten van BASIC wordt geladen vanuit de adressen &H23C7...2427 (zie &H1F66).

Zachte start in BASIC

6200 C3 C6 1F

JP 1FC6. Via dit adres kunt u vanuit een machinetaalroutine op een nette manier terugkeren naar BASIC, waarbij BASIC met een schone stack moet beginnen. Een eventueel in het geheugen aanwezig BASIC-programma wordt niet gewist.

Uitvoerpoort

6203 D3 00

OUT 00

6205 C9

RET

Deze routine maakt het mogelijk vanuit BASIC een waarde naar een uitgangspoort te sturen. In plaats van "00" op adres &H6204 vult BASIC het poortnummer in. De inhoud van de accumulator wordt dan gestuurd naar de uitgangspoort waarvan het nummer is ingevuld op &H6204. Dit nummer en de inhoud van A worden door de interpreter ingevuld als vertaling van de BASIC-opdracht OUT X,A. Daarna wordt naar adres &H6203 gesprongen (zie bij voorbeeld &H32AC: JP 6203).

Na het uitvoeren van de instructie zorgt C9 (RET) voor de terugkeer naar BASIC.

De routine wordt gebruikt bij OUT en WAIT.

Aftrekroutine

6206 D6

----(..)

6213 C9

Op deze adressen staat een aftrekroutine ten behoeve van de functie "deel door 10". Deze wordt bij voorbeeld aangeroepen in &H3658: CALL 6206. De routine gebruikt de adressen &H6207, 620B, 620F en 6212 voor tijdelijke opslag.

RND

De adressen &H6214...623A worden gebruikt voor het genereren van (quasi-)willekeurige getallen met de functie RND (zie de functie die begint op &H4462).

6214

Teller 00...FF. Geeft het aantal keren aan dat RND(1) is gebruikt.

6215(..)

Teller 00...03. Wijst naar één van de posities binnen een groep van 4 waarden in onderstaande tabel.

6216(..)

Teller 00...07. Wijst naar één van de 8 groepen van 4 waarden in onderstaande tabel.

6217 35

Tabel van 8 x 4 waarden, die worden gebruikt voor het berekenen van RND

6236 98

Adres	Inhoud	Omschrijving
6237	(..)	In deze vier adressen wordt een gecodeerde waarde opgebouwd voor de functie RND.
----	(..)	
623A	(..)	Omdat bij het berekenen van elke volgende waarde voor RND(1) gebruik wordt gemaakt van de vorige, kan de random-generator worden "geschud" met de instructie POKE &H6238,PEEK(&H6010).
		<u>Invoerpoort</u>
623B	DB 00	IN 00
623D	C9	RET
		Deze routine maakt het mogelijk vanuit BASIC een waarde van een ingangspoort uit te lezen. In plaats van "00" op adres &H623C wordt door BASIC het poortnummer ingevuld. Daarna wordt naar adres &H623B gesprongen en wordt de momentele waarde van de ingangspoort in de accumulator gezet. Na het uitvoeren van de instructie zorgt C9 (RET) voor de terugkeer naar BASIC.
		Deze routine wordt door de interpreter aangeroepen als vertaling van de BASIC-opdracht V=INP(X). De ingelezen waarde in de accumulator wordt toegekend aan de variabele V.
		<u>USR</u>
		Tabel van adressen van de functies USR0 tot en met USR9. Als geen gebruikersfuncties zijn gedefinieerd, staat op alle adressen het adres &H289C van een foutafhandelingsroutine.
623E	(9C 28)	USR0
6240	(9C 28)	USR1
6242	(9C 28)	USR2
6244	(9C 28)	USR3
6246	(9C 28)	USR4
6248	(9C 28)	USR5
624A	(9C 28)	USR6
624C	(9C 28)	USR7
624E	(9C 28)	USR8
6250	(9C 28)	USR9
		<u>Foutcode</u>
6252	(..)	BASIC-foutcode. Hier staat de code van de laatst opgetreden fout in BASIC (in een programma of in de directe stand).
		<u>Printer</u>
6253	(00)	Printer-kolomteller. Op dit adres houdt de P2000 bij op welke positie van de regel de afdrukkop van de printer zich volgens zijn berekening moet bevinden. Dat hoeft niet per se de werkelijke positie te zijn. De inhoud van dit adres is op te vra-

<u>Adres</u>	<u>Inhoud</u>	<u>Omschrijving</u>
6254	(00)	gen met ? LPOS(X), waarin X een "dummy argument" is (met elke andere letter gaat het ook).
6255	(FF)	Uitvoer-indicator. Op dit adres staat waar de uitvoer naar toe moet. Dit adres wordt onder meer gevuld in de LPRINT-routine (zie &H2AE0). 00 uitvoer naar het scherm 01 uitvoer naar de printer
6256	(0E)	<u>Scherm</u> Hier staat de maximale <u>logische</u> lengte van een regel op het scherm als de P2000 een <u>getal</u> moet afdrukken (&HFF = 255). De inhoud van dit adres bepaalt of Carriage Return (CR) en Line Feed (LF) moeten worden gegeven, en wel in samenhang met de cursorpositie op &H63B4. Dat gebeurt als volgt: als (63B4) + de lengte van het getal + 1 >(6255) dan wordt eerst CR + LF uitgevoerd voordat het getal wordt afgedrukt. Zie bij voorbeeld &H2B31.
		Bij de opdracht PRINT 1,2,3,4 bepaalt de inhoud van dit adres samen met de inhoud van &H63B4 of het volgende getal 14 posities verderop op dezelfde regel moet worden gePRINT of dat CR + LF moet worden gegeven. Ook bepaalt de inhoud van dit adres hoeveel spaties er tussen de getallen moeten worden afgedrukt. Dat gebeurt overeenkomstig de volgende twee regels: 1 Als (&H63B4) \geq (&H6256), doe dan CR + LF. 2 Als (&H63B4) $<$ (&H6256), PRINT dan 14 - (&H63B4) spaties. De spaties worden gePRINT vanaf de momenteel geldende cursorpositie. Hetzelfde geldt voor het afdrukken van strings, bij voorbeeld PRINT"AAAAAA","BBBBBB","CCCCC". Het verschil tussen getallen en strings is dat getallen worden voorafgegaan door een spatie of een minteken en worden gevolgd door een spatie. Het getal 1 vraagt dus 3 posities; 10 vraagt 4 posities enzovoort. Strings vragen even veel posities als ze tekens tellen. Normaal staat in dit adres &H0E (= 14), maar er mag elke waarde staan tussen 0 en 255. Dit werkt als volgt. Na het afdrukken van AAAAA staat de cursor op positie 6 van de regel. Deze waarde wordt afgetrokken van de inhoud van &H6256. Is de uitkomst positief, dan maakt BASIC de inhoud van &H63B4 gelijk aan 14. Dit betekent dat BBBB vanaf de 14e positie wordt afgedrukt. Is de uitkomst 0 of negatief, dan gaat de cursor naar de volgende regel. Is na het PRINTen van BBBB (&H6256) - (&H63B4) nog steeds positief, dan wordt de inhoud van &H63B4 gelijk aan 28 gemaakt. Na het "uitvoeren" van de komma is de inhoud van &H63B4 dus altijd nul of een veelvoud van 14. Maakt u de inhoud van &H6256 = 255, dan begint de volgende string dus steeds 14 posities verder. Als de fysieke regellengte (40 of 80) bereikt is, telt BASIC door op de volgende regel.

Adres	Omschrijving
6257	Vlagadres. Is de vlag = 0, dan moet de cursor vooraan de regel worden gezet door CR + LF.
	<u>CLEAR</u>
6258 (...)	De inhoud van deze adressen geeft het einde van de vrije geheugenruimte en het begin van de stringruimte aan. De grootte van de stringruimte en het hoogste adres dat door BASIC mag worden gebruikt, worden bepaald met CLEAR n1,n2. In &H6258...6259 komt de waarde n2-n1 te staan (voor n2: zie &H63B8). De stringruimte bevindt zich in het hoogste gedeelte van het geheugen. Als n1 niet wordt opgegeven, worden standaard 50 plaatsen gereserveerd (vanaf &H9FCD, &HDFCD of &HFFCD, afhankelijk van de inhoud van &H605C).
	<u>Regelnummer</u>
625A (FF)	Op deze adressen staat het nummer van de BASIC-regel die in uitvoering is. Bij het opstarten van BASIC wordt dit regelnummer op &FFFF (= -1) gezet (in &H2521).
625B (FF)	
	<u>BASIC</u>
625C (47)	Geeft het begin van het BASIC-programma aan. Meestal is dat &H6547.
625D (65)	
	<u>Onbekend</u>
625E (21)	De functie van dit adres is niet zeker. Vermoedelijk wordt het gebruikt om het begin van de volgende buffer aan te geven.
	<u>Buffer</u>
625F (...)	Hier blijft het eerste token of teken van de regel staan die het laatst is uitgevoerd, geLIST of geEDIT of van de opdracht die het laatst in de directe stand is uitgevoerd.
6260 (...)	Buffer (256 bytes) voor BASIC-regels bij het invoeren daarvan (zowel in een programma als in de directe stand) en bij LIST. Na afhandeling van een BASIC-regel worden de eerste twee posities nul gemaakt en kan een deel van de buffer voor andere doeleinden worden gebruikt.

635F (...)	
	<u>Onbekend</u>
6360 (00)	De functie van deze adressen is niet bekend. Niet alle adressen worden echter gebruikt.

63B3 (00)	
	<u>Scherm</u>
63B4 (...)	Hier staat de positie van de cursor op de logische regel. De inhoud kan worden opgevraagd met de BASIC-opdracht PRINT POS(0). Dit levert op: POS(0) = (63B4) + 1. Bij voorbeeld:

Adres	Inhoud	Omschrijving
		PRINT "12345";:PRINT POS(0) levert 6 op. BASIC telt door op de volgende fysieke regels. De inhoud van &H63B4 kan dus liggen tussen 0 en 255 en de uitkomst van POS(0) in principe tussen 1 en 256. Na elke opdracht die CR + LF tot gevolg heeft, wordt de inhoud van &H63B4 nul gemaakt.
		Dit adres wordt ook gebruikt om PRINT TAB(n) uit te voeren. Dat gebeurt als volgt: als n groter is dan (&H63B4), worden n - (&H63B4) spaties gePRINT vanaf de geldende cursorpositie (in &H60B3).
		En tenslotte wordt dit adres gebruikt voor opdrachten zoals PRINT 1,2,3,4 en PRINT "AAAA","BBBB","CCCC". Hierbij wordt de inhoud van &H63B4 0 of een veelvoud van 14 gemaakt (zie &H6256).
		De inhouden van &H63B4 en &H60B3 (de horizontale cursorpositie) zijn vaak aan elkaar gelijk. Ze worden dan ook gelijktijdig opgehoogd. (&H60B3) is echter de werkelijke cursorpositie en (&H63B4) is meer een kopie, die door CR + LF gelijk aan nul wordt gemaakt.
		<u>Variabele</u>
63B5	(..)	<p>Vlagadres voor het opzoeken van een variabele.</p> <ul style="list-style-type: none"> 01 voeg de variabele aan de lijst toe 00 zoek het adres van de variabele in de lijst
		<u>FAC-type-indicator</u>
63B6	(04)	<p>Type-indicator voor de Floating Point Accumulator (FAC) (zie &H6509). In dit adres staat van welke soort variabele sprake is. Dit getal komt in register A te staan bij het aanroepen van een machinetaalroutine door middel van ?USR(X), maar wordt ook binnen BASIC gebruikt, bij voorbeeld voor het berekenen van SIN(X).</p> <p>De type-indicator heeft de volgende betekenis:</p> <ul style="list-style-type: none"> 02 2-byte integer (INT) 03 string (STR) 04 enkeleprecisiegetal met zwevende komma (SNG) 08 dubbeleprecisiegetal met zwevende komma (DBL)
		<u>DATA-vlag</u>
63B7	(00)	<p>Hier staat een vlag die aangeeft of de BASIC-instructie DATA aanwezig is.</p> <ul style="list-style-type: none"> 00 geen DATA 49 wel DATA (zie &H26FE)

Adres	Inhoud	Omschrijving
		CLEAR
63B8	(..)	Hier staat het hoogste adres dat door BASIC mag worden gebruikt. Het adres wordt gevuld met CLEAR n1,n2, waarbij n2 in deze adressen wordt opgeborgen. Voor n1: zie &H6258.
		String
63BA	(..)	Adres van een pointer die wijst naar de eerstvolgende vrije opslagruimte van 3 byte in de hierna volgende stringtabel.
63BB	(63)	
63BC	(..)	Tabel voor stringopslag. Als BASIC bij stringopdrachten (bij voorbeeld bij het optellen van strings of het PRINTen van foutmeldingen) tussenresultaten wil opbergen, gebeurt dat in deze tabel. De totale lengte van de tabel (30 byte) is opgedeeld als 10 x 3 byte. Elke drie byte vormen een zogenaamde "string descriptor":
---	(..)	
63D9	(..)	<ul style="list-style-type: none"> 1e byte lengte van de string 2e byte lage adres (LSB) van de plaats waar de string zelf is opgeborgen. 3e byte hoge adres (MSB) van de plaats waar de string zelf is opgeborgen.
63DA	(..)	In deze adressen worden de gegevens van een string descriptor verzameld voordat deze in zijn geheel wordt overgebracht naar de string-opslagtabel.
63DB	(..)	
63DC	(..)	
63DD	(..)	Het adres van de eerstkomende (lege) positie in de stringruimte (dat wil zeggen de stringruimte bovenin het geheugen - zie &H6258).
63DE	(..)	Bij NEW wordt de inhoud van &H63DD...63DE gelijk gemaakt aan de inhoud van &H63B8...63B9, die met CLEAR is ingesteld.
		Diversen
63DF	(..)	Deze adressen worden voor twee doeleinden gebruikt:
63E0	(..)	<ol style="list-style-type: none"> 1 aangeven van het adres van het token dat in uitvoering is (zie &H2D06); 2 als vlag bij het formateren bij PRINT USING: bij omzetting van een hexadecimaal getal in een ASCII-code bevat deze vlag de aanwijzingen hoe de vorm van het getal moet worden (bij voorbeeld exponentiële notatie, decimale komma's en dergelijke). Een nul hier betekent dat geen formateren nodig is (strikte omzetting van binair in ASCII) (zie &H3EC9).
		Array-variabelen
63E1	(..)	Deze adressen worden gebruikt bij het invoeren van array-variabelen. Zij bevatten het adres van de volgende variabele.
63E2	(..)	

