M. B. IMMERZEEL

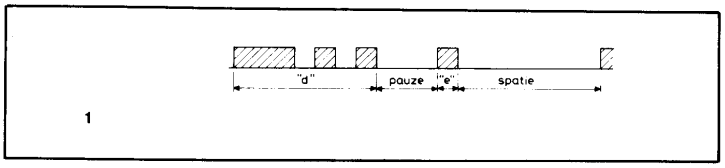
Morse-decodering met de KiM

Het is misschien wel leuk te kunnen zeggen de eigenaar te zijn van een microcomputer (als het zo doorgaat als nu dan is dat over een aantal jaren net zo gewoon als nu het bezit van een zakrekenmachine), nog leuker is het om ook programma's te bezitten en met het apparaat een aantal nuttige en vooral aangename uurtjes te kunnen doorbrengen. Het hiernavolgende programma dat voor de KIM is geschreven kan daartoe misschien wel bijdragen. Het schept de mogelijkheid om ontvangen morsesignalen als 'leesbaar schrift' zichtbaar te maken op het display van de KIM.
Hoewel met een zevensegmentdisplay lang niet alle lees- en lettertekens kunnen worden gevormd en dus voor deze tekens bepaalde symbolen moesten worden gevonden, is het na enig oefenen heel goed mogelijk de in morsecode verzonden berichten van het display te kunnen aflezen.
Niet alleen voor het weergeven van ontvangen berichten kan het programma worden gebruikt, maar ook bij het oefenen in seinen en bij het controleren van het seinschrift. Met hetgene wat door het display wordt weergegeven kan men vaststellen welke fouten er bij het seinen worden gemaakt.

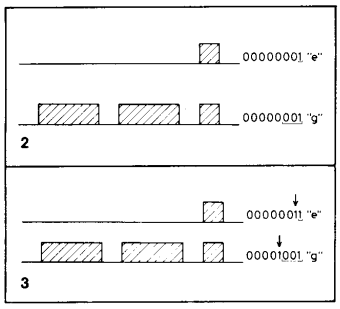
Achtergronden

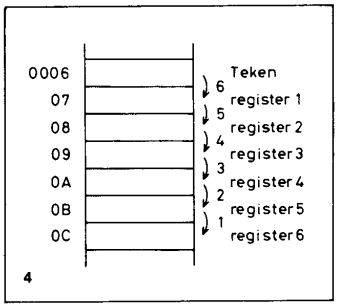
Voor het overbrengen van geschreven teksten langs mechanisch-elektrische weg (telex) of via moderne verbindingswegen (teletext of viewdata bijv.) is het noodzakelijk een codesysteem toe te passen waarbij de schrijf- en leestekens elk door een apart codeteken worden weergegeven. De bekendste van deze codesystemen zijn wel de ASCII-code en de vijf-eenhedencode zoals die bij de telex wordt toegepast (Murray-code).
Het voordeel van deze systemen is dat alle codetekens even lang zijn en elk codeteken in een gelijk aantal, even lange 'eenheden' worden verdeeld (bij telex vijf)' die ook wel 'elementen' worden genoemd. Elk element kan dan 'stroomvoerend' of 'stroomloos' zijn, vergelijkbaar met de '1' en de '0' van het binaire systeem.
De specifieke kenmerken van dit systeem, nl. dat elk codeteken steeds uit een vast aantal elementen bestaat van gelijke lengte en die slechts '0' of '1' kunnen zijn, maken het voor de machine betrekkelijk gemakkelijk om uit de aangeboden codetekens de juiste schrifttekens te herkennen.
Geheel anders ligt het bij het morsecodesysteem. Mogelijk kan worden gesteld dat dit systeem eenvoudiger door de mens met zijn gehoor te analyseren is dan de hiervóór genoemde codes, voor de machine is het echter moeilijk te verteren dat de codetekens onregelmatig zijn voor wat betreft het aantal elementen (van één punt voor de 'e' tot zes punten voor het vergissingsteken) en voor wat betreft de lengte van de elementen. De strepen in deze codetekens zijn immers drie keer zo lang als de punten.
Het zijn niet alleen de punten en de strepen die het teken bepalen. Tussen de elementen van het teken komt een stilte voor die we hier verder 'rust' zullen noemen en waarvan de lengte gelijk aan die van een punt dient te zijn. Tussen de codetekens dient een langere pauze te worden aangehouden. Deze pauze moet even lang duren als een streep.
Een codeteken voor spatie wordt in het algemeen niet verzonden. Hiervoor wordt gewoon een langere pauze gebruikt die tenminste een lengte van vijf punten moet hebben (voorgeschreven is nu zeven punten, vroeger paste men vijf punten toe).
Samengevat ziet het woord 'de' in morsetekens eruit zoals in afb. 1. De 'd' is een streep met twee punten. De rusten zijn één punt lang en de streep drie punten. De 'e' is slechts één punt en wordt met een pauze met een lengte van drie punten gescheiden van de 'd'. Het volgende woord zal slechts beginnen na een pauze met een tijdsduur van zeven punten.
Een ander probleem dat om een oplossing vraagt is de seinsnelheid die voor voor iedere telegrafist weer anders ligt. Het is dan ook noodzakelijk om eerst de lengte van een punt te bepalen. Dit is slechts mogelijk nadat zowel een punt als een streep is ontvangen. Dit programma zal daarom het eerste schriftteken (of in enkele gevallen de eerste paar schrifttekens) niet of anders verminkt weergeven. Bezwaarlijk kan dat niet zijn. Elke uitzending wordt nl. voorafgegaan door een oproep die enkele keren wordt herhaald.

Als een seinschrift niet door een machine is opgewekt zal het in bijna alle gevallen er niet zo uitzien als is voorgeschreven. We zullen er rekening mee moeten houden dat een streep niet altijd drie punten lang is. Bij dit programma is de lengte van twee punten als grens genomen, d.w.z. dat elk element dat langer is dan twee punten als een streep wordt herkend. Dat betekent dan wel dat een seiner niet meer kan worden 'genomen' als hij punten seint die meer dan twee keer zo lang zijn als eerder geseinde punten.
Verder dient het programma rekening te houden met onregelmatigheden in de seinsnelheid. Het verhogen van de seinsnelheid geeft hier geen problemen. Het verlagen van de seinsnelheid



1





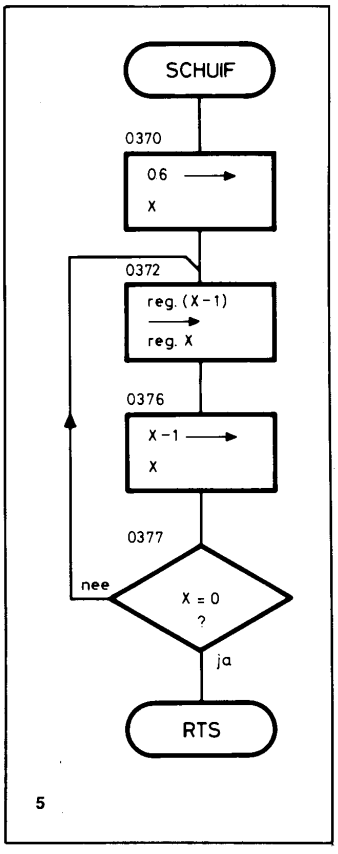
moet in verband met het voorgaande echter zo zijn dat de lengte van de eerstvolgende punt in elk geval kleiner moet zijn dan twee maal de lengte van de voorgaande.

