

MASTER

Een spraakherkenner voor het Busch Timac X-10 afstandsbedieningssysteem

Bloks, R.H.J.

Award date:
1989

[Link to publication](#)

Disclaimer

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

**FACULTEIT DER ELEKTROTECHNIEK
TECHNISCHE UNIVERSITEIT EINDHOVEN**

Vakgroep Medische Elektrotechniek

**Een spraakherkenner voor het
Busch Timac X-10
afstandsbedieningssysteem**

© door R.H.J. Blinks

Verslag van het afstudeerwerk,
uitgevoerd van mei 1986 tot mei 1987
in opdracht van prof. dr. Ir. J.E.W. Beneken
onder leiding van Ir. W.H. Leliveld,
en medebegeleiding door dhr. H.J.M. Ossevoort

**DE FACULTEIT DER ELEKTROTECHNIEK VAN DE TECHNISCHE UNIVERSITEIT
EINDHOVEN AANVAARDT GEEN VERANTWOORDELIJKHEID VOOR DE INHOUD
VAN STAGE- EN AFSTUDEERVERSLAGEN**

SAMENVATTING

In dit afstudeerverslag wordt de ontwikkeling beschreven van een relatief goedkope spraakherkenner, geschikt voor toepassing als hulpmiddel in het dagelijks leven van een motorisch gehandicapt persoon. Hierbij moet dan gedacht worden aan een apparaat dat in staat is om elektrische apparaten binnenshuis te bedienen, daarbij reagerend op gesproken taal.

Uit voorgaande onderzoeken is reeds gebleken, dat het in principe mogelijk en zinvol zou zijn om zo'n spraakherkenner te maken. Er zijn methoden onderzocht waarop spraakherkenning kan plaatsvinden en er is een goede methode uitgekozen. Aan de hand van een bestaande spraakherkenner (de LIS'NER1000) is die methode verder onderzocht.

Op basis van de resultaten hiervan is een spraakherkenner gerealiseerd, die gekoppeld is aan een bestaand binnenshuis afstandsbedieningssysteem, het Busch Timac X-10 systeem. De voor dit systeem benodigde hardware en een gedeelte van de software moest ontwikkeld worden, terwijl de rest van de software herschreven kon worden van de LIS'NER1000. Deze ontwikkelingen worden in dit verslag beschreven.

De spraakherkenner is gebaseerd op een 80C31 microprocessor en werkt met de SP1000, een spraak naar L.P.C. coëfficiënten omzetter. Er is voorzien in een terugkoppeling naar de gebruiker middels een LEDdisplay van 25 LED's, dat altijd de toestand van het gehele systeem weergeeft. Deze LED's zijn op dezelfde manier gegroepeerd als de toetsen op de bestaande netbesturingspost van het Busch Timac X-10 systeem, zodat mensen die reeds ervaring met dit systeem hebben ook heel snel de spraakherkenner kunnen bedienen. Voor de opslag van woorden is een E²PROM gebruikt, die genoeg ruimte biedt voor de opslag van circa 70 woorden (waarvan nu 23 in gebruik).

De software is gedeeltelijk herschreven van de LIS'NER1000 spraakherkenner (de herkenningsalgoritmes), en gedeeltelijk zelf geschreven (de gebruikersinterface en de aansturing van het afstandsbedieningssysteem).

De spraakherkenner is sprekerafhankelijk. Daarom is de mogelijkheid ingebouwd voor de gebruiker om de woorden (maximaal 23) te trainen. Bij het trainen wordt gebruik gemaakt van het 'kiezen en bevestigen'-principe om elk woord afzonderlijk te kunnen trainen en om te voorkomen, dat verkeerde woorden getraind worden. In de normale herkenningsmode kan men met tweewoord commando's alles besturen. Het eerste woord moet een kanaalkeuze zijn en het tweede woord een functiekeuze.

SUMMARY

This report covers the development of a relatively cheap speech recognizing system, which may be used as an aid in activities in daily living for severely disabled people. Our primary goal is a device that is capable of controlling the indoor electrical equipment, on spoken words.

From previous research projects it was found out that basically it would be possible and usefull to create such a speech recognizer. Some of the possible methods for speech recognition were investigated, a choice was made for the method with the best prospects and an already existing speech recognizing system based on that method, called the LIS'NER1000 was used for further investigation.

Based on the results of that investigation a new speech recognizer has been developed, which is connected to an existing indoor environmental control system, the Busch Timac X-10 system. The hardware and a part of the software needed for this system was developed from scratch and the remaining part of the software was rewritten from the LIS'NER1000 software. This report is about these developments.

The speech recognizer is based on a 80C31 microprocessor and the SP1000, a speech to L.P.C. data converter. Feedback to the user has been implemented by a LEDdisplay, containing 25 LED's which reflect the current state of the system. The LED's have been grouped together in the same way as the keys on the existing mains-controlset of the Busch Timac X-10 system. This enables people who already have some experience with this system, to learn how to use the speech recognizer very easily. Words are stored in an E²PROM, which is large enough to store almost 70 words, of which only 23 are currently in use.

The software has been partly rewritten from the LIS'NER1000 speech recognizer (the recognition algorithmes) and the rest (the userinterface and the software to control the remote control system) is completely new.

The speech recognition is person-dependent. For this reason the software allows the user to train the words (max. 23). While training words, the basic idea of choosing and then confirming the choice is used in order to make it possible that each word may be trained alone and to prevent accidental training of wrong words. In normal recognition mode, you may control everything with only two-word commands, in choosing first a channel, then the function it should perform.

INHOUDSOPGAVE

<u>Hoofdstuk</u>	<u>Titel</u>	<u>Blz.</u>
	Titelbladzijde	1
	Samenvatting	2
	Summary	3
	Inhoudsopgave	4
1	Inleiding	6
2	Een spraakherkenner als hulpmiddel voor A.D.L.	7
§ 2.1	Inleiding	7
§ 2.2	Het vooronderzoek	8
§ 2.3	Koppeling aan een afstandsbedieningssysteem	9
§ 2.4	Oprachtoomschrijving	10
3	Spraakherkenning met de SP1000	11
§ 3.1	Methoden voor spraakanalyse	11
§ 3.2	Linear Predictive Coding	12
§ 3.3	Het gebruik van de SP1000	16
4	Systeemopzet en bediening	17
§ 4.1	Het Busch Timac X-10 systeem	17
§ 4.2	De opzet van het nieuwe systeem	19
§ 4.3	Terugkoppeling naar de gebruiker	22
§ 4.4	Invoer van kanalen en functies	24
§ 4.5	Functiekeuzemogelijkheden	25
§ 4.6	Bedieningshandleiding spraakherkenner	28
	Gebruikershandleiding spraakherkenner	28
	De herkenningmode	30
	De trainingsmode	32
5	De hardware	36
§ 5.1	Het analoge ingangscircuit	36
§ 5.2	De interface tussen analoog en digitaal	41
§ 5.3	Het digitale deel van de spraakherkenner	42
6	De software	51
§ 6.1	De oorsprong van de software	51
§ 6.2	De taken van de software	52
§ 6.3	De indeling van het geheugen	54
§ 6.4	De database met referentiewoorden	56
§ 6.4.1	De indeling van woorden in groepen	57
§ 6.4.2	De indeling van een groep in templates	60
§ 6.5	De methode voor het bepalen van woordgrenzen	63
§ 6.6	De tijd- en amplitudenormering	65
§ 6.7	Het Dynamic Time Warping Algoritme	68
§ 6.8	De flowdiagrammen	72
	- Lees set K-parameters	72

Een spraakherkenner voor het Busch Timac X-10 afstandsbedieningssysteem

	- Initialiseer SP1000	72
	- Finite State Machine	72
	- Tijdnormering	73
	- Amplitudenormering	74
	- Lees en verwerk frame	74
	- Luister en herken woord	75
	- Template Matcher	76
	- Pas distance-buffer aan	77
	- Accepteer of verwerp woord	77
	- Dynamic Time Warping	78
	- Bepaling minimum afstand op roosterpunt (X,Y)	78
	- Bepaling afstand op (X,Y) via opgegeven pad	79
	- Bereken locale afstand op roosterpunt (X,Y)	79
	- Interrupt 0 service	79
	- Het laden van een bit	81
	- Interrupt 1 service	81
	- De timer1 interrupt service routine	82
	- Programma-start	83
	- Trainingsmode	83
	- Bied LED (woord) aan voor training	83
	- Neem woord op	84
	- Herkenningsmode	84
	- Verwerk een commando	84
7	Conclusies en aanbevelingen	87
§ 7.1	Bereikte resultaten	87
§ 7.2	Aanbevelingen voor mogelijke verbeteringen	88
8	Literatuurlijst	90

Bijlagen

1	Flowdiagrammen	92
2	Schema van de spraakherkenner	116
3	Onderdelenlijst	117
4	Plot van de printen (layout)	118
5	Frequentiearakteristiek van de analoge ingangstrap	121

HOOFDSTUK 1

INLEIDING

Binnen de vakgroep Medische Electrotechniek van de Technische Universiteit Eindhoven houdt men zich al enige tijd bezig met onder andere het ontwerpen van hulpapparatuur ten behoeve van gehandicapten. Hieruit is de Monoselektor ontstaan, een apparaat waarmee het mogelijk is allerlei functies binnenshuis door middel van slechts één schakelaar te bedienen.

Helaas is niet iedere gehandicapte nog in staat een schakelaar te bedienen. De schakelaar kan daarom diverse uitvoeringsvormen hebben, aangepast aan de handicap, maar zelfs dan blijft het vaak toch nog een moeilijke aangelegenheid om op die manier iets te bedienen, vooral ook omdat de schakelaar meerdere malen bediend moet worden om de gewenste functie uit te voeren.

Dit heeft in de vakgroep geleid tot de vraag of het niet mogelijk was een apparaat te ontwikkelen waarmee men de commando's niet met een schakelaar, maar via spraak kan invoeren. Een deel van de motorisch gehandicapten heeft nog wel het volle spraakvermogen, en voor hen zou een dergelijk apparaat veel eenvoudiger in het gebruik zijn.

Er is toen een onderzoek gestart naar de mogelijkheden van een spraakherkenner en dat heeft uiteindelijk geresulteerd in de realisatie van zo'n apparaat voor toepassing in een bekend en veel gebruikt binnenshuis afstandsbedieningssysteem. De ontwikkeling van de hardware en de software voor deze spraakherkenner wordt in dit verslag beschreven.

HOOFDSTUK 2

EEN SPRAAKHERKENNER ALS HULPMIDDEL VOOR A.D.L.

§2.1 Inleiding

Aan de Technische Universiteit Eindhoven wordt al jaren gewerkt aan de ontwikkeling van hulpapparatuur voor gehandicapten. Het gaat hier vooral om apparatuur voor zgn. A.D.L. toepassingen. A.D.L. staat voor Activities in Daily Living, ofwel activiteiten in het dagelijks leven. Er wordt getracht om zo goed en goedkoop mogelijk apparatuur te ontwikkelen. Dit heeft reeds eerder geleid tot bijvoorbeeld de ontwikkeling van de Monoselector.

Alle tot nu toe gemaakte apparaten hadden echter als nadeel dat de communicatie tussen mens en machine via schakelaars moest verlopen. Deze konden dan eventueel wel worden aangepast aan de handicap van de gebruiker, bijv. druksensors (blaaspijpje), hoofdbewegingsschakelaars o.i.d., maar men moest altijd schakelaars enkele malen bedienen om een gewenste functie te kunnen kiezen en uitvoeren. Bovendien vergden dit soort systemen een training van de gebruiker. De bedieningsmethode beperkt de bewegingsvrijheid van de gebruiker, die toch meestal niet erg groot is. Het streven is echter om de apparatuur zo te maken dat deze zo eenvoudig mogelijk te bedienen is. Hiervoor moeten we dan nagaan wat er voor communicatievormen mogelijk zijn tussen mens en machine, maar vooral welke communicatievorm voor de mens de gemakkelijkste is. Een natuurlijk communicatiemiddel van de mens is de spraak. Het zou dus logisch zijn om te proberen of er met spraakherkenning gewerkt kan worden. Veel gehandicapten, die niet in staat zijn schakelaars te bedienen of alleen op een zeer moeilijke manier hebben nog wel enig spraakvermogen. Voor deze groep is het zeker de moeite waard om te proberen een spraakherkenner te ontwikkelen.

Er zijn echter wel enige nadelen verbonden aan het gebruik van spraakherkenning. Spraakherkenning staat technisch gezien nog in de kinderschoenen. Men was tot enkele jaren geleden nog niet ver genoeg gevorderd om spraakherkenning eenvoudig toe te passen. In tegendeel, de benodigde apparatuur (bestaande spraakherkenners) voor goede herkenning waren erg complex en uitgebreid maar vooral erg duur. De kwaliteit van de

herkenners liet vaak te wensen over. Verder konden de bestaande herkenners geen externe apparaten aansturen, zoals lampen of gordijnsluiters.

De vakgroep E.M.E. is daarom enkele jaren geleden begonnen met een onderzoek naar de mogelijkheden van het gebruik van spraakherkenning voor A.D.L. toepassingen. Het einddoel was het ontwikkelen van een goedkope, kleine spraakherkenner die toch een redelijke herkenningsscore had en die geschikt was om A.D.L. functies te besturen.

In het kader van dit onderzoek is er samenwerking tussen de Technische Universiteit Eindhoven en het I.R.V. (Instituut voor RevalidatieVraagstukken) te Hoensbroek. Daarbij houdt het I.R.V. zich vooral bezig met onderzoek in het veld, terwijl de vakgroep E.M.E. de technische aspecten van het onderzoek onder zijn hoede neemt.

§2.2 Het vooronderzoek

Het onderzoek werd gestart door Hans Bosch (lit [1]) die het hele vooronderzoek voor zijn rekening nam. Hij heeft een literatuuronderzoek gedaan naar de principes van spraakherkenning, met het doel inzicht te krijgen in wat op dit terrein wel en niet mogelijk was met de huidige technologie. Hiervoor zijn een aantal methoden voor spraakherkenning onderzocht op geschiktheid en is ook nagegaan of er al apparatuur op dit gebied te koop was, en wat daar dan de mogelijkheden van waren. Zijn conclusies kwamen in het kort er op neer dat de bestaande spraakherkenners ófwel te duur waren, ófwel te weinig mogelijkheden hadden. Verder zouden er op het gebied van apparatuur en componenten weinig mogelijkheden zijn om een goedkope, maar goede spraakherkenner te kunnen realiseren. De best bruikbare chip is de SP1000, een I.C. van General Instrument (lit.[5]). De SP1000 is een goedkope chip die spraak als invoer heeft en LPC coëfficiënten als uitvoer. Bovendien is deze chip speciaal ontworpen om direkt op een microprocessor te worden aangesloten. De spraakherkenning zelf komt dan helemaal neer op de software, maar dit staat een zeer grote flexibiliteit toe op het gebied van vocabulaire-omvang, drempelwaarden, gebruikte boomstructuren in de herkenning van commando's die uit meer woorden bestaan, e.d. Een bijkomend voordeel was dat er al enkele spraakherkenners bestonden die op deze chip waren gebaseerd. Deze waren uitgevoerd als insteekkaarten voor de IBM en de APPLE 2 computers. Daar de vakgroep E.M.E. op dat moment over een APPLE 2-e beschikte heeft men de betreffende spraakherkenner met de SP1000 aangekocht. Het gaat hier om de LIS'NER1000, een stuk hardware met een grote hoeveelheid software erbij.

Met dit systeem als uitgangspunt is Hans Bosch toen begonnen met de opzet van een spraakherkenner systeem. Hij heeft zich vooral toegelegd op het ontwerp van de analoge ingangstrappen die nodig zijn bij de SP1000. Een aantal mogelijke uitvoeringsvormen zijn getest en met elkaar vergeleken. Verder heeft hij een experimenteersysteem gebouwd met de SP1000 dat voor verdere testdoeleinden gebruikt kon worden.

Het onderzoek werd toen overgenomen door Arjen Traas (lit [2]) die het LIS'NER1000 systeem heeft onderzocht. Hij heeft het hele bijgeleverde programma gedissassembleerd en van commentaar voorzien. Op die manier hebben we inzicht gekregen in de methoden die toegepast moeten worden voor bijvoorbeeld het detecteren van de grenzen van een woord, of de methode om spreesnelheidsvariaties op te vangen bij de herkenning. De door Arjen Traas gemaakte source-listing met commentaar is door mij als basis gebruikt bij het schrijven van de software.

§2.3 Koppeling aan een afstandsbedieningssysteem

Het derde deel van het onderzoek wordt in dit verslag beschreven, namelijk het realiseren van een spraakherkenner die werkt volgens dezelfde methode als die welke Arjen Traas heeft onderzocht. Omdat een alleenstaande spraakherkenner geen functies kan uitvoeren moest deze op één of andere manier worden gekoppeld aan een besturingssysteem. Er moest dus een keuze worden gemaakt uit de bestaande afstandsbedieningssystemen. Die keuze was niet moeilijk: het Busch Timac X-10 systeem. Eén van de redenen voor de keuze van dit systeem is de jarenlange ervaring hiermee in de vakgroep. De Monoselector is namelijk ook bedoeld voor aansturing van dit systeem. In de vakgroep zijn enkele Busch Timac X-10 systemen aanwezig, alsmede informatie over de opbouw en schema's van het systeem. Dit maakt het mogelijk later modificaties aan te brengen in dit systeem en zo de verbinding tussen spraakherkenner en afstandsbedieningssysteem te maken. Zo ontstaat de interessante mogelijkheid om de Monoselector te combineren met de spraakherkenner, waardoor een systeem ontstaat dat bediend kan worden via het toetsenbord van de netbesturingspost, via de ultrasone handzender (afstandsbediening), via de Monoselector en via spraak. Zo ontstaat één afstandsbedieningssysteem met zoveel bedieningsorganen dat er voor bijna elke motorische handicap wel een geschikte bij is.

§2.4 Opdrachtomschrijving

Nu de voorgeschiedenis en het einddoel bekend zijn kunnen we overgaan tot het omschrijven van de opdracht, die ik tijdens mijn afstuderen moest vervullen, en waar de rest van dit verslag over handelt.

Deze luidt als volgt: ontwerp, bouw en test een spraakherkenner die gebaseerd is op de SP1000 van General Instrument, en maak een koppeling tussen de herkenner en het Busch Timac X-10 afstandbedieningssysteem waardoor gesproken commando's ook werkelijk kunnen worden uitgevoerd. Als randvoorwaarden en/of wensen moesten de volgende punten in acht worden genomen:

- Het apparaat moet zo goedkoop mogelijk worden. Dit betekent in de praktijk dat er zo weinig mogelijk onderdelen in gebruikt moeten worden en dat dus zoveel mogelijk taken door de software moeten worden overgenomen.
- Als microprocessor heeft de 80C31 van Intel de voorkeur. Dit is een zeer krachtige 8 bit single-chip-processor met allerlei extra functies aan boord, zoals timers, seriële poort, parallelle I/O poorten, uitgebreide interrupt-mogelijkheden, etc.
- Voor wat betreft de software kon gebruik worden gemaakt van de door A. Traas onderzochte software voor de LIS'NER1000 (geschreven in machinetaal voor de 6502 microprocessor). Dit programma was getest en bleek goed te voldoen (redelijke herkenningsscore). Het was derhalve gewenst, dat deze software werd omgeschreven naar de nieuwe microprocessor en dat, waar dat nodig was nog enkele verbeteringen werden aangebracht.
- Het was gewenst dat de oorspronkelijke functies van het Busch Timac X-10 systeem niet verloren zouden gaan, d.w.z. dat het hele systeem, waar de spraakherkenner op aan werd gesloten nog net zo kon functioneren als daarvoor, maar met als extra faciliteit de spraakinvoer.
- De gebruiker moet zelfstandig woorden kunnen trainen en/of wissen, dus **zonder** hulp van derden. Na het inspreken van een woord moet de gebruiker direkt kunnen zien welk woord er herkend is, en bij foutieve herkenning door een tegengesteld commando het foutieve ongedaan maken.
- Op het gebied van vormgeving, terugkoppeling naar de gebruiker, verdere bedieningsmogelijkheden, koppeling naar het Busch Timac X-10 systeem, voeding en andere eigenschappen van het systeem waren geen eisen gesteld. Uit de verschillende mogelijkheden die hiervoor zijn, moest een weloverwogen keuze worden gemaakt. Hier kom ik later nog op terug.

HOOFDSTUK 3

SPRAAKHERKENNING MET DE SP1000

Voor het analyseren van spraak zijn in de loop der tijd diverse methoden ontwikkeld. Enkele van deze methoden zullen hieronder worden aangegeven met hun voor- en nadelen. De in de spraakherkenner gebruikte methode, LPC analyse, wordt daarna uitvoeriger besproken.

§3.1 Methoden voor spraakanalyse

De meeste technieken zijn ófwel te simpel (te slechte eigenschappen) ófwel ze kosten teveel rekentijd, en zijn dus niet geschikt voor een spraakherkenner die een directe respons moet geven op het ingesproken woord. Enkele methoden zijn:

- Zero crossing rate: Het geluid wordt opgedeeld in segmenten van circa 20 msec., waarna per segment het aantal nuldoorgangen geteld wordt. Een voordeel is de zeer simpele systeemopzet, maar een nadeel is de gevoeligheid voor DC offset, brom, ruis en andere stoorsignalen. Toepassingen worden nog gevonden in bepaling van woordgrenzen en bepaling van stemloos/stemhebbend karakter van bepaalde geluiden.
- Autocorrelatiemethode: Over het geluidssignaal wordt de autocorrelatiefunctie berekend, waarna de bepaalde coëfficiënten als representatie voor het geluid worden genomen. Het voordeel van deze methode is dat de energie van het geluid bepaald wordt in de eerste coëfficiënt, en dat de periodieke eigenschappen van het signaal in de overige coëfficiënten tot uitdrukking komt. Een nadeel van deze methode is dat het bepalen van de autocorrelatiefunctie erg veel rekentijd kost en dat er veel coëfficiënten bepaald moeten worden als een redelijke spraakherkenning gewenst is.
- Filterbankanalyse: Deze methode komt vooral op hardware neer. Met een aantal bandfilters wordt een spectrogram samengesteld van het spraaksignaal. De processor kan dan bijvoorbeeld elke 20 msec. de uitgangswaarden van deze filters uitlezen. Deze methode heeft als voordeel dat er van alles bepaald kan worden, zoals grondtonen, harmonischen, geluidsstrekte, etc. Als nadeel geldt hier, dat er teveel

data is, waardoor een flinke datareductie nodig is. Verder kan de invloed van toonhoogte, accenten en spreesnelheid te groot zijn en zijn de formanten moeilijk te bepalen.

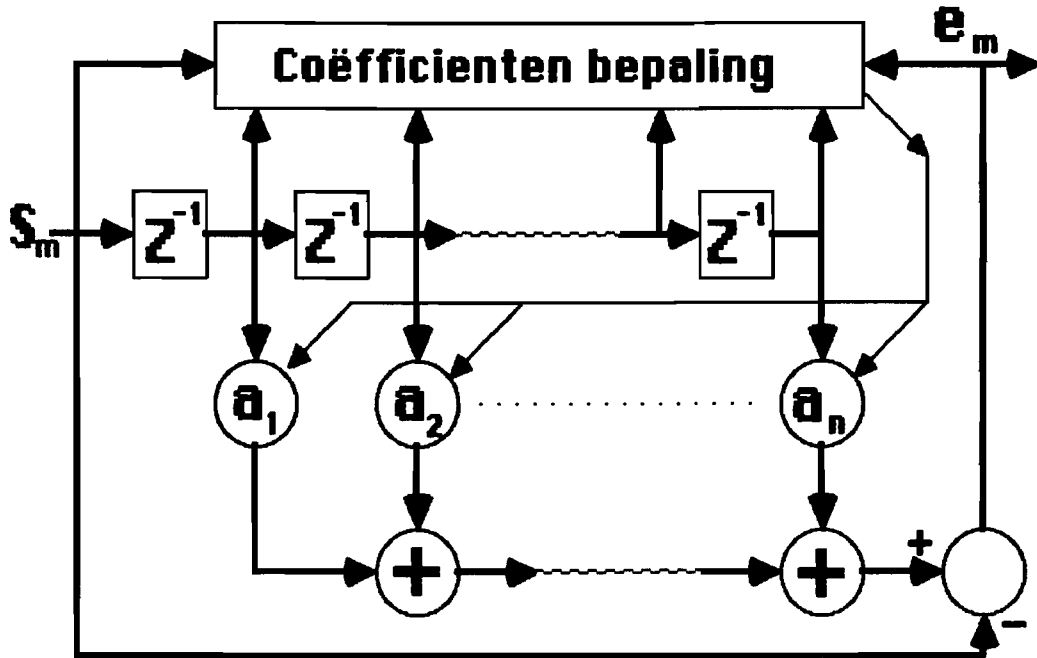
- Fouriertransformatie: Dit is een soort software equivalent van de filterbankenmethode. Het geluid wordt digitaal gesampled en daarna wordt via een FFT algoritme het fourierspectrum of het vermogenspectrum bepaald. Naast de nadelen van de filterbankenanalyse geldt hier ook nog dat er een enorme rekencapaciteit nodig is om per stukje geluid (ca. 20 msec) een volledig fourierspectrum te bepalen.
- LPC analyse: LPC betekent Linear Predictive Coding. In de volgende paragraaf zal hierop verder worden ingegaan.

§3.2 Linear Predictive Coding

LPC analyse is een veel gebruikte analysetechniek bij spraakherkenning en codering en voldoet erg goed. Zoals de naam al suggereert wordt bij deze analysetechniek geprobeerd een voorspelling te maken van het volgende spraaksample op basis van de N voorgaande samples. De achtergronden van de LPC analyse komen in het kort op het volgende neer: Uitgangspunt is het feit dat spraaksignalen slechts langzaam variërende spectrale eigenschappen hebben. Dit betekent dat het mogelijk is het spraaksignaal in korte stukjes van bijvoorbeeld 20 msec. onder te verdelen, waarbij gesteld kan worden dat voor elk van die stukjes signaal (frames) het spectrum constant is. In het tijddomein betekent dat, dat het signaal deterministisch wordt, en dus ook voorspelbaar. Het moet mogelijk zijn om uit de N laatste samples een redelijke voorspelling te maken voor het nieuwe sample. Deze voorspelling zal in het algemeen beter gedaan kunnen worden als N groter wordt. Bij de LPC analyse wordt nu geprobeerd om het nieuwe sample te voorspellen uit de N laatste samples via een lineair model (zie figuur 3.1).

Voor het onthouden van de N laatste samples is een schuifregister nodig van N plaatsen. Deze samples worden eerst elk met hun eigen weegfactor vermenigvuldigd en daarna bij elkaar opgeteld. Deze som wordt nu gebruikt als voorspelling. In formule vorm:

$$S_m^* = \sum_{i=1}^n a_i \cdot S_{m-i} \dots \dots \dots (3.1)$$



Figuur 3.1. LPC predictie filter, waarmee de coëfficiënten bepaald worden.

In figuur 3.1 is te zien dat de berekende voorspelling wordt vergeleken met het werkelijke sample en dat hieruit een errorsignaal $e(m)$ ontstaat. Het is nu de bedoeling om de coëfficiënten a_1 t/m a_n zodanig te bepalen, dat gedurende het hele tijdsinterval van 20 msec. waarvoor ze berekend worden het errorsignaal $e(n)$ minimaal is. In dat geval is het model volgens formule 3.1 een zo goed mogelijk lineair model van het gegeven stukje spraaksignaal. De coëfficiënten a_1 t/m a_n representeren nu een klein stukje spraak en kunnen voor codering en herkenning van spraaksignalen gebruikt worden.

Wiskundig is aan te tonen, dat het bepalen van de coëfficiënten neer komt op het oplossen van een stelsel van N vergelijkingen met N onbekenden voor elk ingelezen sample. In figuur 3.1 gebeurt dit in het blokje "Coëfficiëntenbepaling". Aan de hand van de toestand van het netwerk worden hier de coëfficiënten zodanig aangepast dat voor elk sample het errorsignaal minimaal wordt.

De formule 3.1 stelt eigenlijk een digitaal filter voor met pulsresponsie:

$$h(i) = a_i \dots \dots \dots (3.2)$$

waarbij $h(i)$ de responsie op een deltapuls voorstelt in het tijddomein.

In het frequentiedomein wordt dit:

$$H(z) = \sum_{i=1}^n a_i \cdot z^{-i} \dots \dots \dots (3.3)$$

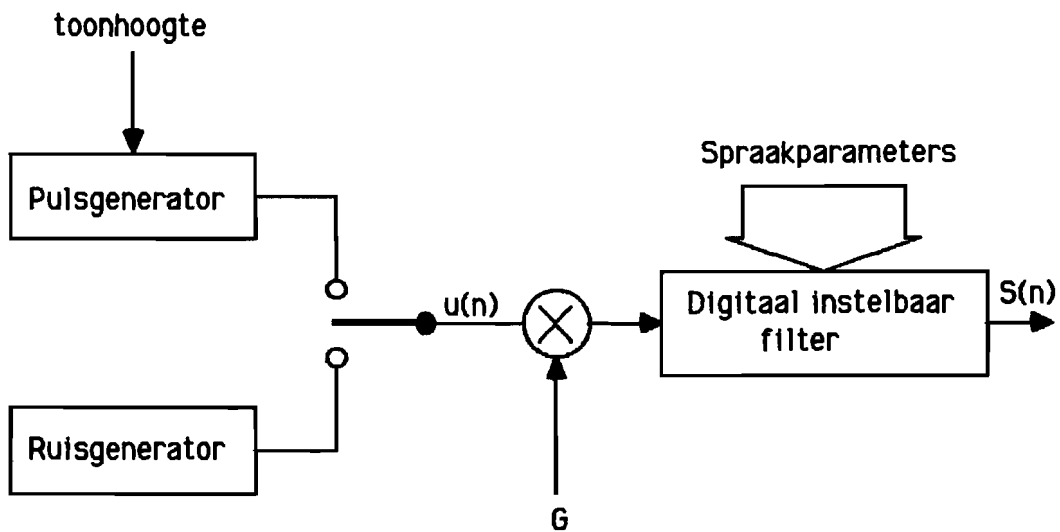
Voor het errorsignaal $e(m)$ geldt:

$$e(m) = S(m) - S^*(m) = S(m) - \sum_{i=1}^n a_i S(m-i) \dots (3.4)$$

De overdracht van het totale systeem van figuur 3.1 is dus:

$$E(z) = 1 - \sum_{i=1}^n a_i z^{-k} \dots \dots \dots (3.5)$$

In figuur 3.2 is een eenvoudig model afgebeeld van een menselijk spraak-productiesysteem. Dit is overgenomen uit lit[1], blz 59.



Figuur 3.2. Een eenvoudig spraakproductiemodel.