Adres	Inhoud	Omschrijving
		<u>Pointer</u>
63E3	(..)	Dit adres wijst naar de plaats waar BASIC zich in de regel van executie bevindt, of naar het einde van een BASIC-opdracht bij afhandelen van FOR.
		<u>DATA</u>
63E5	(..)	De inhoud van dit adres geeft het regelnummer aan waarin de laatste DATA zijn gelezen.
63E6	(..)	
		<u>Diversen</u>
63E7	(..)	De inhoud van dit adres heeft de volgende betekenis: 00 geen FOR in uitvoering (inhoud wordt 00 gemaakt door NEW (zie &H1CF4)) 01 adres gezocht van <u>array</u> -variabele 64 FOR in uitvoering (zie &H272C) 80 FN in uitvoering (zie &H313A) &H64 en &H80 geven tevens aan dat <u>array</u> -variabelen niet zijn toegestaan (zie &H47ED)
		<u>NOP</u>
63E8	(00)	Dit adres wordt niet gebruikt.
		<u>Diversen</u>
63E9	(00)	Hier staat een vlag die aangeeft dat de invoerfase (inputfase, die vooraf gaat aan de executiefase) gaande is. 00 inputfase gaande (zie &H25AA) #00 geen inputfase gaande
63EA	(00)	Dit is eveneens een vlagadres. De inhoud heeft de volgende betekenis: 00 de BASIC-instructie INPUT wordt uitgevoerd 01 de BASIC-instructie READ wordt uitgevoerd (zie &H2C2D) Het adres wordt ook gebruikt bij PRINT USING om ruimte te maken tussen string en variabele.
63EB	(..)	Ook deze adressen worden voor twee doeleinden gebruikt:
63EC	(..)	1 de inhoud bevat het beginadres van BASIC -1; dit adres wordt geïnitieerd tijdens NEW (zie &H1CA0); 2 de inhoud bevat het adres van de code in de buffer achter &H6260 die nu moet worden uitgevoerd (zie &H1EFB)
		<u>AUTO</u>
63ED	(00)	Vlagadres; geeft aan of AUTO R, Δ R in uitvoering is (hierin is R het eerste regelnummer en Δ R het increment (stapgrootte)). 00 geen AUTO in uitvoering #00 AUTO in uitvoering

Adres | Inhoud**Omschrijving**

63EE	(00)	Hier staat het regelnummer dat door de functie AUTO moet worden gebruikt (wordt gevuld in &H256B).
63EF	(00)	
63F0	(00)	Hier staat de stapgrootte bij de uitvoering van AUTO (wordt gevuld in &H2A96)
63F1	(..)	
		Diversen
63F2	...	
63F3	(62)	De inhoud van dit adres is het adres van de code in de buffer achter &H6260 als die gedurende de inputfase wordt uitgevoerd. Tijdens de executiefase staat hier het vigerende regelnummer.
63F4	(..)	
63F5	(..)	Op deze adressen wordt de stackpointer bewaard tijdens de executiefase.
		ERROR
63F6	(FF)	
63F7	(FF)	De inhoud van deze adressen geeft het nummer aan van de BASIC-regel waarin een fout is opgetreden (zie &H2480). Als geen fout is opgetreden, staat hier &FFFF.
		EDIT en LIST
63F8	(FF)	
63F9	(FF)	De inhoud van deze adressen is het nummer van de BASIC-regel die het laatst geEDIT of geLIST is. Deze regel kan op het scherm worden gebracht met EDIT . en met de toets DEF (SHIFT 0 op het kleine toetsenbord). Als het gevraagde regelnummer niet bestaat is de inhoud &FFFF en volgt de foutmelding "Undefined line number".
		ERROR
63FA	(..)	
63FB	(..)	De inhoud geeft het adres aan van het laatste byte dat is uitgevoerd (zie &H249B). Dit adres wordt ook gebruikt om het startadres aan te geven van een BASIC-regel waarin een ERROR is opgetreden (zie &H2A3D).
63FC	(00)	
63FD	(00)	Deze adressen bevatten het adres van de BASIC-regel waarheen moet worden gesprongen bij ON ERROR GOTO (zie &H24B2)
63FE	(00)	Een vlagadres dat aangeeft of de foutafhandelingsroutine is aangeroepen. FF er wordt een fout (ERROR) afgehandeld 00 er wordt geen fout afgehandeld. (Door RESUME wordt de vlag weer op 00 gezet (zie &H2A19))
		Diversen
63FF	(..)	
6400	(..)	De inhoud is het adres van het eerstvolgende token dat moet worden verwerkt.

Adres	Inhoud	Omschrijving
		Het adres wordt ook gebruikt om het adres van de decimale punt in de PRINT-buffer aan te geven (zie &H41C8).
6401	(..)	Hier wordt bij het uitvoeren van STOP en END het laatst uitgevoerde BASIC-regelnummer bewaard (zie &H24AB).
6402	(..)	
6403	...	De inhoud is het adres van het laatste byte dat werd uitgevoerd vóór STOP (zie &H1F56) of ERROR.
6404	(..)	
		<u>Pointers</u>
6405	(..)	De inhoud is een pointer naar het begin van de variabelenruimte, en tevens het eindadres van het BASIC-programma. Deze pointer kunt u verhogen als u achter het BASIC-programma machinetaal wilt opbergen met de bedoeling om die mee weg te schrijven op cassette.
6406	(..)	
6407	(..)	De inhoud van deze adressen is een pointer naar het beginadres van de array-ruimte (geDIMensioneerde variabelen).
6408	(..)	
6409	(..)	De inhoud van deze adressen is een pointer naar het eindadres van de array-ruimte. Dat is tevens het beginadres van de vrije ruimte waarin eigen data kunnen worden opgeslagen of machinetaal kan worden neergezet.
640A	(..)	
640B	(..)	De inhoud van deze adressen is de DATA-pointer. Dit is het adres van het byte volgend op het laatst gelezen DATA-statement. Door RESTORE wordt dit adres gevuld met het beginadres van BASIC minus 1 (doorgaans &H6546) (zie &H1D2C).
640C	(..)	
		<u>Typedeclaratie</u>
640D	(04)	Deze adressen vormen een typetabel van 26 bytes voor variabelen, voor elke letter van het alfabet één (in alfabetische volgorde). De eerste positie in de tabel geeft aan van welk type variabelen zijn waarvan de naam met A (of a) begint; de tweede positie geldt voor variabelennamen die met een B beginnen, enzovoort. Bij het opstarten van BASIC en bij NEW wordt voor alle 26 beginletters 04 ingevuld (enkeleprecisievariabele) (zie &H1CA5). De inhoud van deze adressen kan worden veranderd met DEFSTR, DEFINT, DEFSNG en DEFDBL.
----	(04)	
6426	(04)	
		02 integer 03 string 04 enkeleprecisievariabele 08 dubbeleprecisievariabele
		Deze getallen geven aan hoeveel bytes moeten worden gereserveerd voor de desbetreffende variabelen.
		<u>Buffer</u>
		De adressen &H6427...648E vormen een buffer van 104 bytes voor de opslag van variabelennamen en variabelen die door middel van DEF FN NAAM(X,Y,...) worden opgegeven.

Adres Inhoud	Omschrijving
6427	Dit adres bevat de stackpointer die aangeeft waar de buffer precies op de stack staat.
6428	
6429	...
642A	(..)
642B	...
----	(..)
648E	(..)
648F	(27)
6490	(64)
	Bij het opstarten van BASIC wordt hier het beginadres van de buffer (= &H6427) neergezet. Bij FN van een string wordt dit adres doorgegeven aan &H64FB.
	<u>Hulpbuffer</u>
6491	...
6492	(..)
6493	(..)
64F6	(..)
64F7	(..)
64F8	(..)
64F9	(..)
64FA	(..)
64FB	...
64FC	...
	De adressen &H6491...64F6 vormen een hulpbuffer van 102 bytes voor de tijdelijke opslag van variabelennamen en variabelen die door middel van DEF FN NAAM(X,Y,...) worden opgegeven. In deze hulpbuffer worden de gegevens bijeengebracht voordat ze worden overgebracht naar de echte buffer op &H6429...648E.
	De inhoud van dit adres is de gevulde lengte van de hulpbuffer. Een kopie gaat naar &H6429.
	De eigenlijke hulpbuffer. Vorm en inhoud zijn gelijk aan die van de hoofdbuffer op &H642B...648E.
	<u>Variabelen</u>
	Dit adres bevat een kopie van &H64FA
	De inhoud van dit adres bepaalt of een variabele moet worden opgezocht in de (normale) variabelenlijst (dan is de inhoud gelijk aan de inhoud van &H6407), of dat moet worden gezocht in de FN-variabelenlijst (de inhoud is dan meestal &H642B).
	Vlagadres dat aangeeft waar de variabele moet worden gezocht:
	00 er is geen onderzoek gaande naar FN-variabelen (zoek dus in de normale variabelenlijst)
	#00 er is wel een onderzoek gaande naar FN-variabelen
	De inhoud van dit adres kan &H648F zijn; de inhoud van dat laatste adres is weer &H6427; dat is het beginadres van de hoofdbuffer waarin een string moet worden opgeslagen als FN string is aangeroepen.

Adres	Inhoud	Omschrijving
		<u>Onbekend</u>
64FD	(00)	
----	(00)	
6506	(00)	De betekenis van deze adressen is nog niet bekend.
		<u>TRON/TROFF</u>
6507	(00)	Hier staat de zogenaamde "trace"-vlag. 00 TROFF #00 TRON (bij het intikken van TRON wordt de inhoud &HAF)
		<u>FAC I</u>
6508		Tijdelijke opslag bij rekenroutines bij het ontcijferen van getallen met drijvende komma. Gewoonlijk staat hier het laatste byte dat uit de positie voor het minst significante byte wordt geschoven.
6509	(..)	Eerste floating point ACCumulator (FAC I). Hierin worden de resultaten van berekeningen opgeslagen.
----	(..)	
6510	(..)	De uitkomsten worden als volgt opgeslagen: Adres Dubb. prec. Enk. prec. Integer
		6509 DP (LSB) 650A DP 650B DP 650C DP 650D DP SP (LSB) INT (LSB) 650E DP SP INT (MSB) 650F DP SP 6510 DP (MSB) SP (MSB)
		LSB = Least Significant Byte = byte met de laagste waarde. MSB = Most Significant Byte = byte met de hoogste waarde. De type-indicator staat in &H63B6.
6511	(..)	Teken van het getal bij rekenkundige bewerkingen.
6512		Tijdelijke opslag bij berekeningen in dubbele precisie.
		<u>FAC II</u>
6513	...	Deze buffer heeft dezelfde vorm, lengte en indeling als FAC-I (&H6509...6510). Hij wordt gebruikt voor het optellen van twee getallen die beide van dubbele precisie zijn (zie &H3ADB).
		<u>NOP</u>
651B	(00)	Dit adres wordt niet gebruikt.

Adres | InhoudOmschrijvingPRINT-buffer

651C (...) In deze buffer worden de uitkomsten van berekeningen en dergelijke opgeslagen in ASCII-code. Bij ?FRE(0) wordt bij voorbeeld hier de berekende vrije geheugenruimte in ASCII-code neergezet voordat die op het scherm wordt weergegeven.

FAC-III

6536 (...) Ook deze derde FAC heeft dezelfde vorm, lengte en indeling als FAC-I. Hij wordt onder meer gebruikt voor berekeningen in enkele precisie (zie bij voorbeeld de vermenigvuldigingsroutine voor enkeleprecisiegetallen, opgeslagen in registers BC en DE, vanaf &H35C9).

Onbekend

653E (00)
---- (00)
6542 (00) De functie van deze adressen is onbekend.

Tijdelijke opslag

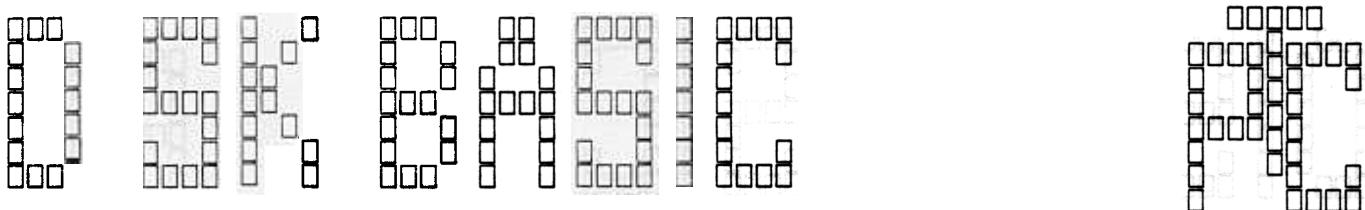
6543 (...) Tijdelijke opslag van MSB van registerwaarden.
---- (...)
6545 (...)

Onbekend

6546 (00) De functie van dit adres is niet bekend.

