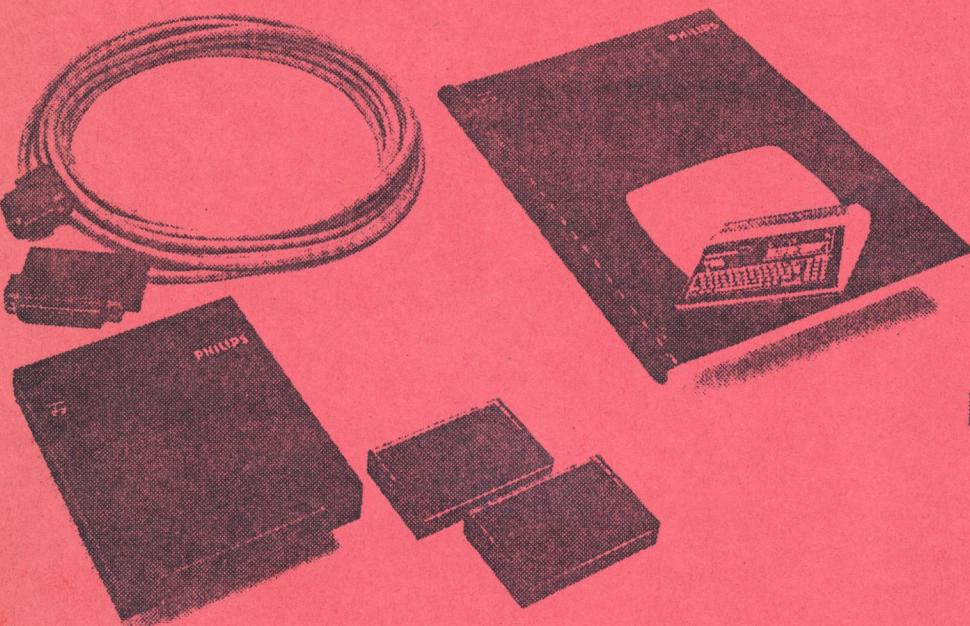
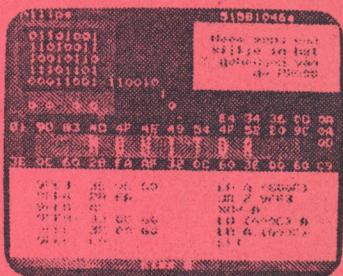


ASSEMBLY

SUBROUTINES



Z80 ASSEMBLER + MONITOR



Z80 DISASSEMBLER



HANS VAN DER VEER

Voor degenen die al een beetje overweg kunnen met assembly, maar nog niet zover zijn dat ze zelf vlot een routine kunnen schrijven volgen hier enkele subroutines voor algemeen gebruik.

Voor gevorderden zijn dit natuurlijk al bekende zaken die ze misschien zelf slimmer zouden aanpakken, maar toch kan een verzameling "standaard"-routines ook voor hen wel handig zijn bij het maken van een programma.

Dit artikel geeft enkele routines voor het scherm. Bij voldoende belangstelling kunnen nog cassette-, reken- en conversieroutines volgen.

Van elke routine is de functie aangegeven en verder nog :

1. De condities waarmee de routine aangeroepen moet worden (IN :).
2. Waar nodig, de toestand van de registers bij het verlaten van de routine (UIT :).
3. De registers, die door de routine veranderd zijn (REG :).

Alle adressen en getallen zijn hexadecimaal en kunnen dus zonder omrekenen met de Assembler van Ron Eijnhoven verwerkt worden.

De listings zijn als volgt ingedeeld :

Links de Object Codes, zoals ze in het programma gebruikt worden.

Rechts de Mnemonics, de Z-80 instrukties voor de Assembler, en daarachter voor zover de ruimte het toelaat de toelichting.

In het midden van de listing de Labels :

Globale labels (mogen van buiten de routine aangeroepen worden) in hoofdletters ;

Locale labels (worden alleen gebruikt in de subroutine) beginnen met een \$-teken.

De namen van de labels zijn in verband met de leesbaarheid soms wat langer dan de lengte van 4 tekens, die de R.E.-Assembler hanteert.