Het model bevat een ruisgenerator voor het opwekken van stemloze klanken en een pulstreingenerator met regelbare frequentie voor de stemhebbende klanken. Verder is er nog een instelbaar digitaal filter aanwezig met p polen en geen nulpunten dat het aangeboden ruis- of pulssignaal zodanig bewerkt, dat daarmee allerlei klanken gemaakt kunnen worden.

Het filter in figuur 3.2 kan gekarakteriseerd worden door de volgende vergelijking:

$$H(z) = \frac{S(z)}{U(z)} = \frac{G}{1 - \sum_{k=1}^p b_k z^{-k}} \dots \dots \dots (3.6)$$

Vergelijken we dit systeem met het menselijk spraakproductiesysteem, dan zien we dat het digitale filter eigenlijk de eigenschappen van de mond-, neus- en keelholte vertegenwoordigt. Als we formule 3.6 vergelijken met formule 3.5 dan valt op, dat deze op een constante na elkaars inverse zijn, indien we de coëfficiënten a_k en b_k aan elkaar gelijk maken:

$$H(z) = \frac{G}{E(z)} \dots \dots \dots (3.7)$$

waardoor $e(m) = G u(m)$, dus het in het predictiefilter ontsane foutsignaal lijkt zeer sterk op het bronsignaal in het spraakproductiemodel. Hieruit volgt, dat als we de coëfficiënten a_1 t/m a_n zodanig bepalen dat het errorsignaal lijkt op het bronsignaal (pulstrein of ruis), we een invers filter hebben bepaald, en dat er dan dus een verband is te leggen tussen het spectrum van het spraaksignaal en de berekende coëfficiënten.

Een methode die in de praktijk bruikbaar is voor het bepalen van de gevraagde coëfficiënten is het minimaliseren van de gemiddelde kwadratische predictiefout. Het bronsignaal is namelijk ófwel een pulstrein en dus het grootste deel van de tijd gelijk aan nul, ófwel ruis en dus niet te voorspellen waardoor een verandering van een coëfficiënt geen invloed meer heeft.

Deze methode voor het berekenen van de coëfficiënten komt in de praktijk neer op het oplossen van een stelsel vergelijkingen. Dat kost dus of een zéér krachtige processor, of heel veel hardware (vermenigvuldigers), of

heel veel tijd. Er zijn echter andere methoden ontwikkeld voor het bepalen van de coëfficiënten die minder van het systeem vergen. Sommige van deze methoden leveren geen echte LPC parameters, maar andere coëfficiënten die echter wel sterk daarmee verbonden zijn. Als voorbeeld kan de traliefiltermethode genoemd worden die in de SP1000 is toegepast. Voor een meer uitgebreide beschrijving van deze techniek en enkele andere wordt verwezen naar lit[1], blz 63-69 en lit[7].

§3.3 Het gebruik van de SP1000

De SP1000 levert steeds na een bepaald tijdsinterval negen getallen, die samen één frame vertegenwoordigen. De eerste acht zijn LPC parameters, het laatste getal is de gemiddelde energie-inhoud van het frame. De lengte van een frame is programmeerbaar (via een interne timer).

De SP1000 moet data in digitale vorm hebben. Er moet dus een externe analoog naar digitaal converter op worden aangesloten. Om deze steeds zo goed mogelijk uit te sturen is voorzien in een automatische volumeregeling. Deze wordt ook regelmatig aangepast. De duur van het tijdsinterval waarna dat gebeurt is programmeerbaar.

De samplefrequentie waarmee de SP1000 data inleest is eveneens door de gebruiker te programmeren. Aangezien er een extern anti-aliasing filter nodig is, moet dit mee veranderen met de samplefrequentie. De besturing die nodig is voor de ADC en de sample-en-hold schakeling wordt geheel door de SP1000 verzorgd.

De SP1000 bevat ook een synthese circuit. Hierbij moet de gebruiker per frame opgeven welk excitatiesignaal er gebruikt moet worden (pultrein of ruis), de waarden van de coëfficiënten en toonhoogte-informatie. In principe zou het mogelijk moeten zijn om met de door de SP1000 zelf berekende coëfficiënten -en met via andere methoden verkregen data omtrent toonhoogte en stemloos of stemhebbend karakter- deze chip spraak te laten genereren. In de praktijk is dit echter erg tegengevallen en tot op heden is dit nog niet goed gelukt (lit[8]).

HOOFDSTUK 4

SYSTEEMOPZET EN BEDIENING

De spraakherkenner is bedoeld voor inbouw in het bestaande Busch Timac X-10 binnenshuis-afstandsbedieningssysteem. Dat legt uiteraard enkele beperkingen en eisen op aan de hele opbouw en layout van het systeem, de maximale hoeveelheid hardware, de manier van terugkoppeling naar de gebruiker, etc.

Voordat begonnen kan worden met de bespreking van het spraakherkennersysteem zal dus eerst enige aandacht moeten worden besteed aan het Busch Timac X-10 systeem.

§4.1 Het Busch Timac X-10 systeem.

Dit afstandsbedieningssysteem bestaat meestal uit een zogenaamde netbesturingspost, een ultrasone afstandsbediening en één of meer elektronische afstandsschakelaars, afstandsdimmers en motorsturingen. Met dit systeem kan men binnenshuis lampen dimmen, gordijnen en deuren openen en sluiten, of bijvoorbeeld de TV aan en uit zetten. Er kunnen in principe maximaal 256 kanalen bediend worden, maar per netbesturingspost niet meer dan 16 tegelijk.

De netbesturingspost heeft een schakelmatrix met 22 druktoetsen. Hiervan zijn er 16, waarmee men een kanaal (overeenkomend met een afstandsschakelaar, -dimmer of motorsturing) kan activeren, terwijl de overige voor de functie invoer zijn (aan/uit/hoger/lager/alle dimmers aan/alles uit).

Iedere keer, als er op één van de 6 functietoetsen wordt gedrukt, zullen alle geactiveerde afstandsschakelaars en -dimmers de betreffende functie uitvoeren. Drukt men daarna op een andere functietoets, dan blijven de eerder geactiveerde kanalen geselecteerd en wordt de nieuwe functie voor diezelfde kanalen uitgevoerd. Drukt men daarentegen op een kanaalkeuzetoets, dan worden alle kanalen gede-activeerd en alleen het kanaal behorende bij de ingedrukte toets wordt geactiveerd. Op deze manier kan de

gebruiker via de 22 toetsen 16 kanalen bedienen. Op de netbesturingspost zit verder nog een zogenaamde huiscode schakelaar. Dat is een 16-standen draaischakelaar waarmee men een zgn. hoofdgroep kan kiezen. Op de elektronische afstandsdimmers en -schakelaars zit ook zo'n 16-standen draaischakelaar en alleen wanneer diens stand overeenkomt met die op de netbesturingspost zal de afstandsdimmer of afstandsschakelaar reageren op de netbesturingspost-signalen. Per instelling van de huiscodeschakelaar heeft men dus de beschikking over 16 kanalen, waarmee het totaal 256 kanalen bedraagt. Deze schakelaar is bedoeld om interferentie tussen meerdere systemen in één huis of gebouw te voorkomen, als deze niet met sperfilters van elkaar kunnen worden gescheiden.

Verder heeft de netbesturingspost nog een ultrasone ingang, waarmee commando's kunnen worden ontvangen van de ultrasone afstandsbediening. Deze commando's worden dan geïnterpreteerd alsof ze op het toetsenbord zijn ingevoerd.

De ultrasone afstandsbediening is een klein, op een batterij werkend apparaat, dat ook een schakelmatrix heeft met dezelfde mogelijkheden als de netbesturingspost. De ultrasone afstandsbediening voert echter geen commando's uit, maar stuurt deze ultrasoon over naar de netbesturingspost, die ze vervolgens decodeert en tenslotte uitvoert. Ook op de ultrasone afstandsbediening ontbreekt elke vorm van terugkoppeling naar de gebruiker, zodat men niet weet welke kanalen op een bepaald moment geactiveerd zijn.

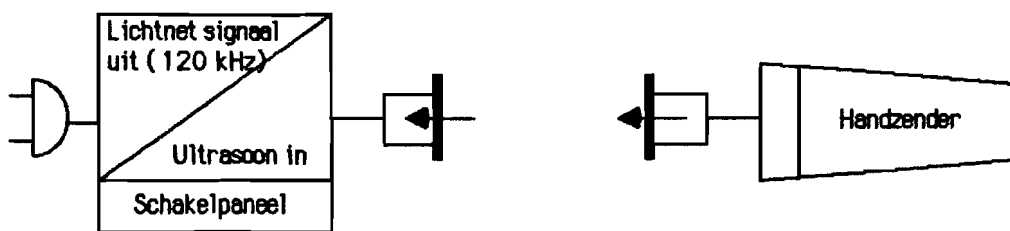
De afstandsschakelaars en -dimmers zijn de eindposten van het systeem. Deze voeren uiteindelijk het commando uit, onder besturing van de netbesturingspost. Op elke afstandsschakelaar of -dimmer zit een tweetal 16-standen schakelaars, waarmee men kan instellen op welk kanaalnummer en huiscodenummer er moet worden gereageerd. De aansturing van deze eindposten vanuit de netbesturingspost gebeurt middels 'hoogfrequent' signalen die via het lichtnet verstuurd worden. Het gaat hier om 120 kHz golfpatronen die steeds vlak na de nuldoorgang van elke periode verzonden worden, dit om storingen tot een minimum terug te brengen. In deze 120 kHz signalen zit dus informatie gecodeerd omtrent huiscode, kanaalnummer en uit te voeren functie.

Naast de hiervoor genoemde drie hoofdbestanddelen zijn er nog diverse andere delen, zoals zeer uitgebreide netbesturingsposten (volledig programmeerbare, met tijdschakelaars, display's, etc.). Daarnaast zijn er nog héél veel hulpstukken, zoals fasenkoppelaars en netfilters. Deze zijn echter voor ons niet van belang en zullen dan ook buiten beschouwing worden gelaten.

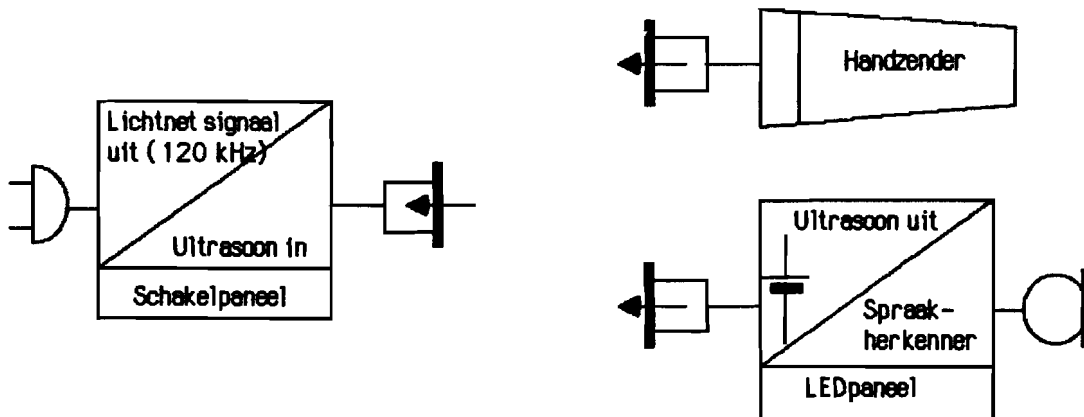
§4.2 De opzet van het nieuwe systeem.

In figuur 4.1a is het voor de functie-invoer verantwoordelijke deel nog eens schematisch weergegeven. Dit gedeelte van het Busch Timac X-10 systeem is voor ons van belang bij het ontwerpen van een spraakherkenner, want die dient ook voor de invoer van opdrachten.

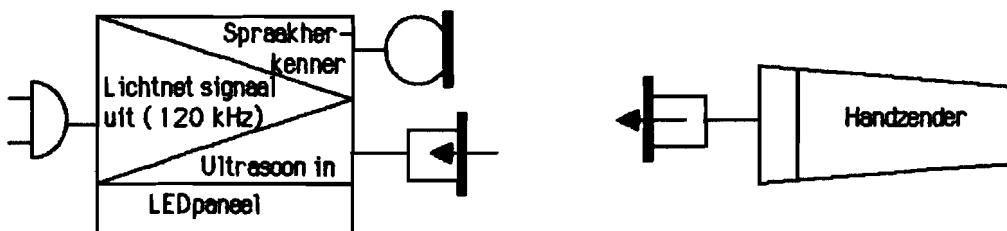
Er zijn nu twee goede mogelijkheden om bij dit systeem een spraak-invoer te realiseren. Deze twee mogelijkheden zijn in figuur 4.1b en 4.1c weergegeven. In het eerste geval wordt er eigenlijk een compleet nieuw apparaatje ontworpen, dat spraak als invoer heeft en ultrasone signalen als uitvoer. Dit apparaat wordt dus als het ware gebruikt parallel aan de ultrasone afstandsbediening. Natuurlijk kan de ultrasone afstandsbediening ook nog gebruikt worden.



Figuur 4.1a. Het bestaande Busch Timac X-10 systeem.



Figuur 4.1b. Mogelijke opzet voor het spraakherkende systeem.



Figuur 4.1c. Tweede mogelijke opzet voor een nieuw systeem.

Bij de tweede mogelijke systeemopzet wordt de spraakherkenner in de netbesturingspost ingebouwd. Zoals in figuur 4.1c is te zien bestaat de installatie nu uit één deel minder dan in het geval van figuur 4.1b. Bij beide systemen werd het nodig geacht een terugkoppeling naar de gebruiker te implementeren in de vorm van een LEDdisplay. Op deze keuze kom ik later in dit hoofdstuk nog terug.

Daar beide in figuur 4.1 voorgestelde mogelijkheden in principe in aanmerking komen moeten de voor- en nadelen van beide systemen goed tegen elkaar worden afgewogen. Beide opzetten hebben hun voor- en nadelen.

De voordelen van de opzet, zoals voorgesteld in figuur 4.1b t.o.v. figuur 4.1c zijn de volgende:

- Het systeem is portable, zodat het bijvoorbeeld ook op een rolstoel kan worden gemonteerd. Het hoeft niet op een vaste plaats te staan en is niet afhankelijk van de aanwezigheid van een vrije lichtnetaansluiting.
- Eventuele (meer valide) huisgenoten worden niet verplicht een ultrasone afstandsbediening aan te schaffen, indien zij ook de lampen willen kunnen aan en uit zetten en gordijnen openen en sluiten.

Tegen de opzet van figuur 4.1b zijn de volgende argumenten:

- De voeding van het systeem kan problemen opleveren. Wanneer men het apparaat een vaste plaats geeft, kan een lichtnet adapter worden gebruikt, waarmee het eerste voordeel al is verdwenen. Bij gebruik op een rolstoel kan eventueel een accu worden gebruikt, maar dat heeft als nadeel, dat het apparaat vast verbonden is met de rolstoel en niet gemakkelijk even kan worden losgemaakt om het ergens anders te gebruiken. Batterijen gebruiken kan ook, maar vanwege het relatief groot verwachte stroomverbruik (100...200 mA, continu) zullen die niet erg lang meegaan.
- Er wordt gebruik gemaakt van ultrasone overdracht van commando's. Deze weg is niet altijd erg betrouwbaar. Als er voorwerpen in de signaalweg staan of de ontvanger en zender staan niet goed op elkaar gericht bestaat de mogelijkheid dat het commando niet goed overkomt. Het hele commando moet dan opnieuw worden ingesproken, hetgeen zeer hinderlijk kan zijn. Ook wanneer gelijktijdig iemand anders de ultrasone afstandsbediening bedient kan er van alles misgaan.
- Men is verplicht om naast de spraakherkenner ook de netbesturingspost aan te schaffen. Deze kost ca. fl. 265,- exclusief B.T.W.
- Er kunnen nu met de spraakherkenner nooit meer dan 16 kanalen worden bediend (net zoals met de ultrasone afstandsbediening), omdat de huiscode niet kan worden beïnvloed.

Een spraakherkenner voor het Busch Timac X-10 afstandsbedieningssysteem

- Een heel groot nadeel is dat de op het display aangegeven toestand van het systeem niet overeen hoeft te komen met de werkelijke toestand. Dit zal ik met een voorbeeld verduidelijken: Stel dat iemand het kanaal 4 activeert door op de ultrasone afstandsbediening toets 4 in te drukken. Vanaf dat moment zal afstandsschakelaar/dimmer 4 geactiveerd zijn en bij de eerst volgende functiekeuze die functie gaan uitvoeren. Als nu de ultrasone afstandsbediening verder niet meer wordt gebruikt en een tijd later wordt door de gebruiker van de spraakherkenner de functie van kanaal 1 ingesproken, dan zijn vanaf dat moment de kanalen 1 en 4 geactiveerd, terwijl op het display alleen kanaal 1 staat aangegeven. De spraakherkenner weet immers niet dat daarvoor al iemand via de ultrasone afstandsbediening een kanaal heeft geactiveerd. Het gevolg is, dat zowel kanaal 1 als 4 op de volgende functie zullen reageren. De gewenste functie wordt dus normaal uitgevoerd, samen met een andere waarvan men het niet weet. Nu is dat op zich nog niet fataal, maar wat als kanaal 4 een buitendeur-opener is, en men hoort de deur niet opengaan....

De systeem opzet van figuur 4.1c heeft meer voordelen dan nadelen. De voordelen zijn:

- De kosten zijn zeer waarschijnlijk lager dan voor het systeem in figuur 4.1b. Men hoeft nu niet de spraakherkenner én de netbesturingspost te kopen.
- Daar de microprocessor van de spraakherkenner nu in de netbesturingspost zit ingebouwd kan deze in principe de huiscode onderscheppen of veranderen. Zodoende is het mogelijk om 256 kanalen te bedienen in plaats van slechts 16.
- De voeding is ook geen probleem meer, omdat het hele systeem toch op het lichtnet moet worden aangesloten (voor de aansturing van de eindposten). Er kan met een kleine trafo worden gewerkt.
- Zodra de spraakherkenner een functiewoord herkent, wordt die functie ook echt uitgevoerd. De zwakke schakel in de keten (ultrasone overdracht) is nu verdwenen.
- Het LEDpaneel geeft nu altijd de actuele toestand van het systeem weer, tenzij men nog een andere netbesturingspost gebruikt die op dezelfde huiscode staat ingesteld. De gebruiker kan de informatie op het display volledig vertrouwen.

Aan deze systeemopzet kleven ook wel enkele bezwaren, maar die zijn zeker niet ernstig:

- De nieuwe netbesturingspost is niet meer via druktoetsen te bedienen, men dient een ultrasone afstandsbediening aan te schaffen (kosten circa fl. 60,-).
- Het systeem is gebonden aan het lichtnet en is dus niet portable.

Deze laatste twee bezwaren kunnen echter verworpen worden, want rond dit tijdstip verschijnt een nieuw soort toetsenbord op de markt met een ingebouwd LEDdisplay. Wanneer we dit zouden gebruiken kan de schakelmatrix gewoon blijven zitten. Verder is het verbonden zijn met het lichtnet niet zo'n groot bezwaar, want mensen die zich slecht kunnen voortbewegen hebben dit apparaat dan meestal naast hun bed of stoel staan, en mensen met een rolstoel kunnen zich wel naar de plaats waar de spraakherkenner staat begeven, om daar een commando in te spreken.

Als we nu alle voor- en nadelen tegen elkaar afwegen, moeten we tot de conclusie komen dat systeem 4.1c meer voordelen dan nadelen heeft, terwijl systeem 4.1b meer nadelen heeft. Bovendien zijn de nadelen van 4.1b ernstiger dan die van 4.1c, en zijn de voordelen van 4.1c weer groter dan die van 4.1b. Daarom hebben we gekozen voor de opzet volgens systeem 4.1c. We gaan dus een netbesturingspost ontwerpen met een ingebouwde spraakherkenner, die als vervanging dient van de oude, en die ook alle functies heeft van de oude netbesturingspost.

§4.3 Terugkoppeling naar de gebruiker.

Nu de opbouw van het systeem vast staat, moet nog een keuze worden gemaakt voor de te gebruiken terugkoppeling naar de gebruiker. In het voorgaande is reeds gesproken over een LEDdisplay, waarmee in feite de gemaakte keuze al is aangegeven, maar deze moet nog wel beargumenteerd worden.

We kunnen de mogelijke terugkoppelingen in twee hoofdgroepen verdelen, namelijk audio en visueel. Onder audio wordt dan verstaan, dat het apparaat in gesproken taal een terugmelding geeft omtrent het gekozen kanaal of functie. Bij visuele terugkoppeling denken we in eerste instantie aan LED's, alfanumerieke displays of 7 segments-displays.

Audio terugkoppeling lijkt op het eerste gezicht grote voordelen te hebben. Zo is er geen oogcontact nodig tussen apparaat en gebruiker. Het gezichtsvermogen van de gebruiker is niet meer van belang. Verstoring van

de audioterugmelding treedt niet op, want we mogen niet uitgaan van storend achtergrond lawaai, omdat dan de spraakherkenning toch al niet meer goed kan functioneren. Verder is audio gemakkelijk in het gebruik (het is eenvoudiger als men in gesproken woord te horen krijgt welk kanaal er is gekozen, dan wanneer men dat zelf aan de hand van een display moet afleiden).

Er zijn echter ook enkele bezwaren aan te voeren. Zo zou de neiging kunnen ontstaan de spreekstem na te bootsen (die waarschijnlijk niet echt goed zou klinken), zodat er een soort commando-toon ontstaat. Dat was niet de bedoeling. Verder is het bijzonder irritant, als het apparaat een woord foutief herkent en dus met een verkeerde terugmelding komt. Als er zich nog meer mensen in de ruimte bevinden, is het zeker denkbaar, dat men het apparaat niet zo snel zal gebruiken, omdat audio (een computerstem) veel opvallender is dan bijvoorbeeld een display. Een ander bezwaar zou zijn, dat audio langzamer is dan visuele terugkoppeling. Tegen de tijd, dat nu de informatie al op het display staat, moet de audio-generator nog beginnen, met het uitspreken van een hele zin. Tenslotte zijn er ook nog twee technische moeilijkheden: Voor het genereren van spraak is erg veel data nodig, en die zal dus ergens vandaan moeten komen. Als we bijvoorbeeld de stem van de gebruiker willen nabootsen, dan moet er een ingewikkelde en veel rekentijd eisende analyse plaatsvinden van de ingelezen spraaksamples. Hiervoor ontbreken in een klein systeempje de mogelijkheden. Als we een simpele vaste terugmelding programmeren (bijv. het kanaalnummer behorend bij het ingesproken woord), dan vergt dat nog steeds erg veel data. Bovendien lijkt een dergelijke simpele melding niet erg zinvol. Als er dan toch audioterugkoppeling wordt gebruikt, doe het dan meteen goed, en spreek de naam van het apparaat uit dat gekozen is. Technisch behoort dat echter in een klein systeem niet tot de mogelijkheden.

Audio terugmeldingen vallen voorlopig af, en dat betekent, dat we nu een keuze moeten maken tussen verschillende soorten displays. Alle displays hebben als voordeel, dat ze snel en onopvallend zijn. Het apparaat (en dus indirect de gebruiker) trekt niet de aandacht. Een algemeen nadeel is, dat de gebruiker een redelijk goed gezichtsvermogen moet hebben, en dat het apparaat dicht bij de gebruiker moet staan (oogcontact).

Een eerste mogelijkheid, die zich aandient is het gebruik van LED's. Als we een display maken, dat bestaat uit een matrix van LED's die in dezelfde layout zijn opgesteld als de druktoetsen op de netbesturingspost, dan kunnen we een systeem maken, waarvan de bediening net zo eenvoudig is als van het bestaande systeem. Een klein nadeel is, dat men zelf moet onthouden welke afstandsschakelaar of -dimmer bij welke LED hoort, maar dat moet bij het

bestaande systeem ook al.

Een alfanumeriek display is een tweede mogelijkheid. Voor de gebruiker is dit veel duidelijker en eenvoudiger (spraak in, tekst uit). In het ideale geval spreekt men een woord in, waarna het herkende woord op het display verschijnt. Allerlei speciale functies en bedieningen kunnen nu veel mooier worden geregeld (zoals trainingsmode, evt. twee woorden onder één LED). Een nadeel is, dat het moeilijker wordt, om bij gelijktijdige activering van meerdere kanalen dit weer te geven. Hiervoor is een LED display het beste geschikt. Een bijna niet te overkomen technisch probleem is de vraag waar de teksten voor het display vandaan moeten komen. Deze moeten natuurlijk per persoon apart programmeerbaar zijn en ook nog te veranderen door de gebruiker. Nu is het invoeren van teksten op dit systeem absoluut onmogelijk. Dat zou via een externe computer moeten gebeuren, waardoor de gebruiker het dus zeker niet zelf kan. Alfanumerieke displays zijn alleen bruikbaar bij vast geprogrammeerde systemen.

Dan is er nog de mogelijkheid van 7-segments displays. Hierop kan slechts één getal worden weergegeven. De mogelijkheid van het weergeven van de toestand van het systeem (welke kanalen zijn er op dit moment geactiveerd) is zeer beperkt. Verder moet de gebruiker nu het verband onthouden tussen een nummer en een woord, en dat is moeilijker dan het onthouden van het verband tussen een LED in een matrix en een woord (dit laatste komt namelijk neer op patroonherkenning, en daartoe blijken de menselijke hersenen beter in staat dan in het onthouden van nummers).

Wanneer we alle mogelijkheden afgaan en de voor- en nadelen tegen elkaar afwegen, dan moeten we tot de conclusie komen, dat het voorlopig het beste is om uit te gaan van een LED display van circa 25 LED's (22 die overeenkomen met de druktoetsen op de oude netbesturingspost en enkele extra voor statusweergave, zoals trainingsmode, e.d). Voor wat betreft de plaatsing van de LED's verdient een layout, die overeenkomt met de plaatsing van de druktoetsen op de netbesturingspost de voorkeur.

§4.4 Invoer van kanalen en functies.

De belangrijkste methode voor invoer van functies is natuurlijk inspreken. Maar ik wil er hier nogmaals op wijzen, dat de spraakherkenner spreker-afhankelijk is en dat wil dus zeggen, dat eventuele huisgenoten of verpleegsters niet van deze faciliteit gebruik kunnen maken. Zij moeten voor het aanzetten van een lamp gebruik maken van een ultrasone afstandsbediening of een andere (gewone) netbesturingspost. We willen

echter de mensen niet verplichten om ook nog een ultrasone afstandsbediening te kopen. Vandaar dat de vraag ontstond of er eventueel ook een druktoets in het apparaat moest worden ingebouwd, waarmee het geheel als monoselektor bediend kon worden (via het scan en bevestigings-principe). Een andere reden voor een extra knop zou kunnen zijn dat hiermee in één keer alle lichten aan kunnen gaan (in noodgevallen) of als alarmknop, speciale trainingsknop of resetknop.

We hebben na overleg echter toch besloten om maar van één speciale druktoets af te zien en alleen via spraak en ultrasoon invoer te werken, omdat:

- De kosten van een complete installatie al ruim fl. 1000,- zullen gaan bedragen, zodat die fl. 60,- extra voor een ultrasone afstandsbediening niet zoveel meer uitmaakt.
- Het monoselektorprincipe vrij omslachtig en langzaam is, en de (valide) huisgenoten dit principe nu ook moeten leren en gebruiken.
- Wat betreft de 'alle lampen aan' functie in noodgevallen er in bijna elke kamer wel een lamp is die niet op het systeem zit (meestal plafondlamp).
- De resetfunctie en de trainingsfunctie ook wel op een andere manier kunnen worden geïmplementeerd. Zie hiervoor de paragraaf over bediening.

We blijven dus bij de keuze van circa 25 LED's op het display, géén speciale knop, maar wel eventueel een speciaal toetsenbord met ingebouwde LED's om de oorspronkelijke functie van de netbesturingspost te kunnen handhaven.

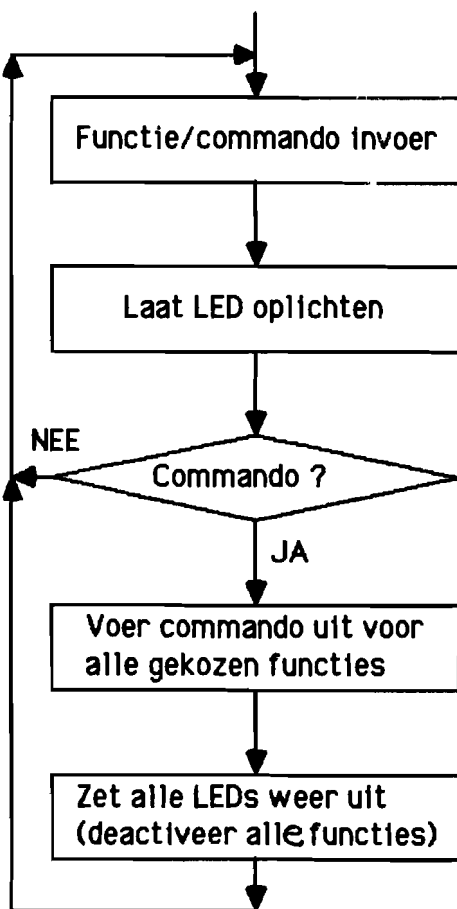
§4.5 Functiekeuzemogelijkheden.

Er zijn twee wezenlijk verschillende methoden om kanalen te kiezen en functies in te voeren (via spraak). De eerste methode houdt in, dat men eerst alle kanalen kiest, die men wil bedienen, waarna de functie wordt gegeven (dus meerdere kanalen tegelijk bedienen). Bij de tweede methode kan er slechts één kanaal tegelijk worden gekozen (zie figuur 4.2).

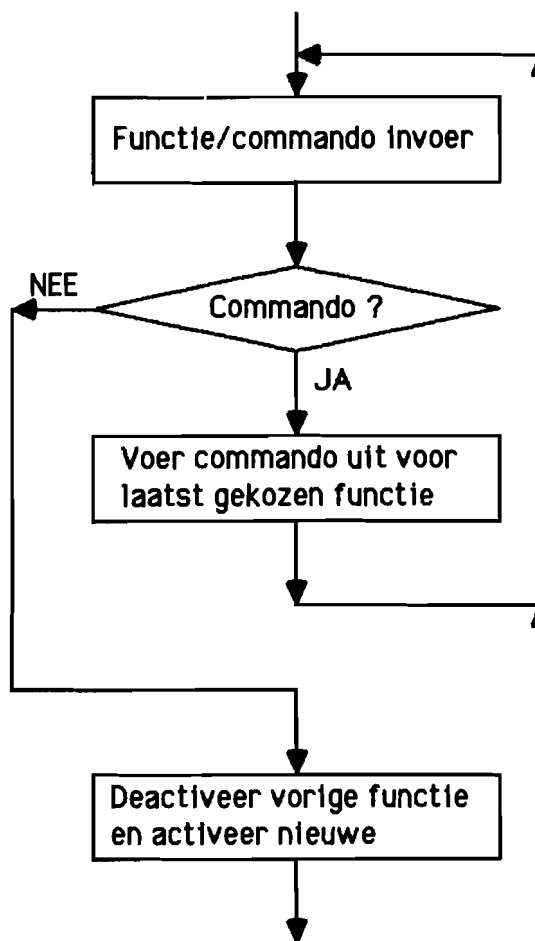
In §4.1 is reeds aan de orde geweest, dat bij het Busch Timac X-10 systeem de mogelijkheid bestaat om meerdere kanalen tegelijk te activeren en een functie te laten uitvoeren. Wanneer we dit ook bij de spraakherkenner willen implementeren schept dat echter toch wel enkele problemen. We moeten namelijk terdege rekening houden met de mogelijkheid, dat er een woord foutief wordt herkend. Dit zou betekenen, dat er ten onrechte een

kanaal wordt geactiveerd. Om deze onacceptabele toestand te voorkomen zou er van een soort van keuze-en-bevestigingsprincipe gebruik moeten worden gemaakt, zoals dat al bij de monoselektor werd gedaan.