Om deze veranderingen van de seinsnelheid te kunnen blijven volgen is het noodzakelijk om steeds de lengte van de laatst ontvangen punt te blijven onthouden. Het kan echter voorkomen dat een ongecontroleerde sleutelbeweging van een minder ervaren seiner een zeer kort contact kan veroorzaken hetgeen dan wordt ervaren als een zeer korte punt. Duurt deze 'punt' korter dan de helft van de daarop volgende punten dan zullen alle volgende code-elementen voor strepen worden aangezien zodat de werking van het programma dan geheel in de war geraakt.
Deze storing kan worden voorkomen door de tijdsduur van elke ontvangen punt te middelen met die van de voorgaande. Op deze manier worden de uitschieters 'gladgestreken' en hebben onregelmatigheden minder invloed op de juiste werking.
Gesteld is reeds dat de seinsnelheid onbeperkt kan worden verhoogd, maar niet onbeperkt kan worden verlaagd. Des ondanks kan een sprongsgewijze snelheidsverandering redelijk worden opgevangen. Rekening houdend met het hiervoor genoemde 'middelen' kan dan ook een stap van bijv. 23 naar 16 woorden per minuut zonder bezwaar worden toegestaan. Regelmatige snelheidsverminderingen kunnen dan ook geen moeilijkheden opleveren.

Het display

Dit programma is gemaakt om de morsetekens als leestekens weer te geven op het KIM-display. De moeilijkheid hierbij is echter dat het slechts een zeven-segmentdisplay is en dus bepaalde letters en leestekens niet kan weergeven. Voor deze letters en leestekens (k;m;n;v;w;x;?;komma;punt en punt-komma) zijn daarom een aantal symbolen verzonnen die zoveel mogelijk op de oorspronkelijke letters en tekens gelijken. Gebleken is dat na een betrekkelijk korte oefentijd te teksten, ondanks de vreemde symbolen, ook bij hoge seinsnelheden soepel zijn te lezen. Het display geeft de teksten weer zoals dat bij een lichtkrant plaatsvindt. Een verschil is echter dat de tekst steeds één plaats verspringt.
De tekens verschijnen na elkaar op het meest rechtse display en verschuiven na ontvangst en vertaling van het volgende morseteken steeds één plaats op naar links. Een spatie wordt weergegeven door het betreffende display uit te laten. In lijst 1 zijn o.a. de gebruikte leestekens met de daarbij behorende morsetekens en displaysymbolen gegeven.

Voor het vertalen van de morsetekens wordt een punt gelezen als '1' en een streep als '0'. Ook voor de KIM is het nodig om over te gaan op codetekens die alle een gelijk aantal elementen bevatten. Hier is het aantal elementen: 8, nl. gelijk aan het aantal bits per geheugenplaats. Doordat alle *lege* geheugenplaatsen van een '0' zijn voorzien zouden, indien de morsetekens zonder meer in enen en nullen werden vertaald, fouten ontstaan. Er zou dan bijv. geen onderscheid meer zijn tussen een 'e' en een 'g' omdat de strepen ('0) en de lege geheugenplaatsen (ook '0') niet meer te onderkennen zijn. In afb. 2 is weergegeven hoe een 'e' en een 'g' in de KIM zouden worden gelezen. Door de tekens vooraf te markeren met een '1' wordt deze fout voorkomen. In afb. 3 is weergegeven hoe de 'e' en de 'g' door de KIM worden vertaald. De markeringen zijn met een pijltje gemerkt.
In lijst 1 zijn onder de kolom 'TEKEN' de codetekens weergegeven waarin de KIM het morseteken vertaalt. Elk morseteken wordt dus in een binair getal vertaald waarvan de kolom 'teken Hex' de hexadecimale waarde weergeeft.
Op het adres 0006 vinden we de geheugenplaats 'TEKEN'. Hierin worden na elkaar de elementen van het codeteken geschoven. Als het teken compleet is wordt het in het eerste van de zes registers overgebracht die worden gebruikt bij het monitorprogramma. Hiervoor zijn de geheugenplaatsen 0007 t/m 000C gereserveerd (afb. 4). De subroutine 'SCHUIF' wordt dan in werking gesteld waardoor, te beginnen bij register 5, de inhouden één plaats worden opgeschoven en zodoende het lichtkranteffect wordt bereikt. De geheugenplaats 'TEKEN' is nu weer beschikbaar voor het volgende codeteken (afb. 5).
Het was niet mogelijk van de subroutine 'SCANDS' van het KIM-monitorprogramma gebruik te maken. De eerste reden hiervoor is dat het programma ook wordt gebruikt bij het meten van de lengte van de elementen en een andere vertraging nodig is dan bij de subroutine CONVD is toegepast.
De tweede reden is dat nu veel meer dan zestien karakters moeten worden weergegeven waardoor per codeteken – en ook voor de displaycode – meer dan vier bits nodig zijn.
Het stroomdiagram voor de subroutine