(Deze tekst is ook gepubliceerd in TRON 4, augustus 1985)

Hans van der Veer - Luytelaer 27 - 5632 BE Eindhoven - Tel. 040 / 417389
27-01-87 - VRIJE PUBLIKATIES - Broederhof 11 - 5504 JC Veldhoven

Laten we beginnen met een routine om het scherm te wissen. We kunnen natuurlijk gebruik maken van de routine in de monitor en dat kan dan als volgt :

```
21 00 50 CLSCR LD HL,5000
3E 18           LD A,18 24 regels
CD 35 00        CALL 0035
C9              RET
```

Hier wordt in HL het adres van het linkerhoekpunt van het scherm gezet en in de accumulator het aantal te wissen regels, waarna de monitorroutine voor wissen wordt aangeroepen met CALL 0035.

Mooi kort, maar het nadeel hiervan is dat de monitorroutine 80 tekens breed wist en dus ook de 2e pagina van het scherm. Als U geen stand-alone programma maakt, is natuurlijk ook de BASIC-routine op adres &H 120F beschikbaar, maar aardiger is het om zelf iets te maken.

De volgende routine doet het eens anders en begint onderaan het scherm (adres &H 5757).

*** Linker schermpagina wissen ***

Grootte 22 bytes

Functie : Wist de linkerhelft van het scherm met een breedte van 40 kolommen. De aanroep met CALL CLSCR wist 24 regels (dus het gehele linkerscherm).

Met CALL CLLOW kan het onderste deel van het (linker) scherm worden gewist : in dit geval moet in de accumulator het aantal regels staan dat gewist moet worden. CLLOW kan handig zijn voor het wissen van foutmeldingen.

IN : Voor CLSCR geen eisen.
(voor CLLOW) A : aantal te wissen regels.

UIT : -

REG : AF, HL

```

3E 18    CLSCR   LD A,18    24 regels
21 57 57 CLLOW   LD HL,5757  Schermadres
C5        PUSH BC   Bewaar BC
06 28    $c11    LD B,28    40 kolommen
48        LD C,B
36 00    $c12    LD (HL),00  Wis locatie
2B        DEC HL    Plaats terug
10 FB    DJNZ $c12
A7        AND A     Carry nul
ED 42    SBC HL,BC  Regel omhoog
3D        DEC A     Teller -1
20 F2    JR NZ $c11
C1        POP BC   Herstel BC
C9        RET

```

Om het gehele scherm te inverteren wordt INVERS aangeroepen.
De aanroep van de VINU-routine (VensterINVers) voor een venster van b.v. 9 breed en 7 hoog, op schermlocatie &H 5520 gaat zo :

3E 07	LD A,07	Hoogte 7
06 09	LD B,09	Breedte 9
21 20 55	LD HL,5520	Schermadres
CD yy xx	CALL VINU	
:	:	
:	:	

Als U de object code direct in DATA-regels van een BASICprogramma zet, moet U voor yy xx het geheugen-adres &H xxyy nemen, waar VINU begint.

Veranderen we in de INVERS routine de instruktie SET 7,(HL) weer door LD (HL),00 dan hebben we een wis het venster-routine gekregen. Een andere routine, die het 8e bit van een schermbyte wijzigt, is de cursor routine voor de P2000. Hier wordt eerst de inhoud van de schermlocatie in de accumulator geladen, dan volgt een XOR (exclusive OR) met &H 80. Tenslotte wordt het resultaat weer in het schermadres teruggezet :

***** Print / Wis Cursor *****

IN : HL
UIT : -
REG : Ongewijzigd

F5	CURSOR	PUSH AF
7E		LD A,(HL)
EE 80		XOR 80
77		LD (HL),A
F1		POP AF
C9		RET

```

IN : Voor INVERS geen eisen
      ( voor VINU ) A : vensterbreedte
                      B : vensterhoogte
                      HL : schermadres
                      van hoekpunt rechtsonder
UIT : -
REG : AF, BC, HL
3E 18    INVERS   LD A,18    24 regels
06 28    LD B,28    40 kolommen
21 57 57 LD HL,5757  Schermadres
C5        PUSH BC   Bewaar BC &
E5        PUSH HL   HL op stack
CB FE    $inv    SET (HL),7  Invert. tot
2B        DEC HL    hele breedte
10 FB    DJNZ $inv  klaar is.
E1        POP HL    Haal eerste
01 50 00 LD BC,0050  locatie op
A7        AND A     en ga een
ED 42    SBC HL,BC  regel omhoog
C1        POP BC
3D        DEC A     Regelsteller
20 EE    JR NZ VINU  <> 0, opn.
C9        RET