Begin BASIC-programma

6547 (...) Op dit adres begint gewoonlijk het BASIC-programma



Adres	Inhoud	Omschrijving
<u>Videogeheugen</u>		
5000	(00)	1e videogeheugen, groot 1920 adressen (24 regels van elk 80 posities). Bij 40 posities per regel zijn dit twee pagina's.
----	(00)	
577F	(00)	Met de instructie POKE kan rechtstreeks informatie op het scherm worden gezet.
5780	(00)	"Scroll"-regel 1e videogeheugen. In deze regel (80 posities) wordt onder meer bij LIST de regel klaargezet die vanaf de onderkant het scherm opschuift.
57CF	(00)	
57D0	(00)	Dit laatste deel van het 1e videogeheugen kan worden gebruikt om bij voorbeeld kleine tabellen op te bergen.
----	(00)	
57FF	(00)	Schrijven naar en lezen in deze ruimte kan leiden tot strepen op het scherm. De informatie kan door de computer worden gewist bij het gebruik van cassettefuncties.
5800	(..)	2e videogeheugen, met dezelfde opbouw als het 1e videogeheugen (totaal 2048 adressen). Bij het T-model ontbreken deze adressen geheel doordat er op deze plaats geen RAM is aangebracht. Bij het M-model bevinden zich hier 4-bit adressen, die de "attributen" voor het 1e videogeheugen bevatten, dat wil zeggen codes die aangeven hoe de tekens van het 1e videogeheugen moeten worden weergegeven (normaal, invers, knipperend, onderstreept). Adres &H5800 correspondeert met &H5700 enz. Omdat de adressen een breedte hebben van 4 bits (i.p.v. 8) zijn ze ook bij het M-model niet voor iets anders te gebruiken.
----	(..)	
5FFF	(..)	
<u>Toetsenbord</u>		
6000	(00)	Toetscodebuffer waarin de codes van maximaal 12 toetsen worden opgeslagen, die al zijn ingedrukt maar nog niet door het programma of door de BASIC zijn verwerkt. Deze adressen worden gebruikt door de monitor.
----	(00)	
600B	(00)	
600C	(00)	Toetsbufferteller. Dit adres bevat het aantal toetscodes dat is opgeborgen vanaf adres 6000 (maximaal 12).
600D	(FF)	Op dit adres staat de code van de laatst ingedrukte toets. De ingedrukte toets wordt niet automatisch verwerkt, maar de toetscode blijft staan zolang de toets ingedrukt is. Wordt de toets losgelaten, dan komt er bij de eerstvolgende toetsenbord-"scan"-routine (d.w.z. binnen 20 ms) FF (=255) te staan.
600E	32	Hier staat de herhalingstijd die geldt voor een ingedrukt gehouden toets. Normaal staat hier &H32 (=48). Verandering van de inhoud wordt meteen weer ongedaan gemaakt door de monitor (zie &H00A3 voor het T- en &H0176 voor het M-model). Als een Floppy Disk Interface is aangebracht, doet dit adres tevens dienst als CTC-teller.

Adres	Inhoud	Omschrijving
600F	(00)	Vlag voor het toetsenbord. Normaal staat er 00. 00 = geen SHIFT of SHIFT LOCK ingedrukt 01 = SHIFT ingedrukt (bit 0 = 1) 04 = SHIFT LOCK aan (bit 2 = 1) 05 = SHIFT en SHIFT LOCK ingedrukt (bit 0 = 1 en bit 2 = 1).
		<u>Teller</u>
6010	(..)	
6011	(..)	Deze twee adressen vormen samen een teller die elke 20 ms wordt opgehoogd. De teller loopt van 0 tot &HFFFF (0...65535) zolang de interrupt aan staat. Dus niet als cassette, schijf of printer werkt. Na zo'n actie loopt de teller verder.
		<u>Monitor</u>
6012	..)	Vlagadres voor de monitor, dat voor verschillende doeleinden wordt gebruikt. Bij voorbeeld voor het automatisch herhalen van een ingedrukt gehouden toets; bit 0 = 1 betekent: "repeat" aan (zie &H006C voor vullen en &H007A...0083 voor testen); bit 1 = 1 betekent "numeric" aan.
		<u>Scherm</u>
6013	(00)	Cursorvorm. Vlagadres voor het toepassingsprogramma. Bij het opstarten van de P2000T wordt dit adres 00 gemaakt; bij de P2000M wordt het 01 (er wordt gecontroleerd of het 2e videogeheugen aanwezig is). 00 betekent dat de cursor als invers blokje zichtbaar wordt gemaakt (P2000T). Een oneven waarde op dit adres maakt bij het T-model de cursor onzichtbaar. Bij het M-model wordt de cursor zichtbaar in de vorm van een streepje. De inhoud kan bij gebruik van DISK BASIC alleen worden veranderd met POKE &H6013,X, waarin X = 00 of 01 is.
6014	(00)	Pointer naar de plaats op het scherm waar de monitor info moet zetten over ondernomen "acties". Deze adressen schijnen niet te worden gebruikt omdat er een fout zit in de desbetreffende monitorroutines. DISK BASIC zet hier 0000 neer (zie &H6D07). Verandert u dat op de systeemschijf bij voorbeeld in &H5000 (of na het laden met POKE &H6014,0:POKE &H6015,&H50), dan verschijnt op de eerste regel van het scherm de letter L als u SHIFT LOCK indrukt en de letter D als de monitor diskroutines uitvoert. De inhoud van &H614...6015 is een offset-waarde. De monitor telt er de inhoud van register E bij op (in &H01C6).
6015	(00)	De adressen worden wel gebruikt door de service-module.

Adres	Inhoud	Omschrijving
		<u>Printer</u>
6016	01	Op dit adres, dat door de monitor wordt gebruikt, staat met welke snelheid de P2000 informatie naar de printer moet sturen ("baudrate"). Met onderstaande formule is dit uit te rekenen (A=PEEK(&H6016)):
		Baudrate = $2400/(1+A)$ of
		A = $(2400/\text{baudrate}) - 1$
		Dit levert het volgende resultaat:
		00 = 2400 Bd
		01 = 1200 Bd (waarde bij opstarten via &H034C)
		03 = 600 Bd
		07 = 300 Bd (waarde bij opstarten via &H0348)
		0F = 150 Bd enz.
		Normaal start de P2000 op met 1200 Bd. Zijn de pennen 2 en 6 van de printerconnector doorverbonden, dan start de P2000 op met 300 Bd. Dit wordt gecontroleerd in &H0344.
		Als bit 7 = 1 wordt niet automatisch CR + LF (Carriage Return plus Line Feed) gegeven.
		<u>NOP</u>
6017	(00)	Deze adressen worden bij DISK BASIC niet gebruikt, tenzij u zelf casetteroutines schrijft. Dan verdient het aanbeveling hiervoor dezelfde adressen te gebruiken als BASIC NL. Zie het adresoverzicht van BASIC NL voor de juiste adressen.
----	(00)	
601F	(00)	
		<u>Interrupties</u>
		Tabel van interruptievectoren voor de Monitor.
		Twee opeenvolgende posities bevatten het adres waar de monitor naar toe moet springen als er een interruptie komt. De eerste twee sprongadressen (&H6130 en &H6137) worden telkens opnieuw geladen als PDOS wordt aangeroepen (zie &HEB1B e.v. in PDOS 1.4 en &HE9DC in PDOS 1.6).
6020	30 61	<u>JP 6130.</u> Interrupt request kanaal 0. Disk-opdracht voltooid of CTC interrupt.
6022	37 61	<u>JP 6137.</u> Interrupt request kanaal 1. Disk niet klaar.
6024	00 00	Interrupt request kanaal 2. Communicatie-interrupt. (niet gebruikt).
6026	38 00	<u>JP 0038.</u> Interrupt request kanaal 4. Interruptie voor toetsenbord en timer.
6028	(00)	
----	(00)	
602F	(00)	Niet gebruikt

Adres | Inhoud**Omschrijving**NOP

- 6030 (00)
---- (00)
6054 (00)
- De adressen 6030...604F zijn interne werkadressen voor de monitor. Ze vormen samen de "record header" of "file descriptor" voor cassettes. Deze gaat vooraf aan elk blok van 1024 bytes dat op cassette wordt weggeschreven.
Bij DISK BASIC worden deze adressen niet gebruikt, evenmin als de adressen &H6050...6054. Bij het schrijven van eigen cassetteroutines moeten deze adressen worden gebruikt voor de functies die de monitor verwacht. Zie hiervoor het overzicht van BASIC NL.

Stack pointer

- 6055 (FB)
6056 (61)
- Op dit adres bewaart de monitor de oude waarde van de stack pointer bij het aanroepen van PDOS (bij CALL 0005, waarbij naar &H044F wordt gesprongen).

NOP

- 6057 (00)
---- (00)
605B (00)
- Deze vijf posities worden niet gebruikt.

Geheugen

- 605C (01)
- Bij het opstarten controleert de monitor hoeveel geheugen aanwezig is en zet het resultaat hier neer. De getallen hebben de volgende betekenis:
- | | |
|----|------------------|
| 01 | 16 Kbyte |
| 02 | 32 Kbyte |
| 03 | 48 Kbyte of meer |

Disk

- 605D (01)
- Als bij het opstarten een Disk Controller en een diskettestation met schijf en gesloten klepje aanwezig zijn, leest de monitor de eerste twee sporen van de schijf in. Daarna controleert de monitor of dat een systeemschijf was door te kijken of op adres &HE000 van de tweede geheugenbank &HF3 staat (zie &H0EC6 e.v.). Is het geen systeemschijf, dan zet de monitor hier 00 neer, in alle andere gevallen 01.
Wat de monitor betreft hoeft dat dus geen systeemschijf te zijn. De eerste twee sporen worden in elk geval ingelezen, onverschillig wat erop staat, en weggeschreven op de adressen &HE000 tot &HFFFF in de tweede geheugen-"bank".
Na het inlezen springt de monitor naar adres &H1010 van de insteekmodule. Disk BASIC controleert op dit adres allereerst de inhoud van &H605D. Is die 00, dan volgt de foutmelding DISK BASIC LOAD ERROR. Dat gebeurt dus als alles in orde is en alleen een verkeerde schijf in disk drive A zit.

- 00 Ingelezen schijf is geen PDOS-systeemschijf
- 01 Geen Disk Controller, geen Disk Drive, geen schijf in Disk Drive A, Disk Drive niet ingeschakeld, klepje open of PDOS-systeemschijf ingelezen.

De 24K DISK BASIC-interpreter controleert of er een systeemschijf in Drive A zit omdat hij anders niet kan werken. Van de circa 24 Kbyte van de interpreter staat 16 Kbyte in ROM (in de insteekmodule). De ontbrekende circa 8 Kbyte wordt ingelezen vanaf de sporen 3, 4 en 5 van de systeemschijf en weggezet op de adressen &H6200...8A90.

Printer

605E E0 6F Beginadres van de eerste printer-vertaal tabel (= &H6FE0).

Gereserveerd

6060 (00)
---- (00)
606F (00) Gereserveerd voor cassetteroutines.

Disk

De adressen &H6070...6086 worden bij het opstarten gevuld door de monitor. De inhoud komt van &H0FE8...0FFE (zie &H0E9A e.v.). DISK BASIC doet dat nog eens dunnetjes over (zie &H106F e.v.), maar modificeert daarna de inhoud van verschillende adressen.

6070 (00)	Transfer-adres (waar de file heen moet). Voor het ontbrekende deel van de interpreter is dit &H6200 en voor PDOS is dit &HE000 in de tweede geheugen-"bank".
6071 (E0)	
6072 09	Lengte opdracht (is altijd &H09).
6073 (42)	Functie-opdracht. 42 = lezen van schijf 45 = schrijven naar schijf
6074 (01)	Nummer van de Disk Drive
6075 (01)	Spoornummer
6076 00	Schijfkant
6077 (...)	Sectornummer (0...&H0F)
6078 01	Transmissiesnelheid ("baudrate") (is altijd 01)
6079 10	Aantal sectoren per spoor (is altijd &H10 = 16)
607A 0E	Ruimte tussen de sectoren ("gap space") (is altijd &H0E)
607B (00)	Datalengte

Adres	Inhoud	Omschrijving
607C	03	Zoeken: aantal bytes van de opdracht (altijd &H03)
607D	0F	code opdracht (&H0F)
607E	(01)	nummer van de disk drive
607F	(..)	sectornummer
6080	02	Recall: aantal bytes van de opdracht (altijd &H02)
6081	07	code opdracht (&H07)
6082	(01)	nummer van de disk drive
6083	03	Specificatie: aantal bytes van de opdracht (altijd &H03)
6084	03	code opdracht (&H03)
6085	60	parameter 1,2 (&H60)
6086	34	parameter 3,4 (&H34)
6087	...	
----	(..)	Werkadressen voor de monitor
608D	(..)	
608E	(..)	Tijdelijke positie van de stack pointer bij het laden van
608F	(..)	de disk-constanten van &H0FE8 naar &H6070 (zie &H0E91 en &H0EDA)
		<u>Scherm</u>
6090	(00)	Vertraging. POKEn van een waarde van 01 tot &HFF op deze positie bouwt een extra vertraging in bij weergeven op het scherm (zowel bij LIST als bij RUN). Hoe hoger de waarde, des te trager het beeld (zie &H69F0). De vertraging komt bovenop de vertraging die het indrukken van SHIFT geeft (01 of 05 op &H600F).
		<u>Disk</u>
6091	(00)	Vlag voor Disk I/O-fout (zie &H69BB).
6092	00	Niet gebruikt.
		<u>Scherm</u>
6093	(00)	Vlagadres voor het uitvoeren van CHR\$(4)CHR\$(Y)CHR\$(X). Bij CHR\$(4) wordt de inhoud 01, na CHR\$(Y) 02 en na CHR\$(X) 03.
6094	(00)	Bevat na CHR\$(4)CHR\$(Y) de waarde Y (regelnummer).
6095	(00)	Bevat na CHR\$(4)CHR\$(Y)CHR\$(X) de waarde X (positie op de regel).
		<u>Disk</u>
6096	(00)	Tijdelijk opbergadres van de laatst ingedrukte toetscode bij het aanroepen van PDOS (zie &H6971 en 6974).