Lijst 1

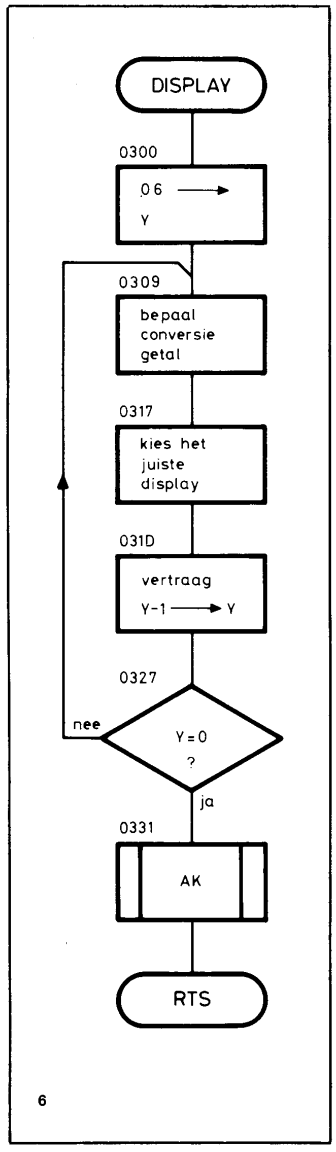
Teken	Morse-code	Inh. 'teken' binair	Inh. 'teken' hex.	Display	Display-code hex.
A	•▬	00000110	06	A	F7
B	▬•••	00010111	17	b	FC
C	▬•▬•	000101001	15	C	B9
D	▬••	00001011	0B	d	DE
E	•	00000011	03	E	F9
F	••▬•	00011101	1D	F	F1
G	▬▬•	00001001	09	g	EF
H	••••	00011111	1F	h	F4
I	••	00000111	07	ı	90
J	•▬▬▬	00011000	18	J	8E
K	▬•▬	00001010	0A	K	F0
L	•▬••	00011011	1B	L	B8
M	▬▬	00000100	04	M	B7
N	▬•	00000101	05	n	D4
O	▬▬▬	00001000	08	o	DC
P	•▬▬•	00011001	19	P	F3
Q	▬▬•▬	00010100	12	q	E7
R	•▬•	00001101	0D	r	D0
S	•••	00001111	0F	S	ED
T	▬	00000010	02	t	F8
U	••▬	00001110	0E	u	9C
V	•••▬	00011110	1E	U	BE
W	•▬▬	00001100	0C	H	FE
X	▬••▬	00010110	16	H	F6
Y	▬•▬▬	00010010	14	Y	EE
Z	▬▬••	00010011	13	Z	DB
1	•▬▬▬▬	00110000	30	1	86
2	••▬▬▬	00111000	38	2	DB
3	•••▬▬	00111100	3C	3	CF
4	••••▬	00111110	3E	4	E6
5	•••••	00111111	3F	5	ED
6	▬••••	00101111	2F	6	FD
7	▬▬•••	00100111	27	7	87
8	▬▬▬••	00100011	23	8	FF
9	▬▬▬▬•	00100001	21	9	EF
0	▬▬▬▬▬	00100000	20	0	BF
?	••▬▬▬•	01110011	73 (33)	?	A7
,	▬▬••▬▬	01001100	4C (2B)	,	8C
/	▬••▬•	00101101	2D	/	D2
.	•▬•▬•▬	01101010	6A (1A)	.	88
:	▬▬▬•••	01000111	47 (26)	:	89
—	▬••••▬	01011110	5E (3D)	-	C0
=	▬•••▬	00101110	2E	=	c—
spatie	•▬••▬	00110110	36		80
begin	▬•▬▬•	00101010	2A	r	D0
eind	•▬▬••	00110101	35	٦	C4
sluiten	•••••▬	01111010	7A (3A)		80
apostroph	•▬▬▬▬•	01100001	61 (11)	'	82
(	▬•▬▬•▬	01010010	52 (31)	]	8F
wacht	•▬•••	00110111	37	⌛	C9
;	▬•▬•▬•	01010101	55 (34)	;	8D

Lijst 2

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			T	E	M	N	A	I	O	G	K	D	W	R	U	S
1		(A_p)	Q	Z	Y	C	X	B	J	P	(.)	L		F	V	H
2	0	9		8			(:)	7			Be	(,)		/	=	6
3	1	(()		(?)	(;)	E	S_p	W_a	2		(S_t)		3	(−)	4	5
4							:						,			
5			(			;									−	
6		A_a									.					
7				?							S_t					

'DISPLAY' is in afb. 6 geschetst en komt in grote lijnen overeen met dat van 'SCANDS'.

De kolom 'inhoud teken Hex' van lijst 1 geeft als laagste waarde Hex 02 voor de 'T' en heeft Hex 7A voor 'sluiten' als hoogste waarde (zie hiervoor ook lijst 2). Dit zou betekenen dat voor de tabel voor de displaycode bijna 128 geheugenplaatsen nodig zijn met meer dan 64 open plaatsen (er zijn 51 verschillende codetekens). De omvang van de tabel kan worden beperkt door bepaalde tekens te 'verplaatsen'. Uit lijst 2 volgt dat hiervoor de tekens op rij 7 moeten worden verminderd met Hex 40, die op rij 6 met Hex 50 en op rij 5 en 4 met Hex 21. In lijst 1 zijn de nieuwe waarden tussen haakjes geplaatst. De displaycodetabel staat in de geheugenplaatsen 02C0-02FF.

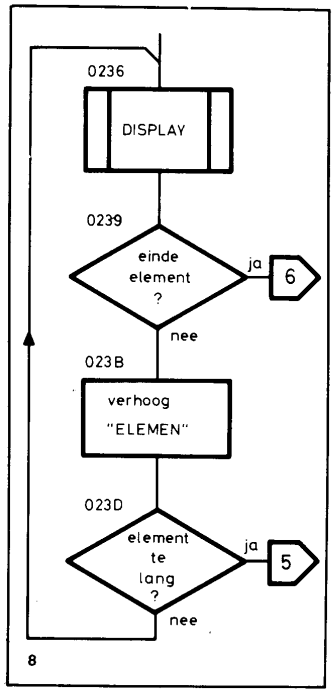


6

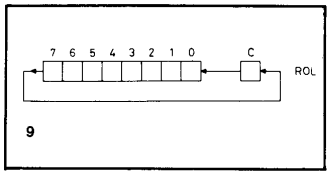
Het hoofdprogramma

Het stroomdiagram van het hoofdprogramma vinden we in afb. 7. Het startadres is 0200 en de seinsleutel komt parallel aan één van de contacten van het toetsenbord (bijv. A17 en A18 van de applicatiesteker). Na het starten van het programma door op de toets 'go' te drukken worden eerst de in- en uitgangsbuffers geïnitieerd en bepaalde hulpregisters gereset. De displayregisters worden geladen met het codeteken voor '=' waarna het programma blijft wachten tot de toets 'go' wordt losgelaten. Na het loslaten wacht het programma op het eerste code-element. Bij deze laatste twee programmastappen wordt de displaysubroutine gebruikt zodat het display oplicht en'======' laat zien. Direct na het indrukken van de seinsleutel, dus bij de start van een element, dooft het rechtse display omdat 'teken' wordt geladen met het codeteken voor 'spatie' (3A) dat middels subroutine 'schuif' in register 1 komt. De programmastap 'reset teken' laadt 'TEKEN' met Hex 01 voor het markeren.