```

(Deze tekst is ook gepubliceerd in IRON 4, augustus 1985)

Hans van der Veer - Luytelaer 27 - 5632 BE Eindhoven - Tel. 040 / 417389
27-01-87 - VRIJE PUBLIKATIES - Broederhof 11 - 5504 JC Veldhoven

Ter afsluiting van deze reeks een routine voor het printen (op het scherm) van een blok tekst.

Deze stond al eens op een prikbord, maar omdat niet iedereen over een modem beschikt, wil ik hem hier toch (iets gewijzigd) laten zien.

***** Print een blok tekst *****

Grootte : 42 bytes

Funktie : Zet een blok tekst op het scherm.

In de tekst kan het einde van een regel worden aangegeven met &H OE en een xx aantal spaties met &H 10 xx.

De tekst afsluiten met &H OF.

IN : BC = geheugenplaats waar de tekst staat.

HL = schermadres, waar de tekst heen moet.

REG : AF, DE, HL

OB	SHTEXT	DEC BC	
54	\$nwlin	LD D,H	Adres begin-
5D		LD E,L	regel in DE
03	\$next	INC BC	Haal volgend
0A		LD A,(BC)	teken tekst
FE OF		CP OF	Einde tekst ?
C8		RET Z	Zoja, Return
FE OE		CP OE	Einde regel ?
28 08		JR Z \$down	
FE 10		CP 10	Spaties ?
28 0D		JR Z \$stab	Zoja, \$stab
77		LD (HL),A	Print teken
23		INC HL	1 plaats
18 EF		JR \$next	verder
EB	\$down	EX DE,HL	In HL beginr.
C5		PUSH BC	bereken adres
01 50 00		LD BC,0050	nw.regel
09		ADD HL,BC	eronder
C1		POP BC	
18 E4		JR \$nwlin	
03	\$stab	INC BC	Haal getal
0A		LD A,(BC)	achter de 10
B7		OR A	Is dit nul
28 E1		JR Z \$next	dan geen spaties
36 20	\$tnx	LD (HL),20	Print spatie
23		INC HL	en ga door
3D		DEC A	tot genoeg.
18 F8		JR NZ \$tnx	
18 D8		JR \$next	Naar volgend
			teken tekst.

Het volgende voorbeeld demonstreert het gebruik van deze routine. De tekst, die op adres &H xxxy in het geheugen staat, gaat naar de plaats &H 5050 op het scherm. De SHTXT (ShowText)-routine staat op adres &H nnzz in het geheugen.

```
01 yy xx      LD BC,TEKST
21 50 50      LD HL,5050
CD zz nn      CALL SHTXT
:
:
```

Bij gebruik van de Assembler hoeven we de geheugenadressen niet allemaal zelf in te vullen. Het voorbeeld wordt dan als volgt ingetypt :

ORG 9800		
LD BC,TEKST		
LD HL,5050		
CALL SHTXT		
RET		
TEKST	ASC Demonstratie	Tekst
	BYTE OE	Eind regel
	ASC Op deze regel	Tekst
	BYTE 10 OC	12 spaties
	ASC 12 spaties	Tekst
	BYTE OF	Eind tekst
	SHTXT DEC BC	
	\$nwlin LD D,H	
	LD E,L	
	INC BC	
	LD A,(BC)	
	:	en zo de rest van routine.

Het programma kan nu gestart worden op &H 9800.

Door uit de SHTXT-routine LD (HL),20 weg te laten, worden geen spaties geprint. Op die manier kan een tekst, die al op het scherm stond, gespaard blijven.