<u>Adres</u>	<u>Inhoud</u>	<u>Omschriiving</u>
		<u>Printer</u>
6097	(00)	Opslagadres voor printer-stuurcodes (zie &H6C5B). Wat er daarna met de inhoud gebeurt is niet duidelijk.
		<u>NOP</u>
6098	(00)	
----	(00)	
60AF	(00)	Deze adressen worden waarschijnlijk niet gebruikt.
		<u>Scherm</u>
60B0	(27)	Fysieke lengte van een regel op het scherm. De waarde 39 (&H27) geeft een breedte van 40 posities; 79 (&H4F) geeft 80 posities. U kunt elke waarde tussen 1 en 255 POKEn, maar bij waarden groter dan 79 overschrijft de cursor dezelfde regel. De breedte kan alleen worden veranderd met POKE &H60B0,X. Zie ook &H639F.
60B1	(..)	Geheugenadres van de cursor (bij voorbeeld &H5000) voor het videogeheugen.
60B2	(..)	
60B3	(..)	Positie van de cursor op fysieke de regel. Deze positie is op te vragen met PRINT PEEK(&H60B3). De inhoud van dit adres ligt altijd tussen 00 en de inhoud van &H60B0. Voor de cursorpositie in de <u>logische</u> regel: zie &H66C3.
60B4	...	Geheugenadres van de cursor. Dit adres schijnt altijd dezelfde inhoud te hebben als &H60B1...60B2.
60B5	(..)	
60B6	(00)	Op dit adres wordt aangegeven of kleine letters automatisch moeten worden omgezet in grote. Omschakelen gebeurt met SHIFT TAB. De inhoud heeft de volgende betekenis:
	00	kleine letters worden omgezet in grote (BASIC-programmeerstand)
	FF	geen omzetting van kleine letters in grote (Tekststand)
		Vult u een andere waarde in dan 00 of FF, dan werkt SHIFT TAB niet meer.
		<u>Disk</u>
60B7	(00)	Dit adres wordt nul gemaakt tijdens het laden van PDOS (zie &H6D32), maar verder lijkt er niets mee te gebeuren.
60B8	(00)	Dit adres wordt gebruikt om de inhoud van register HL tijdelijk op te slaan bij het aanroepen van PDOS (zie &H696D).
60B9	(00)	
60BA	(00)	Dit adres wordt vermoedelijk niet gebruikt.
60BB	(00)	In dit adres wordt de inhoud van register C bewaard bij het aanroepen van PDOS (zie &H697C). Na terugkeer uit PDOS kan de inhoud naar register A gaan (&H69A2 en 69C1).

Adres	Inhoud	Omschrijving
60BC	(00)	Hierin wordt de inhoud van register B bewaard, maar daar gebeurt verder schijnbaar niets mee.
		<u>NOP</u>
60BD	(00)	Deze adressen worden waarschijnlijk niet gebruikt.
60C5	(00)	
		<u>Scherm</u>
60C6	(00)	Hier wordt óf &H5000 (zie &H6BD3) opgeslagen, óf het ver-
60C7	(00)	schil van HL en BC (zie &H6BC9). Behalve dat de inhoud van deze adressen elke keer wordt aangepast als het beeld "scrollt", lijkt er verder met die inhoud niets te worden gedaan.
		<u>NOP</u>
60C8	(00)	
60C9	(00)	Deze adressen worden waarschijnlijk niet gebruikt.
60CA	(00)	
		<u>Disk</u>
60CB	(03)	Tijdelijke opslag van het laagste in te lezen spoornummer bij het laden van het ontbrekende stuk van de interpreter (zie &H101E).
60CC	(06)	Tijdelijke opslag van het hoogste in te lezen spoornummer + 1 bij het laden van het ontbrekende stuk van de interpreter (zie &H1023).
60CD	(00)	Tijdelijke opslag van het eerste laadadres bij het laden van het ontbrekende stuk van de interpreter (= &H6200 - zie &H1019).
60CE	(62)	
		Deze adressen worden ook gebruikt als de vraag "Runtime Support ?" met Y wordt beantwoord (zie &H6E19 e.v.). Dan wordt de inhoud respectievelijk 07, 08 en &HC000. Van hieruit worden deze parameters overgebracht naar het "File Control Block" (FCB) op &H6070...608F.
		<u>NOP</u>
60CF	(00)	
---	(00)	
6111	(00)	Deze adressen worden vermoedelijk niet gebruikt.
		<u>Onbekend</u>
6112	(DD)	
----	(..)	
611D	(F0)	Onbekend

Adres	Inhoud	Omschrijving
		<u>Disk</u>
611E	(..)	Tijdelijke opbergplaats van de stack pointer bij het laden van het ontbrekende deel van de interpreter (zie &H1061 en &H10C0).
6120	(..)	
----	(..)	Stack voor de CP/M-monitorroutine (zie &H0453).
612F	(..)	
6130	3E 01	LD A,01. Zet 01 in accumulator.
6132	D3 94	OUT 94. Schakel om naar 2e geheugen-"bank".
6134	C3 0B EA	CALL EA0B. Roep adres &HEA0B (in PDOS) aan. Enkele hulproutines die worden gebruikt om met PDOS te werken. Telkens als PDOS wordt aangeroepen worden deze routines opnieuw geladen vanuit &HEB34 (PDOS 1.4) of &HE9F5 (PDOS 1.6). Zie ook de interruptievector op &H6020.
6137	F3	Enkele hulproutines die worden gebruikt om met PDOS te werken. Ook deze routines worden telkens opnieuw geladen, en wel vanuit &HEB82 (PDOS 1.4) of &HEA3A (PDOS 1.6). Zie ook de interruptievector op &H6022.
----	(..)	
6174	C9	
		<u>Kladblokruimte</u>
6175	(00)	Vrij voor het opslaan van bij voorbeeld tussenresultaten of eigen machinetaalroutines.
----	(00)	
61FF	(00)	
		<u>Terug naar BASIC</u>
6200	C3 74 8A	JP 8A74. Via dit adres kunt u vanuit een machinetaalroutine terugkeren naar BASIC, waarbij BASIC met een schone stack moet beginnen en de buffers geschoond moeten worden. Een eventueel in het geheugen aanwezig BASIC-programma wordt wel gewist (in tegenstelling tot BASIC NL). Zie ook &H6380.
		<u>NOP</u>
6203	(00)	Deze twee adressen worden niet gebruikt
6204	(00)	
		<u>PDOS</u>
6205	C3 34 69	JP 6934. Via dit sprongadres wordt PDOS aangeroepen (via adres &H0005 in de monitor). Het adres &H6934 kan in principe worden vervangen door een ander sprongadres, bij voorbeeld om zelfgemaakte cassetteroutines aan te roepen.
		<u>Vrij voor sprongadressen</u>
6208	(00)	Deze adressen worden niet gebruikt en zijn vrij voor eigen sprongadressen, korte machinetaalroutines en andere zaken.
----	(00)	
622F	(00)	

Adres | InhoudOmschrijvingBASIC

6230 (00)
---- (00)
625F (00)

Deze adressen worden gebruikt om tijdelijk het eerste deel van de buffer op &H6280 op te slaan bij het opstarten van BASIC (zie &H6E46). Hiermee wordt bereikt dat de eerste 48 adressen van de buffer, die begint op &H6280, met de waarde 00 worden geladen. Na afloop mogen de adressen &H6230... &H625F voor andere doeleinden worden gebruikt.

NOP

6260 (00)
---- (00)
627F (00)

Deze 32 adressen worden waarschijnlijk niet gebruikt.

Buffer en andere zaken

6280(..)
----(..)
637F(..)

Deze adressen vormen een buffer van 256 bytes waarin bij LOAD en SAVE een deel van de index van de schijf wordt gezet (1 sector). Na het uitvoeren van deze instructies wordt een deel van de buffer onmiddellijk voor andere doeleinden gebruikt.
Ook eigen machinetaalroutines mogen gebruik maken van de buffer, als u er maar op bedacht bent dat de gegevens niet permanent blijven staan maar overschreven worden zodra u PDOS aanroept. De monitor-programma's van Ron Eijnthoven gebruiken bij voorbeeld een aantal adressen tot &H62B0 als stack.

De buffer is verdeeld in 8 stukken van 32 bytes. Elk stuk bevat de gegevens van één programma uit de index op spoor 1 van de schijf; dit is het zogenaamde File Control Block (FCB). Het FCB is als volgt opgebouwd:

6280 (00) Type file (meestal 00)
6281(..) 8 posities voor de naam. Is de naam korter, dan wordt hij automatisch aangevuld met spaties (&H20).
6288(..)
6289 (42)
628A (41)
628B (53) 3 posities voor de typeaanduiding, zo nodig aangevuld met spaties. Als bij het wegschrijven van BASIC-programma's geen type wordt opgegeven, zet BASIC hier automatisch BAS neer.
628C (00) Nummer van het extent binnen de file (meestal 00). De Disk Allocation Map (zie &H6290) heeft ruimte voor de nummers van ten hoogste 16 "records" van elk 4 sectoren (1024 bytes). Is het programma of bestand langer dan 16 Kbyte, dan wordt een tweede FCB in de index gereserveerd, dat op dit adres het nummer 01 krijgt.

Adres	Inhoud	Omschrijving
628D	(00)	BASIC kan hier een code zetten waarmee het programma of bestand beveiligd is tegen overschrijven ("write protect"). Anders is dit adres 00.
628E	00	Vermoedelijk niet gebruikt.
628F	(..)	Lengte in sectoren van het programma of bestand. Een sector heeft een lengte van 256 bytes.
6290	(..)	Nummers van de "records" (Disk Allocation Map). Elke record bestaat uit vier sectoren, die verspreid over het spoor staan.
----	(..)	
629F	(..)	De werkelijke volgorde van de sectoren op de schijf is dus anders dan de logische volgorde. Zie hiervoor de spoor- en sectorindeling achterin dit boekje.

De blokken &H62A0...62BF, &H62C0...62DF enz. hebben dezelfde indeling.

Zachte start in BASIC

6380	C3 88 17	JP 1788. Zachte landing in BASIC. Een eventueel in het geheugen aanwezig BASIC-programma wordt <u>niet</u> gewist (zie ook &H6200).
------	----------	---

USR

Tabel van adressen van de functies USR0 tot en met USR9. Als geen gebruikersfuncties zijn gedefinieerd, staat op alle adressen het adres &H1F33 van een foutafhandelingsroutine.

6383	(33 1F)	USR0
6385	(33 1F)	USR1
6387	(33 1F)	USR2
6389	(33 1F)	USR3
638B	(33 1F)	USR4
638D	(33 1F)	USR5
638F	(33 1F)	USR6
6391	(33 1F)	USR7
6393	(33 1F)	USR8
6395	(33 1F)	USR9

Vlagadres

6397	(01)	Vlagadres. De bedoeling is niet duidelijk. De functie hangt waarschijnlijk samen met het volgende adres.
6398	(00)	Ook de functie van dit adres is niet duidelijk (zie bijv. &H709C).

Foutcode

6399	(..)	BASIC-foutcode. Hier staat de code van de laatst opgetreden fout in BASIC (in een programma of in de directe stand).
639A	(00)	NOP

Adres	Inhoud	Omschrijving
		<u>Printer</u>
639B (00)		Printer-kolomteller. Op dit adres houdt de P2000 bij op welke positie van de regel de afdrukkop van de printer zich volgens zijn berekening moet bevinden. Dat hoeft niet per se de werkelijke positie te zijn. De inhoud van dit adres is op te vragen met ? LPOS(X), waarin X een "dummy argument" is (met elke andere letter gaat het ook).
		<u>(L)LIST en (L)PRINT</u>
639C (00)		Uitvoer-indicator. Op dit adres staat waar de uitvoer naar toe moet. Dit adres wordt onder meer gevuld in de LPRINT-routine (zie &H21A4). <p style="padding-left: 40px;">00 uitvoer naar het scherm 01 uitvoer naar de printer</p>
		<u>Printer</u>
639D (38)		Aantal keren dat 14 spaties op een regel moeten worden gegeven bij LPRINT A,B,C... De inhoud van dit adres is altijd een veelvoud van 14 (&H0E). Voor berekening: zie &H63A0. Aanpassing van de regellengte op &H639E met WIDTH LPRINT XX leidt automatisch tot het aanpassen van de inhoud van &H639D.
639E (50)		Maximale lengte van een printer-regel voordat de P2000 automatisch een Carriage Return (CR) en een Line Feed (LF) genereert (logische lengte printerregel). De inhoud is te veranderen met WIDTH LPRINT XX . De inhoud van &H639D wordt automatisch aangepast.
		<u>Scherm</u>
639F (FF)		De maximale lengte van een regel op het scherm (&HFF = 255). De inhoud van dit adres is de <u>logische lengte</u> van de schermregel en bepaalt of Carriage Return (CR) en Line Feed (LF) moeten worden gegeven, en wel in samenhang met de cursorpositie op &H66C3. Dat gebeurt als volgt: als (66C3) >(639F) dan CR + LF. Zie bij voorbeeld &H729E.
63A0 (0E)		Bij de opdracht PRINT 1,2,3,4 bepaalt de inhoud van dit adres samen met de inhoud van &H63B4 of het volgende getal 14 posities verderop op dezelfde regel moet worden gePRINT of dat CR + LF moet worden gegeven. Ook bepaalt de inhoud van dit adres hoeveel spaties er tussen de getallen moeten worden afgedrukt. Dat gebeurt overeenkomstig de volgende twee regels: <ol style="list-style-type: none"> 1 Als (&H66C3)≥(&H63A0), doe dan CR + LF. 2 Als (&H66C3)<(&H63A0), PRINT dan 14-(&H66C3) spaties. De spaties worden gePRINT vanaf de momenteel geldende cursorpositie.