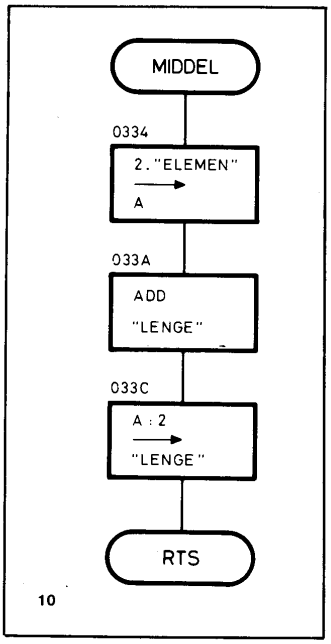
Voor het meten van de lengte van het element wordt eveneens gebruik gemaakt van de subroutine 'DISPLAY', die een aantal malen wordt doorlopen en tijdens elke cyclus de teller 'ELEMEN' verhocgt (afb. 8). Het displayprogramma moet daarvoor een bepaalde lengte hebben en de geheugenplaats 031E is voor de juiste vertraging daarom geladen met Hex FF.
Bij hoge seinsnelheden (30 woorden per minuut en sneller) kan het beter zijn dit adres te laden met Hex 7F. De vertraging wordt daardoor kleiner zodat in een bepaalde tijd een groter aantal metingen plaatsvindt.
Wordt de seinsleutel te lang achtereen ingedrukt gehouden dan volgt er een foutmelding (het achtste bit van 'elemen' wordt 1).
Na het eind van het element, aan het begin van de hierop volgende rust dus, wordt nagegaan of het element een punt of een streep was. Dit is na ontvangst van het eerste element uiteraard niet mogelijk. Pas na het ontvangen van twee elementen van verschillende lengte kan deze meting plaats vinden. De subroutine 'VERGELIJK zorgt er voor dat de lengte van een punt wordt berekend. De geheugenplaats 'LENGE' (0000) wordt dan gevuld met twee keer de lengte van een punt. De inhoud van 'ELEMEN' wordt hiermee vergeleken. Is het element langer (streep) dan volgt CLC, anders SEC. De programmastap 'ROL teken' schuift nu respectievelijk een 0 of een 1 in het eerste bit van 'TEKEN', terwijl alle andere bits naar links opschuiven (afb. 9). Verder zal, indien het element een punt blijkt te zijn, subroutine 'MIDDEL' de gemiddelde waarde berekenen van de nieuwe lengte van de punt met de voorgaande (afb. 10). Het grootste aantal elementen waaruit een morseteken kan bestaan is zes. Voegen we daar het markeringsteken bij dan zijn dus maximaal zeven bits nodig van de acht bits van geheugenplaats 'teken'. Komt het markeringsteken toch in de plaats van het achtste bit dan is er dus sprake van een fout en volgt er een foutmelding. Deze fout zal voorkomen bij telegrafisten die niet genoeg ruimte houden tussen de morsetekens en zo alles aan elkaar 'plakken'. De lengte van de rust wordt op overeenkomstige wijze bepaald als de lengte van het element. Het stroomdiagram is dan ook gelijk aan dat in afb. 8. Voor element en 'ELEMEN' moet nu echter 'RUST' worden gelezen. Als teller dient nu geheugenplaats 'RUST' op adres 0005.
In theorie is de lengte van de rust gelijk aan die van de punt. Beter is het echter daarmee niet te rekenen en steeds de lengte van een rust te middelen met de voorgaande, zoals dat ook met de punt gebeurt. Bij dit programma wordt deze methode dan ook toegepast, zij het dat bij het eerste teken, als de lengte van de rust nog niet bekend is, de lengte van de punt als norm wordt gesteld. In de praktijk blijkt dit goed te voldoen en garandeert dat binnen de kortst mogelijke tijd het eerste juiste karakter op het display verschijnt. De geheugenplaats 'LENGE' op adres 0003 wordt dan gevuld met twee maal de lengte van een rust. De inhoud van 'RUST' wordt hiermee vergeleken en het zal dan blijken of het werkelijk een rust is dan wel een pauze.
In het geval van een rust (einde teken is: neen) wordt na het resetten van 'ELEMEN' en 'RUST' overgegaan tot het meten van het nieuwe element dat nu is begonnen.
In het geval van een pauze is het codeteken compleet (einde teken). Evenals bij het element kan deze pauze erg lang worden aangehouden waardoor de teller 'RUST' gaat vollopen. Dit is echter geenszins als een fout aan te merken en in dit geval geeft de KIM automatisch een spatie.
De volgende programmastap bepaald of de telegrafist een foutmelding heeft