(Deze tekst is ook gepubliceerd in TRON 4, augustus 1985)

De vorige keer hebben we het printen op het scherm besproken, nu komt het invoeren van gegevens via het toetsenbord aan de beurt. Om het simpel te houden laten we het lezen van de Inputpoort buiten beschouwing en beperken we ons tot het gebruik van de routines in de CPU-monitor en Basic Module.

De eenvoudigste invoer is het ophalen van een enkele toetsindruk. Als hierbij niet op het indrukken gewacht hoeft te worden, kan volstaan worden met rechtstreeks te kijken naar de toetsbuffer (adres &H 6000-600B) of, als het alleen om de laatst ingedrukte toets gaat, naar adres &H 600D.

B.v. : LD A,(600D). In A komt nu de toetscode van de laatst ingedrukte toets.

Wilt U het programma op een toetsindruk laten wachten, dan is de aanroep van de monitorroutine op adres &H 0026 voldoende :

CD 26 00 CALL 0026

Na het indrukken van de toets gaat de micro-processor verder met het programma, waarbij de toetscode in de accumulator staat.

De overige registers bevatten na de CALL dezelfde inhoud als voor de aanroep.

Na de aanroep volgt dan de controle of de juiste toets is ingedrukt en als dat niet het geval is, kan een waarschuwing-PIEP gegeven worden met de aanroep van de monitorroutine op &H 0032.

Een voorbeeld is de routine OK?, waarbij het programma met de N-toets (toetscode &H 61) beëindigd wordt en met J (toetscode &H 56) doorgaat naar Label "Jawel".

Alle andere toetsen worden geweigerd :

CD 26 00	OK?	CALL 0026	Wacht op toets
FE 56		CP 56	Is het J ? Zoja
28 08		JR Z Jawel	dan Jawel
FE 61		CP 61	Is het N ? Zoja
C8		RET Z	dan return
18 F1		JR OK?	Andere toets

Jawel
Rest programma

Hans van der Veer - Luytelaer 27 - 5632 BE Eindhoven - Tel. 040 / 417389
27-01-87 - VRIJE PUBLIKATIES - Broederhof 11 - 5504 JC Veldhoven

In deze vorm maakt de routine deel uit van het hoofdprogramma of wordt er met een Jump naar toe gesprongen.

Komt de vraag Ja / Nee vaker voor, dan is het beter er een subroutine van te maken.

Nu wordt indrukken van de N-toets de Carry-vlag gezet en moet na terugkeer uit de OK?-routine beslist worden wat er gaat gebeuren.

***** Wacht op J-of N-toets *****

Grootte 16 bytes.

IN : -	
UIT : A = toetscode	
Carry = 0 als J ingedrukt werd	
Carry = 1 als N ingedrukt werd	
REG : Overige ongewijzigd.	

CD 32 00	fout	CALL 0032	Pieptoon
CD 26 00	OK?	CALL 0026	Wacht op toets.
FE 56		CP 56	Is het J Zoja
28 05		JR Z klaar	dan klaar
FE 61		CP 61	IS het N
20 F2		JR NZ fout	Zoniet
			dan fout
37		SCF	Zet Carry -vlag
C9		RET	Routine return

In het hoofdprogramma wordt deze routine nu als volgt gebruikt :

CD yy xx		CALL OK?	Haal J/N-toets
DA bb aa		JP C Nee	Als Carry =1 dan Nee
...	Ja	Carry = 0 dus Ja
...	Afhandeling
...	Nee	Nee-afhand.

Een programma, dat vriendelijk voor de gebruiker is, laat op het scherm zien welke toets ingedrukt is.

De toetscode moet dan eerst met behulp van een tabel worden omgezet in de ASCII-waarde.

In de BASIC-module bevindt deze tabel zich op adres &H 1814. (vervolg op M.205).

(Deze tekst is ook gepubliceerd in TRON 5, oktober 1985)

Als U in de accumulator de toetscode hebt, gaat het omzetten als volgt :

*** Vertaal toetscode naar ASCII ***

Grootte : 12 bytes
 IN : A = toetscode
 UIT : A = ASCII-waarde
 REG : Overige ongewijzigd.