Omschrijving

Hetzelfde geldt voor het afdrukken van strings, bij voorbeeld PRINT"AAAAAA","BBBBBB","CCCCCC". Het verschil tussen getallen en strings is dat getallen worden voorafgegaan door een spatie of een minteken en worden gevolgd door een spatie. Het getal 1 vraagt dus 3 posities; 10 vraagt 4 posities enzovoort. Strings vragen even veel posities als ze tekens tellen. Normaal staat in dit adres &H0E (= 14), maar er mag elke waarde staan tussen 0 en 255. Dit werkt als volgt. Na het afdrukken van AAAAAA staat de cursor op positie 6 van de regel. Deze waarde wordt afgetrokken van de inhoud van &H63A0. Is de uitkomst positief, dan maakt BASIC de inhoud van &H66C3 gelijk aan 14. Dit betekent dat BBBB vanaf de 14e positie wordt afgedrukt. Is de uitkomst 0 of negatief, dan gaat de cursor naar de volgende regel. Is na het PRINTen van BBBB (&H63A0) - (&H66C3) nog steeds positief, dan wordt de inhoud van &H66C3 gelijk aan 28 gemaakt. Na het "uitvoeren" van de komma is de inhoud van &H66C3 dus altijd nul of een veelvoud van 14. Maakt u de inhoud van &H63A0 = 255, dan begint de volgende string dus steeds 14 posities verder op de logische regel. Als de fysieke regellengte (40 of 80) bereikt is, telt BASIC door op de volgende fysieke regel.

De logische lengte van een regel is maximaal 255 (zie &H639F); de fysieke lengte is 1 tot 80 posities (zie &H60B0).

Onbekend

63A1 (00)	Onbekend (zie &H70B4)
63A2 (00)	Onbekend (zie &H1840)
63A3 (00)	Onbekend
63A4 (00)	Onbekend

CLEAR

De inhoud van deze adressen geeft het einde van de vrije geheugenruimte aan, die door BASIC mag worden gebruikt. Daarboven kunt u machinetaal zetten, zonder dat die door BASIC kan worden overschreven.

Hier is een wezenlijk verschil met BASIC NL. Door middel van de instructie CLEAR,&Hxxxx,nnnn wordt het hoogste adres bepaald dat door Disk-BASIC mag worden gebruikt. Het adres xxxx komt in &H63A5 en 63A6 te staan. Vanaf dit adres naar beneden gerekend is ruimte gereserveerd voor de stack. Deze ruimte wordt bepaald door de waarde nnnn.

Daarna komt, alweer naar beneden toe, de stringruimte. De stringruimte staat dus onder de stack (bij BASIC NL staat de stringruimte boven de stack; zie gebruiksaanwijzing pag. 53).

Bij BASIC NL is de stringruimte ingesloten tussen het met CLEAR gereserveerde adres xxxx en de stack. Wilt u veel met strings manipuleren, dan kunt u de stringruimte vergroten tot nnnn met de instructie CLEAR nnnn,xxxx. De stack schuift dan naar beneden.

Bij Disk-BASIC is de stack ingesloten tussen het met CLEAR gereserveerde adres xxxx en de bovenkant van de stringruimte. Hebt u een programma met veel FOR...NEXT-lussen, GOTO's, GOSUB's enzovoort, dan kunt u de stack vergroten tot nnnn met CLEAR,xxxx,nnnn. De waarde nnnn heeft bij BASIC NL en Disk-BASIC dus een verschillende betekenis. Bij BASIC NL is nnnn de stringruimte, bij Disk-BASIC de grootte van de stack.

Bij Disk-BASIC groeit de stringruimte mee. Die hoeft u dus niet te reserveren. U krijgt pas een foutmelding als de "onderkant" van de stringruimte in botsing komt met de "bovenkant" van het BASIC-programma; dat wil zeggen de arrayruimte. Dan krijgt u de foutmelding "Out of memory". De stringruimte bij Disk-BASIC is dus zo groot als de lengte van het programma en het hoogste, met CLEAR gereserveerde adres toelaten. Zie ook &H66D4.

Regelnummer

63A7 (FF)
63A8 (FF) Op deze adressen staat het nummer van de BASIC-regel die in uitvoering is. Bij het opstarten van BASIC wordt dit regelnummer op &HFFFF (= -1) gezet.

BASIC

63A9 (4C)
63AA (92) Geeft het begin van het BASIC-programma aan. Bij buffers voor drie bestanden is dat &H924C. Bij n bestanden, waarbij n buffers worden gereserveerd, kan het begin van het BASIC-programma worden bepaald met &H8A90 + 296 + n*561 + 1.

Overflow

63AB (D0)
63AC (14)
63AD (00) Beginadres van de foutmelding "Overflow". Deze foutmelding treedt op bij berekening van te grote getallen.

Onbekend

63AD (00)	Onbekend
63AE (00)	Onbekend
63AF (00)	Onbekend
63B0 (00)	Onbekend

<u>Adres</u>	<u>Inhoud</u>	<u>Omschrijving</u>
Buffers		
63B1	90	Eindadres van BASIC-interpreter = begin BASIC-buffer voor ASCII-files (&H8A90).
63B2	8A	
63B3	90	Beginadres BASIC-buffer. Deze buffer heeft een lengte van 40 + 256 = 296 bytes (32 bytes voor het FCB, 8 bytes voor huishoudelijke doeleinden en 256 bytes voor het bufferen van 1 sector). Deze buffer wordt gebruikt bij het inlezen van ASCII-files.
63B4	8A	
63B5	(B8)	Beginadressen van de buffers, gebruikt voor het werken met bestanden. Bij het opstarten kunt u kiezen uit 0...15 bestanden. Hieronder de beginadressen bij het maximum van 15 buffers. Bij minder buffers is de inhoud van de desbetreffende adressen nul. We geven steeds alleen het eerste adres.
----	(..)	
63D2	(AA)	
De buffers hebben een lengte van 561 (= &H231) bytes: 1 byte voor het type file, 32 bytes voor het FCB, 7 bytes voor tijdelijke opslag, 2 x 256 bytes voor bestandsgegevens en 9 bytes die niet worden gebruikt. Werken met 15 bestanden kost dus 8415 bytes geheugenruimte.		
Het beginadres van het BASIC-programma kunt u vinden door het beginadres van de eerstvolgende niet-gebruikte buffer met 1 te verhogen. Bij 0 bestanden is dat dus &H8BB8 + 1 = &H8BB9 (dit adres staat op &H63A9). Bij 3 bestanden: &H924B + 1 = &H924C.		
63B5	B8 8B	1 bestand - &H8BB8
63B7	E9 8D	2 bestanden - &H8DE9
63B9	1A 90	3 bestanden - &H901A
63BB	4B 92	4 bestanden - &H924B
63BD	7C 94	5 bestanden - &H947C
63BF	AD 96	6 bestanden - &H96AD
63C1	DE 98	7 bestanden - &H98DE
63C3	0F 9B	8 bestanden - &H9B0F
63C5	40 9D	9 bestanden - &H9D40
63C7	71 9F	10 bestanden - &H9F71
63C9	A2 A1	11 bestanden - &HA1A2
63CB	D3 A3	12 bestanden - &HA3D3
63CD	04 A6	13 bestanden - &HA604
63CF	35 A8	14 bestanden - &HA835
63D1	66 AA	15 bestanden - &HAA66
Begin BASIC-programma bij 15 bestanden: &HAA66 + 561 + 1 = &HAC98		
63D3	(..)	Aantal gereserveerde buffers (= bestanden) - 1.

<u>Adres</u>	<u>Inhoud</u>	<u>Omschrijving</u>
		<u>Onbekend</u>
63D4	(00)	
----	(00)	
640C	(00)	Deze adressen zijn niet bekend.
		<u>Bestanden</u>
640D	(..)	Bestandstype.
640E	(..)	Hulpbuffer voor bestandsnaam + extension (8 + 3 posities).
----	(..)	Deze wordt gebruikt om namen, korter dan 8 letters, aan te vullen met spaties en eventueel de standaard-extension BAS.
6418	(..)	De complete naam + extension gaat naar het FCB.
6419	00	Nul ter afsluiting
		<u>Onbekend</u>
641A	(00)	
----	(00)	
642D	(00)	Deze adressen zijn nog niet bekend.
642E	3A	De functie van dit adres is niet bekend. Vermoedelijk wordt het gebruikt om het begin van de hiernavolgende buffer aan te geven.
		<u>Buffer</u>
642F	(..)	Hier blijft het eerste token of teken van de regel staan die het laatst in de directe stand is uitgevoerd.
6430	..	Buffer (254 bytes) voor BASIC-regels. Bij het intikken van programmaregels (zie &H656E) wordt in deze buffer de regel in gecodeerde vorm neergezet ("getokeniseerd"). Vanuit deze buffer worden ze overgebracht naar het programma. Hetzelfde gebeurt bij EDIT omdat de regel dan, al of niet gewijzigd, weer in het programma moet worden opgenomen.
----	..)	De buffer wordt ook gebruikt in de directe stand. Dan staat hier de gecodeerde instructie.
652D	..)	
		<u>NOP</u>
652E	(00)	Deze adressen worden waarschijnlijk niet gebruikt.
----	(00)	Vermoedelijk is de voorgaande buffer wat ruimer bemeten
656C	(00)	dan strikt nodig was.
		<u>PRINT-buffer</u>
656D	(2C)	De functie van dit adres is niet precies bekend. Waarschijnlijk markeert 2C (= komma) het begin van de volgende buffer.

Adres Inhoud

656E	(..)	ASCII-buffer (254 bytes) voor BASIC-regels. De buffer wordt voor twee doeleinden gebruikt:
----	(..)	
666B	(..)	1 Bij het intikken van programmaregels en instructies in de directe stand. Voordat deze worden verwerkt, worden ze in gecodeerde vorm overgebracht naar de buffer op &H642E...652D. 2 Bij EDIT en (L)LIST. De gecodeerde programmaregel wordt omgezet in ASCII-code alvorens in leesbare vorm op het scherm te kunnen worden gebracht.

De inhoud van de buffer kan op het scherm worden gebracht met SHIFT 9 op het kleine toetsenbord ("M"). Er verschijnt dan een uitroepsteken op het scherm. De inhoud kan worden gewijzigd met dezelfde instructies als in de EDIT-stand. Na LIST nn of EDIT nn staat regel nn in de buffer, maar zonder regelnummer. Na I(insert) kan een ander regelnummer worden ingetikt, waardoor de programmaregel kan worden gekopieerd. Met de spatiebalk kunt u door de regel lopen en met "D", "I", "C" enz. tekens wissen, tussenvoegen, veranderen enz.

NOP

666C	(00)	Deze adressen worden waarschijnlijk niet gebruikt.
----	(00)	Vermoedelijk is ook de voorgaande buffer wat ruim uitgevallen.
66C2	(00)	

Scherm

66C3		Hier staat de positie van de cursor in de logische regel. De inhoud kan de waarde 0...255 hebben, afhankelijk van de logische lengte die in &H639F staat. De positie van de cursor kan worden opgevraagd met PRINT POS(X), waarin "X" een dummy-argument is (elke andere letter werkt ook). De positie in de <u>fysieke regel</u> staat op &H60B3.
------	--	--

Variabele

66C4	(..)	Vlagadres voor het opzoeken van een variabele. 00 voeg de variabele aan de lijst toe 01 zoek het adres van de variabele in de lijst
------	------	---

FAC-type-indicator

66C5		Type-indicator voor de Floating Point Accumulator (FAC). In dit adres staat van welke soort variabele sprake is. Dit getal komt in register A te staan bij het aanroepen van een machinaalroutine door middel van ?USR(X), maar wordt ook binnen BASIC gebruikt, bij voorbeeld voor het berekenen van SIN(X).
------	--	---

Adres	Inhoud	Omschrijving
		De type-indicator heeft de volgende betekenissen: 02 2-byte integer (INT) 03 string (STR) 04 enkeleprecisiegetal met zwevende komma (SNG) 08 dubbeleprecisiegetal met zwevende komma (DBL)
		<u>Vlagadres</u>
66C6	(..)	Onbekend (zie &H1A1F)
66C7	(..)	Onbekend (zie &H1A22)
		<u>BASIC</u>
66C8	(..)	De inhoud van dit adres wijst naar de plaats in de BASIC-regel vóór de plaats waar de volgende instructie begint die moet worden uitgevoerd. In de directe stand wijst de inhoud van dit adres naar de overeenkomstige positie in de buffer vanaf &H642E.
		<u>Codering getallen</u>
		Bij DISK BASIC worden getallen in gecodeerde vorm in het programma opgenomen; dit in tegenstelling tot BASIC NL waarbij getallen in ASCII-code in het programma staan. Op deze adressen worden de gecodeerde waarden verzameld voordat ze in de programmaregel worden opgenomen. Deze adressen worden ook gebruikt bij berekeningen in de directe stand.
66CA	(..)	Code voor de grootte van het getal. De codes hebben de volgende betekenis: 11...1A integers 0...9 (code = &H11 + n) 0F integers 10...255 (decimaal) 1C integers 256...32767 (decimaal) 1D enkeleprecisiegetallen 1F dubbeleprecisiegetallen
66CB	(..)	Aantal posities voor het getal. 02 integers 04 enkeleprecisiegetallen 08 dubbeleprecisiegetallen
66CC	(..)	
----	(..)	8 posities voor het gecodeerde getal
66D3	(..)	
		<u>BASIC</u>
66D4	(..)	Hier staat het hoogste adres + 1 van de stringruimte. Dat is tevens het eerste adres van de stack. De inhoud van het adres kan worden veranderd met CLEAR,xxxx,nnnn. De inhoud wordt dan xxxx (hoogste gereserveerde adres voor BASIC) minus nnnn (grootte van de stack). Bij starten en
66D5	(..)	

Adres Inhoud

Omschrijving

RESET wordt de stackgrootte op 516 gezet. Het eerste adres van de stack is dan &HFDFFC. Zie ook &H63A5.