In figuur 4.2 zijn de genoemde twee bedieningsmethodes in de vorm van een flowdiagram weergegeven. Om een keuze te maken tussen de twee voorgestelde methodes moeten we dus nagaan welke in de praktijk het snelste en eenvoudigste is. Een redelijk veel voorkomende praktijksituatie is, dat men een aantal lampen in huis wil aandoen. Met de methode van figuur 4.2a (meerdere kanalen) kan dat in één commando. Via methode b moet per lamp een commando worden gegeven. Aan de andere kant kan gesteld worden, dat lampen meestal op afstandsdimmers worden aangesloten en deze dus via het 'alle dimmers aan' functie allemaal tegelijk kunnen worden aangezet. Een groot nadeel van de meer-kanalen- tegelijk-methode is het feit, dat na het uitvoeren van een functie alle kanalen moeten worden gede-actieveerd, omdat ze anders nooit meer kunnen worden uitgezet (bij het kiezen van een nieuw kanaal wordt dit mede-geactiveerd, maar de al geactiveerde kanalen worden niet afgezet).



Figuur 4.2a. Meerdere kanalen tegelijk



Figuur 4.2b. Eén kanaal tegelijk

Dit nadeel heeft de tweede methode niet. Na het kiezen van een kanaal en het uitvoeren van een functie blijft dat kanaal geactiveerd en kan men gewoon een andere functie voor datzelfde kanaal invoeren. Als bij het dimmen van een lamp de helderheid iets te ver is opgelopen, kan men dus gewoon een inverse dimfunctie inspreken tot de helderheid goed is. Bovendien heeft deze methode als voordeel, dat er niet met bevestigingen hoeft te worden gewerkt. Stel dat de spraakherkenner een kanaalkeuzewoord foutief herkent, dan zal dat door de gebruiker worden opgemerkt, doordat op het display de verkeerde LED oplicht. Door dan het gewenste woord nogmaals in te spreken (tot de herkenning wel goed gaat) kan heel eenvoudig een kanaal worden gekozen. Dat ook bij functiewoorden geen bevestiging nodig is, is iets moeilijker in te zien. De mogelijke fouten die hier optreden, zijn het verwisselen van de woorden AAN, UIT, HOGER en LAGER (de gebruiker kan natuurlijk zelf bepalen welke woorden hij/zij aan deze functies wil toekennen). Foutieve herkenning van 'aan' als 'uit' of omgekeerd is niet erg, omdat dan een lamp bijvoorbeeld uit zou blijven terwijl men hem aan wilde zetten. Door nogmaals 'aan' in te spreken gaat de lamp toch nog aan. Hoger en lager zijn functies die alleen werken op afstandsdimmers en deze zijn meestal aangesloten op lampen, waardoor foutieve herkenning geen ernstige gevolgen zal hebben en eenvoudig ophefbaar is door het goede woord nogmaals in te spreken. Een apart geval treedt op bij de functies 'lampen aan' en 'alles uit'. Dit zijn bij de bestaande netbesturingspost direct uitgevoerde functies, d.w.z. dat de bijbehorende toets zowel kanaalkeuze-toets als functietoets tegelijk is. Om te voorkomen, dat bij foutieve herkenning één van deze functies spontaan wordt uitgevoerd moeten deze worden opgesplitst in 'lampen' gevolgd door 'aan' enerzijds en 'alles' gevolgd door 'uit' anderzijds. Hierbij dienen 'aan' en 'uit' dus als een soort van bevestiging voor de kanaalkeuze, net zoals bij de gewone kanalen.

Alleen bij deuropeners is het gevaar aanwezig dat deze opengaat of blijft staan door foutieve herkenning of zelfs zonder dat de gebruiker het weet. Deuropeners werken vaak maar één kant op (ze kunnen niet uit zichzelf de deur sluiten). Er is hier dus geen inverse-functie om een foutieve herkenning op te heffen. Men moet echter eerst voor het kanaal van de deuropener kiezen en daarna een functie inspreken om hem te openen, hetgeen de kans op het per ongeluk openen van een deur heel klein maakt. De kans is en blijft aanwezig dat het gebeurt en het moet daarom worden afgeraden om een deuropener via spraak te bedienen, als men niet zelf in staat is de deur weer te sluiten.

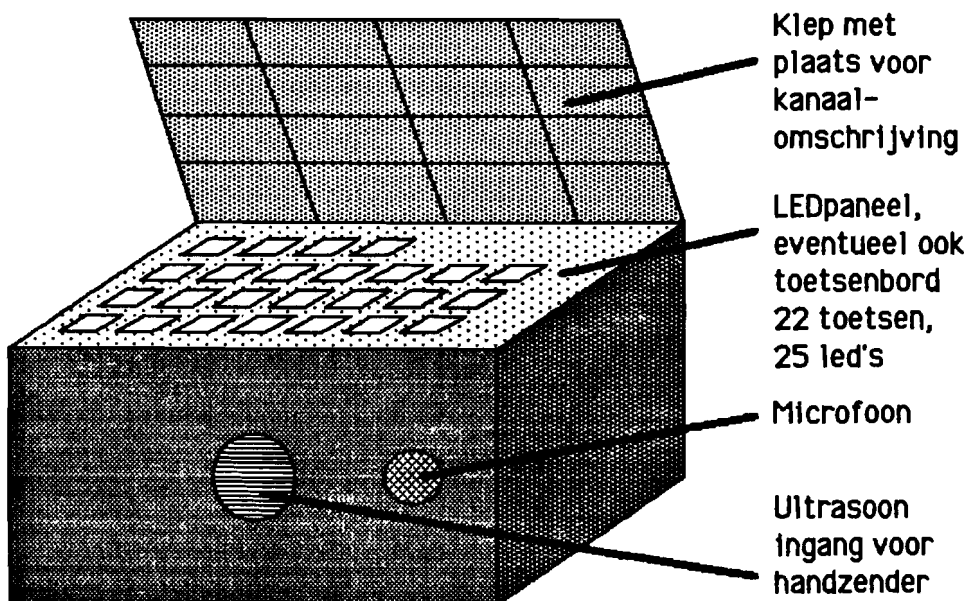
§4.6 Bedieningshandleiding spraakherkenner.

In deze paragraaf zal een onafhankelijke bedieningshandleiding worden gegeven voor het werken met de spraakherkenner voor het Busch Timac X-10 afstandsbedieningssysteem. Deze handleiding kan onafhankelijk van de rest van het verslag worden gelezen en moet iedereen die hem gelezen heeft in staat stellen, het apparaat te bedienen.

Twee instellingen, namelijk de looptijd in de trainingsmode en de huiscode zijn niet met software instelbaar (dus ook niet via spraakinvoer), maar via druktoetsen. Deze instellingen moeten éénmaal door bijvoorbeeld een kennis worden gedaan, en behoeven daarna nooit meer te worden veranderd. Alle andere instellingen kunnen door de gebruiker zelf worden veranderd, volledig via spraakinvoer.

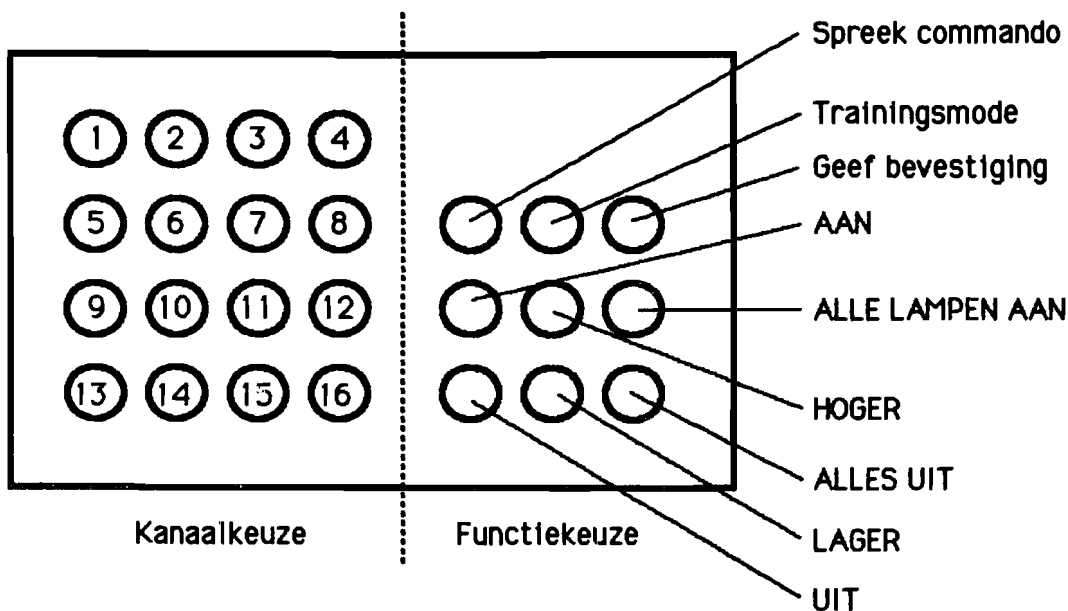
Gebruikershandleiding spraakherkenner.

In figuur 4.3 is de nieuwe netbesturingspost schematisch weergegeven. Hierin is duidelijk te zien, dat we de layout van de bestaande netbesturingspost hebben overgenomen.



Figuur 4.3. De nieuwe netbesturingspost met spraakherkenner.

Het hele systeem ziet er nog net zo uit als vroeger. Pas bij een nadere beschouwing van het toetsenbord/LED-paneel is te zien dat het hier om een nieuw systeem gaat. De schakelmatrix bestaat uit 25 LED's en 22 toetsen. De toetsen zitten op dezelfde plaats en hebben dezelfde functie als bij het oude systeem. De LED's bij een kanaaltoets geven aan of dat betreffende kanaal wel of niet is geactiveerd. De LED's bij de functietoetsen geven aan welke functie nu wordt uitgevoerd of het laatst is uitgevoerd. Dan blijven er nog drie LED's over die nog niet zijn verklaard.



Figuur 4.4. De LED/schakelmatrix van de nieuwe netbesturingspost

De eerste LED geeft aan of er een commando mag worden ingesproken (de spreekLED). Wanneer deze oplicht is de spraakherkenner actief en kan er dus een woord worden ingesproken. Een dergelijk signaal aan de gebruiker is nodig, omdat de processor vaak met andere taken bezig is, en dan even niet naar spraak kan luisteren. Steeds nadat de spraakherkenner een woord heeft ingeladen moet dit aan het herkenningsalgoritme worden doorgegeven om daar verder te worden onderzocht. Dit herkennen kost tijd, (tot 1 sec als men veel woorden heeft ingesproken). Verder kan de processor bij de ontvangst van ultrasone signalen ook niet naar spraak luisteren, omdat het inladen van het ultrasoon signaal teveel tijd kost. Tenslotte moet in de trainingsmode de gebruiker woorden op het juiste moment inspreken. Om te voorkomen dat de woorden helemaal verkeerd worden ingesproken moet de gebruiker goed weten wanneer hij of zij mag spreken.

De tweede LED is de trainings-indicator. Wanneer deze LED brandt,

1700-120-1000?

bevindt het apparaat zich in de trainingsmode. Zie hiervoor verder het deel over de training.

De derde en laatste LED die nog verklaard moet worden is de bevestigingsLED. Deze wordt alleen gebruikt in de trainingsmode. Omdat training van woorden in principe voor de ingebouwde database van het systeem een destructief karakter heeft, moeten hierbij enige beveiligingen worden ingebouwd, waardoor het per ongeluk in de trainingsmode geraken en iets wissen of hertrainen praktisch onmogelijk wordt. Eén van deze beveiligingen is een bevestiging voor elk te trainen woord. Wanneer deze LED oplicht, wacht het systeem op een bevestiging. Indien deze niet binnen een bepaalde tijd komt wordt verder gegaan zonder te trainen. Voor verdere informatie wordt verwezen naar het deel over training.

In hetvolgende wordt enkele malen gesproken over het 'maken van geluid' door de gebruiker, dit ter bevestiging van keuzes of het stoppen van een dim-functie of het starten van de trainingsmode bij het aanzetten van het apparaat. Onder het 'maken van geluid' wordt niets anders verstaan dan het genereren van een kort stukje geluid, dat minstens 10 dB boven het achtergrondlawaai ligt (ter plaatse van de microfoon natuurlijk). De spraakherkenner kan dat detecteren en dan al of niet een bepaalde actie ondernemen.

De herkenningsmode:

Tijdens normaal gebruik bevindt het apparaat zich altijd in de herkenningsmode. Men kan nu via spraak, het toetsenbord of de ultrasone afstandsbediening commando's invoeren, die dan ook worden uitgevoerd. Via het LEDdisplay kan de toestand van het systeem worden afgelezen.

Wanneer het apparaat wordt aangezet, en er bevinden zich reeds woorden in het interne geheugen, dan zal na circa 2 seconden de spreekLED oplichten. Verder geeft geen enkele andere LED licht, want er is geen kanaal geactiveerd. Men kan nu via spraak één van de 16 kanalen kiezen, die op de linkerzijde van het display zijn weergegeven, of één van de functies 'lampen' en 'alles'. Er wordt hier nogmaals op gewezen, dat de woordkeuze geheel vrij is aan de gebruiker, en dat de hier gebruikte namen voor functies en kanalen slechts een keuze van de auteur zijn. Kiest men een kanaal, dan gaat de bijbehorende LED branden. Als er een verkeerde LED gaat branden, is de woordherkenning mislukt en moet het woord voor het gewenste kanaal nogmaals worden ingesproken. Iedere keer als men een kanaalkeuzewoord

inspreekt, dan zullen alle eerder gekozen kanalen worden gede-activeerd (LED's gaan uit) en alleen het zojuist ingesproken kanaal wordt actief (LED gaat aan). Verder gaat bij een kanaalkeuze natuurlijk ook een eventueel nog oplichtende functieLED uit.

Nadat er een kanaal is gekozen kan een functie worden ingegeven. Dit kan één van de woorden aan, uit, hoger of lager zijn. Zodra de spraakherkenner één van deze woorden herkent, zal de bijbehorende LED oplichten en wordt de functie uitgevoerd. Als het woord verkeerd was herkend, kan men direkt de inverse functie ingeven. Na de uitvoer ervan branden er drie LED's, namelijk de spreekLED, de gekozen kanaalLED en de gekozen functieLED. Nu zijn er drie mogelijkheden. Ten eerste kan men een nieuwe functie inspreken. De bijbehorende LED zal gaan branden en de vorige gaat uit. De functie wordt uitgevoerd en het systeem is in een identieke toestand gekomen. De tweede mogelijkheid is een nieuw kanaal kiezen. In dat geval wordt hetzelfde gereageerd als in de al beschreven reset-situatie. Ten derde kan men circa 10 seconden niets doen, waarna het systeem via een time-out automatisch terugkomt in de resetsituatie. In deze situatie is geen enkel kanaal geactiveerd. Deze time-out is ingebouwd als extra beveiliging om te voorkomen dat de spraakherkenner uit achtergrondgesprekken woorden oppikt en die als functies ziet. Door deze beveiliging moeten kanaal en functie binnen een korte tijd na elkaar komen.

Een probleem treedt nog op als er een dimfunctie wordt gekozen. Deze kan wel worden gestart, maar hoe lang moet het dimmen duren? Het is niet erg praktisch als men voor elke dim-stap het hele commando opnieuw moet inspreken. Hiervoor is de volgende oplossing gevonden: De spraakherkenner begint met de uitvoer van de dimfunctie en blijft dimmen (met dezelfde snelheid als het oorspronkelijke systeem) totdat er een geluidsenergiestijging van minstens 10 dB wordt geconstateerd. Men kan bijvoorbeeld het woord 'stop' uitspreken om dit te bewerkstelligen.

Niet alle woorden kunnen te allen tijde worden herkend. Kanalen kunnen altijd worden ingesproken. Functies daarentegen alléén wanneer er een kanaal actief is. Bovendien kan men na uitvoeren van een functie (als er een functieLED brandt) alleen de inverse daarvan geven, en niet een willekeurig ander. Het woord, behorende bij het starten van de trainingsmode kan te allen tijde worden ingesproken. Zie hiervoor het deel over de trainingsmode.

Men kan alleen woorden inspreken als de spreekLED oplicht. Als deze LED uit is, wordt er geen spraakdata door de microprocessor ingelezen en heeft spreken dus ook geen zin.

Uit het bovenstaande is het wel duidelijk dat er maar één kanaal tegelijk via spraak kan worden geactiveerd. Dat geldt echter niet voor de

ultrasone afstandsbediening en het toetsenbord. Deze reageren nog net zo als in het bestaande Busch Timac X-10 systeem. Als met de ultrasone afstandsbediening of het toetsenbord een kanaal wordt gekozen, dan zal deze mede worden geactiveerd, dus samen met de reeds geactiveerde kanalen. Dit betekent, dat het dus mogelijk is om bijvoorbeeld kanalen 1, 4, 5, 7, 10 en 16 eerst te kiezen en vervolgens deze allemaal tegelijk eenzelfde functie te laten uitvoeren. Op het LEDdisplay zullen dan alle kanaalLED's gaan branden die behoren tot geactiveerde kanalen. Kanalen worden pas gede-actieveerd als na het uitvoeren van een functie een nieuw kanaal wordt gekozen (zoals in het bestaande systeem) of als er een time-out optreedt (circa 10 sec). Ook hier kan te allen tijde een nieuw kanaal worden gekozen. Maar verder kan hier ook elke functie door elke andere willekeurige functie gevolgd worden. Bovendien geldt voor de ultrasone afstandsbediening en het toetsenbord, dat hier 'alle lampen aan' en 'alles uit' niet zijn opgesplitst in twee aparte toetsen, zoals dat bij spraak nodig bleek te zijn. Net zoals bij het oude systeem geeft één druk op de knop hier direkt het gewenste resultaat.

Omdat het apparaat verschillend moet reageren op commando's via spraak enerzijds en commando's van de ultrasone afstandsbediening of toetsenbord anderzijds is het wat moeilijk in te zien wat er gebeurt als deze gelijktijdig ingegeven worden. In het programma van de microprocessor zijn enkele voorzieningen getroffen om dit goed te laten verlopen. Voor de gebruiker is het handig om te weten, dat steeds wanneer er via spraak een kanaal wordt gekozen, alle oplichtende kanaalLED's doven en alleen de LED van het gekozen kanaal gaat branden. Als er via de ultrasone afstandsbediening of het toetsenbord een kanaal wordt gekozen, zal de bijbehorende LED ook gaan branden, maar de al oplichtende LED's blijven aan (en daarmee de kanalen geactiveerd).

De trainingsmode:

In de trainingsmode kunnen de verschillende woorden die de spraakherkenner later moet kunnen herkennen worden getraind. Daar het trainen van een woord de referentiewoorden in het geheugen overschrijft en dus een destructief karakter heeft bij onjuist gebruik moet deze goed beveiligd worden tegen per ongeluk trainen van een woord.

Er zijn drie mogelijke wegen, langs welke men in deze mode terecht kan komen. De eerste is vrij eenvoudig, maar voor de gebruiker niet interessant. Als het apparaat voor de allereerste keer wordt ingeschakeld zal er nog geen enkel woord in het geheugen zijn opgeslagen. Omdat in dat geval

spraakherkenning toch onmogelijk is, komt het apparaat dan automatisch in de trainingsmode op. Dit zal steeds gebeuren, zolang er nog geen woorden zijn opgeslagen in het geheugen. Ook wanneer men door het wissen van woorden (zie verderop in deze beschrijving) het hele geheugen heeft gewist, zal het apparaat niet meer naar de herkenningmode terugkeren, totdat er minstens één woord is ingesproken en opgeslagen.

De tweede methode om in de trainingsmode te komen is het inspreken van het woord, behorende bij de functie 'training'. Dit is de normale manier om vanuit de herkenningmode in de trainingsmode te komen. Men moet natuurlijk wel eerst even dat woord hebben ingesproken, voordat de spraakherkenner het kan herkennen. Het verdient dus aanbeveling om dit woord meteen te trainen (het is echter niet noodzakelijk). Om te voorkomen dat de herkenner te gemakkelijk in de trainingsmode geraakt, moet datzelfde woord nogmaals binnen circa 10 seconden worden ingesproken. Tijdens deze 10 seconden lichten de bevestigingsLED en de trainingsLED op. Indien de bevestiging niet komt, of er komt een ander woord, dan wordt niet naar de trainingsmode overgegaan. Om via spraak in deze mode te komen moet er natuurlijk wel eerst een woord voor worden ingesproken. Bovendien moet het ook mogelijk zijn om een training te starten als de gebruiker bijvoorbeeld hees wordt en de herkenner het helemaal laat afweten. Voor dat geval is de derde mogelijkheid om in de trainingsmode te komen ingebouwd. Deze houdt in, dat men binnen 2 seconden na het aanzetten van het apparaat even wat geluid moet maken (een 'schreeuw' van 10 dB boven het achtergrond lawaai is voldoende, maar een redelijker alternatief is het uitspreken van het woordje 'training'). Het apparaat test namelijk in de eerste 2 seconden na het aanzetten op zo'n stijging van de geluidsenergie en gaat -indien die inderdaad wordt gedetecteerd- over in de trainingsmode (dit verklaart meteen waarom het apparaat na het aanzetten even 'niets lijkt te doen'). Daar het aan en uitzetten van het apparaat moet gebeuren door het insteken en uittrekken van de stekker uit de wandcontactdoos kan dit over het algemeen, gezien de doelgroep, niet door de gebruiker zelf gebeuren. Er zal dan een tweede persoon aan te pas moeten komen, maar deze methode is ook alleen voor noodgevallen bedoeld.

Zodra men via één van de bovenstaande drie manieren in de trainingsmode is beland, zal de trainingsLED branden en begint het apparaat één voor één alle kanalen en functies waar een woord bij hoort aan te bieden aan de gebruiker.

Elk programmeerbaar (inspreekbaar) kanaal of functie zal nu éénmaal worden aangeboden aan de gebruiker voor training, te beginnen met de 16

normale kanalen, gevolgd door respectievelijk de trainingsfunctie, aan, uit, hoger, lager, alle lampen aan en alles uit. Dit gebeurt via de LED's. Van een woord dat wordt aangeboden, zal de LED knipperen met een frequentie van circa 2 Hz. De LED blijft circa 2 seconden knipperen (dit is eventueel nog instelbaar). Wordt er binnen die 2 seconden niet gereageerd door de gebruiker, dan zal worden doorgeswitcht naar het volgende kanaal of functie, totdat alle LED's geweest zijn. Op dat moment wordt de trainingsmode verlaten en keert het systeem terug in de herkenningmode, mits er natuurlijk minstens één woord in het geheugen staat. Onder 'reageren' wordt overigens verstaan, dat men even wat geluid produceert, zodat ter plaatse van de microfoon een geluidsenergie-stijging ontstaat, die minimaal 10 dB is. Reageert de gebruiker inderdaad op die manier binnen 2 seconden, dan stopt de LED met knipperen en blijft constant branden. De bevestigingsLED gaat eveneens branden, ten teken dat het systeem nu op een bevestiging wacht. De gebruiker heeft nu 10 seconden tijd om nogmaals wat van zich te laten horen. Is hij of zij 10 seconden stil, dan wordt met de volgende LED doorgeswitcht. Komt de bevestiging wel in 10 seconden, dan gaat de bevestigingsLED uit en wordt de opnameprocedure gestart. Op dit moment bevindt het apparaat zich in de stand, waarin de referentiewoorden kunnen worden aangemaakt. Wat men nu insprekt wordt later voor de herkenning gebruikt, en het is dus van het grootste belang, dat er duidelijk en goed wordt gesproken. Verder is het noodzakelijk, dat er geen achtergrondlawaai aanwezig is.

De opnameprocedure dient zowel om woorden op te nemen als om woorden te wissen. Het verdient aanbeveling om woorden, die niet meer gebruikt worden te wissen, omdat de spraakherkenning daardoor sneller en betrouwbaarder wordt. Het opnemen van een woord gaat als volgt:

- Men wacht tot de spreekLED gaat branden.
- Nu moet binnen 10 seconden het woord worden ingesproken.
- Nadat het woord is ingesproken moet de spreekLED even uitgaan (circa een halve seconde). Indien dat niet gebeurt is het woord te zacht ingesproken, en moet het worden overgedaan.
- Het bovenstaande wordt nogmaals herhaald. Men moet een woord dus tweemaal inspreken. Het systeem 'middelt' deze twee woorden en maakt daar een referentiewoord van.
- Als het woord tweemaal is ingesproken wordt de opnameprocedure verlaten en gaat het systeem verder met het aanbieden van de volgende LED.

Een bestaand woord kan ook gewist worden. In feite gaat dit op dezelfde

Een spraakherkenner voor het Busch Timac X-10 afstandsbedieningssysteem

manier als hierboven beschreven voor opnemen, alleen moet men nu in de daarvoor bestemde 10 seconden géén woord inspreken, maar volkomen stil zijn. Bovendien hoeft dit niet nog een tweede maal te worden herhaald, zoals bij een echt woord. Als de microfoon tijdens de eerste 10 seconden per ongeluk toch geluid opvangt (bijvoorbeeld een voorbijrijdende auto), dan 'denkt' de processor dat het een woord is en zal men dat 'woord' nog een tweede maal moeten inspreken. Door tijdens die tweede opname alsnog 10 seconden stil te zijn, zal het woord toch gewist worden, ondanks het per ongeluk opgevangen achtergrondlawaai.

Zoals reeds eerder vermeld, zal in de trainingsmode de trainingLED oplichten. Dit kan misschien tot enige verwarring aanleiding geven als de trainingsfunctie wordt aangeboden om te worden getraind. Deze LED zal dan knipperen, net zoals alle andere LED's wanneer die worden aangeboden. Het enigste moment in de trainingsmode, waarop de trainingsLED dus niet constant brandt, is wanneer de trainingsfunctie zélf wordt aangeboden.

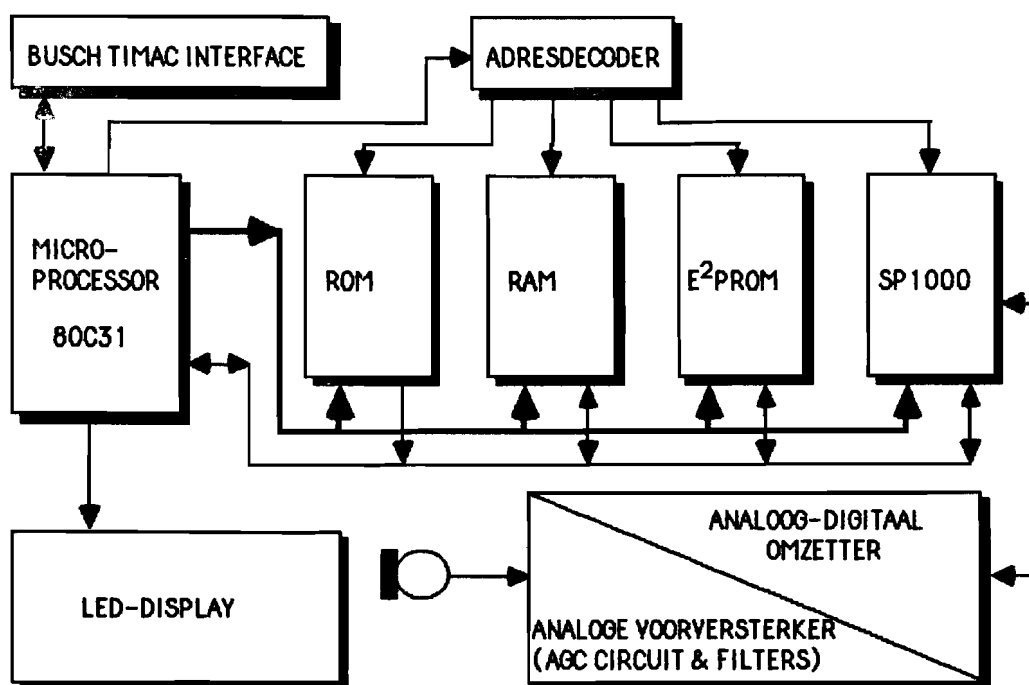
Een praktische aanwijzing voor het trainen is de volgende: wanneer men bij het aanbieden van kanalen en functies in de trainingsmode een woord wil kiezen (als de bijbehorende LED aan de beurt is), laat de LED dan eerst enkele malen knipperen (3 à 4 keer) alvorens de keuze te maken (door iets te zeggen). Het apparaat reageert dan veel beter. Men heeft tijd genoeg om te reageren (elke LED blijft 2 seconden knipperen, en die tijd is eventueel nog te vergroten). Iets dergelijks geldt voor het inspreken van woorden. Als de spreekLED gaat branden heeft men 10 seconden de tijd om te spreken. Neem die tijd ook en spreek het woord goed in. Stiltes voor het woord worden vanzelf door de microprocessor weggehaald, maar de processor is beter in staat om de woordgrenzen te bepalen als men vóór het woord even stil is (minimaal 0.5 seconden).

Met deze praktische tip zijn we aan het einde van de bespreking van het hele systeem gekomen qua opzet en bedieningsmogelijkheden.

HOOFDSTUK 5

DE HARDWARE

Zoals reeds eerder vermeld bestaat de elektronica uit twee grote delen, namelijk het analoge ingangscircuit en het digitale besturingsgedeelte. De interface tussen deze twee delen wordt gevormd door de SP1000, de spraak naar LPC omzetter. In de onderstaande figuur (5.1) is het blokschema van het totale systeem weergegeven. De hierin aangegeven blokken komen overeen met de circuits in het gedetailleerde schema in bijlage 2 (qua plaatsing). In dit hoofdstuk zullen deze blokken apart worden besproken en zal een motivering worden gegeven voor de keuze van de componenten.