ES	GTAS	PUSH HL			
21 14 18		LD HL,1814	Adres ASCII-		
			tabel		
85		ADD L			
6F		LD L,A			
30 01		JR NC pak'm	Stoptoets :		
			jump		
24		INC H			
7E	pak'm	LD A,(HL)	ASCII-waarde		
E1		POP HL	Haal HL terug		
C9		RET			

Toetscodes met een eigen routine omzetten heeft alleen maar zin in een stand-alone programma, of als U zowel ASCII-waardes als toetscodes (b.v. voor funktietoetsen) nodig hebt. Gebruikt U alleen ASCII dan is een CALL 104D eenvoudiger : de BASIC verzorgt nu de invoer en de vertaling van de toets.

Tot slot van deze aflevering een voorbeeld van een uit meerdere toetsindrukken bestaande invoer. Met de routine GETAL kan een getal worden ingevoerd, bestaande uit een van te voren opgegeven aantal cijfers.

Maar let wel op : het grootste getal wat de routine kan innemen is 65535 (&H FFFF).

De routine gebruiken voor meer dan 5 cijfers heeft dus geen zin. Alvorens de routine aan te roepen moet eerst in het B-register het aantal cijfers (1-5) ingevuld worden, en in HL het schermadres van de plaats waar de invoer te zien moet zijn, bijvoorbeeld :

```
LD B,03
LD HL,5700
CALL GETAL
```

Tijdens de invoer kan met de backspace een verkeerde aanslag gekorrigeerd worden.

(Deze tekst is ook gepubliceerd in IRON 5, oktober 1985)

Hans van der Veer - Luytelaer 27 - 5632 BE Eindhoven - Tel. 040 / 417389
 27-01-87 - VRIJE PUBLIKATIES - Broederhof 11 - 5504 JC Veldhoven

Na het indrukken van de Enter-toets volgt een return naar het aanroepende programma : in het DE-register staat dan het ingevoerde getal (binair).

Het minst significante byte van het getal staat ook in de accumulator, maar is als uitkomst natuurlijk alleen betrouwbaar als het kleiner is dan 256.

Dit kan na het verlaten van de GETAL-routine gecontroleerd worden door de carry-vlag te testen.

Globaal gezien werkt de routine als volgt :

De invoer wordt in eerste instantie op het scherm gezet met de gnum (get number)-subroutine.

Nadat de Enter-toets is ingedrukt worden de tekens van het scherm gelezen en met de conversieroutine convrt omgezet van ASCII naar binair. De getalwaarde wordt hierbij bepaald met een lus, waarin telkens met een factor 10 vermenigvuldigd wordt.

Om het C-register in tact te houden (makkelijk voor opslag van een waarde die na de aanroep van GETAL nog weer nodig is), staat aan het begin en eind van de routine PUSH BC, resp. POP BC.

GETAL maakt gebruik van de routine CURSOR, die ik in de vorige aflevering besproken heb.

***** Getalinvoer *****

Grootte : 90 bytes

Funktie : Invoer van een getal, bestaande uit naar keuze 1-5 cijfers. Maximale getalgrootte 65535, of 255 bij uitlezen van A.

IN : B = aantal cijfers dat verwacht wordt.

HL = schermlocatie, waar de cursor moet komen te staan.

UIT : DE = resultaat

Als DE >= 256 dan

Carry = 1

Als resultaat < 256 dan Carry = 0 en ook in A het resultaat.

REG : AF, DE, HL

C5	GETAL	PUSH BC	
48		LD C,B	
E5		PUSH HL	Bewaar schermlocatie
CD tt ss		CALL gnum	Neem getal in
CD yy xx		CALL CURSOR	Wis cursor
E1		POP HL	Haal beginplaats
11 00 00		LD DE,0000	Maak resultaat nul
78		LD A,B	Maximum aantal
91		SUB C	-de rest = aantal
47		LD B,A	ingetypte cijfers
3E 00		LD A,00	Geen invoer ? dan
28 18		JR Z exit	A = 0 en klaar.