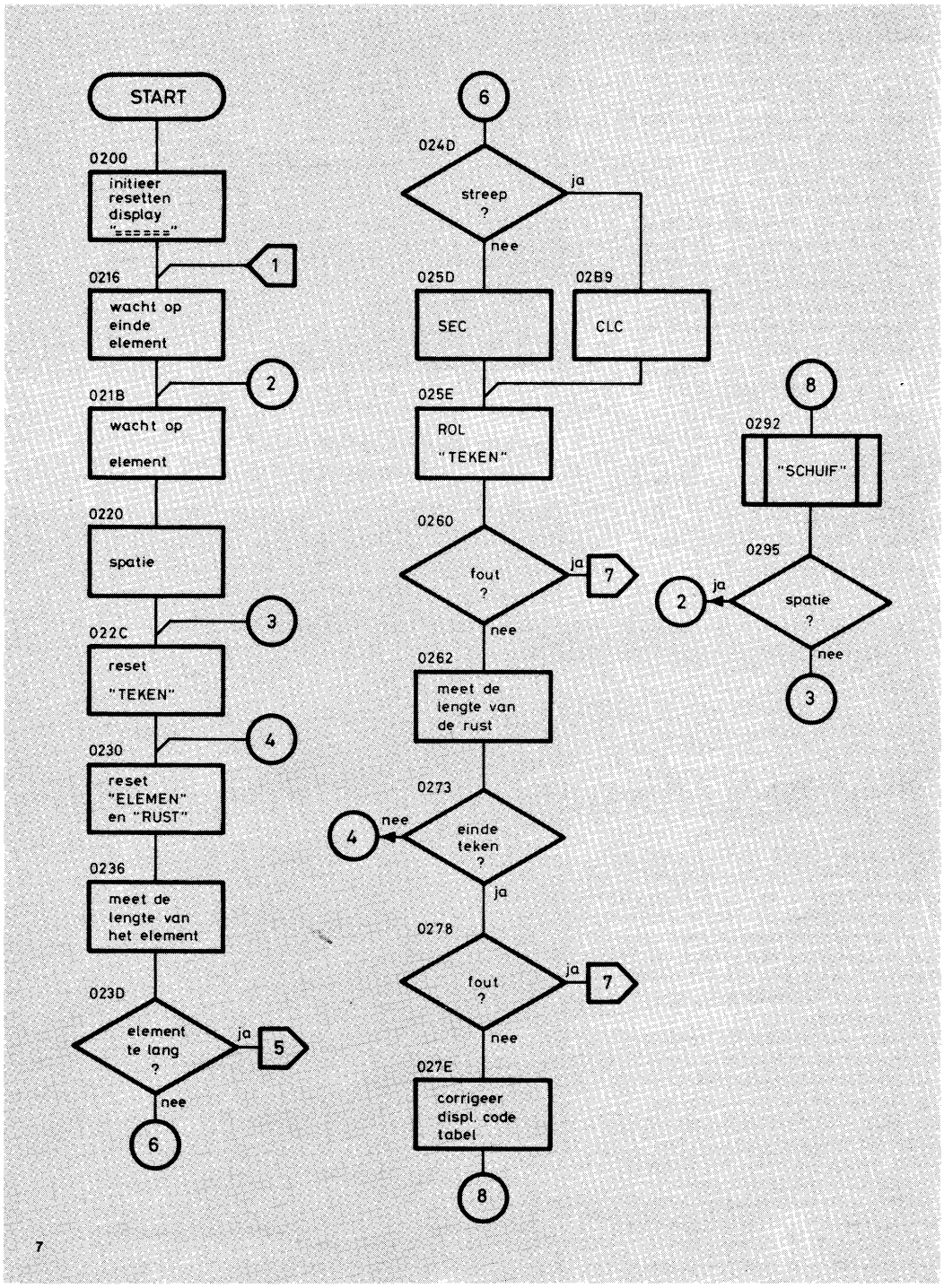
8



9



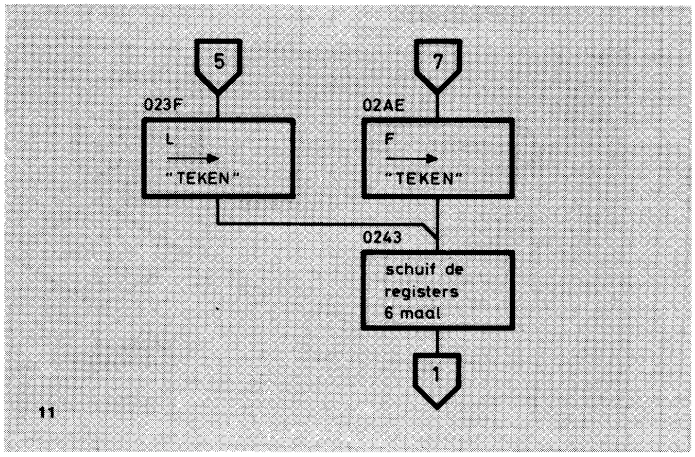
10


START
0200
initieer
resetten
display
"======"
0216
wacht op
einde
element
021B
wacht op
element
0220
spatie
022C
reset
"TEKEN"
0230
reset
"ELEMEN"
en "RUST"
0236
meet de
lengte van
het element
023D
element
te lang
?
ja
nee
1
2
3
4
5
6
024D
streep
?
ja
nee
025D
SEC
02B9
CLC
025E
ROL
"TEKEN"
0260
fout
?
ja
nee
7
0262
meet de
lengte van
de rust
0273
einde
teken
?
nee
ja
4
0278
fout
?
ja
nee
7
027E
corrigeer
displ. code
tabel
8
0292
"SCHUIF"
0295
spatie
?
ja
nee
2
3
8
7

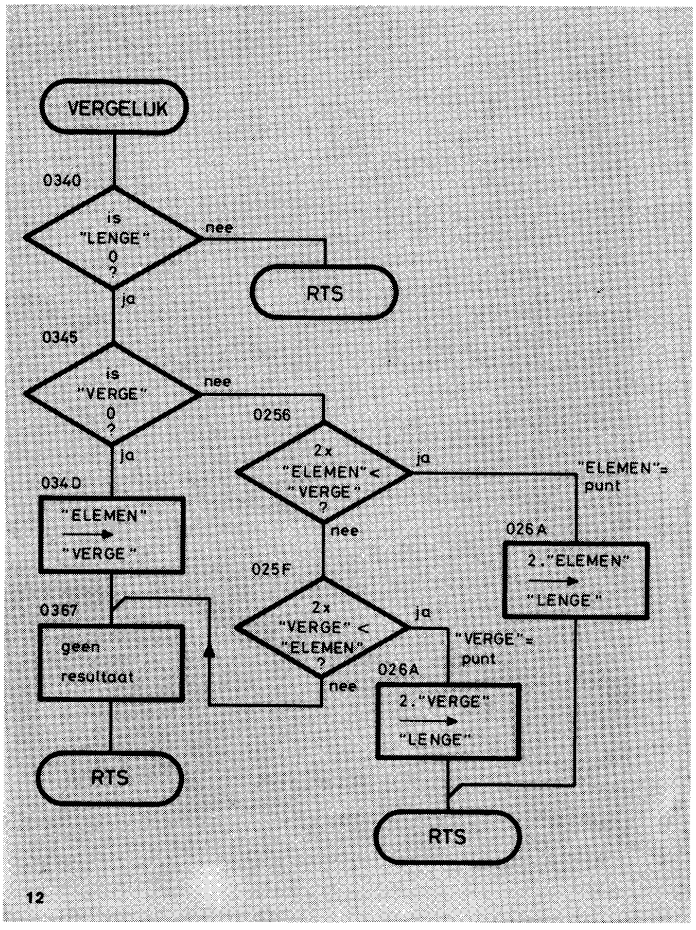
gedaan (zes punten). Zo ja dan geeft het display 'FFFFFF'. In het andere geval wordt nagegaan of de inhoud van 'TEKEN' moet worden gewijzigd om de omvang van de displaycodetabel te beperken (lijst 2). waarna de subroutine 'SCHUIF' de inhoud van de displayregisters opschuift en de inhoud van 'TEKEN' in het eerste displayregister brengt.
Wat de lengte van de pauze betreft zijn er nu nog twee mogelijkheden:

a. De pauze duurt ongeveer drie keer zo lang als de rust. In dit geval wordt na het resetten van 'TEKEN', 'ELEMEN' en 'RUST' overgegaan tot het meten van het eerste element van het volgende codeteken dat nu al is begonnen.
b. De pauze is langer (zeven maal zo lang als de rust of veel langer). Dit betekent het eind van een woord. Het programma blijft nu eventueel wachten op het eerste element van het volgende codeteken en zal dan, na eerst een spatie te hebben gegeven resetten als onder a en de lengte van dat element gaan meten.

Als het goed is duurt de pauze tussen twee letters van een woord ongeveer even lang als drie rusten. Deze pauze kan dan worden gevonden door te gaan vergelijken met vier maal de lengte van een rust. De meeste telegrafisten maken deze pauze echter te lang (uit angst te gaan 'plakken'?). Het gevolg zou kunnen zijn dat de KIM na elke letter een spatie geeft. Het is daarom beter te gaan vergelijken met zes maal rust. Hiertoe dient de subroutine 'LANG' op adres 038B, die kan worden aangeroepen door Hex 8B te plaatsen op adres 0298.
Voor hen die het seinen met de KIM willen beoefenen is het echter ook mogelijk te vergelijken met vier rusten ('kort') of vijf rusten ('midden') door respectievelijk Hex 7C of 80 op adres 0298 te plaatsen.
In het voorgaande was drie keer sprake van een foutmelding, één keer voor het te lang indrukken van de seinsleutel, dan voor een te groot aantal elementen in 'TEKEN' in daarna voor het seinen van het codeteken voor 'fout'.
In het eerste geval laat het display 'LLLLLL' zien en in de andere gevallen 'FFFFFF'. Hiertoe wordt het betreffende codeteken in 'TEKEN' geladen en door het zes maal doorlopen van de subroutine 'SCHUIF' in alle zes de displayregisters gebracht. Het programma gaat nu terug naar 'wacht op einde element' (afb. 11).