Figuur 5.1. Blokschema van het spraakherkende systeem.

§5.1 Het analoge ingangscircuit

De taken van het circuit zijn als volgt te omschrijven:

- Het versterken van het microfoonsignaal tot een voor de analog naar

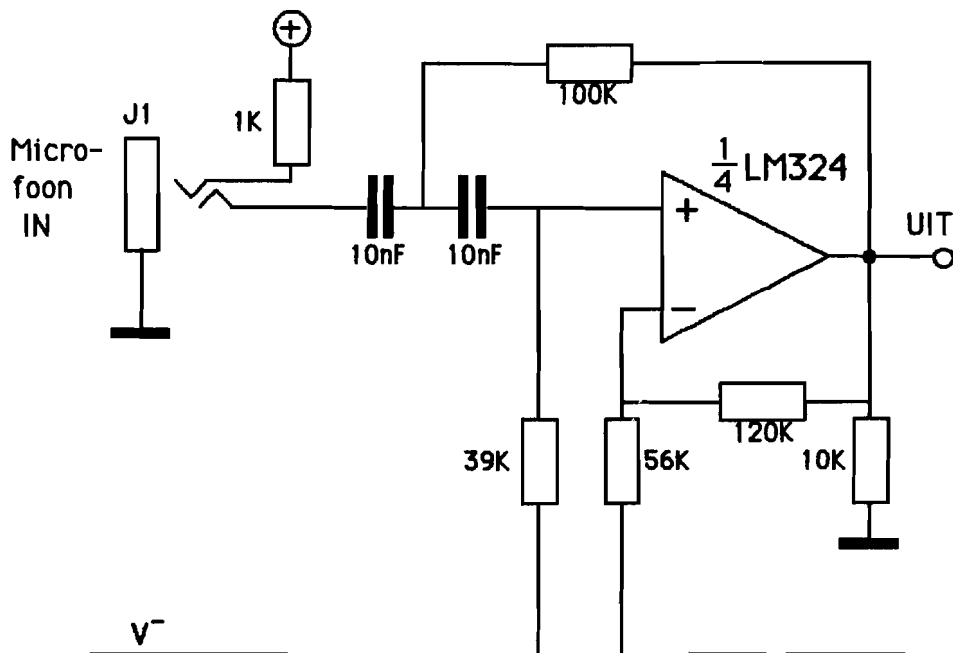
digitaal omzetter verwerkbaar niveau.

- Het uitvoeren van de automatische versterkingsregeling, onder besturing van de SP1000.
- Het bandbegrenzen van het ingangssignaal tot het voor spraakherkenning spectraal interessante gebied. Dit functioneert tevens als anti-aliasing filter. (In de volgende trap wordt het analoge signaal bemonsterd).

De eerste trap van het ingangscircuit (zie figuur 5.2) is een actief hoogdoorlaatfilter van orde 2 (12 dB/octaaf). Uit de componentenwaarden blijkt, dat het hier om een Butterworth filter gaat met als kantelfrequentie $f_0 = 255 \text{ Hz}$, en versterkingsfactor $K = 3.14$ (zie lit[1]). De belangrijkste functie van deze trap is het onderdrukken van laagfrequente stoorsignalen zoals brom, trillingen veroorzaakt door langsrijdende auto's, etc.

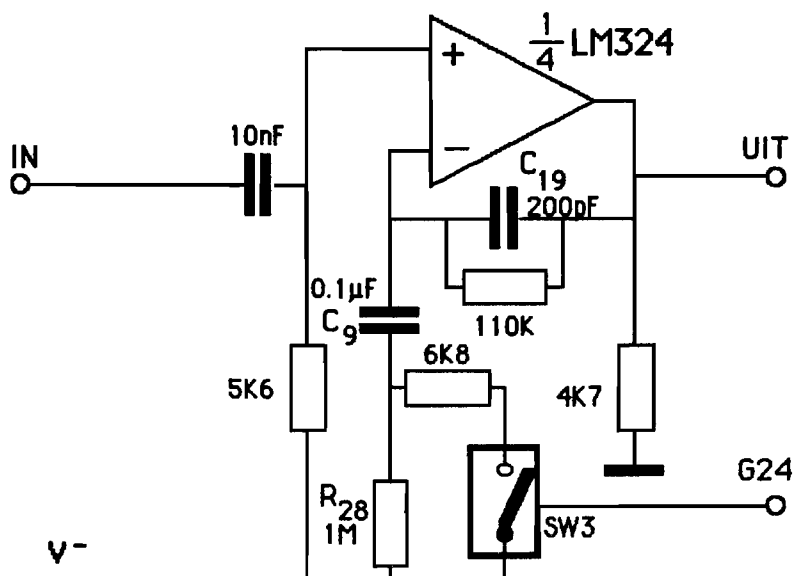
Wat opvalt aan de OPAMP schakeling is een weerstand van 10 k Ω tussen de uitgang en massa. Deze is volgens de fabrikant van de LM324 (lit [3]) nodig om bij capacitieve belasting van de OPAMP een bias-stroom in te stellen. Deze bias dient om cross-over vervorming in de uitgangstrap van de OPAMP te voorkomen.

De lijn V^- staat op een potentiaal van 2.5 volt (halve voeding). Deze dient om het werkpunt van de OPAMP schakeling in te stellen. Hierop wordt later in dit hoofdstuk nog ingegaan.



Figuur 5.2. Ingangstrap van de analoge voorversterker.

Na de ingangstrap volgt een enkelvoudig RC netwerk met eveneens een hoogdoorlaatwerking. De kantelfrequentie hiervan is $f_0 = 2,8 \text{ kHz}$. Dit netwerk dient als een pre-emphasis om de voor spraak belangrijke frequenties rond 3 kHz extra te 'versterken' ten opzichte van de lagere frequenties.



Figuur 5.3. De tweede trap van de voorversterker is een AGC circuit.

De tweede actieve trap (zie figuur 5.3) is een deel van de automatische volumeregeling. De elektronische schakelaar wordt bestuurd door de G24 lijn van de SP1000, hetgeen betekent dat het verschil in versterking tussen het open en gesloten zijn van deze schakelaar 24 dB ofwel een factor 16 moet zijn. Als de schakelaar open staat, geldt voor de versterking $k=1.01$. Als de schakelaar gesloten is wordt dat $k=17.1$

Door de aanwezigheid van de condensatoren zullen deze waarden enigszins frequentie-afhankelijk zijn, maar gezien de kantelfrequenties van de RC netwerken zal deze afhankelijkheid minimaal zijn. Alleen C_{19} heeft een hoge kantelfrequentie tot gevolg en maakt van de hele trap een hoogdoorlaatfilter met een kantelfrequentie $f_0 = 8 \text{ kHz}$. Deze condensator is nodig voor vergroting van de stabiliteit en heeft, gezien de hoge kantelfrequentie, geen invloed op het spectraal interessante gebied. Condensator C_9 dient alleen om te voorkomen dat er een DC verandering optreedt bij het openen of sluiten van de schakelaar. De functie van R_{28} is ervoor te zorgen dat als de schakelaar open staat, condensator C_9 aan die kant toch op massa-potentiaal blijft. Bij afwezigheid van deze weerstand

zou C_9 zich ontladen over zijn interne lekweerstand en dat zou bij het sluiten van de schakelaar een spanningspiek op de uitgang van het circuit teweeg kunnen brengen.

De derde trap (zie figuur 5.4) wordt voorafgegaan door een passieve omschakelbare verzwakker. De verzwakking is instelbaar op 0 dB of 6 dB. Aangezien er ook een koppelcondensator aanwezig is, gelden deze waarden alleen voor frequenties die ver boven het 3dB punt van het RC netwerk liggen. Deze kantelfrequentie ligt echter erg laag (≈ 16 Hz), zodat dit filter geen invloed heeft op de belangrijke frequenties.

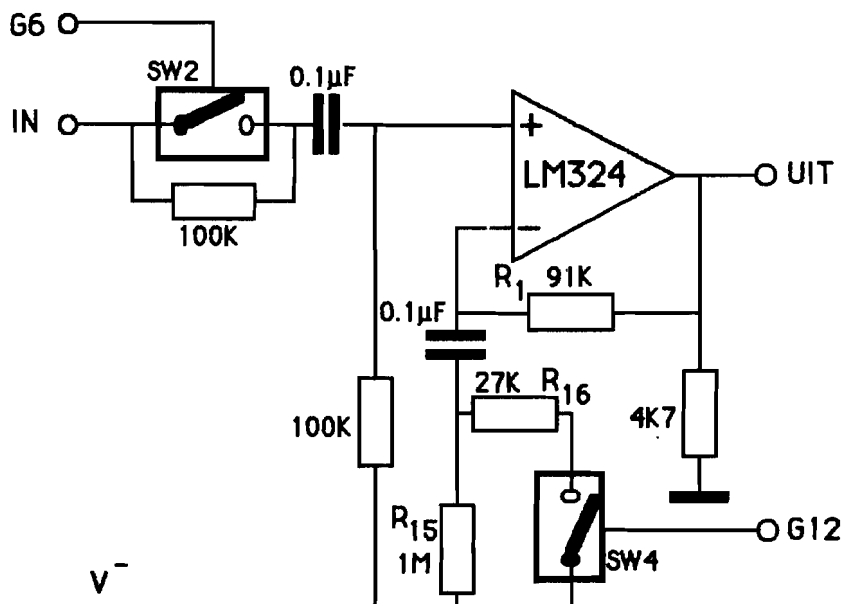
De derde trap bestaat uit een AGC circuit, waarvan de versterking instelbaar is op twee waarden, die 12 dB uit elkaar liggen. Als de schakelaar open is, is de versterking:

$$K = \frac{R_1 + R_{15}}{R_{15}} = \frac{91 + 1000}{1000} = 1,1$$

Wanneer de schakelaar gesloten is, wordt dit:

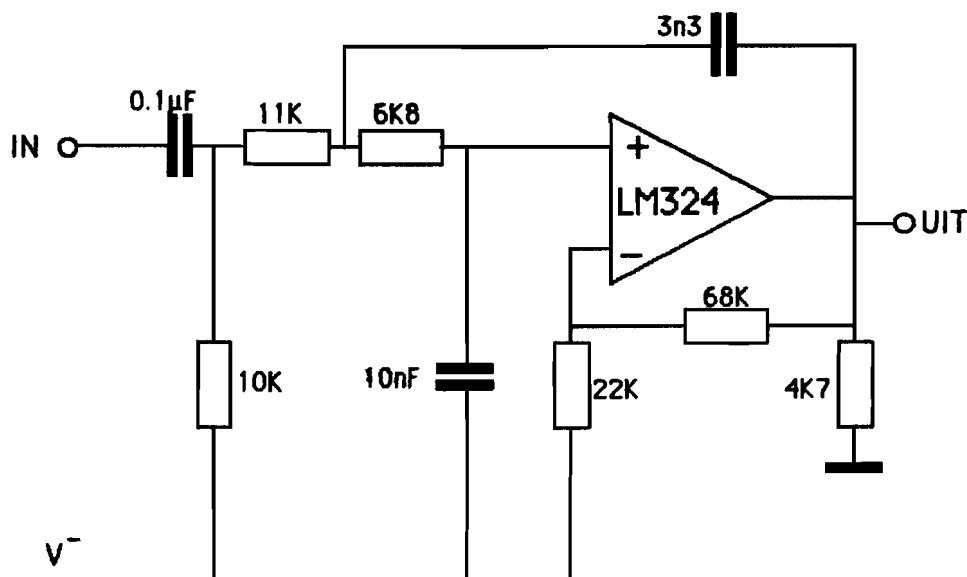
$$K = \frac{R_1 + \frac{R_{15} \cdot R_{16}}{R_{15} + R_{16}}}{\frac{R_{15} \cdot R_{16}}{R_{15} + R_{16}}} \approx \frac{R_1 + R_{16}}{R_{16}} = \frac{91 + 27}{27} = 4,4$$

Inderdaad is het verschil in versterking een factor 4, ofwel 12 dB.



Figuur 5.4. De derde trap van het analoge ingangscircuit.

De laatste filtertrap (zie figuur 5.5) is via een eerste orde hoogdoorlaatfilter met kantelfrequentie $f_0 = 160$ Hz gekoppeld aan de vorige. Het betreft hier weer een simpel RC netwerk. De laatste trap is het enigste laagdoorlaatfilter in de hele analoge ingangsketen. Het is een tweede orde Butterworth filter dat als anti-aliasing filter fungeert.



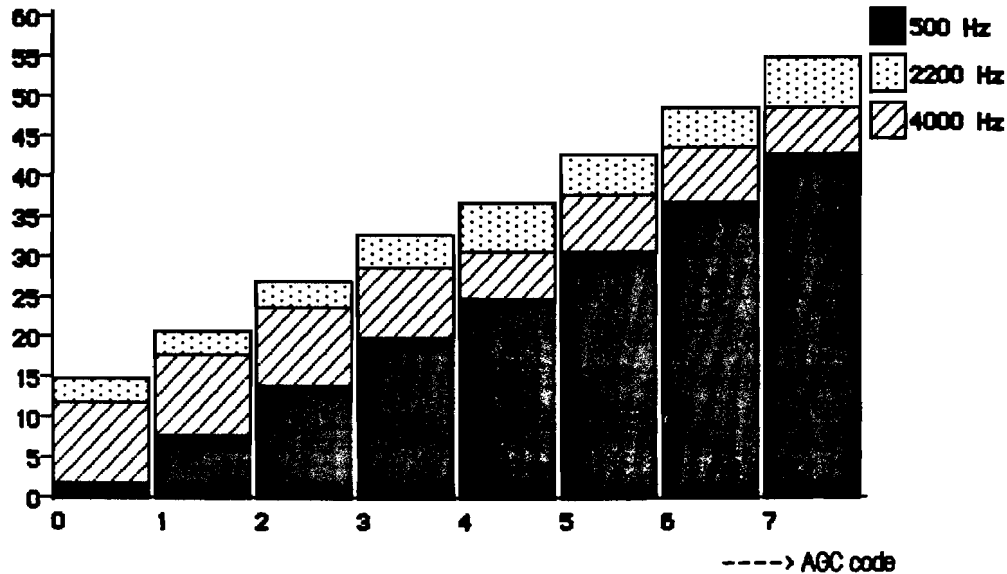
Figuur 5.5. De laatste trap is een laagdoorlaatfilter van orde 2.

Aangezien het analoge uitgangssignaal na de laatste trap bemonsterd moet worden, moet het spectrum ook begrensd worden. Met de gegeven componentwaarden heeft het filter een 3 dB punt bij $f_0 = 3,2$ kHz. Voor de gebruikte samplefrequentie (6215 Hz) is dit eigenlijk een te hoge waarde.

Het hele analoge circuit is opgebouwd rond 4 OPAMP's van het type LM324 (deze zitten met 4 in een IC). Deze OPAMP heeft als voordeel, dat hij al bij heel lage voedingsspanningen goed werkt en een vrij groot uitstuurbereik heeft (van 1.5V boven de negatieve voeding tot praktisch de positieve voeding). Om bij gebruik van een enkele voedingsspanning toch een goed instelpunt te krijgen wordt een kunstmatige 'middelste voedingslijn' gemaakt via een weerstandsdeler en een buffercondensator. Omdat de instelling hiervan vrij kritisch bleek is deze met een instelpotmeter uitgevoerd. In figuur 5.6 is de versterkingsfactor van de analoge ingangstrap weergegeven als functie van de AGC instelling voor drie verschillende frequenties. In bijlage 5 is nog een tabel opgenomen met de versterkingsfactoren voor drie verschillende frequenties als functie van de drie stuurbits G66, G12 en G24. Hieraan is te zien dat vooral bij de hogere

frequenties een afwijking gaat optreden ten opzichte van de ideale kromme, maar dat over het algemeen de meetwaarden minder dan 1.5 dB van de ideale karakteristiek afliggen.

versterkingsfactor (dB)



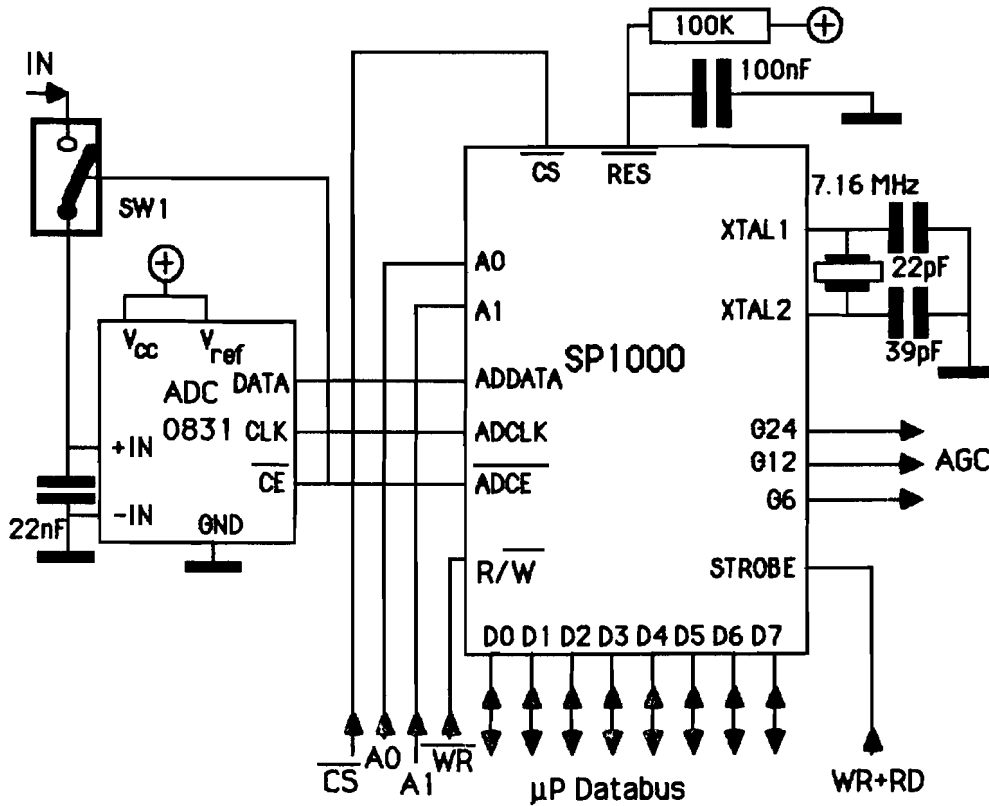
Figuur 5.6. Versterkingsfactor als functie van de AGC instelling bij drie frequenties.

§5.2 De interface tussen analoog en digitaal

Nadat het signaal de analoge filtertrappen is gepasseerd wordt het bemonsterd met een analoge schakelaar, gevolgd door een holdcondensator. De waarde die voor deze condensator gekozen moet worden is afhankelijk van de samplefrequentie, de tijdsduur dat de schakelaar gesloten is, en de belasting (ontlaadstroom). In deze schakeling zijn al deze grootheden constant, zodat ook de condensator een vaste waarde kan krijgen. In de praktijk bleek 22 nF een waarde te zijn die goed voldoet (Bij een samplefrequentie van 6215 Hz).

Over de condensator ontstaat nu een nulde-orde-hold bemonsterde versie van het analoge uitgangssignaal van de voorversterker (zie figuur 5.7). Dit signaal wordt nu toegevoerd aan de verschil-ingang van een seriële 8 bit analoog naar digitaal omzetter (ADC). Deze ADC heeft een instelbare referentiespanningsingang, waarmee de gevoeligheid kan worden ingesteld (zie lit[4]). Als de referentiespanning V_{ref} bedraagt, dan zal een ingangsspanningsvariatie van $0 \dots V_{ref}$ een uitgangscodewariatie opleveren van $0 \dots 255$. Aangezien de analoge voorversterker een maximaal uitgangsniveau heeft van circa 5 volt, moet ook de referentiespanning deze waarde hebben.

Het is belangrijk, dat deze vrij van ruis, brom en andere stoorsignalen is, daar elke variatie op deze ingang direct doorwerkt op de uitgangscodes.



Figuur 5.7. Analooq naar LPC data omzetter.

De ADC heeft een seriële uitgang en moet via een tweetal stuurlijnen worden aangestuurd. Het betreft hier de signalen \overline{CE} , waarmee een nieuwe conversie gestart kan worden, en CLK waarmee de bits een voor een naar buiten kunnen worden geschoven. De databits verschijnen dan op de D uitgang. Al deze stuursignalen voor de ADC worden opgewekt door de SP1000 (zie lit[5]). De SP1000 zal met de ingestelde samplefrequentie het \overline{CE} signaal van de ADC en de bemonsteringsschakelaar even activeren, en kort daarna de naar digitaal omgezette spanningswaarde inklokken en verwerken.

§5.3 Het digitale deel van de spraakherkenner.

Het digitale deel is opgebouwd rond een microprocessor van het type 80C31, een RAM (werkgeheugen) en een EPROM (programmegeheugen). De

microprocessor heeft een groot aantal taken te vervullen. Deze zijn in grote lijnen als volgt te omschrijven:

- Het verzamelen van LPC data van de SP1000 en het verwerken ervan.
- Het inlezen van (seriële) data van het custom-IC in de Busch Timac hoofdpot en het verwerken daarvan.
- Het weergeven van de toestand van het afstandsbedieningssysteem op een LEDdisplay, zodat de gebruiker kan zien welke kanalen en functies op een gegeven moment geactiveerd zijn.
- Het genereren van een seriële code die (nadat hij op een 120 kHz golf is gemoduleerd) via het net naar de afstandsschakelaars en -dimmers van het systeem wordt getransporteerd om deze te besturen.

Deze taken moeten door de processor worden verricht. Bij de keuze van de processor is het daarom logisch na te gaan in hoeverre deze in staat is om al deze taken ook echt uit te voeren, en hoeveel externe hardware daar dan eventueel nog voor nodig is. Uit de genoemde taken kunnen we een aantal eisen voor de processor afleiden:

Vooraf het inlezen en genereren van de seriële codes is een taak die alleen door een relatief **snelle microprocessor** gedaan kan worden.

Het LEDdisplay bestaat uit 8 rijen en 4 kolommen. Deze 4 kolommen worden één voor één gedurende korte tijd van spanning voorzien, waarbij de spanningsniveau's die op dat moment op de rijen aanwezig zijn bepalen welke LED's van zo'n kolom oplichten (zie hiervoor de beschrijving van het LEDdisplay verderop in dit hoofdstuk en figuur 5.8). De display software moet dus regelmatig (100.....1000 x per seconde) de volgende kolom van het display opzetten. De exacte frequentie is niet echt van belang, zolang die maar stabiel is (anders varieert de display helderheid). Het is daarvoor dus nodig dat er een timer in het systeem aanwezig is, om een interrupt te genereren als de tijd gekomen is om de volgende LEDkolom aan te sturen. Voor het aansturen van het display, de interfacelijnen naar de Busch Timac X-10 netbesturingspost e.d. zijn **een flink aantal I/O pinnen** nodig op de processor. Dit beïnvloedt mede de keuze van de processor.

Voor de geheugen-groottes waren bij aanvang van het project de volgende verwachtingen opgesteld:

- 4k.....8k RAM, als werkgeheugen voor de processor
- 2k.....4k EPROM, voor de opslag van min. 25 referentiewoorden
- 4k.....8k EPROM, als programma-geheugen
- 2 bytes voor adressering van de SP1000

De processor moet dus **minimaal 20 KByte kunnen adresseren.**

Als we al deze eisen combineren, kunnen we concluderen dat de eenvoudige controllers, zoals de 8039 niet bruikbaar zijn, en dat ook de gewone processoren, zoals de 6502 niet voldoende mogelijkheden hebben om hier als systeempromotor te worden gebruikt. De 80C31 van Intel is echter een microcontroller die genoeg mogelijkheden en snelheid biedt om hier te worden ingezet. De klokfrequentie kan tot maximaal 12 MHz worden opgevoerd, waarbij de processor snel genoeg is om alle real-time processen goed af te werken. Verder bevat deze processor een tweetal multimode timers/counters van elk 16 bit die met een maximale klokfrequentie van 1 MHz geklokt kunnen worden. Ook de aansturing van de externe interface-circuits is geen probleem, dankzij de vele vrije I/O lijnen op deze processor.

Als programmeergeheugen is achteraf een 4 kByte EPROM nog net voldoende gebleken, maar om eventuele toekomstige uitbreidingen nog mogelijk te maken kan er beter een 8 kByte EPROM worden gebruikt. Voor het prototype is een 27C256 gebruikt (dat is een 32 kByte EPROM).

De spraakherkenningsalgoritmes vergen momenteel ca. 1.5 kByte aan tijdelijke data-opslagruimte. Het zou dus voldoende zijn om een 2 kByte RAM geheugen in het circuit op te nemen, maar om tijdens de ontwikkelingsfase problemen met te weinig RAM te voorkomen is voor het prototype uitgegaan van een 8 kByte IC.

Verder bevat het circuit nog een E²PROM (Elektrically Erasable Programmable Read Only Memory) voor de opslag van referentiewoorden. Elk referentiewoord bestaat uit 108 bytes (zie hiervoor hoofdstuk 6, de software). Totaal zijn er -voorlopig althans- 25 woorden die getraind kunnen worden. Samen met enkele belangrijke systeemvariabelen kost de opslag hiervan zo'n 2700 bytes. E²PROM's zijn veel duurder dan bijvoorbeeld RAM's of gewone EPROM's van dezelfde grootte. Een veel gebruikt en goed verkrijgbaar type E²PROM is de 2816 van Intel. Deze is echter slechts 2 kByte, maar heeft al een opvolger van 8 kByte (2864), die hier is gebruikt voor de opslag van referentiewoorden. Binnenkort zal deze ook in CMOS uitvoering leverbaar zijn, waardoor het stroomverbruik van de hele schakeling aanzienlijk zal afnemen. Elk byte in de 2864 kan 10000 keer worden overschreven en onbepaald worden uitgelezen. De toegangstijd voor schrijven is slechts 2 msec., waarmee deze E²PROM tot de snelste behoort. Er dient op te worden gelet dat voor deze E²PROM de Intel versie wordt gebruikt en niet een andere, omdat uit de diverse fabrikantgegevens is gebleken, dat deze onderling niet uitwisselbaar zijn (dus E²PROM's met hetzelfde typenummer, maar van verschillende fabrikanten zijn niet

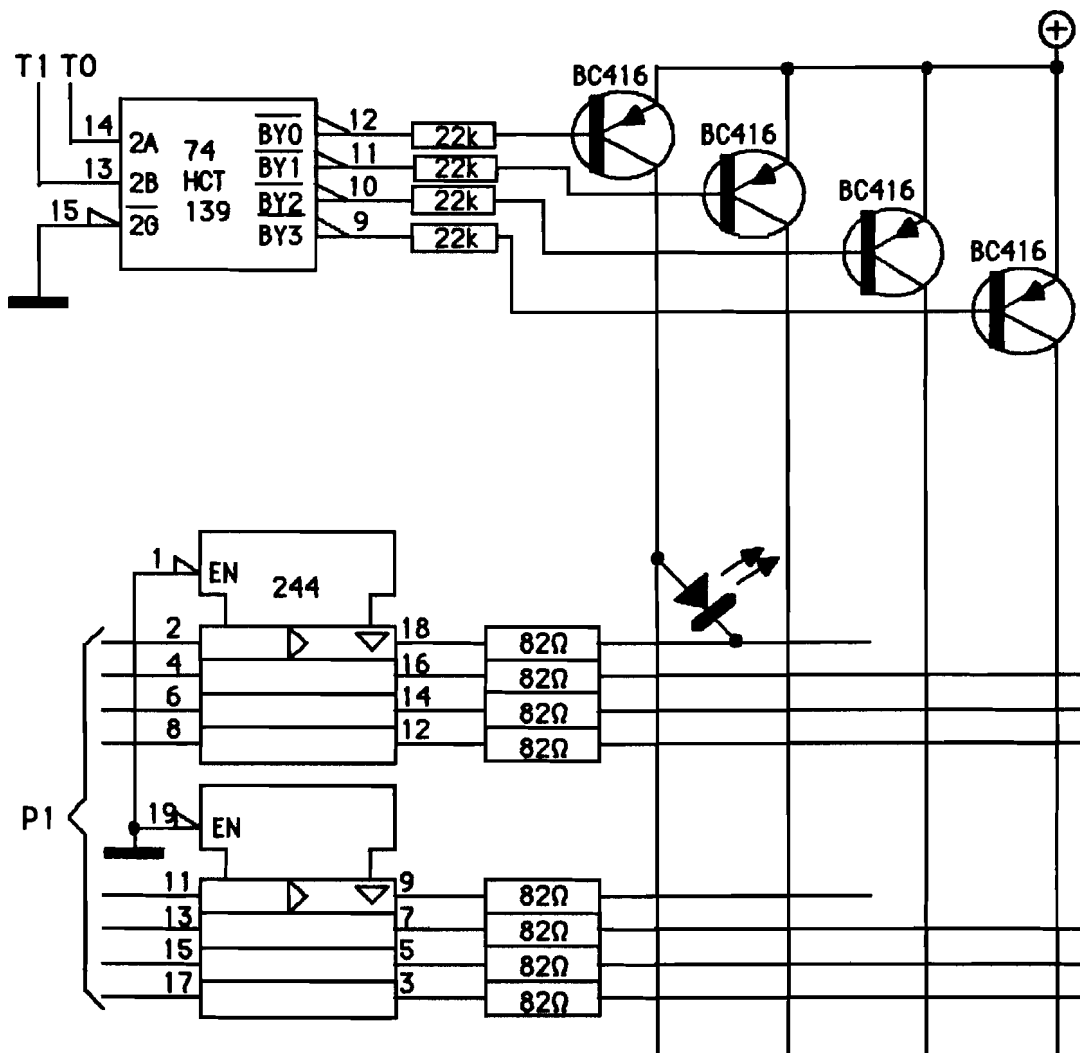
hetzelfde I). Het verschil is gelegen in de manier waarop de microprocessor kan detecteren of de E²PROM al klaar is met een interne schrijfcyclus. Deze duurt, zoals al vermeld, ca. 2 msec.. Gedurende die tijd kan de E²PROM niet worden gebruikt voor het lezen of schrijven van andere adressen. De microprocessor moet na het geven van een schrijfo opdracht aan de E²PROM dus wachten tot deze klaar is. Sommige E²PROM's hebben hiervoor op pin 1 een speciaal READY signaal staan, dat de processor kan uitlezen. Andere hebben helemaal geen terugmeldingssignaal. Bij deze typen moet de processor gewoon 2...10 msec. wachten en hopen dat dan alles goed gegaan is. Bij de Intel versie is echter weer een ander systeem bedacht, namelijk dat van de "Intelligent polling". Nadat de processor een schrijfcommando heeft gegeven, kan het einde van een schrijfcyclus gedetecteerd worden door de E²PROM uit te lezen. Zolang de E²PROM namelijk nog bezig is zal die bij een leesopdracht hetzelfde byte naar buiten sturen, dat hij op dat moment intern aan het wegschrijven is, maar met bit 7 geïnverteerd. Is de E²PROM klaar, dan leest de processor natuurlijk gewoon het weggeschreven byte terug. Door dus steeds de E²PROM uit te lezen en te testen of bit 7 van het ingelezen byte wel of niet geïnverteerd is t.o.v. het weggeschreven byte, kan worden nagegaan of de E²PROM nog met een interne schrijfcyclus bezig is.