String

66D6	...	Adres van een pointer die wijst naar de eerstvolgende geldige 3 bytes in de hierna volgende stringtabel.
66D7	(66)	
66D8	...	Tabel voor het opslaan van stringdescriptors. Als BASIC bij stringopdrachten (bijvoorbeeld bij het optellen van strings of het PRINTen van foutmeldingen) strings moet opzoeken in de stringruimte of in het programma, gebeurt dat door middel van deze tabel. De totale lengte van de tabel (30 byte) is opgedeeld als 10 x 3 byte. Meestal staat hier 04 4E 17, hetgeen verwijst naar "Ok" op &H174E. Elke drie byte vormen een zogenaamde "string descriptor":
----	(..)	
66F5	(..)	
		1e byte lengte van de string 2e byte lage adres (LSB) van de plaats waar de string zelf is opgeborgen. 3e byte hoge adres (MSB) van de plaats waar de string zelf is opgeborgen.

66F6	(..)	In deze adressen worden de gegevens van een string descriptor verzameld voordat deze in zijn geheel wordt overgebracht naar de string-opslagtabel op &H66D8.
66F7	(..)	
66F8	(..)	
66F9	(..)	Het adres van het laatst gelezen teken in de stringruimte (dat wil zeggen de stringruimte bovenin het geheugen - zie &H63A5).
66FA	(..)	Bij NEW wordt de inhoud van &H66F9...66FA gelijk gemaakt aan de inhoud van &H66D4...66D5, die met CLEAR is ingesteld.

Diversen

66FB	(..)	Deze adressen worden voor twee doeleinden gebruikt:
66FC	(..)	1 aangeven van het <u>adres</u> van het token dat in uitvoering moet worden genomen (zie &H2526); 2 als vlag bij het formateren bij PRINT USING: bij omzetting van een hexadecimaal getal in een ASCII-code bevat deze vlag de aanwijzingen hoe de vorm van het getal moet worden (bij voorbeeld exponentiële notatie, decimale komma's en dergelijke). Een nul hier betekent dat geen formateren nodig is (strikte omzetting van binair in ASCII) (zie &H7FCA).

Onbekend

66FD	(00)	Deze functie van deze adressen is onbekend.
66FE	(00)	(Zie &H4BBC en &H4BC7)

<u>Adres</u>	<u>Inhoud</u>	<u>Omschrijving</u>
		<u>Pointer</u>
66FF	...	De inhoud van dit adres bevat het eindadres van de laatst uitgevoerde BASIC-opdracht.
6700	(..)	
		<u>DATA</u>
6701	...	De inhoud van dit adres geeft het regelnummer aan waarin de laatste DATA zijn gelezen.
6702	(..)	
		<u>FOR...NEXT</u>
6703	..)	Hier staat een vlag die aangeeft of FOR in uitvoering is. 00 geen FOR in uitvoering 01 FOR in uitvoering (soms staat hier &H80 (zie &H298E) of &H64 (zie &H1CD3))
		<u>INPUT en READ</u>
6704	(00)	Vlagadres voor INPUT (00) en READ (01).
		<u>BASIC</u>
6705	(..)	De inhoud van dit adres bevat het beginadres van de volgende BASIC-regel die moet worden uitgevoerd. De inhoud van dit adres wordt bij NEW gelijk gemaakt aan het begin van het BASIC-programma -1 (= &H924B bij drie bestanden).
6706	(..)	
		<u>Onbekend</u>
6707	(00)	Onbekend
		<u>AUTO</u>
6708	(00)	Vlagadres dat aangeeft of AUTO R, ΔR in uitvoering is (hierin is R het eerste regelnummer en ΔR het increment, ofwel de stapgrootte). 00 geen AUTO in uitvoering #00 AUTO in uitvoering; de waarde in dit adres geeft de stapgrootte bij AUTO aan.
6709	(00)	
670A	(00)	Het regelnummer dat door de functie AUTO moet worden gebruikt (wordt gevuld in &H18F5).
670B	(00)	
670C	(..)	De stapgrootte bij AUTO (wordt gevuld in &H2154).
		<u>Diversen</u>
670D	(..)	De inhoud van dit adres is het adres van de code in de buffer achter &H642E als die gedurende de inputfase wordt uitgevoerd.
670E	(64)	
670F	(..)	Op deze adressen wordt de stackpointer bewaard tijdens de executiefase.
6710	(..)	

Adres	Inhoud	Omschriiving
		<u>Foutafhandeling</u>
6711	(FF)	De inhoud van deze adressen geeft het nummer aan van de BASIC-regel waarin een fout is opgetreden (zie &H1811). Als er geen fout is opgetreden staat hier &FFFF.
		<u>EDIT en LIST</u>
6713	(00)	De inhoud van deze adressen is het nummer van de BASIC-regel die het laatst geEDIT of geLIST is. Deze regel kan op het scherm worden gebracht met EDIT . (EDIT + punt).
6714	(00)	
6715	(..)	Adres van de volgende instructie die moet worden uitgevoerd (in het programma of in de buffer vanaf &H642E).
6716	(..)	
		<u>Foutafhandeling</u>
6717	(00)	Deze adressen bevatten het adres van de BASIC-regel waarheen moet worden gesprongen bij ON ERROR GOTO (zie &H3FD1)
6718	(00)	
6719	(00)	Een vlagadres dat aangeeft of de foutafhandelingsroutine is aangeroepen. FF er wordt een fout (ERROR) afgehandeld 00 er wordt geen fout afgehandeld. (Door RESUME wordt de vlag weer op 00 gezet).
		<u>BASIC</u>
671A	(..)	De inhoud is het adres van het eerstvolgende token dat moet worden verwerkt.
671B	(..)	
671C	(..)	Hier wordt bij het uitvoeren van STOP en END het laatst uitgevoerde BASIC-regelnummer bewaard (zie &H1821).
671D	(..)	
671E	(..)	De inhoud is het adres van het laatste byte dat werd uitgevoerd vóór STOP.
671F	(..)	
		<u>Pointers</u>
6720	...	De inhoud is een pointer naar het begin van de variabelenruimte, en tevens het eindadres van het BASIC-programma. Deze pointer kunt u verhogen als u achter het BASIC-programma machinetaal wilt opbergen met de bedoeling om die mee weg te schrijven op schijf.
6721	..	
6722	(..)	De inhoud van deze adressen is een pointer naar het beginadres van de array-ruimte.
6723	(..)	
6724	(..)	De inhoud van deze adressen is een pointer naar het eindadres van de array-ruimte. Dat is tevens het beginadres van de vrije ruimte waarin eigen data kunnen worden opgeslagen of machinetaal kan worden neergezet. Het is ook het onderste adres van de stringruimte.
6725	(..)	

Adres	Inhoud	Omschrijving
6726	(..)	De inhoud van deze adressen is de DATA-pointer. Dit is het adres van het byte volgend op het laatst gelezen DATA-statement. Door RESTORE wordt dit adres gevuld met het beginadres van BASIC minus 1 (bij 3 bestanden &H924B). Bij RESTORE n (n = regelnummer) komt hier het beginadres van de desbetreffende DATA-regel minus 1 te staan.
6727	(..)	
		<u>Typedeclaratie</u>
6728	(04)	Deze adressen vormen een typetabel van 26 bytes voor variabelen, voor elke letter van het alfabet één (in alfabetische volgorde). De eerste positie in de tabel geeft aan van welk type variabelen zijn waarvan de naam met A (of a) begint; de tweede positie geldt voor variabelennamen die met een B beginnen, enzovoort. Bij het opstarten van BASIC en bij NEW wordt voor alle 26 beginletters 04 ingevuld (enkeleprecisievariabele) (zie &H1CA5). De inhoud van deze adressen kan worden veranderd met DEFSTR, DEFINT, DEFSNG en DEFDBL.
----	(04)	
6741	(04)	
		02 integer 03 string 04 enkeleprecisievariabele (bij opstarten) 08 dubbeleprecisievariabele
		Deze getallen geven aan hoeveel bytes moeten worden gereserveerd voor de desbetreffende variabelen.
		<u>Onbekend</u>
6742	(00)	
----	(00)	
6745	(00)	De functie van deze adressen is niet bekend.
		<u>Buffer</u>
6746	(00)	Deze adressen vormen een buffer van 100 bytes voor het opslaan van variabelennamen en variabelen die door middel van DEF FN NAAM (X,Y,...) worden opgegeven. De buffer heeft dezelfde lengte en vorm, en meestal ook dezelfde inhoud, als de buffer op adres &H67AE.
----	(00)	
67A9	(00)	
		Dit adres bevat de stackpointer die aangeeft waar de buffer precies op de stack staat.
67AA	(42)	
67AB	(67)	
		Dit adres bevat de lengte van het gevulde deel van de buffer (deze lengte wordt geteld vanaf &H67AE, zodat 100 byte beschikbaar is voor de opslag van variabelen). Bij overschrijding volgt de foutmelding "Illegal function call".
67AC	(..)	
67AD	(..)	
		Dit is de eigenlijke variabelenbuffer voor 100 byte. Elke variabele begint met zijn type (1 byte), gevolgd door zijn naam (2 ASCII-bytes) en zijn waarde:
6812	(..)	

Adres | Inhoud

Omschrijving

		2 bytes voor integers 4 bytes voor enkele precisiegetallen 8 bytes voor dubbele precisiegetallen
6813	(..)	Begin van de variabelenruimte ?
6814	(..)	
		<u>Onbekend</u>
6815	(..)	
----	(..)	De functie van deze adressen is nog niet bekend.
6849	(01)	
		<u>Beveiliging</u>
684A	(00)	Als de inhoud van dit adres ongelijk nul is, kan het programma niet worden geLIST. Ook EDIT, POKE en PEEK geven een "Illegal function call". De inhoud van dit adres wordt &HFE als een programma, weggeschreven met de toevoeging ',P', wordt ingelezen. Zo'n programma kan wel worden uitgevoerd, maar er kan niets aan worden veranderd. Zie PTC PRINT nr. 5 om de beveiliging op te heffen.
		<u>Onbekend</u>
684B	(..)	
----	(..)	De functie van deze adressen is niet bekend.
685B	(..)	
		<u>TRON/TROFF</u>
685C	(00)	Hier staat de zogenaamde "trace"-vlag. 00 TROFF #00 TRON (bij het intikken van TRON wordt de inhoud &HAF)
		<u>Tijdelijke opslag</u>
685D		Tijdelijke opslag bij rekenroutines bij het ontcijferen van getallen met drijvende komma. Gewoonlijk staat hier het laatste byte dat uit de positie voor het minst significante byte is geschoven.
		<u>FAC I</u>
685E	(..)	Eerste Floating point ACCumulator (FAC I). Hierin worden de resultaten van berekeningen opgeslagen.
----	(..)	
6865	(..)	De uitkomsten worden als volgt opgeslagen:

Adres	Dubb. prec.	Enk. prec.	Integer
655E	DP (LSB)		
685F	DP		
6860	DP		
6861	DP		
6862	DP	SP (LSB)	INT (LSB)
6863	DP	SP	INT (MSB)
6864	DP	SP	
6865	DP (MSB)	SP (MSB)	

LSB = Least Significant Byte = byte met de laagste waarde.

MSB = Most Significant Byte = byte met de hoogste waarde.

De type-indicator staat in &H66C5.

Diversen

- | | | |
|------|-----|---|
| 6866 | ..) | Teken van het getal bij rekenkundige bewerkingen. |
| 6867 | (.. | Tijdelijke opslag bij berekeningen in dubbele precisie. |

FAC II

- | | | |
|------|------|---|
| 6868 | ... | Deze buffer heeft dezelfde vorm, lengte en indeling als |
| ---- | (..) | FAC-I (&H685E...6865). Hij wordt gebruikt voor het optellen |
| 686F | (..) | van twee getallen die beide van dubbele precisie zijn. |

Onbekend

- | | | |
|------|------|--|
| 6870 | (..) | De functie van dit adres is nog niet bekend. |
|------|------|--|

PRINT-buffer

- | | | |
|------|------|---|
| 6871 | (..) | In deze buffer worden de uitkomsten van berekeningen opgeslagen in ASCII-code. Bij ?FRE(0) wordt bij voorbeeld hier |
| ---- | (..) | de vrije geheugenruimte in ASCII-code neergezet voordat die |
| 688A | (..) | op het scherm wordt weergegeven. |

FAC III

- | | | |
|------|------|---|
| 688B | (..) | Ook deze buffer heeft dezelfde vorm, lengte en inhoud als |
| ---- | (..) | FAC I (&H685E...6865). |
| 6892 | (..) | |

Onbekend

- | | | |
|------|------|--|
| 6893 | (00) | |
| ---- | (00) | De functie van deze adressen is nog niet bekend. |
| 689D | (00) | |

INP en OUT

- | | | |
|------|----------|-------------|
| 689E | CD 6A 2B | CALL 2B6A |
| 68A1 | 32 A5 68 | LD (68A5),A |
| 68A4 | DB 00 | IN 00 |
| 68A6 | C3 E1 28 | JP 28E1 |

Adres | Inhoud**Omschrijving**

Deze functie wordt gebruikt bij de BASIC-instructie INP. Op adres &H68A5 wordt in plaats van 00 het nummer van de aangeroepen poort ingevuld.

68A9 CD 54 2B CALL 2B54
68AC D3 00 OUT 00
68AE C9 RET

Deze functie wordt gebruikt voor de BASIC-instructie OUT. Daarmee kan vanuit BASIC een waarde naar een uitgangspoort worden gestuurd. De routine die begint op &H2B54 vult op adres &H68AD het nummer van de aangeroepen poort in. De inhoud van de accumulator wordt naar de desbetreffende poort gestuurd.

68AF DB 00 IN 00
68B1 AB XOR E
68B2 A0 AND B
68B3 CA AF 68 JP Z 68AF
68B6 C9 RET

Deze functie wordt waarschijnlijk gebruikt bij de BASIC-instructie WAIT.

Niet gebruikt

68B7 (00)
---- (00)
68FF (00) Deze adressen worden waarschijnlijk niet gebruikt.