11



12

De subroutine 'vergelijk'

De subroutinevergelijk dient uitsluitend om bij het ontvangen van de eerste code-elementen te bepalen welk element een punt is. Is de lengte van een punt eenmaal bekend en 'LENGE' is geladen met een waarde die twee maal zo groot is dan doet deze subroutine geen dienst meer. Direct bij de start van de subroutine wordt dan ook vastgesteld of 'LENGE' geladen is ja of nee (afb. 12).

Na ontvangst van het eerste element zijn 'LENGE' en 'VERGE' nog leeg en wordt de inhoud van 'ELEMEN' in 'VERGE' (0001) geladen. Uiteraard is het nog niet bekend of dit een punt of een streep is. De inhoud van 'VERGE' wordt verder gebruikt om de lengte van het volgende element of volgende elementen hiermee te vergelijken. De programmastap 'geen resultaat' stelt vast dat de lengte van de punt nog niet bekend is waarna wordt teruggesprongen naar 'wacht op element'.

Na ontvangst van het tweede element is de situatie dus zo dat in 'VERGE' de lengte van het eerste element is vastgelegd en in 'ELEMEN' de lengte van de tweede. Er zijn nu vier mogelijkheden:

a. VERGE: punt, ELEMEN: punt { 2 × ELEMEN > VERGE 2 × VERGE > ELEMEN
b. VERGE: punt, ELEMEN: streep { 2 × ELEMEN > VERGE 2 × VERGE < ELEMEN
c. VERGE: streep, ELEMEN: punt { 2 × ELEMEN < VERGE
d. VERGE: streep, ELEMEN: streep { 2 × ELEMEN > VERGE 2 × VERGE > ELEMEN

Duidelijk is dat alleen in de situaties b en c de lengte van een punt wordt gevonden en 'LENGE' wordt gevuld met een waarde die overeenkomt met twee maal de lengte van een punt.

In de gevallen a en d wordt geen resultaat bereikt en wordt gewacht op een volgend element. Dit zal zich herhalen totdat zich de situatie b of c voordoet.

Nabespreking van het programma

Het gehele programma is in lijst 3 gegeven. De aandachtige beschouwer zal hierin een aantal schoonheidsfoutjes kunnen ontdekken. Het enige gevolg van deze foutjes is dat er een paar geheugenplaatsen meer gebruikt zijn dan strikt noodzakelijk was. Ik heb er dan ook geen voldoende reden in gezien om het gehele programma daarvoor te gaan overschrijven met de kans om hierbij fouten te maken die niet altijd even eenvoudig weer zijn op te sporen.

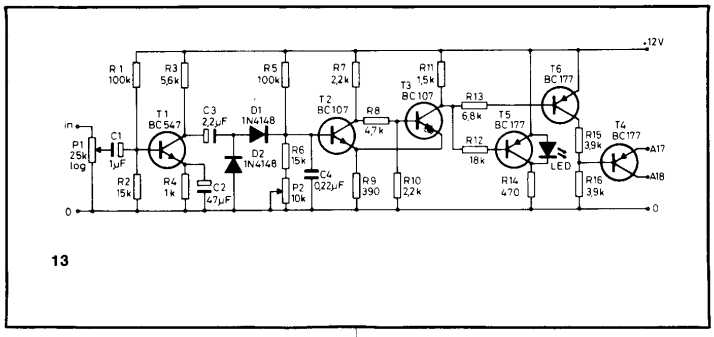
In het kort volgen hierbij nog enige belangrijke punten:

- Starten op adres 0200.
- Seinsleutel tussen de punten A17 en A18 van de applicatiesteker.
- Na ontvangst van twee verschillende code-elementen (een punt en een streep) worden de karakters op het display zichtbaar.
- Bij een zeer langzame seiner laat het display 'LLLLLL' zien.

- Bij een 'plakker' of een foutmelding 'FFFFFF'.
- Een toename van de seinsnelheid wordt zonder meer geaccepteerd.
- Een sprongsgewijze vermindering kleiner dan 1/3 van de seinsnelheid ook.
- Bij grote seinsnelheden Hex 7F op adres 031E.
- Bij seinoefeningen Hex 7C op adres 0298.
- Stoorimpulsen (bijv. bij ontvangst van radiosignalen) kunnen het programma verstoren. Alle elementen worden dan als strepen geregistreerd. In dat geval opnieuw het programma starten.

Hulpschakeling

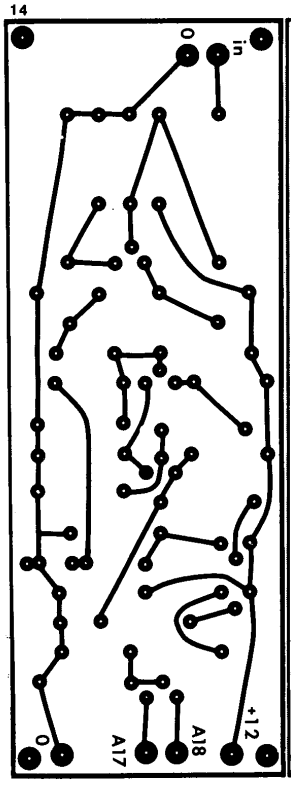
Een schakeling waarmee signalen uit een ontvanger kunnen worden omgezet tot impulsen voor de KIM is geschetst in afb. 13. De schakeling is gevoelig maar heeft een te lage ingangsimpedantie om rechtstreeks op de uitgang van een detector te kunnen worden aangesloten. Voor de gelijkspanning kan de 12 V worden gebruikt die wordt geleverd door het KIM voedingsapparaat. Met de potmeter P1 wordt de ingangsgevoeligheid gere-