Voor de uitlezing is een display aanwezig van 25 LED's. Deze LED's worden gemultiplexed aangestuurd in een matrix van 4 kolommen en 8 rijen (er zijn dus 7 plaatsen in de matrix onbezet). Voor dit principe van displaysturing is gekozen, omdat dat in zowel de hardware als het stroomverbruik van de schakeling een besparing oplevert. Voor de aansturing van 25 LED's zouden anders 3 of 4 registers nodig zijn, terwijl nu maar één buffer-IC gebruikt wordt. De stroombesparing treedt op omdat er nu maximaal 8 LED's tegelijk aan staan, in plaats van 25. De prijs die daarvoor betaald moet worden is een kleinere lichtopbrengst (elke LED staat maar een gedeelte van de tijd aan), maar indien gebruik wordt gemaakt van kleine rode high-efficiency LED's is de lichtopbrengst ruim voldoende.

Voor de buffer, die de rijen van de matrix aanstuurt is een 74HCT244 genomen. Dat is een niet-inverterende buffer. De reden hiervoor is, dat de microprocessor bij een reset (dus ook bij het inschakelen) alle uitgangslijnen hoog maakt. Door nu de rijen via een niet inverterende buffer aan te sturen zullen de LED's hierdoor tijdens een reset uit gaan, waarmee tevens een hinderlijk inschakelverschijnsel wordt voorkomen. De kolommen van de matrix worden ook aangestuurd door de processor. Voor 4 kolommen zijn 2 stuurbits nodig, die met een 2 naar 4 lijnen decoder (74HCT139)

Een spraakherkenner voor het Busch Timac X-10 afstandsbedieningssysteem

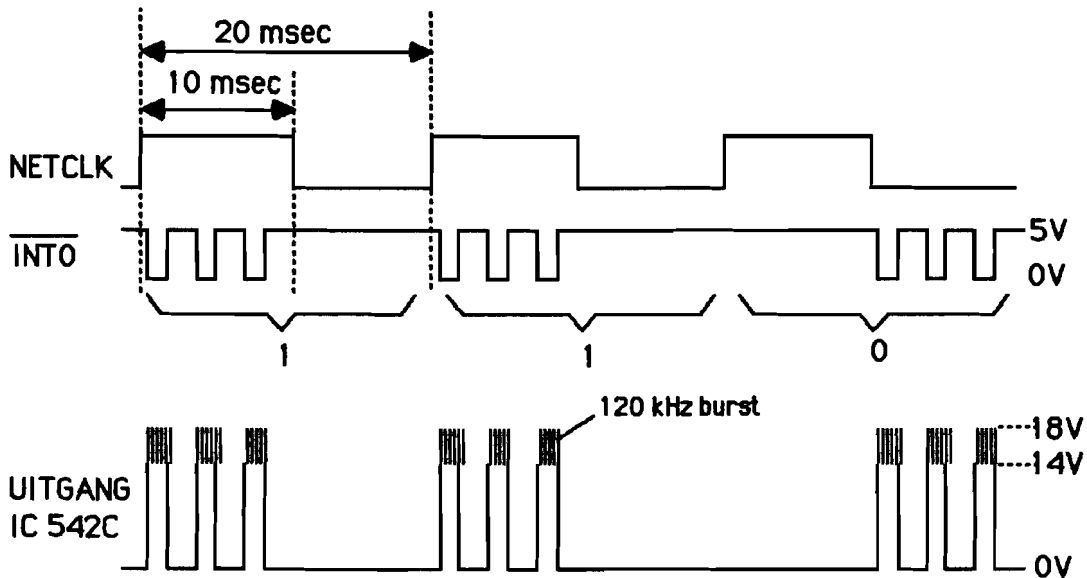
worden omgezet naar een van de 4 kolommen. Daar een kolom 8 LED's kan bevatten, die elk 5...15 mA opnemen is het wel duidelijk dat de decoder deze stroom niet kan leveren. Vandaar dat de kolommen via een transistor worden aangestuurd. Met de lijnen T0 en T1 kan de processor een van de 4 kolommen kiezen. De gekozen kolom wordt voorzien van 5 volt, waarna via poort P1 de 8 LED's van die kolom kunnen worden bestuurd. Een nul op poort P1 betekent dan dat de corresponderende LED oplicht. Om de stroom door de LED's te begrenzen zijn in serie met het buffer-IC nog een achttal kleine weerstanden van $82\ \Omega$ opgenomen.



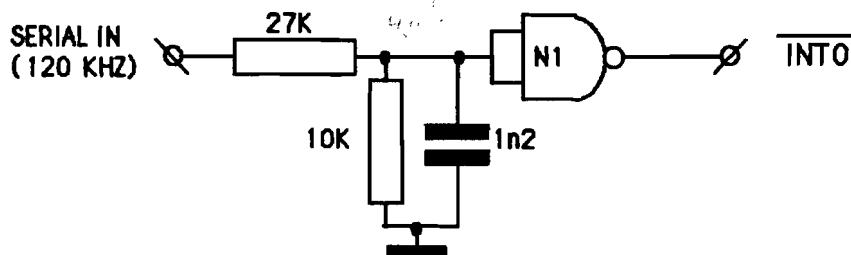
Figuur 5.8. Het LEDdisplay.

Verder bevat het digitale deel nog enkele interfacecircuits naar de bestaande Busch Timac X-10 netbesturingspost. In deze laatste bevindt zich een IC dat ultrasone signalen kan ontvangen van de handzender en het

toetsenbord aftast, en dat bij detectie van een ingegeven commando een 120 kHz code genereert, die via een uitgangsversterkertrap over het lichtnet wordt verstuurd naar de afstandsschakelaars, -dimmers, etc. Op dit punt vindt de koppeling tussen spraakherkenner en afstandsbedieningssysteem plaats. Daartoe wordt de genoemde signaaluitgang van het IC losgehaald van de 120 kHz versterkertrap en naar de processorkaart gevoerd. De ingang van de 120 kHz versterkertrap wordt doorverbonden met de signaaluitgang van de processorkaart.



Figuur 5.9. De signaalformen van en naar de netbesturingspost.

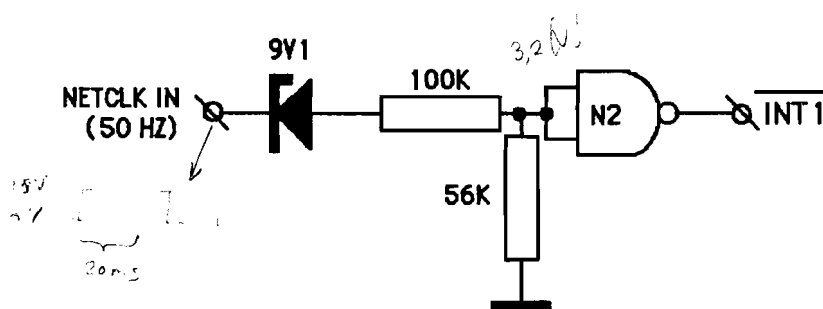


Figuur 5.10. Conversie van 18V, 120kHz naar TTL niveau zonder 120 kHz component.

Het door de netbesturingspost opgewekte 120 kHz codesignaal wordt dus aan de processorkaart toegevoerd. Dit signaal bevat overigens ook een laagfrequent component (zie figuur 5.9 voor de optredende golfvormen). Op de processorkaart wordt het signaal eerst met behulp van een verzwakkend laagdoorlaatfilter van de 120 kHz component ontdaan en teruggebracht tot het voor TTL ongevaarlijk spanningsniveau van 5V (zie figuur 5.10). Via een

NAND poort wordt het signaal gebufferd en geïnverteerd en zo geschikt gemaakt voor de actief lage interruptieingang van de processor. Die laatste kan nu via interrupts het begin van een 120kHz signaal detecteren en zo commando's ontvangen die van het toetsenbord of de ultrasone handzender afkomstig zijn.

Het genereren van een seriële 120 kHz code voor verzending over het lichtnet is iets moeilijker. Het is namelijk van groot belang, dat de codes precies op het goede moment worden verzonden, namelijk vlak na de nuldoorgang van elke netperiode. Elk bit moet zelfs drie maal worden verzonden, op gelijk verdeelde (tijds-)afstanden binnen een periode. (zie figuur 5.9). Dit is nodig om ook apparaten die op een andere fasegroep zijn aangesloten te kunnen besturen (via een fasenkoppelaar). De microprocessor moet dus kunnen nagaan wanneer er een nuldoorgang optreedt in het netsignaal. Nu zit er al een nuldoorgangsdetector in de netbesturingspost, omdat die ook dezelfde codes moet kunnen opwekken (synchroon met het net). Dat signaal kan nu ook door de processor gebruikt worden. Daartoe moet het wel eerst worden omgezet naar een voor de processor verwerkbaar signaalvorm. Het enige signaal uit de netpost dat hiervoor bruikbaar is, heeft een waarde van $18 V_{tt}$. Dit moet dus eerst worden omgezet naar TTL niveau (figuur 5.11). Via een 9,1V zener en een weerstandsdeler wordt het signaal omgezet naar circa $4V_{tt}$ met een gemiddelde waarde van 2V. Via een NAND wordt het zo ontstane signaal gebufferd (dit dient tevens als bescherming van de processor tegen hoge netspanningspieken) en naar een interruptieingang gevoerd.

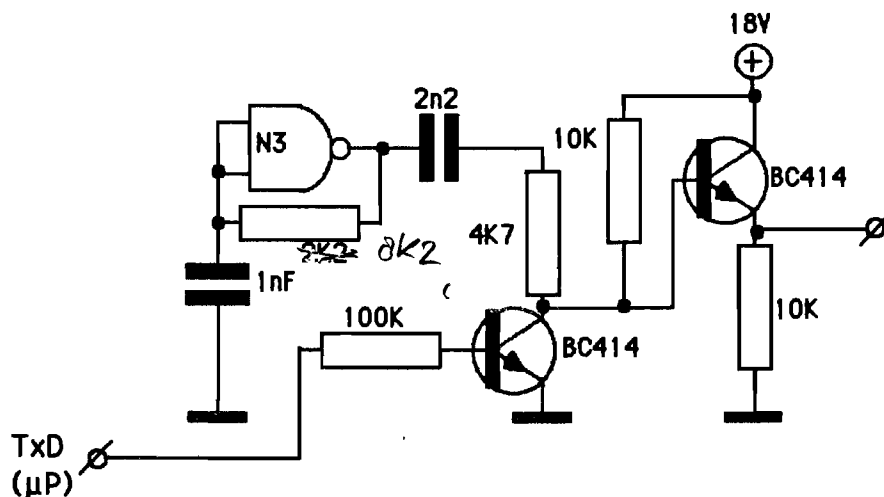


Figuur 5.11. Conversie van 18V netklok signaal naar TTL niveau.

Het genereren van een 120 kHz wisselspanning is voor een microcontroller iets teveel gevraagd. Vandaar dat er een aparte vrij lopende 120 kHz oscillator in het circuit is opgenomen. Deze is opgebouwd met een NAND Schmitt-trigger (zie figuur 5.12). Het uitgangssignaal van deze oscillator wordt via een koppelcondensator en een weerstandsdeler naar de +18V

Een spraakherkenner voor het Busch Timac X-10 afstandsbedieningssysteem

omgezet naar een niveau dat voor de 120 kHz versterker in de netbesturingspost voldoende is om uitgestuurd te worden. Omdat die versterker het signaal nogal belast en dat hier niet toelaatbaar is, wordt er nog een emittervolger opgenomen in de signaalweg. Deze levert genoeg vermogen om de 120 kHz uitgangstrap aan te sturen. Natuurlijk moet het 120 kHz oscillatorsignaal door de microprocessor aan- en uit kunnen worden gezet. Hiertoe kan de processor via een transistor het signaal op de basis van de emittervolger kortsluiten naar massa, waarbij tevens het uitgangssignaal van de signaalgenerator op massaniveau komt te liggen, waarmee we precies dezelfde golfvorm hebben opgewekt als de netbesturingspost.



Figuur 5.12. Conversie van TTL signaal naar gemoduleerd 120 kHz signaal op 18V niveau.

De 8031 heeft in principe twee gescheiden geheugenbereiken. Er kan een programmeergeheugen en een datageheugen worden aangesloten van elk 64 kByte (tegelijk). Bij het uitlezen van instructies uit het programmeergeheugen wordt het $\overline{\text{PSEN}}$ lijntje laag gemaakt, terwijl bij gewone lees- en schrijfcycli gebruik wordt gemaakt van de $\overline{\text{WR}}$ en $\overline{\text{RD}}$ signalen. Voor het programmeergeheugen is een 27C256 in gebruik, welke dus via het $\overline{\text{PSEN}}$ signaal wordt geselecteerd. Het dataRAM-adresbereik is opgedeeld in 8 stukken van elk 8 kBytes. Hiervan zijn er drie bezet door IC's die door de processor moeten kunnen worden aangesproken. Dit zijn de 62C64 (externe RAM), de 2864 (E2PROM) en natuurlijk de SP1000.

De memory map van het hele systeem wordt nu als volgt:

PROGRAMMA: 0000H 7FFFH = ROM (27C256)

Een spraakherkenner voor het Busch Timac X-10 afstandsbedieningssysteem

DATARAM:	0000H 1FFFFH	=	RAM (62C64)
	2000H 3FFFFH	=	Laadpuls voor schuifregister
	4000H 5FFFFH	=	E2PROM (2864)
	6000H 7FFFFH	=	SP1000
	8000H FFFFFH	=	Schuifklok voor schuifregister

De SP1000 heeft in feite maar 4 bytes nodig. Dit betekent dat de SP1000 om de vier bytes opnieuw terugkomt, dus dat adresseringen in het gebied 6000 6003 hetzelfde opleveren als bijvoorbeeld in het gebied 7438H 743BH. Het uitdecoderen van de adressen van de processor naar enable- signalen voor een van deze IC's gebeurt met een twee naar vier lijnen decoder, waarvan er nog een over was (de andere helft is gebruikt in de display sturelektronica).

Het laatste stuk van de hardware dat nog niet besproken is, is het schuifregister, waarmee de processor extra informatie kan inlezen, die nodig is voor de huiscode- en looptijdstelling. In hoofdstuk 4 is al ingegaan op de 4 bit huiscode, die ook over het net moet worden verstuurd. Daartoe moet de processor wel de mogelijkheid hebben, om de stand van de huiscodeschakelaar in te lezen. Ook is het gewenst om de snelheid, waarmee in de trainingsmode de functies achter elkaar worden aangeboden, in te stellen naar persoonlijke voorkeur (voorlopig via dipswitches). Dit is mogelijk gemaakt met behulp van een 8 bits schuifregister, waarmee deze instellingen seriëel kunnen worden ingelezen. In eerste instantie lijkt het seriëel inlezen van deze data een omslachtige methode, want het zou net zo goed parallel kunnen met een tri-state latch (er is een vrij stuk in het adresbereik). De seriële methode is echter gekozen op praktische gronden. Het hele circuit moet worden ingebouwd in de bestaande behuizing van de netbesturingspost en daarvoor zijn twee printjes ontworpen, waarop het hele circuit moet passen. Nu passen net niet alle digitale IC's op een printje, waardoor de hele databus alleen voor dit ene IC naar het tweede printje zou moeten worden doorverbonden. Op de seriële manier vergt dit echter slechts twee extra lijntjes. Bovendien worden de printen er aanzienlijk eenvoudiger door.

Hiermee is het hele circuit besproken. Op de hardware in de netpost zal ik niet ingaan, omdat een beschrijving daarvan in de literatuur gevonden kan worden (zie lit[6]).

HOOFDSTUK 6

DE SOFTWARE

In dit hoofdstuk zal een uitgebreide bespreking volgen van de software. Eerst zal een globaal overzicht worden gegeven van wat de software moet doen. Daarna zal worden ingegaan op de spraakherkenningsalgoritmes op een algemene manier (de principes). Hierna worden dan de flowdiagrammen van de software van de spraakherkenner besproken. Tenslotte komen dan nog de overige programmadelen aan de orde, zoals de gebruikersinterface (trainen, inlezen commando's, aansturing Busch Timac X-10 systeem, etc.).

§6.1 De oorsprong van de software.

Het uiteindelijke systeem bevat circa 4 kByte aan software. Deze kan worden opgesplitst in twee delen, namelijk de spraakherkennings-software enerzijds en de gebruikersinterface anderzijds. Beide delen zijn ongeveer 2 kByte. De software bepaalt uiteindelijk de functie van het apparaat. Vandaar dat bij de ontwikkeling van de software veel aandacht moet worden geschonken aan een overzichtelijke en gestructureerde opbouw, die nog voor uitbreiding geschikt is. Voor de ontwikkeling van de spraaksoftware is uitgegaan van het LIS'NER1000 systeem voor de APPLE computer. Dit systeem bestaat uit een stuk hardware (SP1000 en analoog Ingangscircuit) en een aantal programma's in BASIC en machinetaal. De BASIC programma's dienen voor de interface naar de gebruiker, terwijl de machinetaal (6502) programma's voor de spraakherkenning zorgdragen. De hierbij gebruikte algoritmes voor spraakherkenning zijn in eerste instantie overgenomen in de Busch Timac X-10 spraakherkenner. Reeds eerder is vermeld, dat we gebruik maken van een 80C31 microprocessor. Deze is echter zodanig anders van structuur en mogelijkheden, dat de software als het ware helemaal opnieuw geschreven moest worden, waarbij de programmatuur van de LIS'NER1000 als voorbeeld gebruikt werd. Het nieuwe programma heeft dan ook een aantal grote veranderingen ondergaan qua structuur, terwijl ook nog een aantal verbeteringen zijn aangebracht aan de algoritmes. Het oorspronkelijke herkenningsalgoritme op de APPLE was 4 kByte. Het nieuwe algoritme op de

80C31 is slechts 2 kByte en is bovendien sneller.

De rest van de software, namelijk de interface naar de gebruiker en het Busch Timac X-10 systeem bestond natuurlijk nog niet in de LIS'NER1000 software. Dit software pakket, ter grootte van 2 kByte moest ook worden ontwikkeld.

§6.2 De taken van de software.

De software moet de volgende taken verrichten:

- Het initialiseren van de systeem bouwstenen en datastructuren in de interne database van het systeem.
- Het laden van LPC parameters van de SP1000. Dit moet in het ideale geval continu gebeuren. Daar de processor ook nog andere taken heeft kan hij niet constant data inlezen van de SP1000.
- Het bepalen van woordgrenzen uit de ingelezen data, en hieruit dan weer detecteren of er al een heel woord is ingelezen en indien dat zo is dit woord doorspelen aan de herkenning algoritmes voor verdere verwerking.
- Het herkennen van woorden die van het woordgrensdetectie-algoritme afkomen met behulp van de in het geheugen opgeslagen referentiewoorden en het vervolgens al of niet accepteren van deze woorden.
- Het bij geaccepteerde woorden behorende commando uitvoeren en het LEDdisplay aanpassen aan de nieuwe situatie.
- Het aansturen van het LEDdisplay. Dit display wordt gemultiplext in 4 rijen en 8 kolommen aangestuurd (zie hoofdstuk 5). De software moet regelmatig de volgende kolom van het display aansturen. Deze taak moet altijd ongehinderd doorgaan, ongeacht wat er verder in het systeem gebeurt. De minste hapering of onregelmatigheid is direct te zien op het display in de vorm van knipperen of variërende helderheid van de LED's.
- Het inlezen van commando's die afkomstig zijn van de Busch Timac X-10 netbesturingspost. Deze scant continu het toetsenbord en kan ook ultrasone commando's ontvangen. Zodra een toets of ultrasoon commando wordt gedetecteerd, wordt dit door de ingebouwde micro-elektronica omgezet naar een 120 kHz signaal op het lichtnet. Dit signaal wordt nu echter alleen maar aan de processor toegevoerd, die het weer decodeert en aan de hand daarvan het LEDdisplay aanpast en eventueel het bijbehorende commando uitvoert.
- Het genereren van 120 kHz signalen op het lichtnet voor de aansturing van afstandsschakelaars en afstandsdimmers. De netbesturingspost, die deze taak eerst voor zijn rekening nam, kan nu zelf geen signalen meer op het

Wijst wel!

- lichtnet zetten, omdat zijn uitgangssignaal niet meer naar de uitgangsversterker wordt gevoerd, maar naar de processorkaart.
- Het trainen van woorden (het aanpassen van de interne database en het garanderen van de structuur daarvan). Hiertoe moeten de kanalen en functies die getraind kunnen worden aan de gebruiker worden aangeboden. De gebruiker moet een woord al of niet voor training kunnen selecteren, waarna het woord één of meerdere keren wordt ingesproken. Tenslotte wordt het -eventueel na middeling- opgeslagen in de database van referentiewoorden.

Een additioneel probleem is nog, dat het genereren van de 120 kHz signalen op het lichtnet simultaan moet kunnen gebeuren met de andere genoemde taken. De reden hiervoor is de volgende: elke boodschap die men over het net wil versturen moet minimaal twee maal worden verstuurd, omdat anders de afstandsdimmers en -schakelaars er niet op reageren. Eén bericht duurt 220 msec. (hier kom ik later in dit hoofdstuk nog op terug). Dat betekent dat het systeem steeds 440 msec bezig is met het versturen van boodschappen over het lichtnet. Als er via het toetsenbord of de ultrasone afstandsbediening een dimcommando wordt ingevoerd en de processor is bezig om dat over te zenden, dan is hij niet in staat om tegelijk na te gaan of die toets op een bepaald moment nog steeds is ingedrukt (dus of het dimcommando nog moet worden voortgezet). Hiervoor moet namelijk de seriële commando-ingang worden uitgelezen (via welke commando's van de netbesturingspost binnenkomen) en dat is een zéér tijdrovende aangelegenheid. Dit zou dan tot gevolg hebben, dat de processor beurtelings een bericht uitzendt over het lichtnet en dan weer naar de commando-ingang kijkt of er nog iets binnenkomt en indien dat zo is het bericht decodeert en uitvoert. Het vervelende hierbij is, dat de tijd tussen twee over het lichtnet verstuurd berichten te groot wordt, met als gevolg dat het dimmen heel onregelmatig en met haperingen verloopt. Bij motorsturingen voor bijvoorbeeld gordijnen wordt het nog veel erger: daar start en stopt de motor steeds, hetgeen de levensduur van de motor niet ten goede komt. Bij spraakcommando's treedt een soortgelijke situatie op. Als een dimcommando wordt gegeven kan dit worden gestopt door iets in te spreken, bijvoorbeeld het woord STOP. Alleen het maken van geluid is al voldoende. De processor moet dan natuurlijk wel controleren of de gebruiker het dim commando wil stoppen en moet dus de geluidsterkte meten. Dat kost tijd en als dat niet tegelijk met het genereren van de 120 kHz signalen kan, dan ontstaat dezelfde situatie als hiervoor beschreven en verlopen alle dimcommando's haperend. Vandaar, dat het absoluut

noodzakelijk is dat het genereren van 120 kHz signalen op het lichtnet als een achtergrondproces wordt uitgevoerd, simultaan met andere processor-operaties.

§6.3 De indeling van het geheugen.

De interne database met referentiewoorden voldoet aan een bepaalde structuur, die door het programma gewaarborgd dient te worden, d.w.z. dat als er nieuwe data in het geheugen wordt geschreven, of als er iets uit weggehaald wordt, dan dient de software ervoor te zorgen, dat deze opbouw niet verandert, en tevens dat alle bijbehorende variabelen worden aangepast.

FFFFH	In gebruik voor de aansturing van het schuifregister. Met een extra adresdecoder zou hier nog meer geheugen o.i.d. kunnen worden aangesloten.
8000H	SP1000, de spraak naar LPC omzetter.
6000H	De basis van de spraakherkenning.
4000H	E ² PROM, voor de opslag van referentiewoorden en bijbehorende variabelen.
2000H	In gebruik voor de aansturing van het schuifregister.
0000H	RAM, voor opslag ingelezen frames en werkruimte voor processor.

Figuur 6.1. De hoofdindeling van het externe datageheugen van de 80C31.

Het externe geheugen van de processor is opgedeeld in een aantal gebieden, die elk hun eigen functie hebben. Deze gebieden zijn in figuur 6.1 in kaart gebracht.

De bovenste helft van het adresbereik van de processor wordt gebruikt voor het inladen van de data van het schuifregister (looptijdstelling en huiscodemeting), terwijl de onderste helft weer is verdeeld in vier even grote stukken van elk 8 kByte. Daarvan zijn er twee bezet met geheugen, één met de SP1000, en één weer voor het inladen van data van het schuifregister

(hiervoor zijn drie signalen nodig, waar ik later nog op terug kom).

De gewone variabelen voor alle systeemfuncties zijn opgeslagen in de interne RAM van de processor en gaan dus verloren bij spanningsuitval. De RAM bevat alle grote variabelen (array's en werkruimte) die nodig zijn bij de spraakherkenning behalve de variabelen, die betrekking hebben op de referentiewoorden. Deze laatste zijn namelijk allemaal opgeslagen in het niet-vluchtige geheugen, de E²PROM. Dat is natuurlijk ook niet zo verwonderlijk, daar deze gegevens niet verloren mogen gaan als de spanning uitvalt of als het apparaat wordt uitgezet. Bij de referentiewoorden horen nog een aantal variabelen, die ook in de E²PROM zijn opgeslagen. Ten slotte kan de processor nog testen of de E²PROM al is geïntialiseerd aan de hand van twee testbytes, die een bepaalde waarde moeten hebben.

In de externe RAM zijn een aantal buffers gedefinieerd, die de processor gebruikt als werkruimte en voor opslag van ingelezen frames.

000H	LPC-buffer, inleesbuffer voor LPC Data
009H	FRAMES, buffer voor 12 genormeerde frames
075H	FRMSUM, buffer voor 16 16-bit waarden tijdens middelen van frames bij tijdnormering.
087H	FRDATA, buffer waar frames worden opgeslagen als de processor staat te luisteren en probeert een woord te herkennen. Er is plaats voor maximaal 100 ruwe frames.
40BH	EGYBUF, voor tijdelijke opslag energiewaarden van te middelen frames.
47FH	NOISBF, opslag laatste 16 'silence'-energiewaarden
48FH	DISTBF, bevat tijdens template matching algoritme de drie kleinste woordafstanden die al gevonden zijn.
4B7H	TSTORE, tijdelijke opslag genormaliseerd woord bij training (voor middeling indien >1 keer inspreken).
523H	GRID, gebruikt tijdens Dynamic Time Warping Algoritme Werkruimte, waarin dit algoritme wordt uitgevoerd. Elk punt bevat minimale afstand vanaf (1,1) tot dat punt.
675H	Niet in gebruik

Figuur 6.2. De indeling van het RAM werkgeheugen.

De LPC buffer bevat het laatste van de SP1000 ingelezen frame. Frames worden bij het inlezen eerst geplaatst in de FRDATA buffer, waar de ruwe frames staan.

Wanneer er voldoende frames zijn ingelezen (overeenkomend met de waarde van de variabele NRMEAN), dan worden deze gemiddeld en het resultaat wordt in de LPC buffer gezet. Normaal is deze variabele 1, zodat frames direkt van de FRDATA buffer naar de LPC buffer worden gecopiëerd.

Op het nieuwe frame in de LPC buffer vinden dan een aantal bewerkingen plaats, waarna het resultaat weer in de FRDATA buffer wordt verwerkt. Elk ingelezen en verwerkt frame wordt uiteindelijk achterin FRDATA geplaatst, waardoor deze buffer tenslotte een compleet ruw woord zal bevatten (nog niet genormeerd). Op dat moment worden de normeringsroutines aangeroepen, die de ruwe data vanuit FRDATA in tijd en amplitude normeren, en het resultaat opslaan in de buffer FRAMES. Deze buffer bevat dus het eindresultaat van de woord-detectie en normering (precies 12 frames van elk 9 bytes). De inhoud van deze buffer wordt nu in het DTW algoritme gebruikt om te vergelijken met de referentiewoorden. De werkruimte voor het DTW algoritme is de buffer GRID, die alle roosterpunten bevat van het DTW rooster. Zie hiervoor verder de beschrijving van het DTW algoritme. De andere buffers, die nog in de RAM aanwezig zijn worden alle gebruikt om middelingsoperaties uit te voeren, bijvoorbeeld het bepalen van de achtergrondruis uit de energiewaarden van de laatste 16 stilte-frames, of het middelen van complete woorden (templates), zoals dat bij training van woorden moet gebeuren.

§6.4 De database met referentiewoorden.

Om de spraakherkenning te versnellen en meer betrouwbaar te maken, zijn de verschillende woorden die in het systeem zijn opgeslagen ingedeeld in groepen van gelijksoortige of bij elkaar behorende woorden. De processor hoeft niet op elk willekeurig moment elk woord te kunnen herkennen. Als er bijvoorbeeld nog geen kanaal is gekozen, dan is het niet nodig, dat de processor functiewoorden kan herkennen, zoals AAN of LAGER. Door deze woorden nu in aparte groepen te zetten en vervolgens de eigenschap in te bouwen, dat de processor alleen woorden uit bepaalde groepen mag herkennen (afhankelijk van de toestand van het systeem) kan de spraakherkenning zowel versneld als meer betrouwbaar worden. In het volgende zal nu eerst de indeling in groepen worden besproken, en daarna de opbouw van een groep in woorden en bijbehorende variabelen.

§6.4.1 De indeling van woorden in groepen.