BASIC

6900 (AF) Begin tweede deel DISK BASIC-interpreter

Sprongadressen en andere zaken

6909 C3 D3 6E JP 6ED3 Vervolg van de routine die begint op &H6900.
690C 00 00 00 Niet gebruikt.
690F (01) Vlagadres.
6910 C3 22 69 JP 6922 Zie aldaar.
6913 C9 00 00 RET Niet-gebruikt sprongadres.
6916 C3 D6 69 JP 69D6 Zet een teken op het scherm.
6919 C3 CB 6C JP 6CCB Haal een teken (in ASCII-code) op.
691C C3 AD 6C JP 6CAD Is er een toets ingedrukt ? Zo nee, dan terug.
691F C3 D6 69 JP 69D6 Zet een teken op het scherm.
6922 C3 00 6D JP 6D00 Wis het scherm, vraag aantal gewenste bestanden en laad desgewenst "Runtime support".
6925 C9 RET Niet gebruikt.
6926 95 6F Plaats van de stoptoets in de toetscodetabel.

<u>Adres</u>	<u>Inhoud</u>	<u>Omschrijving</u>
6928	C9 60 10	<u>RET</u> Als C9 wordt veranderd in C3 ontstaat een sprongadres naar de disk-routines in de eerste ROM. Dit adres wordt echter niet aangeroepen.
692B	C3 22 6C	<u>JP 6C22</u> Druk een teken af op de printer.
692E	C3 77 6E	<u>JP 6E77</u> Roep toetsenbord-statusroutine aan. RET als geen toets is ingedrukt. Bepaal anders de ASCII-code. Spring naar &H405E als de STOP-toets is ingedrukt.
6931	C9	<u>RET</u> Niet gebruikt.
6932	D3 6F	Begin conversietabel voor het scherm (= &H6FD3).
6934	C3 6D 69	<u>JP 696D</u> Roep PDOS aan. Naar dit adres wordt alleen gesprongen in &H6205. Zie aldaar.
6937	C9	<u>RET</u> Niet gebruikt.
6938	3D 6F	Beginadres toetscodetabel (= &H6F3D). Dit adres wordt echter niet aangeroepen.
693A	C9	<u>RET</u> Niet gebruikt.
693B	(00)	Wordt door PDOS gebruikt om de inhoud van register HL te bewaren.
693C	(00)	
693D	----	Tekst: PHILIPS DISK BASIC
696C	----	
69D6	(C5)	Opstartroutine voor DISK BASIC. Deze adressen worden alleen bij het opstarten gebruikt. Daarna mogen ze worden overschreven door eigen routines.
6ED3	----	
		<u>Toetscodetabel</u>
6F3D	(07)	Doordat de toetscodetabel in RAM staat kan voor elke toets een andere ASCII-code worden gedefinieerd (met POKE &H6F3D+n,x waarin n de toetscode is (zie Tabel 5 van de gebruiksaanwijzing P2000T) en x de gewenste ASCII-code).
6FCC	----	
		<u>Schermvertaaltabel</u>
6FD3	(7B)	Tabel voor het vertalen van ASCII-codes voor het scherm. De tabel bestaat uit vijf byte-paren. De eerste ASCII-code van elk paar wordt vertaald in de tweede voordat die naar het scherm wordt gestuurd. De tabel vertaalt het pondteken in een hekje, , in een apostrof enz. DISK BASIC zoekt in te tabel totdat hij de code 00 tegenkomt.
6FDF	(00)	De tabel kan op een andere plaats worden gezet, mits op &H6932/33 het nieuwe beginadres wordt gezet.

Printer-vertaaltabellen

6FE0 (00)

700C (20)

De tabel bestaat uit twee secties die achter elkaar staan. De eerste sectie wordt gebruikt als de twee pennen aan de onderkant van de moederprint (bereikbaar via een stop aan de onderkant van de P2000) niet zijn doorverbonden; de tweede sectie als ze wel zijn doorverbonden.

De eerste sectie is bestemd voor het gebruik van printers die geen stapje terug kunnen doen (in het algemeen matrix-printers); de tweede sectie is voor printers die dat wel kunnen (letterwiel- of golfbalprinters). Dit is gedaan omdat matrixprinters gelijktijdig een letter met accent of onderstrekking kunnen afdrukken; letterwiel- en golfbalprinters drukken eerst de letter af, doen dan een stapje terug en zetten vervolgens het accent erop of het streepje eronder.

De eerste sectie bestaat uit paren van twee bytes en begint met een getal dat het aantal byteparen van de eerste sectie aangeeft. Het eerste byte van elk paar is het byte dat vertaald moet worden; het wordt vervangen door het tweede byte van het paar.

Het begin van de tweede sectie kan worden bepaald met: beginadres eerste sectie + 2 x het aantal byteparen + 1.

DISK BASIC heeft standaard geen eerste sectie (inhoud &H6FE0 = 00). U kunt hier een eigen vertaal tabel voor matrixprinters neerzetten volgens bovenstaande opbouw.

De tweede sectie bestaat uit twee sub-secties. De eerste subsectie begint op &H6FE1 en bestaat uit trio's (drie bytes); de sectie begint weer met een getal dat het aantal trio's aangeeft (06 op &H6FE1). Het eerste byte van elk trio is het karakter dat moet worden vertaald. In plaats van dat karakter worden de twee volgende bytes naar de printer gestuurd, met daar tussenin een "backspace" (stapje terug). Daardoor is het mogelijk bij voorbeeld het hekje te vertalen in de letter e, gevolgd door een backspace en daarna door een accent aigu. Is bit 7 van het tweede of derde byte 1, dan zendt de P2000 eerst de "escape"-code (&H1B = 27) naar de printer. Daardoor is het mogelijk automatisch CHR\$(27) naar een printer te sturen, als die deze escape-code kent.

De tweede subsectie van de tweede sectie heeft dezelfde structuur als de eerste hoofdsectie: eerst een getal dat het aantal byteparen aangeeft en daarna dat aantal byteparen. Het eerste byte wordt vervangen door het tweede. De tweede subsectie begint op &H6FF4 en bestaat uit &H0C = 12 byteparen.

<u>Adres</u>	<u>Inhoud</u>	<u>Omschrijving</u>
700D	(00)	Na het laatste byte van de tweede subsectie is er een vrije ruimte van 115 bytes voor een eigen printer-vertaal tabel of voor uitbreiding van de bestaande.
707F	(00)	Vult u zelf de eerste sectie in met een eigen vertaal tabel, dan dient u de tweede sectie te verplaatsen tot na de eerste.
		<u>Omzetting HEX in decimaal</u>
7F9F	CD 22 78	Startadres van een routine die hexadecimale getallen omzet in decimale. Het hexadecimale getal moet in HL staan. De decimale waarde komt in ASCII-code in de PRINT-buffer te staan (adressen &H6874...6879).
		<u>Einde BASIC</u>
8A8D	C3 67 37	<u>JP 3767</u> - Laatste regel van de BASIC-interpreter.
		<u>Buffers</u>
8A90	(00)	Buffer van 296 bytes voor het inlezen van programma's.
-----		Op de eerste 32 bytes komt het File Control Block te staan.
8BB7	(00)	Daarna volgt één sector van het programma (256 bytes).
8BB8	(00)	Eerste bestandsbuffer, in totaal 561 bytes lang.
-----		Op de eerste staat het type bestand en op de volgende 32 posities staat het FCB van het desbetreffende bestand. Dan volgen 7 bytes waarvan de betekenis niet helemaal duidelijk is (tijdelijke opslag). Daarna komen 512 bytes met bestandsgegevens (2 sectoren) en tenslotte 9 lege posities.
8DE8	(00)	De plaats waar DISK BASIC is gebleven met het lezen van gegevens in de bestandsbuffer is te vinden met de instructie PRINT HEX\$(VARPTR(#n)), waarin n het nummer van het bestand is.
8DE9	(00)	Ruimte voor ten hoogste nog 14 buffers van elk 561 bytes.
-----		Voor het begin van deze buffers: zie &H63B5.
AC97	(00)	Het beginadres van het BASIC-programma is afhankelijk van het aantal gereserveerde buffers. Dat beginadres is te berekenen met het volgende eenvoudige programmaatje:
		10 INPUT"Aantal bestanden"; A%
		20 B\$=HEX\$(&H8A90+296+A%*561+1)
		30 PRINT"Begin BASIC-programma ="B\$
		40 GOTO 10

Spoor- en sector-indeling van schijven

De schijven, die worden gebruikt bij 24K DISK BASIC, hebben 35 of 40 sporen ("tracks") van 16 sectoren. Elke sector heeft een lengte van 256 bytes. De opslagcapaciteit per spoor is dus $16 \times 256 = 4096$ bytes (4 Kbyte) en per schijf 35×4 Kbyte = 140 Kbyte of 40×4 Kbyte = 160 Kbyte.

Het aantal sporen wordt bepaald door de constructie van de diskettestations ("Floppy Disk Drives") en door PDOS. Als de diskettestations 40 sporen aankunnen, kan PDOS eenvoudig zo worden veranderd dat dit besturingssysteem eveneens met 40 sporen werkt.

Bij het wegschrijven van programma's of bestanden gebruikt PDOS niet eenvoudig zoveel sectoren als nodig zijn, maar een veelvoud van "records" van 4 sectoren ofwel 1024 bytes. Een programma van bij voorbeeld 200 bytes neemt dus 1 record van 1024 bytes (vier sectoren) in beslag, evenals een programma van 1024 bytes. Bovendien sluiten de sectoren niet op elkaar aan; ze zijn verspreid over het spoor. Met andere woorden: de logische volgorde van de sectoren is niet gelijk aan de fysieke volgorde. Record nr. 04 bestaat bij voorbeeld uit de sectoren 1, 7, 13 en 3 (in die volgorde) van spoor 2 en record 05 uit de sectoren 9, 15, 5 en 11 van spoor 2.

Het eerste spoor van werkschijven (niet van de systeemschijf) bevat de index. Hierin staan de FCB's ("File Control Blocks") van de programma's en bestanden. Elk FCB bestaat uit 32 posities en ziet er (als u het met één van de monitor-programma's bekijkt) bij voorbeeld als volgt uit:

00	4D	4F	4E	49	54	4F	52	MONITOR
45	42	41	53	00	00	00	1B	EBAS
0E	0F	10	11	12	13	14	00	
00	00	00	00	00	00	00	00	

Op de posities 2 tot en met 9 staat de naam (in dit geval MONITORE). Als de naam korter is dan 8 tekens, wordt hij automatisch aangevuld met spaties. Op de 16e positie staat het aantal sectoren (in dit geval &H1B = 27; de lengte van het programma is dus minder dan $27 \times 256 = 6912$ bytes. Vanaf positie 17 staan de nummers van de records. In de tabel op de volgende bladzijde kunt u zien dat 8 sectoren van spoor 4, alle 16 sectoren van spoor 5 en 4 sectoren van spoor 6 zijn gebruikt; samen $7 \times 4 = 28$ sectoren, één meer dan strikt genomen nodig zou zijn. De laatste sector van record &H14 wordt dus niet gebruikt.

De posities 17 tot en met 32 van het FCB noemt men de "Disk Allocation Map", ofwel de toewijzing van sectoren aan het programma of bestand. Is zo'n file langer dan 16 Kbyte, dan wordt een tweede "entry" in de index geopend.

Bij het werken met bestanden wordt in de bestandsbuffers het FCB bijgehouden. PDOS noteert hier dus waar wat staat, om het later weer te kunnen vinden. De instructie CLOSE maakt dat het bijgewerkte FCB wordt opgenomen in de index op het eerste spoor. Vergeet u CLOSE, dan is het bestand later niet meer helemaal, of helemaal niet meer te vinden.

Overzicht spoor- en sectorindeling

Decimaal	1	7	13	3	9	15	05	11	2	8	14	104	10	16	6	12
Hexadec.	00	06	0C	02	08	0E	04	0A	01	07	0D	03	09	0F	05	0B
Spoor																
1	00				01				02				03			
2	04				05				06				07			
3	08				09				0A				0B			
4	0C				0D				0E				0F			
5	10	10	10	10	11	11	11	11	12	12	12	12	13	13	13	13
6	14	14	14	14	15	15	15	15	16	16	16	16	17	17	17	17
7	18	18	18	18	19	19	19	19	1A	1A	1A	1A	1B	1B	1B	1B
8	1C	1C	1C	1C	1D	1D	1D	1D	1E	1E	1E	1E	1F	1F	1F	1F
9	20				21				22				23			
10	24				25				26				27			
11	28				29				2A				2B			
12	2C				2D				2E				2F			
13	30				31				32				33			
14	34				35				36				37			
15	38				39				3A				3B			
16	3C				3D				3E				3F			
17	40				41				42				43			
18	44				45				46				47			
19	48				49				4A				4B			
20	4C				4D				4E				4F			
21	50				51				52				53			
22	54				55				56				57			
23	58				59				5A				5B			
24	5C				5D				5E				5F			
25	60				61				62				63			
26	64				65				66				67			
27	68				69				6A				6B			
28	6C				6D				6E				6F			
29	70				71				72				73			
30	74				75				76				77			
31	78				79				7A				7B			
32	7C				7D				7E				7F			
33	80				81				82				83			
34	84				85				86				87			
35	88				89				8A				8B			
36	8C				8D				8E				8F			
37	90				91				92				93			
38	94				95				96				97			
39	98				99				9A				9B			
40	9C				9D				9E				9F			