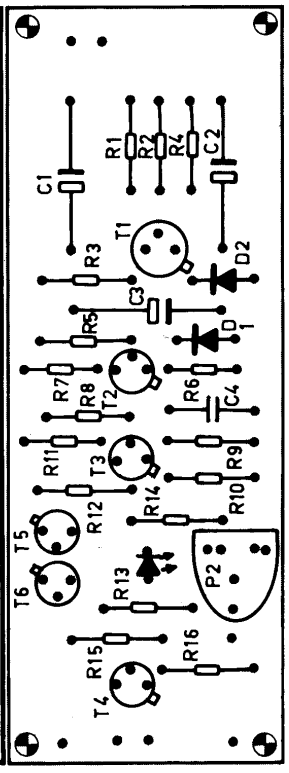
13

geld. Het signaal dat uit het laagfrequentgedeelte van de ontvanger wordt afgetakt (eventueel van de luidsprekeraansluiting) wordt door T1 versterkt en door de dioden D1 en D2 gelijkgericht. De transistoren T2 en T3 zijn als Smitttrigger geschakkeld. In rusttoestand is T3 geleidend en daardoor ook T5 en T6. De LED wordt door T5 kortgesloten en is dus uit. Door het geleiden van T6 is T4 gesperd (dit ervaart de KIM als '0'). De transistor T4 is op A17 en A18 van de applicatiesteker aangesloten (denk om de juiste polariteit, A17 is emittor).
Komt een voldoende groot signaal op de ingang dan gaat T2 geleiden. T6 spert waardoor T4 opengaat en A17 en A18 praktisch kortsluit. Dit wordt door de KIM ervaren als '1'. Ook T5 spert, zodat de Led die hiermee parallel is geschakeld, oplicht.
Voor het afregelen van P2 zorgt men dat er geen signaal aan de ingang aanwezig is. Potmeter P2 wordt nu zo ingesteld dat de LED net niet oplicht. Daarna wordt het signaal aan de ingang aangesloten en P1 opengedraaid totdat de LED de morsetekens door oplichten zichtbaar maakt.
In afb. 14 is de print getekend en in afb. 15 de componentenopstelling.