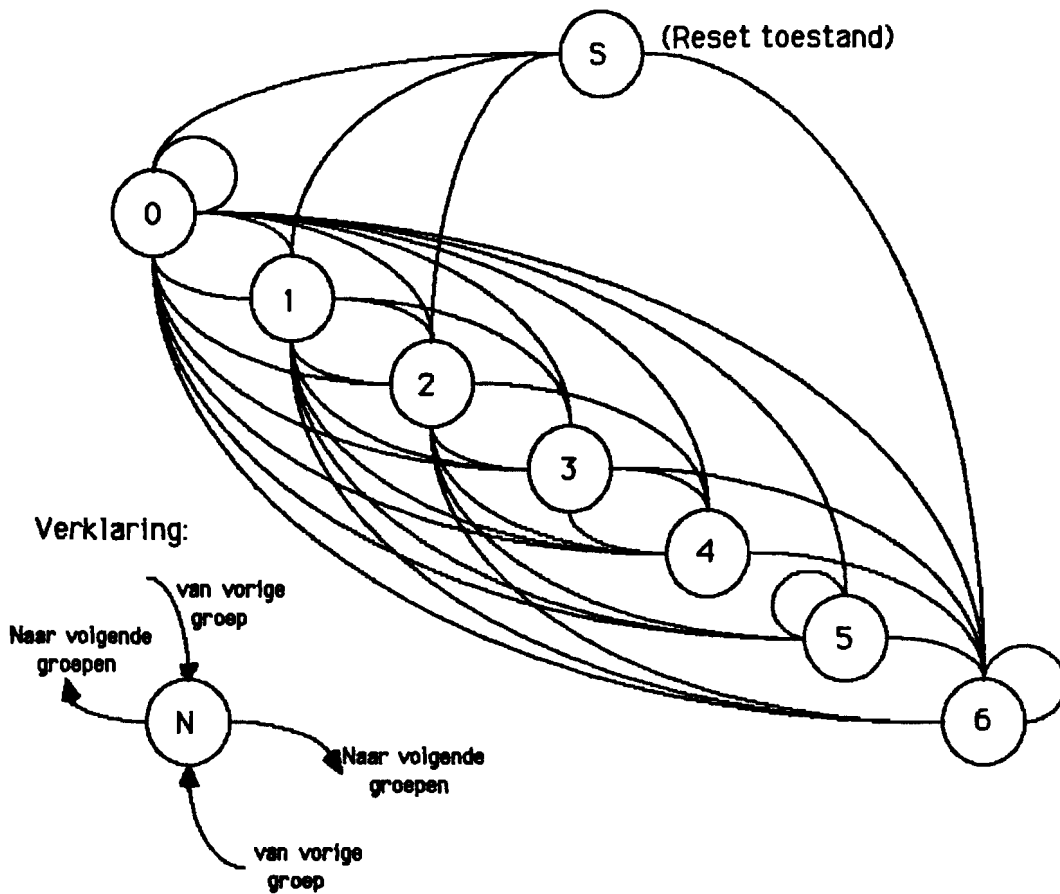
Er kunnen maximaal 8 groepen met woorden zijn. Deze zijn genummerd van 0 tot 7. In de huidige systeemopzet zijn 7 van de 8 groepen bezet. Alle woorden binnen één groep zijn gelijksoortige of bij elkaar behorende woorden. In onderstaande tabel zijn de woorden van elke groep weergegeven.

GROEP:	INHOUD AAN WOORDEN:
GROEP 0	16 WOORDEN VOOR KANAALKEUZE
GROEP 1	'ALLE LAMPEN'
GROEP 2	'ALLES'
GROEP 3	'AAN'
GROEP 4	'UIT'
GROEP 5	'HOGER' & 'LAGER'
GROEP 6	'TRAINING'

Tabel 6.1. De woorden die in elke groep te vinden zijn.

Nadat de processor een woord uit een bepaalde groep heeft herkend, kan hij alleen woorden uit bepaalde andere groepen herkennen, waarbij die nieuwe mogelijke groepen bepaald worden door de groep van het laatst herkende woord. Dit kan worden uitgebeeld in een soort boomstructuur.

In figuur 6.3 zijn alle verbanden tussen de groepen aangegeven. De cirkels stellen de groepen voor (S is de reset toestand, waar het systeem ook na een time-out in terechtkomt). Bogen, die horizontaal vertrekken van een cirkel zijn de uitgaande lijnen. De verticaal aankomende lijnen zijn de inkomende lijnen. Het principe is nu als volgt: nadat een woord van groep X is herkend, kan de processor daarna alleen maar woorden herkennen, die behoren tot een van de groepen, die via een horizontale bij groep X vertrekkende boog met groep X verbonden zijn. Er kan dus ook gezegd worden, dat op een bepaald moment een woord uit groep Y alleen herkend kan worden indien er een boog horizontaal vanuit de huidige systeemtoestand vertrekt en verticaal bij groep Y binnenkomt. Verder komt het systeem na een time-out van circa 10 seconden altijd terecht in de toestand S, de reset-toestand.



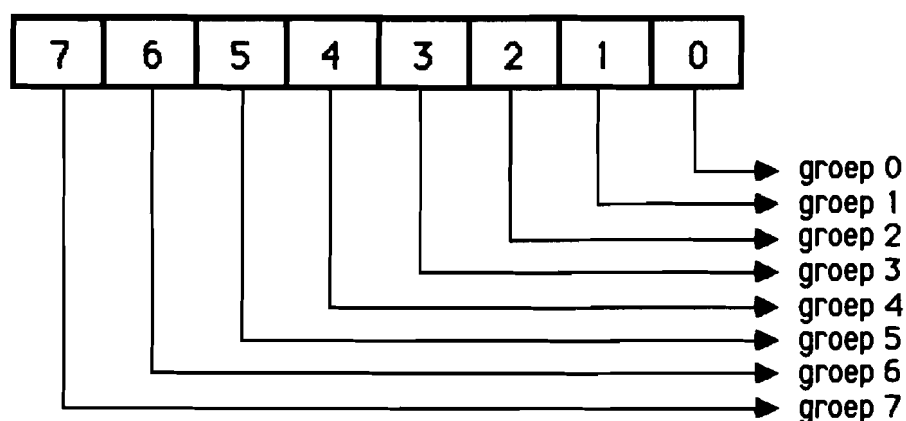
Figuur 6.3. De onderlinge afhankelijkheid van de groepen.

Huidige groep:	Mogelijke vervolgroepen:	SYN byte:
Start	0, 1, 2 en 6	01000111 = 47H
0	0, 1, 2, 3, 4, 5 en 6	01111111 = 7FH
1	0, 2, en 3	00001101 = 0DH
2	0, 1 en 4	00010011 = 13H
3	0, 1, 2, 4 en 6	01010111 = 57H
4	0, 1, 2, 3 en 6	01001111 = 4FH
5	0, 1, 2, 5 en 6	01100111 = 67H
6	0, 1, 2 en 6	01000111 = 47H
7	_____	_____
Time-out	0, 1, 2 en 6	01000111 = 47H

Tabel 6.2. Groepen en hun onderlinge verbanden in tabelvorm.

Men kan deze structuur ook in een tabel weergeven. Dat ziet er dan uit als in tabel 6.2. Hierin is tevens nog iets nieuws aangegeven, namelijk het SYN byte. Dit byte is bij elk woord in het geheugen aanwezig en bepaalt, zoals de tabel al doet vermoeden, welke van de mogelijke 8 groepen voor het volgende woord in aanmerking komen. Elk bit in dit byte komt overeen met één van de 8 groepen, die in principe gebruikt kunnen worden. Steeds wanneer de processor een woord herkend heeft, wordt het bijbehorende SYN byte uitgelezen en opgeslagen in een systeemvariabele. Als er later weer een nieuw woord moet worden herkend, zal de processor alleen die groepen doorzoeken, die hun overeenkomstige bit in die systeemvariabele op 1 hebben staan. Op die manier wordt de herkenningprocedure niet alleen sneller, maar ook betrouwbaarder, want met minder woorden om te testen wordt de kans op fouten ook kleiner. De opbouw van het SYN byte is in figuur 6.4 nogmaals schematisch weergegeven.

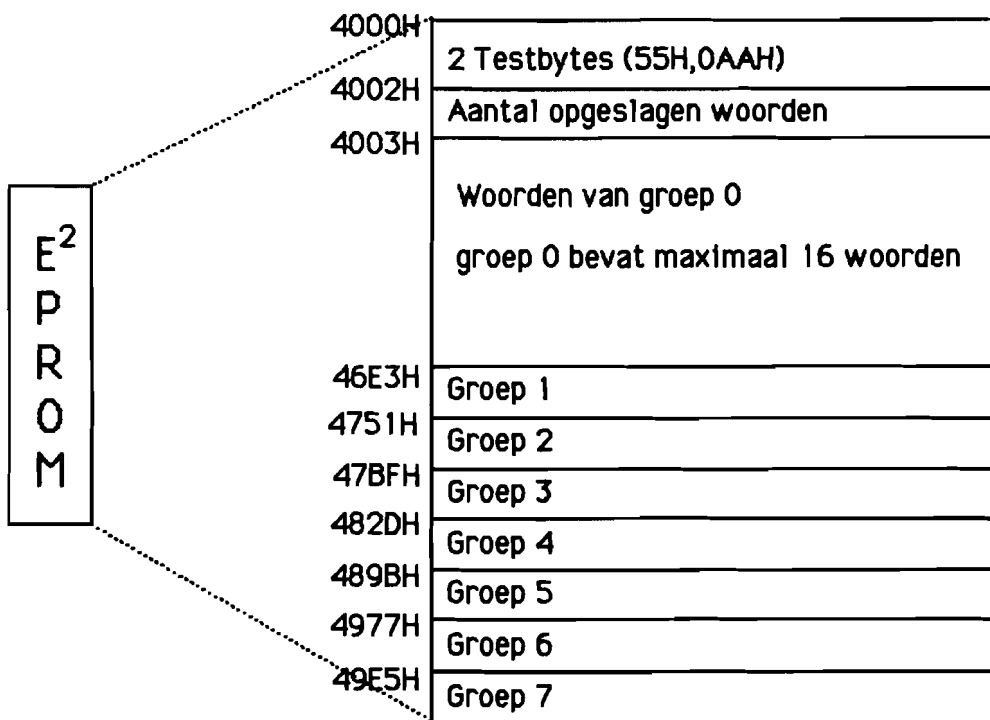
Het SYN byte:



Figuur 6.4. De opbouw van het SYN byte.

Nu de opdeling in woorden is besproken kan nog worden aangegeven, hoe deze in het geheugen zijn opgeslagen. Voor de opslag is een niet vluchtig geheugen gebruikt, namelijk een E²PROM van 8 Kbyte. Deze bevat ten eerste een tweetal testbytes, waaraan de processor kan zien of deze E²PROM al is geïntialiseerd. Vervolgens komt een byte waarin het exacte aantal woorden staat opgeslagen, dat op een bepaald moment in de E²PROM aanwezig is. Vanaf het derde byte komen dan de groepen. Aangezien alle groepen een vaste lengte hebben, die aan de processor bekend is, hebben alle groepen, en daarmee ook alle woorden een vast startadres in het geheugen. Groep 0 bevat 16 woorden, die per stuk wel of niet getraind kunnen zijn. Ook als een

bepaald woord niet is getraind wordt daarvoor toch ruimte gereserveerd! De groepen 1, 2, 3, 4 en 6 bevatten elk slechts één woord en groep 5 heeft er twee. In figuur 6.5 is de hoofdindeling van de E²PROM in groepen aangegeven, samen met de bijbehorende adressen (zie ook tabel 6.1). De reden waarom de woorden 'AAN' en 'UIT' niet samen in één groep zitten is dat na het woord 'ALLE LAMPEN' alléén 'AAN' mag worden herkend of een kanaalkeuzewoord, maar niet 'UIT'. Dat is namelijk een illegaal commando voor het Busch Timac X-10 afstandsbedieningssysteem. Iets dergelijks geldt voor het woord 'ALLES'. Hierna mag alleen 'UIT' komen of een kanaalkeuzewoord, maar niet het woord 'AAN'. De woorden 'AAN' en 'UIT' zijn dus niet gelijksoortig en mogen niet in dezelfde groep zitten. Om precies dezelfde reden zitten de woorden 'ALLE LAMPEN' en 'ALLES' ook niet in één groep. Voor 'HOGER' en 'LAGER' geldt dit niet, want deze woorden kunnen toch al niet herkend worden na 'ALLE LAMPEN' of 'ALLES'.



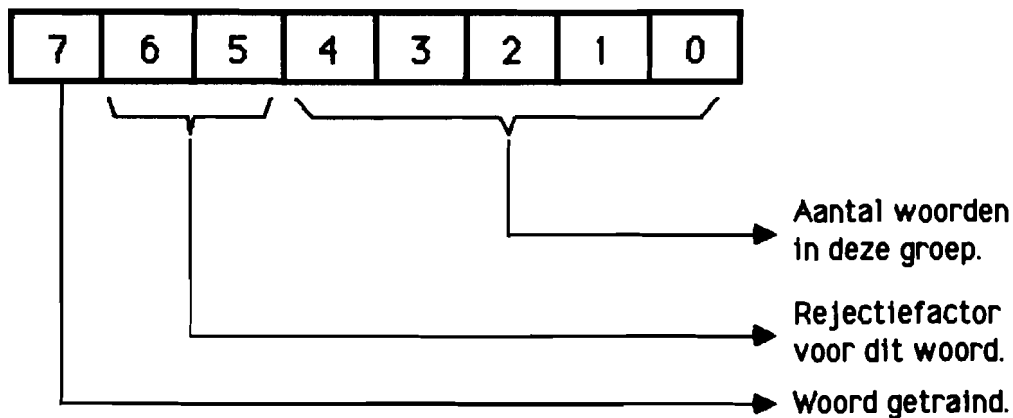
Figuur 6.5. De indeling van de E²PROM, die de referentiewoorden bevat.

§6.4.2 De indeling van een groep in templates.

Binnen een groep kan weer een verdeling worden gemaakt in woorden. Elk woord bestaat uit 2 bytes voor variabelen en een template. De eerste variabele zijn we al tegengekomen, namelijk het SYN byte. De tweede variabele is het zogenaamde HDR byte (header). De opbouw van dit byte staat

In figuur 6.6. De onderste vijf bits (0..4) geven aan hoeveel woorden er in deze groep kunnen worden opgeslagen (de maximale grootte van een groep), maar deze worden door de software verder helemaal niet gebruikt. In toekomstige uitbreidingen kan dit echter wel van pas komen. De bits 5 en 6 bevatten de zogenaamde rejectiefactor voor dit woord. Deze factor bepaalt de strengheid waarmee de processor het woord moet beoordelen na een mogelijke herkenning. Het is namelijk zo dat de processor bij het herkennen van woorden alle toegestane woorden afgaat en via het DTW algoritme vergelijkt met het ingesproken woord. Daarbij worden de drie beste referentiewoorden uitgekozen en onthouden. Achteraf vindt dan nog een beoordeling plaats om het beste gevonden woord te accepteren of te verwerpen, aan de hand van de gevonden drie 'afstanden' bij het DTW algoritme. Hierbij kan de processor meer of minder streng optreden, hetgeen bepaald wordt door de rejectiefactor. Zie verder de bespreking van het betreffende stuk programma verderop in dit hoofdstuk.

Het HDR byte:

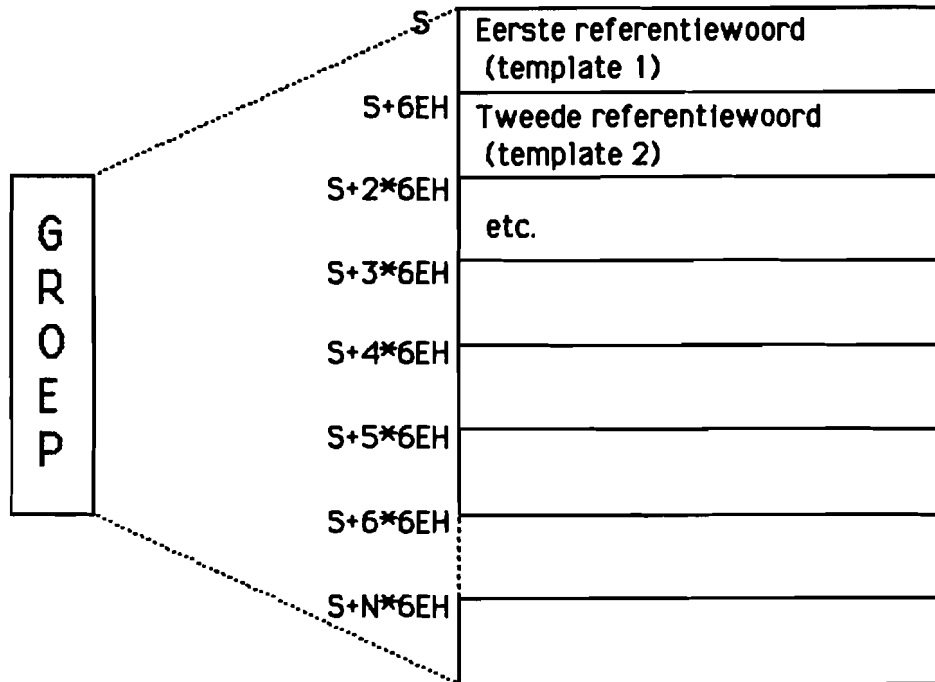


Figuur 6.6. Het HDR byte, een variabele die bij een woord hoort.

Het hoogste bit van het HDR byte geeft aan of het betreffende woord wel of niet is getraind, dus of de data in het template-field wel of niet geldig is. Een logische 1 betekent een geldig woord. Bij het template matching algoritme, dat op zijn beurt het DTW algoritme aanroept, worden van de toegestane groepen alleen die woorden getest, waarvan dit bit op 1 staat. Dit maakt het ook heel gemakkelijk om een woord te wissen. Het enigste wat er moet gebeuren is het op nul zetten van dit bit en het woord is er niet meer.

Na het HDR byte begint het template field, waarin dus uiteindelijk het woord is opgeslagen. Een template bestaat uit 12 genormeerde frames van

elk 9 bytes en neemt dus 108 bytes in beslag. Het template field is het enigste dat de gebruiker zelf kan programmeren in de trainingsmode. De inhoud van het SYN byte en het HDR byte is voor de gebruiker niet toegankelijk en dus zeker niet te veranderen. Deze variabelen worden door de processor bij het initialiseren van de E²PROM éénmalig ingesteld en worden daarna niet meer veranderd (met uitzondering van bit 7 van het HDR byte).

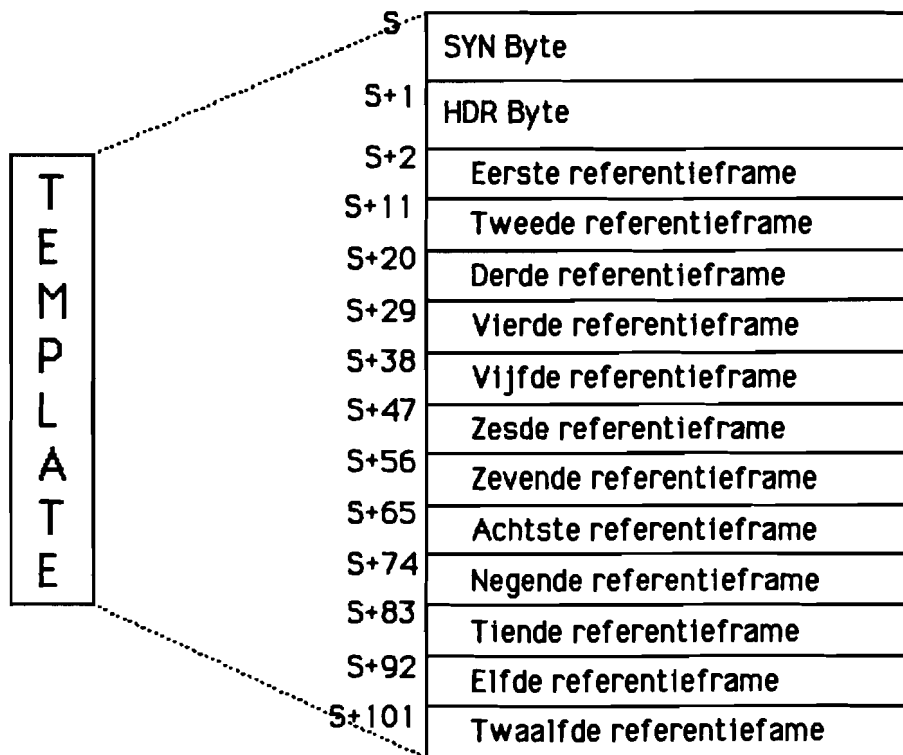


Figuur 6.7. De woorden zijn verdeeld in verschillende groepen.

De referentietemplates worden door de processor zelf aangemaakt in de trainingsmode als men een woord inspreekt. De LPC data die de processor tijdens de normale herkenningmode van de SP1000 inleest wordt -na woordgrensdetectie- ook omgezet in templates die daarna via het DTW algoritme met de referentiewoorden vergeleken kunnen worden.

De opbouw van een referentiewoord in het geheugen is in figuur 6.8 te zien. Hierbij kan nog worden opgemerkt, dat alle constanten zoals bijvoorbeeld het aantal genormeerde frames en het aantal LPC parameters per frame in het programma met labels zijn gedefinieerd en dus heel eenvoudig te veranderen zijn. Om bijvoorbeeld het genormeerde aantal frames te veranderen in 14 hoeft alleen het bijbehorende label te worden veranderd. Alle buffers in het geheugen en de groep lengtes in de E²PROM worden vanzelf aangepast. De hier steeds gebruikte waarde van 12 is waarschijnlijk (maar niet zeker) de optimale waarde, en is overgenomen van het oorspronkelijke

programma (de LIS'NER1000).



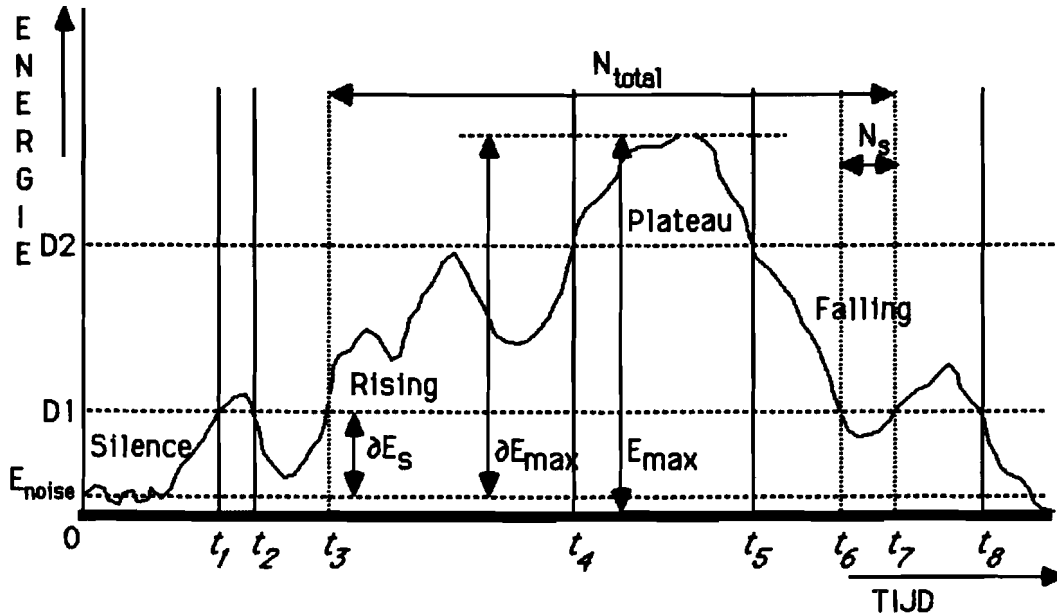
Figuur 6.8. De opbouw van een referentiewoord in het geheugen.

§6.5 De methode voor het bepalen van woordgrenzen.

De processor krijgt elke 20 msec. van de SP1000 een frame binnen, bestaande uit 9 bytes. De eerste 8 bytes zijn LPC parameters. Het 9^e byte is een zogenaamd energy byte, dat aangeeft wat de gemiddelde energie was in het afgelopen 20 msec. interval. Uit deze datastroom zal de processor moeten bepalen wanneer er een woord begint en eindigt, de tussenliggende frames opslaan voor latere bewerking en deze bij detectie van een woord-einde doorgeven aan de bewerkings- en herkenningsalgoritmes. Het feit, dat de processor alle tussenliggende frames moet opslaan (dus de frames, waar het woord uit opgebouwd is), impliceert tevens dat woorden niet erg lang mogen zijn. Momenteel ligt de grens bij 100 frames, ofwel 2 seconden. In het volgende zal de detectie van woordgrenzen worden besproken.

De detectie van de woordgrenzen is gebaseerd op de geluidsenergiestijging die zich tijdens een woord voordoet. Aan de hand van de energie die de SP1000 levert kan de processor dus zien of er mogelijk een woord wordt uitgesproken. In figuur 6.9 is het verloop van de energie bij het uitspreken

van een woord weergegeven. We zien, dat voor en na het woord de energie inhoud van de frames laag is, maar niet nul. Het te allen tijde aanwezige achtergrondlawaai kan derhalve de woordgrensdetectie verstoren. Om hier iets aan te doen houdt de processor een lopend gemiddelde bij van de energiewaarden van de laatste 16 frames die nog niet tot een woord behoren (in de figuur aangegeven met E_{noise}). Alle verdere energiewaarden worden dan gemeten ten opzichte van deze achtergrondenergie, zodat langzame variaties hierin geen invloed meer hebben op de detectie.



Figuur 6.9. Het verloop van de geluidsenergie tijdens het uitspreken van een woord.

Voor de detectie wordt gebruik gemaakt van een Finite State Machine, kortweg FSM. Deze heeft vier toestanden, genaamd silence, rising, plateau en falling. Deze namen hebben uiteraard betrekking op het energieniveau. In rust bevindt deze FSM zich in de toestand silence. De processor berekent nu voor elk binnenkomend frame de energie t.o.v. de achtergrondenergie en pas wanneer deze een bepaalde drempel $D1$ overschrijdt (met waarde ΔE_s boven het achtergrondenergieniveau) gaat de FSM over in de toestand rising, anders blijft de toestand silence en bepaalt de processor opnieuw een lopend gemiddelde voor de achtergrondenergie. In het voorbeeld van figuur 6.9 gaat de toestand over in rising op $t=t_1$. Vanaf dit moment worden alle binnenkomende frames opgeslagen (maximaal 100) totdat het woord is gedetecteerd of blijkt dat er helemaal geen woord komt. Als de energie nu weer daalt tot onder de drempel $D1$ zonder eerst boven de drempel $D2$ te zijn

uitgekomen, wordt er van uitgegaan dat er storing optrad (dichtvallende deur o.i.d.) en begint de processor weer helemaal opnieuw met de woordgrensdetectie (zoals op $t=t_2$). Stijgt de energie echter tot boven de tweede drempel (D2), dan gaat de toestand van de FSM over in plateau (in de figuur op $t=t_4$) en wordt het frame waarbij de toestand overging in rising ($t=t_3$) als een voorlopig begin van een woord aangemerkt. De processor blijft nu frames opnemen. Op een bepaald moment zal de energie weer dalen tot onder de drempel D2 (op $t=t_5$). De FSM gaat dan in de toestand falling en het frame waarbij dat gebeurt wordt als mogelijk einde van een woord gemerkt. Dit is echter nog niet helemaal zeker, zodat de processor gewoon doorgaat met opnemen van frames. Als de energie nu weer stijgt tot boven de drempel D2, dan wordt de toestand weer plateau en herhaalt het geheel zich. Zet de daling van de energie daarentegen door tot onder de eerste drempel (D1), zoals op $t=t_6$, dan is het vrijwel zeker dat het laatste frame waarbij de toestand van de FSM overging van plateau naar falling een wordeinde is. Maar voordat deze grenzen ook inderdaad worden goedgekeurd moet er aan een aantal strenge eisen voldaan zijn:

- De piek energie (E_{max}) van het woord moet boven een bepaalde minimum waarde liggen, die overigens veel hoger is dan drempel D2 in de figuur.
- Hetzelfde geldt voor de piek energie t.o.v. de achtergrondruis (∂E_{max}).
- De lengte van het hele woord (N_{total}) moet minimaal 4 frames zijn.
- Aan het einde van het woord moeten minimaal 10 silence frames komen (N_s), anders wordt de energiedaling toegeschreven aan een in het woord voorkomende rust of lettercombinatie met weinig energie-inhoud. Deze 10 frames duren samen 200 msec. De waarde van 10 frames is overgenomen uit de LIS'NER1000 en kan eventueel veranderd worden.

Tenslotte worden alle frames vanaf het begin van het gevonden woord nogmaals door de FSM geleid, waarbij echter het achtergrondruisniveau van tevoren gelijk is gemaakt aan de piek energie minus 42 dB. Dit betekent dus dat alle frames, waarvan de energie meer dan 42 dB onder de piek energie ligt alsnog worden verworpen. Pas als na deze slag weer aan de hiervoor genoemde eisen is voldaan worden de resterende frames als 'het woord' bij elkaar genomen en doorgegeven aan de voorberekingsalgoritmes (tijd- en amplitudenormering) waarna de herkenning kan plaatsvinden.

§6.6 De tijd- en amplitudenormering.

De frames, die van de woordgrensdetectie komen vormen samen een

woord, dat uit minimaal 4 en maximaal 100 frames bestaat. Stel nu dat we geen tijdnormering toe zouden passen. Dan zouden de referentiewoorden en de te herkennen woorden alle verschillende lengtes hebben die konden liggen tussen 4 en 100 frames. Zelfs één woord kan bij verschillende spreek-snelheden grote lengteverschillen vertonen. Hoe kunnen we nu woorden vergelijken, die totaal verschillend van lengte zijn? Als iemand het woord 'lamp' snel of langzaam uitspreekt mag dat voor de spraakherkenner niets uitmaken, het blijft hetzelfde woord. Om deze redenen wordt er een tijdnormering toegepast. De binnenkomende woorden worden door middelingsprocessen omgezet naar een constante lengte van 12 frames. Het vergelijken van templates gaat dan ook veel eenvoudiger, omdat nu zowel testwoord als referentiewoord altijd dezelfde lengte hebben.

Het tijdnormeringsalgoritme zet een willekeurig aantal frames om in een constant aantal (hier 12). Het principe van het middelingsproces is als volgt: Als we N frames hebben, dan zetten we eerst elk frame 12 keer achter elkaar, zodat we $N \cdot 12$ frames krijgen, verdeeld in N groepen van 12 identieke frames. Deze $N \cdot 12$ frames verdelen we dan in 12 stukken van elk N frames, waarna we per groep het gemiddelde nemen over die N frames. Dit geheel gebeurt dan natuurlijk voor elke LPC parameter van een frame. In de praktijk is er te weinig geheugen om elk frame 12 keer op te slaan, en moet er dus een equivalent worden genomen, dat geen geheugen kost. Als we nagaan wat er in werkelijkheid gebeurt in het hiervoor genoemde algoritme, dan komt dat neer op het middelen van een aantal naburige frames met een bepaalde weegfactor. Stel dat we uitgaan van 23 frames. Dan beginnen we met 12 keer frame 1 en 12 keer frame 2 achter elkaar te zetten. Als we hier dan een groep van 23 frames af halen om te middelen, dan hebben we dus eigenlijk berekend:

$$\text{NormFrame1} := \frac{12 \cdot \text{Frame1} + 11 \cdot \text{Frame2}}{23}$$

Er is nu nog maar $24-23=1$ frame over, dus voor het volgende genormeerde frame moeten we eerst weer 12 keer het derde en 12 keer het vierde woordframe achter de rij plaatsen. Nemen we nu weer een groep van 23 frames weg, dan krijgen we dus:

$$\text{NormFrame2} := \frac{1 \cdot \text{Frame2} + 12 \cdot \text{Frame3} + 10 \cdot \text{Frame4}}{23}$$

Op deze manier kunnen we doorgaan tot alle frames berekend zijn. We zien dat er in feite niets anders gebeurt als het middelen van naburige frames met bepaalde weegfactoren. Deze weegfactoren zijn vrij eenvoudig te berekenen. Het tijdnormeringsalgoritme doet dan ook niets anders als het bepalen van de genormeerde frames met formules als de twee bovenstaande, waarbij de weegfactoren ter plekke worden berekend. Met het vermenigvuldigen van een frame met een weegfactor wordt overigens bedoeld, dat elke parameter van dat frame met die factor vermenigvuldigd wordt.