----- Conversie ASCII - Binair -----

7E		LD A,(HL)	Lees ASCII van het scherm
D6 30		SUB 30	ASCII naar binair
4F		LD C,A	
E5		PUSH HL	Schermlocatie op stack bewaren
EB		EX DE,HL	
29		ADD HL,HL	2 x
5D		LD E,L	De waarde met 2 x
54		LD D,H	bewaren in DE
29		ADD HL,HL	Wordt nu dus 4 x
29		ADD HL,HL	Wordt nu dus 8 x
19		ADD HL,DE	Wordt nu dus 10 x
59		LD E,C	
16 00		LD D,00	
19		ADD HL,DE	
EB		EX DE,HL	
E1		POP HL	Schermlocatie terug
23		INC HL	Volgend cijfer
10 EB		DJNZ convrt	
AF		XOR A	
92		SUB D	
7B		LD A,E	

: --- Einde conversie ---

C1	exit	POP BC	
C9		RET	

---- Get routine ----

Hoofdingang bij Label gnum.

79	back	LD A,C	Al op de voorste
B8		CP B	plaats ?
28 26		JR Z fout	zoja, dan fout
AF		XOR A	
77		LD (HL),A	Wis de cursor
2B		DEC HL	Plaats terug en
77		LD (HL),A	wis schermlocatie
0C		INC C	Teller + 1

(vervolg op M.207)

(Deze tekst is ook gepubliceerd in TRON 5, oktober 1985)

CD 26 00	gnum	CALL 0026	Toetsinvoer
FE 34		CP 34	Enter-toets ?
CB		RET Z	Zoja, dan klaar
FE 2C		CP 2C	Backspace ?
28 EA		JR Z back	Zoja, dan back
OC		INC C	Teller nul dan
OD		DEC C	is het maximale
28 10		JR Z fout	aantal cijfers bereikt : andere toets vraag.
CD zz vv		CALL GTAS	Omzetten naar ASCII
FE 30		CP 30	Check of cijfer
38 09		JR C Fout	>= 0 en
FE 3A		CP 3A	>= 9 is.
30 05		JR NC fout	
77		LD (HL),A	Print op scherm
23		INC HL	1 plaats verder
OD		DEC C	Teller -1
18 DF		JR gnum	en de volgende
CD 32 00	fout	CALL 0032	Piep-routine
18 DD		JR input	

&H xxxy is het adres waar de routine CURSOR in het programma staat
 sstt is het adres van gnum
 vvzz is het adres van GTAS

In de Get-routine wordt het C-register gebruikt voor het bijhouden van een tellertje van het aantal cijfers dat is ingevoerd.
 Bij het begin van de invoer is C = B = maximum aantal.

(Deze tekst is ook gepubliceerd in TRON 5, oktober 1985)

NOOT van VRIJE PUBLIKATIES.

Om met een machinetaalprogramma in BASIC te kunnen werken, kan met het volgende programma de machinetaal in het geheugen worden gezet. Als voorbeeld nemen we de INVERS-routine (zie M.202).

```
10 DATA 3E,18,06,28,21,57,57,C5,E5,CB,FE,2B,10
20 DATA FB,E1,01,50,00,A7,ED,42,C1,3D,20,EE,C9
30 FOR I = 0 TO 25 : READ A$
40 POKE &H 6150 + I, VAL ("&H" + A$ )
50 NEXT I
60 DEFUSR 1 = &H 6150
```

De machinetaal komt vanaf plaats &H 6150 in het geheugen te staan.
 Dit zijn een aantal vrije plaatsen in de Administratieruimte van de Monitor.

Door intypen van PRINT USR 1(0) of A = USR 1(0), in de direkte stand (of verwerkt in een Basicprogramma), wordt het machinetaalprogramma uitgevoerd. In bovenstaand geval wordt het gehele scherm geinverteerd. Worden op regel 10 de data ,CB,FE, vervangen door ,36,00, dan wordt het gehele scherm gewist.

Enkele vrije ruimten te gebruiken voor de opslag van o.a. machinetaal zijn :
 &H 6150-61FF (176x), &H 6360-63B3 (84x), &H 6070-608F (32x),
 &H 6113-612F (29x) , &H 6057-605B (5x) , &H 60F7-60FB (5x).

Hans van der Veer - Luytelaer 27 - 5632 BE Eindhoven - Tel. 040 / 417389
 27-01-87 - VRIJE PUBLIKATIES - Broederhof 11 - 5504 JC Veldhoven