14



15



```
0200 20 8C 1E START  JSR  INIT1      ┐
0203 A9 00           LDA  #$00       │
0205 85 00           STA  LENGE      │ Initialiseren
0207 85 01           STA  VERGE      │
0209 85 03           STA  LENGR      │
020B 85 04           STA  VERGR      ┘
020D A2 06      RESET LDX  #$06      ┐
020F A9 2E           LDA  #$2E       │
0211 95 06      RESET1 STA TEKEN,X   │ DISPLAY '- - -'
0213 CA              DEX             │
0214 D0 FB           BNE  RESET1     ┘
0216 20 00 03 MORSEA JSR  DISPL      ┐ Wacht op einde
0219 D0 FB           BNE  MORSEA     ┘ element
021B 20 00 03 MORSE1 JSR  DISPL      ┐ Wacht op element
021E F0 FB           BEQ  MORSE1     ┘
0220 A9 3A           LDA  #$3A       ┐
0222 85 06           STA  TEKEN      │ Spatie
0224 20 70 03        JSR  SCHUIF     ┘
0227 EA EA EA        NOP's
022A EA EA           NOP's
022C A2 01      MORSE2 LDX #$01      ┐
022E 86 06           STX  TEKEN      │
0230 A2 00      MORSEB LDX #$00      │ Reset
0232 86 02           STX  ELEMEN     │
0234 86 05           STX  RUST       ┘
0236 20 00 03 MORSE3 JSR  DISPL      ┐
0239 F0 12           BEQ  VERG1      │ Meet de lengte
023B E6 02           INC  ELEMEN     │ van het element
023D 10 F7           BPL  MORSE3     ┘
023F A9 1B           LDA  #$1B       ┐ Element te lang
0241 85 06           STA  TEKEN      ┘
0243 A0 06      MORS11 LDY #$06      ┐
0245 20 70 03   MORS12 JSR SCHUIF    │ Schuif de
0248 88              DEY             │ registers
0249 D0 FA           BNE  MORS12     │ 6 maal
024B F0 C9           BPL  MORSEA     ┘
024D A2 00      VERG1 LDX  #$00      ┐
024F 20 40 03        JSR  VERGEL     │ Bepaal, streep of
0252 F0 C7           BEQ  MORSE1     │ punt
0254 C5 02           CMP  ELEMEN     │
0256 30 61           BMI  MORSE7     ┘ Streep in ELEMEN
0258 A2 02           LDX  #$02       ┐ Punt in ELEMEN
025A 20 34 03        JSR  MIDDEL     ┘
025D 38              SEC             ┐ Punt is 1
025E 26 06      MORSE8 ROL TEKEN     ┘
0260 30 50           BMI  MORS10     ┐
0262 20 00 03 MORSE4 JSR  DISPL      │
0265 D0 37           BNE  VERG2      │
0267 E6 05           INC  RUST       │ Meet de lengte
0269 10 F7           BPL  MORSE4     │ van de rust
026B A5 03           LDA  LENGR      │
026D D0 04           BNE  MORSE9     ┘
026F A5 00      MORSE5 LDA  LENGE    ┐ Inhoud LENGE in
0271 85 03           STA  LENGR      ┘ LENGR
0273 C5 05      MORSE9 CMP  RUST     ┐ Bepaal rust of pauze
0275 10 30           BPL  MORS3A     ┘ Rust
0277 38              SEC             ┐ Pauze (einde teken)
0278 A5 06           LDA  TEKEN      ┘
027A C9 70           CMP  #$70       ┐ Foutmelding?
027C 10 30           BPL  VERM5      ┘
027E C9 60           CMP  #$60       ┐
0280 10 08           BPL  VERM2      │
0282 C9 40           CMP  #$40       │
0284 30 0C           BMI  VERM4      │
0286 E9 21           SBC  #$21       │ Corrigeer de display-
0288 10 06           BPL  VERM3      │ code tabel
028A E9 50      VERM2 SBC  #$50      │
028C 10 02           BPL  VERM3      │
028E E9 3F      VERM1 SBC  #$3F      │
0290 85 06      VERM3 STA  TEKEN     ┘
0292 20 70 03   VERM4 JSR  SCHUIF    ┐ Schuif de registers
0295 A5 03           LDA  LENGR      ┘
0297 20 8B 03        JSR  LANG       ┐ Pauze of spatie
029A 30 20           BMI  JUMP       ┘
029C 10 8E           BPL  MORSE2     ┐ Spatie
029E A2 03      VERG2 LDX  #$03      ┘ of pauze
02A0 20 40 03        JSR  VERGEL     ┐
02A3 F0 CA           BEQ  MORSE5     │ Is lengr gevuld?
02A5 D0 CC           BNE  MORSE9     ┘
02A7 A2 05      MORS3A LDX #$05      ┐
02A9 20 34 03        JSR  MIDDEL     │ Rust
02AC D0 82           BNE  MORSEB     ┘
02AE C9 7F      VERM5 CMP  #$7F      ┐ Indien geen fout-
02B0 D0 DC           BNE  VERM1      ┘ melding, corrigeer
02B2 A9 1D      MORS10 LDA #$1D      ┐
02B4 85 06           STA  TEKEN      │ Foutmelding
02B6 4C 43 02        JMP  MORS11     ┘
02B9 18         MORSE7 CLC           ┐ Streep in elemen
02BA 90 A2           BCC  MORSE8     ┘
02BC 4C 1B 02   JUMP  JMP  MORSE1    ┐ Tussensprong voor 029A
02BF -- -- -- -- -- -- -- --         ┐
02C0 -- -- -- F8 TBL                 │
02C3 F9 B7 D4                        │
02C6 F7 90 DC                        │
02C9 EF F0 DE                        │
02CC FE D0 9C                        │
02CF ED -- -- 82                     │
02D2 E7 DB EE                        │
02D5 B9 F6 FC                        │
02D8 8E F3 88                        │
02DB B8 -- -- F1                     │ Displaycodetabel,
02DE BE F4 BF                        │ '- - -' wil zeggen: inhoud
02E1 EF -- -- FF                     │ is niet van belang
02E4 -- -- -- 89                     │
02E7 87 -- -- --                     │
02EA D0 8C -- --                     │
02ED D2 C8 FD                        │
02F0 86 8F -- --                     │
02F3 A7 8D C4                        │
02F6 80 C9 DB                        │
02F9 -- -- 80 --                     │
02FC CF C0 E6                        │
02FF ED                              ┘
0300 A9 7F      DISPL LDA  #$7F      ┐ PAD op uitgangen
0302 8D 41 17         STA  PADD.     ┘
0305 A2 09            LDX  #$09      ┐ Eerste display
0307 A0 06            LDY  #$06      ┘ 6 display's
0309 B9 06 00 DISPLA  LDA  TEKEN,Y   ┐ register naar accu
030C 84 FC    DISPL1  STY  TEMP      ┘ Save Y
030E A8               TAY            ┐ Haal de display-
030F B9 C0 02         LDA  TBL,Y     ┘ code uit de tabel
0312 A0 00            LDY  #$00      ┐ Display uit
0314 8C 40 17         STY  PAD       ┘
0317 8E 42 17         STX  PBD       ┐ Kies display
031A 8D 40 17         STA  PAD       ┘ Display aan
031D A0 FF            LDY  #$FF      ┐
031F 88       DISPLB  DEY            │ Laat het display
0320 D0 FD            BNE  DISPLB    ┘ een ogenblik aan
0322 E8               INX            ┐ Stel X in op het
0323 E8               INX            ┘ volgende display
0324 A4 FC            LDY  TEMP      ┐ Stel Y in op het
0326 88               DEY            ┘ volgende register
0327 D0 E0            BNE  DISPLA    ┐ Laatste display?
0329 8E 42 17         STX  PBD       ┘ Display uit
032C A9 00            LDA  #$00      ┐ PAD naar ingang
032E 8D 41 17         STA  PADD      ┘
0331 4C FE 1E         JMP  AK        ┐ Naar 'Any Key?'
0334 B5 00    MIDDEL  LDA  LENGE,X   ┐
0336 18               CLC            │ Vermenigvuldig met 2
0337 0A               ASL            ┘
0338 CA               DEX            ┐ Stel vorige waarde in
0339 CA               DEX            ┘
033A 75 00            ADC  LENGE,X   ┐ Tel de vorige waarde op
033C 4A               LSR            ┘ en deel door 2
033D 95 00            STA  LENGE,X   ┐ Zet het resultaat weg
033F 60               RTS            ┘ en terug
0340 B5 00    VERGEL  LDA  LENGE,X   ┐ Is LENGE of LENGR
0342 F0 01            BEQ  VERGE1    ┘ gelijk aan 0?
0344 60               RTS            ┐ nee
0345 E8       VERGE1  INX            ┐
0346 B5 00            LDA  LENGE,X   │ Is VERGE of VERGR
0348 F0 03            BEQ  VERGE2    ┘ gelijk aan 0?
034A 4C 56 03         JMP  VERGE3    ┐ Nee
034D E8       VERGE2  INX            ┐
034E B5 00            LDA  LENGE,X   │ Inhoud van ELEMEN
0350 CA               DEX            │ naar VERGE
0351 95 00            STA  LENGE,X   │
0353 4C 67 03         JMP  VERGE4    ┘
0356 E8       VERGE3  INX            ┐
0357 B5 00            LDA  LENGE,X   ┘ ELEMEN in accu
0359 0A               ASL            ┐ x2
035A CA               DEX            ┘
035B D5 00            CMP  LENGE,X   ┐ 2. ELEMEN-VERGE
035D 30 0B            BMI  VERGE5    ┘ Resultaat negatief?
035F B5 00            LDA  LENGE, X  ┐ VERGE in accu
0361 0A               ASL            ┘ x2
0362 E8               INX            ┐
0363 D5 00            CMP  LENGE,X   ┘ 2 VERGE-ELEMEN
0365 30 03            BMI  VERGE5    ┐ Resultaat negatief?
0367 A9 00    VERGE4  LDA  #$00      ┘ Geen resultaat
0369 60               RTS
036A CA       VERGE5  DEX
036B CA               DEX
036C 95 00            STA  LENGE, X
036E 60               RTS
036F -- --            --
0370 A2 06    SCHUIF  LDX  #$06      ┐ 6 registers te schuiven
0372 B5 05    SCHUI 1 LDA  TEKEN-1,X ┐ Verplaatst een
0374 95 06            STA  TEKEN,X   ┘ register
0376 CA               DEX            ┐ Laatste?
0377 D0 F9            BNE  SCHUI1    ┘
0379 60               RTS            ┐ Terug
037A -- -- --         --             ┘
037C 0A       KORT    ASL            ┐ 2 x LENGR (4 x rust)
037D C5 05            CMP  RUST      ┘ Vergel. 4 x rust met pauze
037F 60               RTS
0380 4A       MIDDEN  LSR            ┐ LENGR:2 (1 x rust)
0381 85 0D            STA  SAVE      ┘ 1 x rust in SAVE
0383 0A               ASL            ┐ 2 x rust in accu
0384 0A               ASL            ┘ 4 x rust in accu
0385 18               CLC            ┐
0386 65 0D            ADC  SAVE      ┘ 5 x rust in accu
0388 C5 05            CMP  RUST      ┐ Verg. 5 x rust met pauze
038A 60               RTS            ┘
038B 85 0D    LANG    STA  SAVE      ┐ 2 x rust in SAVE
038D 0A               ASL            ┘ 4 x rust in accu
038E 18               CLC            ┐
038F 65 0D            ADC  SAVE      ┘ 6 x rust in accu
0391 C5 05            CMP  RUST      ┐ Verg. 6 x rust met pauze
0393 60               RTS            ┘
```

Gebruikte geheugenplaatsen

0000	LENGE	0007	displayregister 1
0001	VERGE	0008	displayregister 2
0002	ELEMEN	0009	displayregister 3
0003	LENGR	000A	displayregister 4
0004	VERGR	000B	displayregister 5
0005	RUST	000C	displayregister 6
0006	TEKEN	000D	save
		00FC	temp