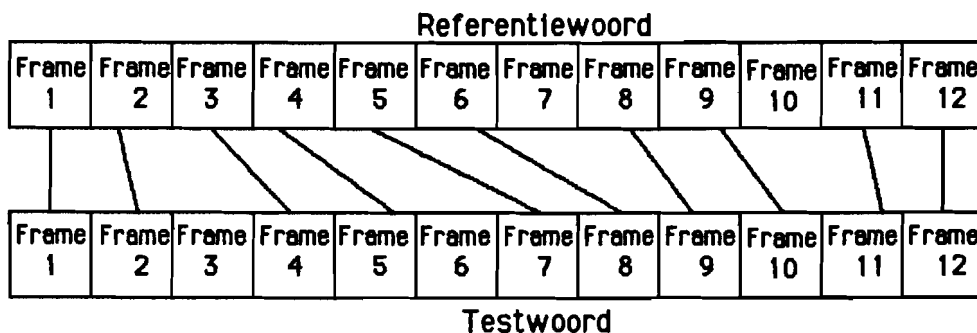
Er is nu een tijdnormeed woord ontstaan. De LPC parameters van dit woord hebben een waarde die ligt tussen -64 en +63. De energieparameter ligt echter tussen 0 en 127. Als we later in het DTW algoritme de frames willen vergelijken moeten alle getallen waar deze uit bestaan in hetzelfde bereik liggen, dus -64...+63. In het DTW algoritme worden alle 9 parameters namelijk gelijkwaardig behandeld. Bij de detectie

van woordgrenzen is al vermeld, dat frames met een energie die meer dan 42 dB onder de piekenergie ligt verworpen worden. Dit betekent, dat van de frames die uiteindelijk door de tijdnormering komen het verschil tussen minimum en maximum energie nooit meer dan 42 dB kan zijn. Voor de energieparameter geldt, dat elke verhoging met 1 overeenkomt met 1.5 dB, zodat de waarde van de energieparameter niet meer dan $42/1.5 = 28$ kan variëren. We willen echter graag dat deze parameter hetzelfde bereik heeft als de andere LPC parameters omdat anders de invloed van de energie bij het DTW algoritme klein is t.o.v. die van de overige parameters. In de amplitude-normering wordt het kleine bereik van 28 stappen, dat zich ergens in het gebied van 0 tot 127 bevindt zodanig geëxpandeerd, dat de energie na afloop ook van -64 tot +63 loopt.

Amplitudenormering is veel eenvoudiger als tijdnormering. Als eerste wordt het maximum bepaald van alle energiewaarden, die in de ingeladen frames voorkomen. Vervolgens wordt een ruisniveau gedefinieerd als zijnde 30 dB (20 binaire stappen) onder deze piek energie (al deze waarden zijn overgenomen van het oorspronkelijke LIS'NER1000 programma). Het zo bepaalde ruisniveau wordt daarna van de energie van elk frame afgetrokken, waarbij eventuele negatieve resultaten op nul worden afgerond. Alle energiewaarden liggen dan dus in het interval [0,19]. Van deze waarden wordt tenslotte 10 afgetrokken en het resultaat met 4 vermenigvuldigd om een bereik-expansie te krijgen tot de grenswaarden -40 en +40. De energieparameter heeft nu een redelijk bereik, dat zeker geschikt is om aan het DTW programma door te geven. Het amplitudenormeringsalgoritme is volledig overgenomen van de LIS'NER1000 programmatuur, inclusief de erin voorkomende constanten.

§6.7 Het Dynamic Time Warping Algoritme.

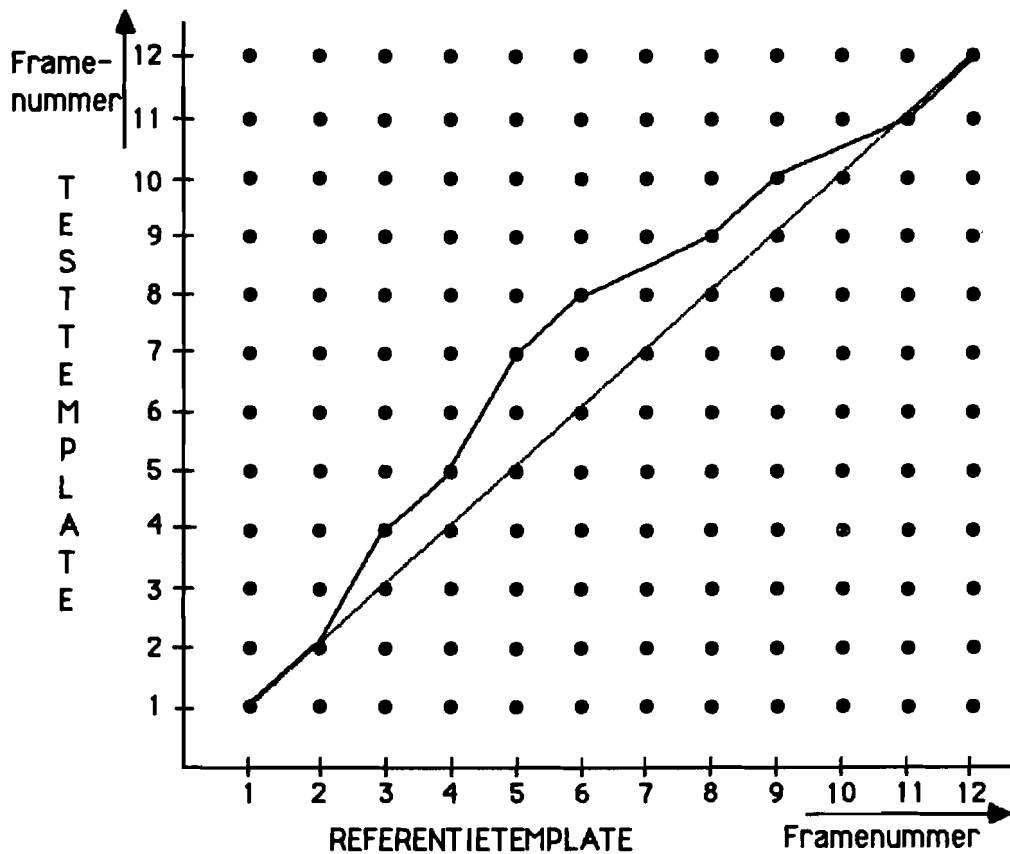
Na de amplitude- en tijdnormering is er een template ontstaan, dat uit 12 frames van 9 bytes bestaat, net zoals de referentietemplates. In principe zou de processor nu alleen nog even alle toegestane referentietemplates af moeten gaan en ze vergelijken met het zojuist uit het ingesproken woord berekende template, om te bepalen welk woord men heeft ingesproken. Hierbij treedt echter een probleem op, namelijk dat van de spreesnelheid. Die kan variëren, van woord tot woord, maar ook binnen één woord. Als een woord in zijn geheel langzamer of sneller wordt uitgesproken dan in de trainingsmode, dan wordt dat al volledig opgevangen door de tijdnormering. Maar als de spreesnelheid binnen een woord varieert, dan kan de herkenning niet plaats vinden door zomaar alle referentietemplates te vergelijken met het template van het ingesproken woord, dat ik voortaan het testtemplate zal noemen. In figuur 6.10 is dit uitgebeeld. Als het bovenste template het referentietemplate is, en het onderste het testtemplate, en als het testwoord in het begin langzamer en aan het eind sneller is uitgesproken dan het referentiewoord, dan zou eigenlijk eerst het testwoord in het begin in elkaar moeten worden gedrukt en aan het eind worden uitgerekt, voordat men frame voor frame met elkaar kan vergelijken.



Figuur 6.10. Het vergelijken van woorden met variërende spreesnelheid.

In de praktijk komt dit er op neer, dat er ter plaatse van zo'n uitrekking maximaal één frame van een template overgeslagen wordt bij het vergelijken, zoals ook duidelijk uit de figuur blijkt. Dit principe kan heel goed in een tweedimensionale figuur (rooster) worden uitgezet. In figuur 6.11 zijn langs de horizontale as de 12 frames van het referentietemplate uitgezet en langs de verticale as de 12 testframes. Wanneer er geen spreesnelheidsvariëaties optreden, is de weg langs welke de frames van de

twee woorden moeten worden vergeleken de rechte lijn van (1,1) naar (12,12). Deze is aangegeven met een stippelijntje. Treden er wel variaties in spreesnelheid op, zoals in de situatie van figuur 6.11, dan wordt een andere weg gevolgd. De met een doorgetrokken lijn aangegeven weg komt overeen met de in figuur 6.10 weergegeven situatie. In het algemene geval zal de processor zelf moeten bepalen welke weg er gevolgd moet worden, met andere woorden: de processor zal moeten bepalen óf, en waar er precies spreesnelheidsvariatiaties optreden, en daarvoor dan correcties toepassen. Het algoritme dat hiervoor zorgt is het DTW algoritme (Dynamic Time Warping).



Figuur 6.11. Tweedimensionale weergave van het vergelijken van frames.

Als we frames willen vergelijken en bepalen hoeveel twee frames op elkaar lijken, dan zullen we eerst een maat moeten vaststellen om aan te geven in hoeverre twee frames op elkaar lijken. Een voor de hand liggende maat is de afstand tussen de frames als deze als 9 dimensionale vectoren beschouwd worden (een frame bestaat uit 9 bytes). De formule hiervoor is :

$$d(F1,F2) = \sqrt{(B_{1,1}-B_{2,1})^2 + \dots + (B_{1,9}-B_{2,9})^2}$$

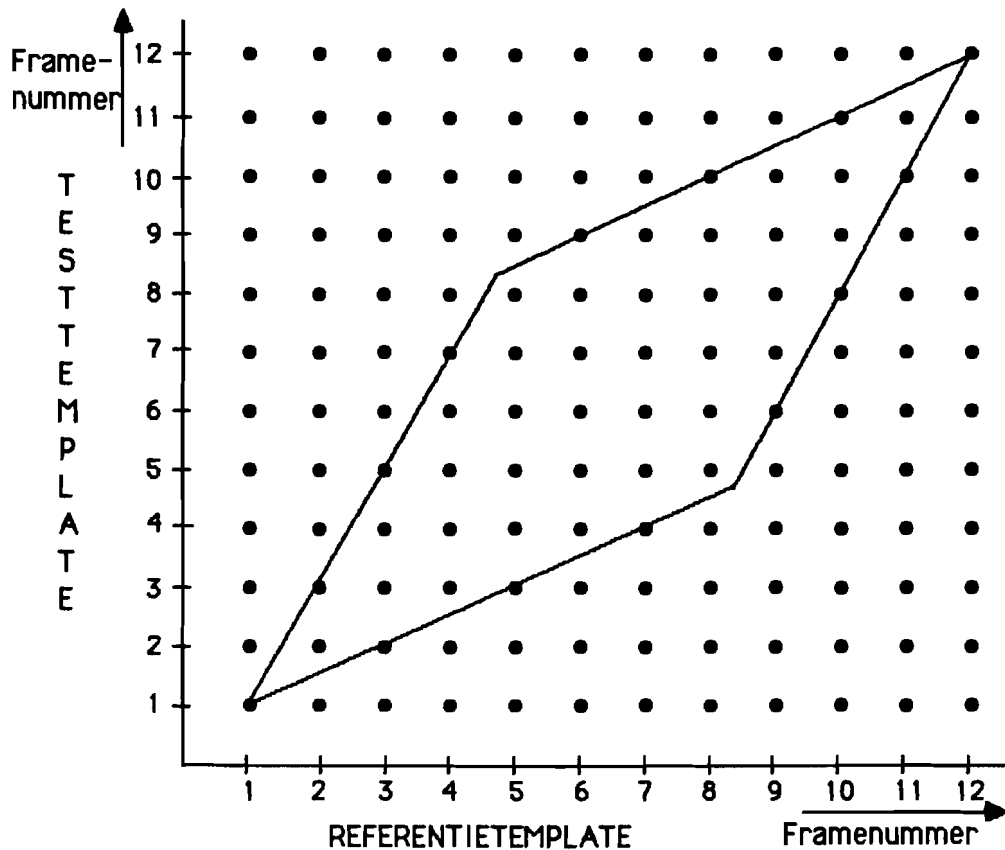
waarin $B_{n,m}$ het m^e byte van het n^e frame voorstelt en $d(F1,F2)$ de afstand tussen twee frames. De formule bevat vermenigvuldigingen en een wortel. Beide bewerkingen zijn tijdrovend, en omdat deze bewerkingen zéér vaak uitgevoerd moeten worden tijdens een herkenningprocedure is deze formule niet erg geschikt. Vandaar dat er gebruik wordt gemaakt van een iets andere 'afstand'-formule. De afstand tussen twee frames wordt gedefinieerd als de som van de absolute waarden van de verschillen tussen de overeenkomstige bytes van de twee frames. In formulevorm wordt dat:

$$d(F1,F2) = \sum_{i=1}^9 |B_{1,i} - B_{2,i}|$$

Deze formule voldoet aan de wiskundige principes van het begrip afstand. Er komen alleen optellingen in voor en het vergelijken van frames kan daardoor zeer snel gebeuren. We kunnen deze definitie nu ook uitbreiden tot het geval van een heel template. De afstand tussen twee templates is de som van de afstanden van de vergeleken frames. Het probleem is nu nog om te bepalen langs welke weg het rooster van (1,1) naar (12,12) moet worden doorlopen.

De oplossing is vrij eenvoudig: zet op alle roosterpunten de locale afstand neer tussen het bijbehorende testframe en het referentiefraam. Bepaal vervolgens voor alle mogelijke wegen van (1,1) naar (12,12) de totale afstand (som van alle locale afstanden) en neem daarvan het minimum. Deze minimale afstand behoort tot de best mogelijke weg door het rooster. Nu is het afgaan van alle wegen een erg tijdrovende zaak. Er is een algoritme dat in staat is om in een zogenaamde graafstructuur, waarvan dit rooster een voorbeeld is, op een snelle manier het minimale pad te bepalen (DFS algoritme, ofwel Depth First Search). Verder kan er nog worden bespaard op rekentijd door niet alle roosterpunten toe te staan. Bijvoorbeeld het punt (1,12) zal nooit voorkomen in de praktijk, omdat dat zou betekenen dat men in het begin het woord héél langzaam had uitgesproken en aan het eind zéér snel, met een factor $12 \cdot 12 = 144$ verschil in spreeknelheid. Feitelijk komt de helling van de gevolgde weg overeen met de verschillen in spreeknelheid ten opzichte van die tijdens de training. Door deze te begrenzen op de waarden 1/2, 1 en 2 kan de gevolgde weg door het rooster nooit buiten het in figuur 6.12 aangegeven parallellogram komen, terwijl er praktisch geen concessies worden gedaan ten aanzien van de herkenning zelf. Ook zonder deze begrenzing zal de gevolgde weg bijna nooit buiten dit parallellogram komen, omdat daar een snelheidsvariatie binnen een woord van zeker een

factor 4 nodig is.



Figuur 6.12. Het parallellogram, waarbinnen de gevolgde weg moet liggen.

Bij nadere beschouwing van figuur 6.12 blijkt dat niet alle wegen van (1,1) naar (12,12) over evenveel roosterpunten lopen. De weg die over een rechte lijn loopt raakt 12 punten, maar een weg die over de buitenrand van het parallellogram loopt raakt slechts 8 punten. Bij het minimum pad algoritme zou dit betekenen dat er een voorkeur bestaat voor paden die langs de rand van het parallellogram lopen, dus paden met veel hellingen van 1/2 en 2. Om dit tegen te gaan moeten we aan de hellingen weegfactoren toekennen, waarmee dan een correctie kan worden toegepast op de afstanden. De verhouding tussen het aantal punten (12/8) is een goede waarde voor de verhouding tussen de weegfactoren. Aan lijnstukken met helling 1 kennen we daarom een weegfactor 2 toe, en aan lijnstukken met helling 1/2 of 2 een weegfactor 3. In het oorspronkelijke LIS'NER1000 programma waren deze factoren 2,1 en 1 voor respectievelijk hellingen 1/2, 1 en 2, en dus niet optimaal. Het totale probleem van het bepalen hoeveel twee templates op elkaar lijken kan nu dus als volgt worden omschreven: Vind door het rooster zoals voorgesteld in figuur 6.12 een pad van (1,1) naar (12,12) dat binnen het

aangegeven parallellogram ligt, zodanig dat de som van de locale afstanden, vermenigvuldigd met de weegfactoren van de gevolgde hellingen over alle door het pad geraakte punten minimaal is. Het algoritme dat hieruit is ontstaan is het Dynamic Time Warping algoritme (DTW).

§6.8 De flowdiagrammen

De flowdiagrammen van het volledige programma zijn in bijlage 1 opgenomen. De meeste van deze diagrammen spreken wel voor zich, maar sommige vragen toch wat meer uitleg. In het volgende zullen de diagrammen besproken worden, voor zover dat nodig is.

LEES SET K-PARAMETERS

In lit.[5] staat nauwkeurig omschreven hoe de processor data kan uitlezen van de SP1000. Het programma is daarvan direkt overgenomen en behoeft dus geen verdere uitleg.

INITIALISEER SP1000

Het initialiseren komt neer op het schrijven van een aantal waarden naar de interne besturingsregisters van de SP1000. Het principe is net als bij het lezen: geef eerst het registernummer op aan de SP1000, vervolgens de data die erin moet komen te staan en wacht dan tot de SP1000 een ready-sigitaal afgeeft. Voor de waarden en functies van de registers wordt verwezen naar lit.[5]. De initialisatie is zodanig dat de duur van één frame 20 msec is en dat de AGC elke 10 msec aangepast wordt. Verder bedraagt de samplefrequentie 6129 Hz. De frameduur en AGC-update snelheid mogen zonder verdere aanpassingen aan de hardware worden veranderd, maar de samplefrequentie mag niet lager worden dan 6 kHz, in verband met de analoge ingangstrap. Deze is bandbegrensd op ca. 3 kHz.

FINITE STATE MACHINE

De finite state machine kan in 4 toestanden verkeren, te weten silence, rising, plateau en falling. Aan de ingang van het FSM programma wordt afhankelijk van deze toestand gesprongen. In alle 4 stukjes programma wordt

dan als eerste de nieuwe toestand van de FSM bepaald, aan de hand van de energie van het laatste nieuwe frame. Verder worden er een drietal pointers bijgehouden, die naar bepaalde frames wijzen. Dit zijn CFRMPT, die wijst naar het huidige frame (na afloop wijst deze pointer naar de eerste vrije plaats in de framedata-inlaadbuffer), LFALPT welke wijst naar het laatste falling frame van het woord, voor zover dat nu bekend is, en FFALPT welke wijst naar het laatste voorkomende frame, waarbij de toestand van de FSM overging van plateau in falling. Indien de toestand terugkomt in silence vanuit falling, dan wordt de zogenaamde cyclus flag geset om aan te geven dat nu de hele energiecyclus die nodig is voor een woord is doorlopen en dat mogelijk nu een woord is gedetecteerd. Op dat moment worden de pointers aangepast, zodat alle tot nu toe ingeladen data zeker tot het ruwe woord zal behoren. Vervolgens zal het geheel onverstoord verder gaan, want het woord hoeft nog niet te zijn afgelopen. Als er nu minstens 10 silenceframes komen zal de eindtest beginnen, en anders begint het hele detectie-algoritme van voren af aan met het tweede deel van het woord. Het al ingeladen deel staat nog steeds in de buffer en wordt niet overschreven. Wanneer er later 10 silence frames worden gedetecteerd zal de gehele buffer worden doorgegeven aan de normeringsalgoritmes. Het aantal silence frames wordt bijgehouden in NRSILE. Deze variabele wordt steeds als de FSM zich in de toestand silence bevindt met 1 opgehoogd en anders op 0 gezet. Verder wordt in de variabele NRTOTL nog het totale aantal ingeladen frames bijgehouden. De FSM houdt ook nog de piekenergie bij en berekent daaruit de 'ruisvloer'. Dat is een energieniveau dat 42 dB onder de piek energie ligt en dat later gebruikt wordt om alle frames met een te lage energie-inhoud weg te gooien. De FSM bepaalt of er een volledig en goed woord is gedetecteerd. Hiertoe wordt door de FSM een flag, de FPOKE (First Pass O.K.) flag geset of gereset. Als aan de eisen is voldaan wordt de flag geset. Verder is er nog een overflow flag, waarmee wordt aangegeven dat de databuffer vol is. Dit treedt op bij woorden die langer zijn als 2 seconden. Deze woorden zijn altijd ongeldig.

TIJDNORMERING

Op het principe van de tijdnormering is in het voorgaande al uitgebreid ingegaan. Hier zal dan ook alleen de implementatie worden uitgelegd. Voor de uitvoering van middelingsoperaties op frames is een buffer nodig waarin plaats is om de som van een aantal frames op te slaan (9 getallen van elk 16 bit). Hiervoor dient de framesombuffer (FRMSUM). Er worden vier variabelen gebruikt voor de bepaling van de weegfactoren: N = het genormeerde aantal

frames (12), G = het aantal goede frames waar we van uitgaan, en dat moet worden teruggebracht tot N , verder C die als teller dient voor het aantal al berekende genormeerde frames, en A als hulpvariabele (zie flowdiagram). Voor elk ruw frame wordt nu afzonderlijk de bijbehorende weegfactor berekend. Dit gaat via het eerder uitgelegde principe. Vervolgens wordt dat frame met die weegfactor vermenigvuldigd en opgeteld bij de inhoud van de framesombuffer. Zodra er genoeg frames bij elkaar zijn opgeteld, vindt er nog een deling plaats door het aantal ruwe frames, om de middeling compleet te maken waarna een nieuw genormeed frame is bepaald en het kan worden opgeslagen in de daarvoor bestemde buffer.

AMPLITUDE-NORMERING

De amplitudenormering is een erg eenvoudig algoritme. Het principe is al besproken en het flowdiagram is heel eenvoudig te doorzien als men die bespreking erbij houdt.

LEES EN VERWERK FRAME

Deze routine wordt aangeroepen vanuit **LUISTER & HERKEN**, de hoofd-routine voor de spraakherkenning, als de SP1000 een melding geeft, dat er weer een frame klaar is (dus elke 20 msec.). De routine begint met het inlezen van dat frame van de SP1000. Voor wat betreft het inlezen zijn er 2 mogelijkheden. De eerste is dat ingelezen frames direkt worden verwerkt, de andere houdt in dat er een aantal frames worden ingeladen, die vervolgens worden gemiddeld tot één frame met het tijdnormeringsalgoritme waarna dat éne frame wordt verwerkt. Dit kan worden ingesteld met de variabele **NRMEAN**, die aangeeft hoeveel frames er moeten worden gemiddeld tot één. Als deze variabele de waarde 1 heeft, wordt er dus niet gemiddeld. Dit is overigens de waarde die hier gebruikt wordt. Deze instelmogelijkheid is overgenomen van de LIS'NER1000 maar wordt dus verder niet gebruikt. Wat precies het effect op de spraakherkenning is, is niet onderzocht en wordt aan een eventuele opvolger overgelaten. Indien er niet wordt gemiddeld staat de middelingsteller (die aangeeft hoeveel te middelen frames er al zijn ingeladen) altijd op 0. Dit betekent dus dat het frame, nadat het in de LPC buffer is gezet, wordt doorgegeven aan de FSM. Is na afloop de toestand **silence**, dan wordt er waarschijnlijk geen woord ingesproken en gaat de processor een nieuw achtergrondgemiddelde bepalen dat als ruisniveau kan dienen (alle in de FSM gebruikte drempels zijn t.o.v. dit ruisniveau). Als de toestand niet **silence** is, wordt het frame opgeslagen in de **FRDATA** buffer als

zijnde een frame, dat bij het woord hoort. Op dit punt eindigt deze routine, tenzij er volgens de FSM een woord is gedetecteerd. Op dat moment wordt namelijk de tweede slag gestart, waarbij alle zojuist geladen frames nogmaals door de FSM worden geleid. Hierbij is echter het achtergrondruis-niveau op 42 dB onder de piek energie van de ingeladen frames gezet, om het dynamisch gebied waarover de energiewaarden zich uitstrekken in te perken. De middeling van frames vindt namelijk per parameter plaats. De LPC parameters van frames met een lage energie-inhoud zouden dan een te grote invloed krijgen. Bovendien is bij te lage energiewaarden de LPC data niet meer te vertrouwen, omdat dan achtergrondlawaai een te grote invloed krijgt. Frames, die bij de tweede slag niet meer voldoen worden weggegooid door alle erna komende frames naar voren te schuiven en daarbij de foute frames dus gewoon te overschrijven. Na afloop wordt getest of er nog wel voldoende frames overgebleven zijn en indien dat zo is worden deze aan de tijd- en amplitudenormerings-algoritmes toegevoerd.

LUISTER EN HERKEN WOORD

Dit is de hoofdroutine voor het laden en herkennen van woorden. In rust bevindt de processor zich bijna constant in dit stukje programma. Om te beginnen wordt de achtergrondenergie-buffer met 7FH gevuld, evenals de gemiddelde achtergrondenergie-waarde zelf (dat is de variabele, die steeds bepaald wordt door middeling van de waarden in de buffer). De waarde 7FH is de maximale waarde die kan optreden (de energie loopt van 0 tot 127 =7FH). Dit betekent dat in het begin geen enkel frame een energie heeft die hoog genoeg is om de FSM van toestand te laten veranderen. Na 16 silence frames (320 msec.) is de buffer echter al weer met normale waarden gevuld, zodat de spraakherkenning kan beginnen. Door deze manier van initialiseren worden enkele vervelende 'software-inschakel-verschijnselen' voorkomen. Na een verdere initialisatie van de benodigde pointers en variabelen wordt de time-out teller op 10 seconden gezet, mits het systeem geen 120 kHz signalen over het lichtnet aan het verzenden is. De timer die voor het zenden gebruikt wordt is namelijk dezelfde als de time-out timer, bij gebrek aan vrije timers in de microprocessor. De timer1-interrupt wordt dus voor twee taken tegelijk gebruikt, waarbij met een flag aangegeven wordt welke taak op een bepaalde moment is geactiveerd. Na afloop van een zendoperatie wordt de timer automatisch weer als time-out teller opgezet met een tijd van 10 seconden.

Terug in het flowdiagram zien we dat nu de 'spreekLED' aan gaat om de

gebruiker te laten zien dat er nu een commando mag worden ingesproken. Een andere belangrijke systeem flag is de speech-flag. Deze geeft aan of een commando van spraak enerzijds of van het toetsenbord of de ultrasone afstandsbediening anderzijds afkomstig is. De flag wordt op dit punt in het programma geset, ten teken dat het een spraakcommando is, maar als er op interruptbasis een commando van de netbesturingspost binnenkomt (seriële commando), wordt de flag gereset. De SP1000 wordt weer geïnitieerd en het wachten op frames kan beginnen. Zodra de SP1000 een 'frame klaar'-melding geeft wordt dat met de reeds besproken routine ingeladen en verwerkt. Tevens wordt getest of er misschien al een seriële commando is ingeladen en indien dat zo is, wordt de speechflag gereset en de routine verlaten voor de commando-uitvoer. Is er geen commando ingeladen en ook geen time-out dan gaat de processor weer staan wachten op een volgend frame. Zodra er een woord klaar is of er een time-out optreedt zal de 'spreekLED' uitgezet worden en de time-out teller gestopt, om te voorkomen dat er nu nog een time-out conditie op kan treden. Indien het systeem nu in de trainingsmode verkeert moet de routine meteen verlaten worden (geen herkenning). In de herkenningmode moet, indien er geen time-out is opgetreden het ingeladen template vergeleken worden met de referentietemplates, waarna het best lijkende woord door het accepteer-of-verwerp algoritme wel of niet wordt goedgekeurd. Indien het woord wordt verworpen, of als er een time-out was opgetreden wordt de groepstatus (SYN) gereset op de beginwaarde, terwijl anders de groepstatus wordt opgehaald uit de data van het beste gevonden woord.

TEMPLATE MATCHER

De template matcher gaat alle in het geheugen aanwezige woorden af, die zich bevinden in groepen, waarvan het overeenkomstige bit in de groepstatus (SYN byte) geset is, en bepaalt met behulp van het DTW algoritme de drie woorden die het meest lijken op het ingesproken woord. De afstanden en woord- en groepsnummer van deze drie woorden is opgeslagen in de distancebuffer. In het begin wordt deze geïnitieerd op 7FFFH, een zeer grote afstand, die met het DTW algoritme nooit bereikt kan worden. In de buitenste programmalus worden nu alle groepen afgegaan. Als een groep zijn bijbehorende bit in SYN op 0 heeft staan, wordt de hele binnenste lus overgeslagen, anders worden alle woorden in die groep afgegaan. Per woord wordt het volgende gedaan: De variabele RBASE (Reference BASE pointer) krijgt een waarde die gelijk is aan het beginadres van het referentiwoord. Dan wordt het HDR byte opgehaald, waaraan gezien kan worden of het woord

eigenlijk wel getraind is. Als dat zo is, wordt het DTW algoritme gestart, en het resultaat (een afstand) wordt, mits klein genoeg op de juiste plaats in de distancebuffer gezet. Daarna wordt RBASE verhoogd om naar het volgende referentiewoord te wijzen. Wanneer alle woorden geweest zijn is de binnenste lus afgesloten en komt de volgende groep aan de beurt. Als die allemaal geweest zijn is de database doorlopen en zijn de drie best lijkende woorden bepaald.

PAS DISTANCE-BUFFER AAN

Het op de juiste plaats in de distance-buffer zetten van een afstand is vrij eenvoudig. Alle afstanden staan op volgorde van grootte, dus de processor hoeft alleen maar alle afstanden af te gaan, te beginnen met de kleinste, en te testen of de gevonden nieuwe afstand kleiner is dan die in de buffer. Indien dat zo is, moeten alle resterende afstanden in de buffer één plaats opschuiven in de richting van groter wordende afstanden en op de vrijgekomen plaats kan dan de nieuwe afstand worden geplaatst. Tevens moeten in de buffer ook nog het woord- en groepsnummer van het betreffende woord erbij worden gezet, anders weet de processor later nog niet welk woord nou het beste was. Deze nummers zijn beschikbaar in de lusvariabelen van de template matcher routine, waarin deze routine wordt aangeroepen. De distance buffer is vier plaatsen groot, maar de vierde plaats is bedoeld als data-afvalbak en zijn inhoud is niet te vertrouwen. Deze is nodig om de software eenvoudiger te kunnen houden.