Nr.	Token	Disk-BASIC code	Cass.-BASIC adres	Cass.-BASIC code	Cass.-BASIC adres
1	AND	F7	----	C2	344E
2	ABS	FF 86	76AD	CE	371E
3	ATN	FF 8E	8795	DA	457F
4	ASC	FF 95	4CA4	E5	4C34
5	AUTO	AB	212C	AA	2A6E
6	CLOSE	C3	38BD		
7	CONT	9A	40BA	A6	1D82
8	CLEAR	92	4162	AB	1E4C
9	CINT	FF 9C	77C0	DC	3831
10	CSNG	FF 9D	783A	DD	38AB
11	CDBL	FF 9E	7866	DE	38D7
12	CVI	FF AB	3651		
13	CVS	FF AC	3654		
14	CVD	FF AD	3657		
15	COS	FF 8C	86ED	D7	44D7
16	CHR\$	FF 96	4CB4	E6	4C44
17	CALL	B6	3B11		
18	COMMON	B8	1FFE		
19	CHAIN	B9	3B94		
20	CLOAD			AC	4E7B
21	CSAVE			AD	4E5D
22	DATA	84	1FFE	83	293A
23	DIM	86	46C1	85	4791
24	DEFSTR	AD	1EE4	97	284D
25	DEFINT	AE	1EE7	98	2850
26	DEFSNG	AF	1EEA	99	2853
27	DEFDBL	80	1EED	9A	2856
28	DEF	98	2937	A3	30E5
29	DELETE	AA	2D63	A9	33D5
30	END	81	4063	80	1D33
31	ELSE	A2	2000	92	293C
32	ERASE	A6	411B	96	1DE4
33	EDIT	A7	42B5	9C	1A48
34	ERROR	A8	2121	9D	2A63
35	ERL	D6	73D4	B6	----
36	ERR	D7	73D1	B7	----
37	EXP	FF 8B	859E	D6	43C4
38	EOF	FF AF	3050		
39	EQV	FA	----	C5	3630
40	FOR	82	1CD1	81	272C
41	FIELD	C0	38FB		
42	FILES	C6	3543		
43	FN	D3	7BAA	B1	1C8B
44	FRE	FF 8F	4EB8	CF	4E3F
45	FIX	FF 9F	78B3	DF	3924
46	GOTO	89	1FAC	88	28F4
47	GO TO	89	1FAC		
48	GOSUB	8D	1F94	8C	28E3
49	GET	C1	8812		
50	HEX\$	FF 9A	4A51	E2	4A02
51	INPUT	85	2374	84	2BF1
52	IF	8B	2162	8A	2AA4
53	INSTR	DA	774C	B9	----
54	INT	FF 85	78C6	CD	3937
55	INP	FF 90	689E	DO	1801
56	IMP	FB	----	C6	37BD
57	INKEY\$	DD	7997		
58	KILL	C8	3519		
59	LET	88	202A	87	295F
60	LINE	B1	2306	9B	2BB4
61	LOAD	C4	376F		
62	LSET	C9	3965		
63	LPRINT	9E	21A4	A2	2ADE
64	LLIST	9F	2B76	A8	3331
65	LPOS	FF 9B	28D7	D1	3084
66	LIST	93	2B7B	A7	3336
67	LOG	FF 8A	7509	D5	3583
68	LOC	FF B0	3117		
69	LEN	FF 92	4C98	E0	4C28
70	LEFT\$	FF 81	4D0A	E8	4C9A
71	LOF	FF B1	312F		
72	MERGE	C5	3830		

Nr.	Token	Disk-BASIC code	Cass.-BASIC adres	Cass.-BASIC code	Cass.-BASIC adres
73	MOD			FC	----
74	MKI\$	FF	B2	3638	
75	MKSS	FF	B3	363B	
76	MKD\$	FF	B4	363E	
77	MID\$	FF	83	4D45	EA 4CD5
78	NEXT			83	41F1
79	NULL			96	40CE
80	NAME			C7	337B
81	NEW			94	3F81
82	NOT			D5	77B9
83	OUT			9D	68A9
84	ON			95	2097
85	OPEN			BF	33D0
86	OR			F8	----
87	OCT\$	FF	99	4A4B	E1 49FC
88	OPTION			BA	2F1C
89	PUT			C2	8811
90	POKE			99	2DA7
91	PRINT			91	21AC
92	POS	FF	91	28DD	D2 308A
93	PEEK	FF	97	2D9D	DB 17F9
94	READ			87	243E
95	RUN			8A	1F7E
96	RESTORE			8C	4043
97	RETURN			8E	1FE3
98	REM			8F	2000
99	RESUME			A9	20E4
100	RSET			CA	3964
101	RIGHT\$	FF	82	4D3B	E9 4CCB
102	RND	FF	88	8651	D4 4462
103	RENUM			AC	2DD8
104	RESET			CC	3500
105	RANDOMIZE			BB	2F66
106	STOP			90	405E
107	SWAP			A5	40DD
108	SAVE			CB	3872
109	SPC(D4	7C8E
110	STEP			D1	7A6C
111	SGN	FF	84	76C2	CC 3733
112	SQR	FF	87	8541	D3 4370
113	SIN	FF	89	86F3	D8 44DD
114	STR\$	FF	93	4A57	E3 4A08
115	STRING\$			D8	754F
116	SPACE\$	FF	98	4CED	E7 4C7D
117	SYSTEM			BD	34D2
118	TRON			A3	40D7
119	TROFF			A4	40D8
120	TAB(DO	783A
121	TO			CE	77C0
122	THEN			CF	7881
123	TAN	FF	8D	8780	D9 456A
124	USING			D9	75B5
125	USR			D2	7A65
126	VAL	FF	94	4D66	E4 4CF6
127	VARPTR			DC	796B
128	WIDTH			A1	2B18
129	WAIT			97	2AFD
130	WHILE			B4	3A79
131	WEND			B5	3A9C
132	WRITE			B7	3E87
133	XOR			F9	----
134	+			F2	----
135	-			F3	----
136	*			F4	----
137	/			F5	----
138	↑			F6	----
139	÷			FD	----
140	↓			DB	----
141	>			EF	----
142	=			FO	----
143	<			F1	----
* werkt niet in Cassette-BASIC				CB	37EA

ROM-routines uit de BASIC-NL-module

Adres Routine

- 2631 Zoek regel. Entry: in DE te zoeken regelnr. Exit: NC=Niet gevonden. C=Wel gevonden. Dan: BC=adres juiste regel, HL=adres regel daarop. Routine kort en goed leesbaar.
- 2CF2 Bereken de expressie vanaf HL. Resultaat (string of getal) in FAC1. Routine doet zijn best en keert zonder foutmelding terug bij het eerste niet-te-berekenen-teken, bijv. een komma, dat dan in (HL) staat.
- 38F2 Test type FAC1 (63B6). Type Mismatch als het geen string (3) is.
- 1D03 Vergelijk (HL) met het byte direct na de CALL (dus in het aanroepende programma!). Verschil = Syntax error. Gelijk = JP 2833
- 2833 Volgende teken in prg.regel vanaf HL, >< spatie of TAB. C=Cijfer, NC= geen cijfer, Z=nul. Algemeen om het volgende teken te pakken: springt over spatie's en TABS heen.
- 1F0A Huidige regelnr. (625A) in HL, Z=directe stand, NZ=Prg. stand.
- 28AC "VAL" van rijtje ASCII-codes vanaf HL. Komt in DE terecht. Geen cijfers: 0 en NC. Syntax error als >65535, geen negatieve getallen. Zie ook 3D0B. Prima geschikt voor adressen of regelnummers, maar kan niet rekenen.
- 28D4 RUN Entry met NC: RUN "..."; NZ: RUN <regelnr>; C&Z: RUN. Evt. string of getal moet vanaf HL staan. C&Z te verkrijgen met 2833!
- 295F LET Entry: HL op teken na LET (dus eerste teken variabele). Exit: DE=VARPTR. Maar de toekenning heeft dan al plaatsgehad.
- 3303 Bereken expressie vanaf HL. Komt in A en DE. >256=Syntax error.
- 373D Test het getal in FAC1 op TRUE/FALSE. Z=FALSE, NZ=TRUE.
- 1900 INPUT met "?". Exit: HL=adres buffer. C=STOP ingedrukt geweest. Loopt door in:
- 1911 INPUT zonder "?". Verder idem als 1900.
- 1A9B (of via 60E0) EDIT Buffer wordt opgehaald als begin >< 0. Wordt aange-roepen door 1900 en dus ook door 1911.
- 2E71 DE/HL. Levert SNG-getal op in FAC1. Overflow of Division by Zero kan optreden en forceert een BASIC-foutmelding.
- 2F2E Zet het teken in A om naar een hoofdletter.
- 3383 PRINT alle bytes vanaf HL, afgesloten door een "echte" nul.
- 3831 CINT FAC1 -> INT in FAC1 en in HL. Geen strings natuurlijk. Let op een Overflow!
- 39BA DE := DE*BC. Positieve integers 0<INT<65535, >FFFF = Subscript o.o.range Wordt gebruikt om een arrayelement te vinden, vandaar.
- 3A06 FAC1 := HL := HL*DE. Ook negatief mag, Overflow -> SNG in FAC1.
- 3D0B Leest ASCII-codes vanaf HL -> FAC1. Getal mag gevolgd worden door een typeteken. Geen typeconversie, dus slijt alles (losse cijfers zonder typeteken wordt dus INT, zie ook 2BAC). Rekent echter niet.
- 3E92 PRINT "in..." met als regelnr. het getal in HL (evt. FFFF).
- 4BE7 Verplaats string. HL op descriptor, DE op bestemming. Exit: DE op adres na de verplaatste string.
- 4E5D CSAVE. Entry: HL op eerste teken na CSAVE, dat ook in A moet staan (via bijv. 2833). Vult volledige header op 6130, evt. na gebruik van de Stringsaveroutines. Dan volgt CALL 60CC = recorder. Door hier tijdelijk C9 = RET neer te zetten wordt de recorder afgevangen, header is dan nog steeds ongewijzigd. Exit: HL op teken na laatst leesbare teken (meestal is dat dus na de stringexpressie).
- 3E9E PRINT het getal in HL, rechts uitgelijnd, totaal zes tekens inc. spatie.
- 3EA9 STR\$ FAC1 (alle types) -> ASCII vanaf 651C, beeindigd met een echte nul. In HL staat na afloop 651C, dus direct geschikt voor CALL 3383.
- 3F88 Zet A in ASCII in het geheugen vanaf (HL+1), altijd 2-cijferig dus bijv. 06. 00<A<99!

38AB CSNG

344E SNG ADD

3630 SNG DIV

35C9 SNG MULT

VARPTR opzoeken. Naam van de variabele vanaf adres HL, geen spaties vooraf, arrayelementen (A(7)) mogen ook. VARPTR komt terug in DE. Een nog onbekende variabele wordt bijgemaakt. Na afloop staat HL na de variabele-naam. (63E7)=0 bij entry: als beschreven. Wanneer (63E7)=1 bij de entry wordt de VARPTR van het array met de genoemde naam in BC teruggegeven. BC wijst dan naar het aantal dimensies.

NEW Initialisatie systeemvariabelen. Let op: de stack wordt in zijn geheel gewist en de stackpointer net onder de stringruimte gezet. Het is echter toch een echte CALL, de routine keert netjes terug.

25FB Update line pointers vanaf adres DE. Meestal LD DE,(625C): CALL 25FB.

25F1 Beeindig een LOAD-opdracht: update line pointers en naar directe stand. Meestal: LD DE,(625C): PUSH DE: JP 25F1 of JP 25F2 zonder PUSH DE.

Haal string uit FAC1: beginadres komt terug in DE, ie letter in A. I11. function call als de lengte nul is. Filenaam voor recorder!

4E3F *FRE* Type in (63B6). Schuift de stringruimte aan wanneer nodig.

4A15 Verplaats string naar de stringruimte, volledig automatisch incl. Out of string space e.d. als dat nodig is. Entry: HL op stringdescriptor Exit: DE wijst naar nieuwe descriptor die nog over de oude heen moet worden geschreven! (BC) geeft het teken na de verplaatste string.

1CFD Vergelijk HL met DE. Z als HL=DE. S&C als HL<DE. NZ&P&NC als HL>DE.

15D3 Print de header op 6030. Let er op dat het aantal blokken ongewijzigd wordt overgenomen! (routine kijkt niet naar de echte lengte). = CALL 15D3 + "Hier overheen (j/n)?" en afhandeling vraag. Komt de routine terug dan was het "Ja", bij "Nee" volgt een BASIC-foutmelding. Er wordt continu naar de cassettestatus gekeken (w/g cass, w/g stopje). Behandel de string in FAC1 als naam met evt. extensie en zet die netjes ge"headerd" op 6130 neer. Een "oude", daar staande extensie wordt alleen overschreven als er een nieuwe wordt opgegeven. Moet worden voorafgegaan door CALL 16B3 om restanten van een eerder opgegeven naam te wissen.

293A Zoekt nul of ":" (End Of Line) vanaf HL.

1688 CLOAD-routine die de parameters bij CLOAD en CSAVE (1059 en 105c) uitzoekt en in de header op 6130 zet. Dus HL op 6130, lengte op 6132, 6134 en 6143 en de meegegeven naam. Extra: in A de extensiecode (een zo meegegeven extensie wordt overschreven als er in de naam een extensie wordt meegegeven). 0=DBL, 1=BAS, 2=INT, 3=STR, 4=SNG. Deze code wordt standaard op 1 (BAS) gezet, met CALL 168A wordt de meegegeven code gebruikt.

104A PRINT het teken in A. Identiek aan PRINT CHR\$(A).

104D Wacht op toets en geeft ASCII-code terug in register A.

1053 Foutmeldingen >= 64 (STOP). Foutcode in A meegeven. ON ERROR GOTO werkt

1056 Initialisatie systeemvariabelen, maar GEEN "NEW" (BASIC blijft intact).

105F LPRINT het teken in A (vergelijk 104A).

1062 Toetsindruk? NZ = toets ingedrukt.

1071 Inhoud cassette net zoals ZOEK.

1074 Cassette wissen ? (J/N). "J" moet nog los ingetikt worden.

1059 CLOAD HL=begin, DE=max.eind+1, A=ie letter naam.

105C CSAVE HL=begin, DE=eind, FAC=Hele naam in stringvorm.

1077 LPRINT venster (= PRINT CHR\$(5)).

247D Foutmeldingen < 64. Code in register E meegeven. ON ERROR GOTO werkt.