ACCEPTTEER OF VERWERP WOORD

Nadat met de template matcher de drie best lijkende woorden zijn bepaald moet de processor nog nader onderzoeken of het best lijkende woord wel of niet mag worden geaccepteerd. Zonder meer het best lijkende woord eruit pikken is natuurlijk niet aanvaardbaar, daar de herkenner dan bij elk woord iets zal doen, terwijl dat woord misschien wel helemaal geen commando is. De hier gebruikte methode is de volgende: Als de afstand van het best lijkende woord groter is dan een bepaald maximum, wordt dat woord zonder meer verworpen. Is die afstand kleiner dan een bepaalde ondergrens, dan wordt het woord zonder meer geaccepteerd. Ligt de afstand tussen die twee grenzen, dan wordt nagegaan of de som van de afstanden van het op één na en het op twee na best lijkende woord groter is dan de afstand van het best lijkende woord, vermenigvuldigd met een bepaalde factor. Indien dat zo is wordt het woord geaccepteerd, anders niet. De twee grenzen en de

vermenigvuldigingsfactor worden bepaald door de in het HDR byte opgeslagen rejectiefactor. Dit is een 2 bits getal, waarmee dus vier verschillende waarden kunnen worden ingesteld. In een tabel kunnen dan de twee grenzen en de vermenigvuldigingsfactor worden gevonden. Deze waarden bepalen dus eigenlijk de strengheid waarmee het systeem mogelijk herkende woorden beoordeelt op geldigheid. Rejectiefactor 0 betekent dat het systeem bijna alles goedkeurt, bij 1 is de herkenning normaal, bij 2 wordt het al moeilijk om een woord herkend te krijgen, maar is dus ook de foutenkans heel klein, en bij 3 is de herkenning zo streng, dat het bijna onmogelijk is om het apparaat te gebruiken. De rejectiefactoren staan nu altijd allemaal op 1, en deze kunnen door de gebruiker **niet** veranderd worden. Het experimenteren met deze waarden wordt overgelaten aan een mogelijke opvolger. Het zou misschien nuttig zijn om de rejectiefactor bij groep 6 (het woord TRAINING) op 2 te zetten. De kans dat men door foutieve herkenning in de trainingsmode terecht komt wordt daardoor nog verder verkleind.

DYNAMIC TIME WARPING

De principes van dit algoritme zijn in het voorgaande reeds uitgebreid besproken. Hier zal ik me dan ook beperken tot de implementatie ervan. In het werkgeheugen bevindt zich een array dat alle roosterpunten kan bevatten die in het 13 by 13 rooster voorkomen. Per roosterpunt is opslagruimte voor een 16 bits getal nodig. Het rooster is 13 punten groot omdat ook de punten (0,1), (0,0) en (0,1) nodig zijn voor het algoritme. Deze worden daarvoor gewoon op 0 geïnitieerd. De X-coördinaat geeft het referentieframe aan, de Y-coördinaat het testframe. In de buitenste lus loopt de X van 1 tot 12, en gaat dus één voor één alle referentieframes af. Voor elke waarde wordt via een tabel bepaald wat de minimum en maximum Y waarden zijn van de legale roosterpunten (op deze manier wordt het parallellogram bepaald). In de binnenste lus loopt de Y coördinaat nu van deze minimum naar de maximum waarde, waarbij voor elk punt eerst de locale afstand wordt berekend tussen referentieframe X en testframe Y en vervolgens met behulp van deze waarde de minimum totale afstand op (X,Y). Als op die manier het punt (12,12) wordt bereikt is dus de minimale totale afstand van (1,1) naar (12,12) bekend en daarmee de afstand tussen de twee templates.

BEPALING MINIMUM AFSTAND OP ROOSTERPUNT (X,Y)

Men kan in principe op 3 manieren op het punt (X,Y) komen, namelijk via (X-1,Y-1), via (X-2,Y-1) en via (X-1,Y-2). Deze drie paden hebben elk hun

eigen nummer gekregen. Pad 0 loopt onder 45°, pad 1 onder helling 2 en pad 3 onder helling 1/2. Aangezien de beginpunten van deze drie paden al aan de beurt zijn geweest, zijn de totale minimum afstanden op deze punten ook al bekend. Zoals het flowdiagram al laat zien, wordt eerst een minimum afstand geïnitialiseerd op 0FFFFH, het maximum dat mogelijk is. Vervolgens wordt in een lus via de drie mogelijke paden de afstand op (X,Y) berekend en het minimum daarvan genomen. Als het beginpunt buiten het parallelogram valt is een pad natuurlijk illegaal en telt niet mee. De na drie paden gevonden minimum afstand is dan de uiteindelijke totale minimale afstand van (1,1) naar (X,Y) en deze wordt dan ook opgeslagen op die roosterlocatie. Bij de volgende twee X-coördinaten kan dit punt weer als startpunt dienen.

BEPALING AFSTAND OP (X,Y) VIA OPGEGEVEN PAD

Bij binnenkomst in deze routine is de locale afstand op punt (X,Y) al berekend. Uit het gegeven padnummer kan nu heel eenvoudig het startpunt worden bepaald. Dit is (X-1,Y-1) voor padnummer 0, (X-1,Y-2) voor pad 1 en (X-2,Y-1) voor pad 2. De weegfactor is 2 voor pad 0 en 3 voor de andere twee paden. Na vermenigvuldiging van de weegfactor met de locale afstand hoeft het resultaat alleen nog maar te worden opgeteld bij de totale minimale afstand van het startpunt, die op dat moment al berekend is en dus in het array zit opgeslagen. Alleen wanneer het startpunt een illegaal punt is, moet dit met een flag worden aangegeven en is het resultaat ongeldig.

BEREKEN LOCALE AFSTAND OP ROOSTERPUNT (X,Y)

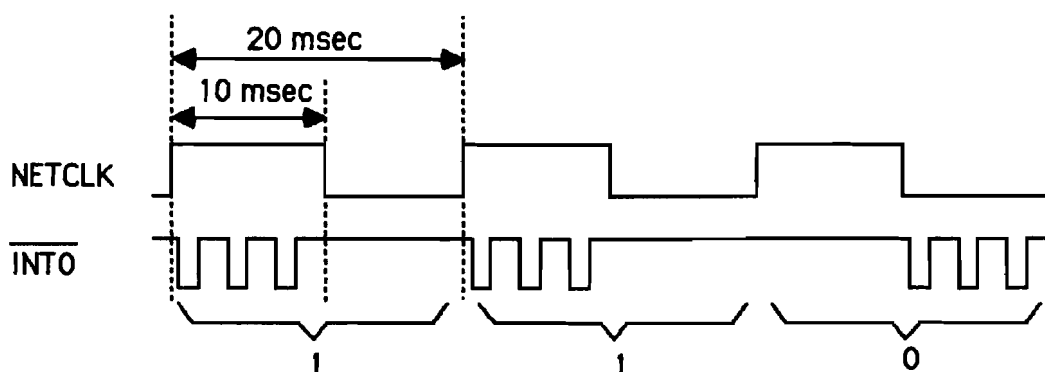
Op het allerdiepste niveau in de spraakherkennerroutines moet de afstand tussen twee frames worden berekend. Zoals reeds eerder in de bespreking naar voren is gekomen is die als volgt gedefinieerd: de som van de absolute waarden van de verschillen van de overeenkomstige bytes van de frames. Het flowdiagram spreekt voor zich zelf en behoeft geen verdere uitleg.

INTERRUPT 0 SERVICE

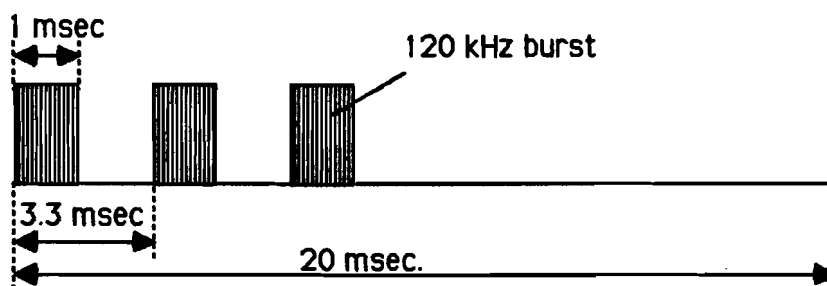
De interrupt 0 wordt gebruikt voor het inladen van seriële commando's, die afkomstig zijn van het custom IC in de bestaande netbesturingspost. Dit IC krijgt zijn commando's weer van het toetsenbord of de ultrasone ingang. Het uitgangssignaal van dit IC wordt door de hardware voorbereid op een zodanige manier, dat er pure bit-informatie overblijft (nullen en enen) die

dan aan de $\overline{\text{INTO}}$ Ingang van de processor worden toegevoerd. Op de eerste flank van een binnenkomend seriëel signaal zal de processor een interrupt krijgen, waarna de service routine gestart wordt. Deze kan dan de code inlezen en decoderen, waarna de normale programma-uitvoer hervat wordt.

In figuur 6.13 is de golfvorm van het signaal op de interrupt ingang weergegeven. Een logische 1 bestaat uit drie bursts van 120 kHz met een lengte van 1 msec. die precies gelijk zijn verdeeld over de 10 msec van een halve netperiode, gevolgd door een halve netperiode van rust. Bij een logische 0 komt eerst de halve periode rust en dan pas de drie bursts van 120 kHz. De reden dat er steeds drie bursts moeten komen op gelijke afstanden is de volgende: Het dan mogelijk is om het signaal via een fasenkoppelaar door te voeren naar een netgroep die op een andere fase werkt (RST). Aan de hand van het netclock signaal dat ook aan de processor wordt toegevoerd kan deze zien of er al een halve periode voorbij is, en dus wanneer er een volledig bit is ingeladen. De bittijd is altijd precies gelijk aan de tijdsduur van één netperiode. Een nieuw signaal kan beginnen op een willekeurige nuldoorgang van het netsignaal (dus elke 10 msec.).



Figuur 6.13a. De opbouw van een seriëel commando.



Figuur 6.13b. Een logische 1, opgebouwd uit 3 bursts, gevolgd door een rust.

In de interrupt service routine gaat de processor derhalve eerst na wat de waarde is van het netclock signaal. Als deze waarde bijvoorbeeld 1 is, dan betekent dat, dat bij iedere opgaande flank van het netclock signaal een volledig bit is ingeladen en het volgende bit begint. Op basis van de toegevoerde timingsignalen kan de processor nu één voor één alle bits binnenladen en testen op geldigheid. Als alle bits zijn ingeladen wordt het zo onstane commando in een variabele USBYTE opgeslagen en de flag GOTCOM geset. Deze flag heeft tot gevolg dat in de hoofdroutine voor de herkenning meteen wordt teruggekeerd naar de commando-uitvoer routine. Als de processor nog geen tijd had om een eventueel vorig seriële commando uit te voeren is deze flag al geset en wordt het zojuist ontvangen commandobyte verder genegeerd. Het laden van bits gebeurt via polling, dus niet op interruptbasis. Tijdens het laden is interrupt0 daarom gedisableed. De processor kan om bepaalde redenen uit synchronisatie raken met het binnenkomende signaal, bijvoorbeeld doordat hij bezig is een andere belangrijke taak die eerst af moet komen, of omdat de interrupt ge-enabled wordt op een moment dat er al een half commando is binnengekomen. Er kan echter nooit een verkeerd commando worden ontvangen op die manier, omdat de signaalvorm die hoort bij het startsignaal van een commando uniek is en in de rest van de code niet voorkomt.

HET LADEN VAN EEN BIT

Uit de figuren op de vorige bladzijde is al duidelijk geworden wat er moet gebeuren. De processor moet gedurende de twee volgende halve netperioden tellen hoeveel pulsen er binnenkomen. Per halve periode zijn er steeds twee mogelijkheden: géén enkele puls of 3 pulsen. Na afloop kan dan nagegaan worden wat voor een bit er is ingeladen. De combinatie 3 pulsen, gevolgd door geen puls komt overeen met een logische 1. Eerst geen pulsen gevolgd door 3 in de tweede halve periode betekent een 0, terwijl 3 pulsen gevolgd door nogmaals 3 pulsen een startbit voorstellen. Alle andere combinaties leveren een illegaal bitpatroon op.

INTERRUPT 1 SERVICE

De interrupt1 wordt gebruikt voor het synchroniseren van de door de processor zelf gegenereerde 120 kHz signalen. Deze moeten steeds net na een nuldoorgang beginnen. Op de **TNT1** ingang krijgt de processor het netclock signaal toegevoerd, waarmee deze dus weet wanneer er een nuldoorgang is geweest.

Als eerste wordt deze interrupt verder gedisabled, want anders zou elke 20 msec. de interrupt opnieuw komen. Verder wordt een pulsteller op 6 gezet. Het signaal kan opgebouwd gedacht worden uit 2 groepjes van 3 pulsen die per groepje wel of niet aanwezig mogen zijn. Bij een 1 is alleen het eerste groepje aanwezig, bij een 0 alleen het tweede en bij een startbit beide groepjes. De interrupt1 server zet twee flags die aangeven welk type bit er in deze periode verzonden moet worden. De flag STARTB geeft aan dat het een startbit is. Als STARTB nul is, geeft TXBIT aan wat de waarde is van het te verzenden bit (0 of 1). De exacte timing van pulsen en tussenliggende rustperiodes van één bit wordt verzorgd door timer1. Daartoe wordt timer1 op 0 gezet en met software een timer1-interrupt gegenereerd. Meteen nadat de interrupt1 service routine verlaten wordt zal nu de timer1 interrupt service routine gestart worden.

DE TIMER1 INTERRUPT SERVICE ROUTINE

In de timer1 server zal nu het opgegeven bit worden uitgezonden over het lichtnet. De flag 'timing-delay' geeft aan of het zojuist afgetelde tijdsinterval een rustperiode was of een pulstijd. Indien het een pulstijd was komt er nu dus een rustperiode. De pulsteller kan met 1 verlaagd worden en indien hij daardoor niet op nul komt is het bit nog niet afgelopen en moet de pulsuitgang hoog worden (rust) en wordt timer1 opnieuw gestart met als tijdstelling de tijdsduur van een rust. Komt de bitteller wel op nul, dan is een volledig bit uitgezonden en wordt timer1 gestopt, de pulsuitgang in de ruststand gezet en interrupt1 weer aangezet, waardoor op de volgende negatieve flank van het netclock signaal door deze interrupt het volgende bit gestart kan worden.

Als bij binnenkomst in deze routine de 'timing-delay'-flag wel gezet is, dan moet er nu een puls komen. Daartoe wordt de timer1 op de tijdsduur van een puls ingesteld (1 msec) en de pulsuitgang gelijk gemaakt aan de inverse waarde van het te verzenden bit (1 = rust, 0 = signaal) waarna de routine weer verlaten wordt. Na verloop van 1 msec. zal de volgende timer1 interrupt de puls beëindigen.

De timer1 server kan ook worden aangeroepen wanneer er niet wordt gezonden, namelijk bij gebruik als een time-out counter in de hoofdroutine voor herkenning. In dat geval wordt er alleen een software teller verlaagd en indien die dan de nulstand bereikt een time-out gegenereerd.

PROGRAMMA-START

Dit is het startpunt van het hele programma. Als het apparaat wordt aangezet begint hier de uitvoering. Als eerste moet natuurlijk het hele systeem worden geïnitieerd, te beginnen met de hele processor en de SP1000. Vervolgens moet de E²PROM getest worden. Als de testbytes niet goed zijn of het aantal getrainde woorden staat op nul, dan wordt de hele E²PROM opnieuw geïnitieerd en naar de trainingsmode overgegaan. Het is dus niet mogelijk in de herkenningmode te komen, zolang er niet minstens één woord is getraind. Als de E²PROM wel goed is geïnitieerd gaat het systeem circa 2 seconden staan testen op een stijging in geluidsenergie. Als die inderdaad wordt waargenomen wordt de trainingsmode gestart, anders komt het systeem in de normale herkenningmode. Vanuit de herkenningmode is het ook mogelijk om in de trainingsmode te komen.

TRAININGSMODE

In deze mode kan de gebruiker de woorden die bij de LED's behoren trainen. Het programma begint met het aanzetten van de trainingsLED en het aanbieden van de eerste LED. De gebruiker kan nu kiezen: wel of niet trainen. Als er voor training wordt gekozen moet het woord 2 keer worden ingesproken waarbij de 2 ontstane templates gemiddeld worden. De opname routine kan ook verlaten worden bij een time-out. Als dat gebeurt moet het woord juist gewist worden. Wissen kan gebeuren door bit 7 van het HDR byte op nul te zetten en indien het bit nog niet nul was ook nog de variabele die het aantal woorden in de E²PROM bijhoudt te verlagen. Het opslaan van een woord kan gebeuren door bit 7 van het HDR byte op 1 te zetten en indien dat bit daarvoor 0 was ook nog het aantal woorden met 1 te verhogen. Verder moet het aangemaakte template natuurlijk worden opgeslagen op de daarvoor bestemde plaats in de E²PROM.

Vervolgens is het dan de beurt aan de volgende LED. Als alle LED's geweest zijn wordt de trainingsmode verlaten.

BIED LED (WOORD) AAN VOOR TRAINING

De LED van de aangeboden functie zal eerst 1 maal knipperen (200 msec), waarbij de processor data inleest van de SP1000 om de achtergrondenergie te bepalen. Vervolgens gaat diezelfde LED nogmaals knipperen (maximaal 5 keer), waarbij de processor nog steeds energie meet, maar nu om te kijken of er een energiestijging plaatsvindt. Zodra een stijging tot boven een bepaalde

drempelwaarde wordt geconstateerd stopt de LED met knipperen en gaat continu branden. De processor wacht nu tot de geluidsenergie weer daalt tot het normale niveau, zet dan de 'bevestigingsLED' aan en gaat staan wachten op een bevestiging (geluidsenergiestijging) of een time-out. Bij één van de twee gebeurtenissen wordt de 'bevestigingsLED' uitgezet en de routine verlaten. In de aanroepende routine wordt dan bij een time-out verder gegaan met de volgende LED of bij een bevestiging de opnameprocedure gestart.

NEEM WOORD OP

De opnameprocedure bestaat uit het maken van twee opnamen (via de routine LUISTER EN HERKEN) en het middelen van de op die manier aangemaakte templates. Bij een time-out moet de routine direct worden verlaten en als een woord te lang is wordt dit met een knipperende LED aangegeven. Het woord moet dan opnieuw worden ingesproken (sneller). Tussen de twee opnames wacht de processor circa een halve seconde omdat alles anders zo snel gaat dat de gebruiker niet eens in de gaten heeft dat hij of zij al voor de tweede keer moet inspreken. (Men heeft niet kunnen zien dat de 'spreekLED' uit is geweest).

HERKENNINGSMODE

Dit is de normale gebruikers-mode. Er wordt meteen begonnen met het aanroepen van de LUISTER EN HERKEN hoofdroutine waar de processor bijna al zijn tijd in doorbrengt. Bij terugkomst van deze routine zijn er drie mogelijkheden: er is een spraakcommando gedetecteerd, er is een seriële commando ingeladen of er is een time-out opgetreden. In het laatste geval worden alle LED's uitgezet en wordt de kanaal & functiestatus gereset (alle kanalen niet-actief). Bij een spraakcommando wordt dit uitgevoerd mits de rejectflag niet is geset. Tenslotte wordt een seriële commando altijd direct uitgevoerd, maar als het een 'ALLE LAMPEN AAN' of 'ALLES UIT' commando betreft moet het worden opgesplitst in twee aparte commando's voor de verwerking. Dit laatste is nodig omdat bij spraak deze commando's in tweeën moesten worden gesplitst. Zie daarvoor verder het hoofdstuk over de bediening (hoofdstuk 4).

VERWERK EEN COMMANDO

Voordat we de commando-verwerking kunnen bespreken moet eerst nog

even een belangrijke variabele worden besproken, namelijk de commando-status. Dit byte geeft precies aan in welke toestand het systeem verkeert voor wat betreft de functiekeuze. De waarde geeft aan wat het laatst verwerkte commando was:

00	=	ALLES	04	=	LAGER
01	=	ALLE LAMPEN	05	=	HOGER
02	=	AAN	06	=	TRAINING
03	=	UIT	80H	=	Geen commando

De commandoverwerking gaat aan de hand van het groepsnummer van het woord na wat voor een soort woord het is. Als het een kanaalkeuze betreft die via spraak is ingegeven worden alle LED's uitgezet. Als het een kanaalkeuze is via een seriële commando gebeurt dat alleen als dit de eerste kanaalkeuze is sinds de laatste keer dat er een commando werd uitgevoerd (net zoals bij het bestaande systeem in dat geval alle geactiveerde kanalen worden ge-deactiveerd). De commandostatus wordt altijd gereset bij een kanaalkeuze. De processor verzendt net vóór een commando altijd de kanaalnummers, tenzij die sinds de vorige commando-uitvoer niet zijn veranderd. Als na een commando-uitvoer een nieuw kanaal wordt gekozen moeten alle kanalen later bij de volgende functie weer worden uitgezonden. Dat wordt aangegeven met de 'verzend kanalen' flag.

Bij een kanaalwoord (een woord uit groep 0, 1 of 2) worden alle kanaalLED's uitgezet, alsmede de 'ALLES' en 'ALLE LAMPEN' - LED's. Verder moet dan nog de LED van het gekozen kanaal worden aangezet. Bij een woord uit groep 1 of 2 wordt de betreffende LED aangezet en de commandostatus krijgt de juiste waarde.

De woorden van groep 3, 4 en 5 zijn functiewoorden. Zij hebben het genereren van lichtnet besturingssignalen tot gevolg. Voordat er signaal gegenereerd kan worden moet er mistens één kanaal zijn geselecteerd. Verder moet het een geldig commando betreffen. Zo is bijvoorbeeld ALLE LAMPEN HOGER ongeldig en in dat geval gebeurt er dus niets. Dit soort combinaties zijn mogelijk als spraak en toetsenbord door elkaar gebruikt worden. Bij een woord uit groep 3-5 wordt eerst nagegaan of het een geldig commando is en indien dat zo is wordt de commandostatus aangepast en het commando uitgevoerd.

Bij het trainingwoord (groep 6) zullen de eerste keer de 'bevestigingsLED' en de 'trainingsLED' aan gaan, terwijl alle andere LED's uitgaan. De commandostatus krijgt de waarde 6 om aan te geven dat dit de eerste keer was dat het woord TRAINING werd ingesproken. Bij de tweede keer gaat dan

de 'bevestigingsLED' uit en wordt het trainingsprogramma gestart. Het is niet mogelijk om via het toetsenbord of de ultrasone afstandsbediening in de trainingsmode te komen.

Hiermee zijn we aan het einde van de software gekomen. Van het oorspronkelijke LIS'NER1000 programma zijn alleen de principes overgenomen. Het programma op zich is eigenlijk helemaal opnieuw ontwikkeld. Het DTW algoritme is zelfs volgens een geheel nieuwe implementatie uitgevoerd. De sourcelisting is ruim van commentaar voorzien, zodat een mogelijke opvolger vrij snel in de software ingewerkt kan worden, mochten er ooit nog fouten in ontdekt worden.

HOOFDSTUK 7

CONCLUSIES EN AANBEVELINGEN

§7.1 Bereikte resultaten

Wanneer we de opdrachtomschrijving naast de rest van dit verslag houden kunnen we concluderen dat het werk zeker geslaagd is. Er is een spraakherkenner gerealiseerd, die voldoet aan alle opgelegde eisen. De herkenner is gekoppeld aan het bekende Busch Timac X-10 afstandsbedieningssysteem, waardoor er een gebruiksklaar prototype is ontstaan. Er is erg veel aandacht besteed aan de manier waarop het apparaat moet kunnen worden bediend, waarbij natuurlijk speciaal gelet is op de bedoelde gebruikersgroep. We kunnen de behaalde resultaten het beste puntsgewijs opsommen:

- De herkenner vervangt de bestaande netbesturingspost.
- De oude taken van de netbesturingspost zijn niet aangetast. Het toetsenbord en de ultrasone ingang zijn nog steeds aanwezig en bruikbaar, zodat bijvoorbeeld ook combinatie met de Monoselector mogelijk is.
- Voeding vindt plaats via het lichtnet (trafo). Er zijn dus geen batterijen nodig. Voor de aansturing van afstandsschakelaars en dimmers is toch altijd een aansluiting op het lichtnet nodig.
- Een LEDdisplay met 25 LED's geeft altijd de actuele toestand van het hele systeem weer (terugkoppeling naar gebruiker).
- De bediening via toetsenbord of ultrasone afstandsbediening blijft precies zoals bij het bestaande systeem. Bij bediening via spraak kan slechts één kanaal tegelijk worden geactiveerd en zijn de directe functies ('alle lampen aan' en 'alles uit') opgesplitst in twee-woord commando's.
- Er kunnen maximaal 23 woorden worden getraind.
- De gebruiker (gehandicapte) kan zelf die woorden trainen en wissen.
- In de trainingsmode wordt van het bekende 'kiezen en bevestigen'-principe gebruik gemaakt om foutieve trainingen te voorkomen.
- In de normale gebruiks-mode kan men gewoon het gewenste kanaal inspreken, gevolgd door de gewenste functie en er is direkt resultaat. Bij mogelijke foutieve herkenning kan men het inverse of het gewenste commando inspreken om alsnog tot het goede resultaat te komen.

Wat betreft de aanbevelingen, gedaan in het verslag van mijn voorganger, lit[2], kan het volgende worden opgemerkt:

- Een standby toestand voor de spraakherkenningssoftware is niet nodig. Het apparaat staat altijd aan; men kan praktisch op elk moment een woord inspreken en het opgenomen vermogen is zeer klein (ca. 1 Watt). Voeding vindt bovendien plaats via het lichtnet, zodat dit geen probleem is.
- Er is nog geen databank met testwoorden aangelegd. Zie §7.2
- De weegfactoren in het DTW algoritme zijn nu aangepast. De andere fouten die nog in het LIS'NER1000 programma voorkwamen zijn ook verbeterd. Er is echter niet getest of dit ook inderdaad een verbetering in de herkenningsscore opleverde.
- Het DTW algoritme is helemaal opnieuw geschreven. De nieuwe versie is veel overzichtelijker en gemakkelijker te veranderen dan de oude versie. Bovendien is het programma korter en gebruikt veel minder variabelen en toestanden dan de LIS'NER1000 versie.

§7.2 Aanbevelingen voor mogelijke verbeteringen

De gerealiseerde spraakherkenner is waarschijnlijk nog lang niet optimaal. In het volgende zullen een aantal aanbevelingen worden gedaan, zowel op het gebied van hardware als software, die mogelijk verbeteringen in de herkenning tot gevolg kunnen hebben. Verder zijn er ook nog enkele algemene aanbevelingen:

- Er is tot op dit moment nog geen mogelijkheid om de herkenningsscore te meten, omdat er geen databank is met woorden om te testen. Deze moet dus worden aangelegd. Daarvoor moeten er eerst een aantal referentiewoorden worden bepaald waarmee voortaan wordt getest. Misschien zijn er al een aantal gedefinieerd in de literatuur?
- Als er een lijst met referentiewoorden is gemaakt moet met behulp daarvan een soort testprocedure worden ontwikkeld om de herkenningsscore te meten. Een mogelijke methode zou kunnen inhouden dat een aantal proefpersonen de hele referentielijst inspreekt op band, gevolgd door in willekeurige volgorde enkele honderden woorden uit die lijst. De herkenner kan dan met de band getraind en getest worden.
- De analoge ingangstrap is overgenomen van de LIS'NER1000 kaart. In lit[1] zijn een aantal mogelijke configuraties gegeven voor het ingangscircuit. Deze zijn door mij niet uitgetest. Het is mogelijk dat gebruik van deze circuits een verbetering in de herkenning oplevert.
- Indien bij de spraakherkenner een externe microfoon wordt gebruikt, dient

er rekening mee te worden gehouden, dat alle punten in de herkenner, dus ook de massa van de microfoon **aanraakgevaarlijk** zijn. De elektronica die al in de netbesturingpost aanwezig was wordt namelijk direkt uit het lichtnet gevoed, dus niet via een trafo. Vanwege de koppeling tussen de spraakherkenner en deze elektronica is nu elk punt in principe gevaarlijk! Het verdient aanbeveling hier iets aan te doen.

- Alhoewel het schuifregister reeds aanwezig is, wordt de informatie daaruit omtrent looptijdinstelling en huiscode nog niet gebruikt. Deze eenvoudige aanpassing moet zo snel mogelijk gemaakt worden.
- In het hoofdstuk over de software (H6) is reeds de mogelijkheid ter sprake geweest, dat tijdens het inladen van frames steeds N frames door middeling worden teruggebracht tot 1 frame, dat dan wordt opgeslagen. Het is mogelijk dat hiermee een verbetering in de herkenning optreedt. Dit moet nog onderzocht worden.
- Een groot aantal variabelen en waarden in de software zijn met labels instelbaar. Hierbij moet gedacht worden aan bijvoorbeeld de drempelwaarden in de woordgrensdetectie-algoritmes, het genormaliseerde aantal frames, e.d. De invloed hiervan is nog niet bekend.
- De rejectiefactoren zijn in een tabel in de EPROM opgeslagen. Deze hebben invloed op de strengheid van de herkenning. De optimale instelling hiervan moet nog bepaald worden.
- Bij sommige LED's zou het gewenst zijn als er meerdere woorden aan konden worden toegekend. Hierbij moet vooral gedacht worden aan AAN, UIT, HOGER en LAGER. Wanneer bijvoorbeeld gordijensluiters zijn aangesloten zou men deze graag bedienen via een commando als 'Gordijnen open' in plaats van 'Gordijnen aan', terwijl bij een lamp weer het woord 'aan' gewenst is.
- In het trainingsalgoritme kan een extra stukje software worden ingebouwd, dat als bij het tweede keer inspreken van een woord dit woord teveel afwijkt van het eerste, een van de twee ingesproken woorden negeert en de gebruiker voor de derde keer het woord laat inspreken.
- Bij het inladen van frames kan ook iets intelligenter te werk worden gegaan: men kan het zo programmeren, dat een frame alleen wordt overgenomen van de SP1000 als het voldoende afwijkt van het vorige. Op die manier worden dus feitelijk de veranderingen in het ingesproken woord overgenomen en niet de statische delen. Deze methode is door P. Neutelings toegepast (lit[8]) en gaf inderdaad enige verbetering in de herkenningsscore.

HOOFDSTUK 8

LITERATUURLIJST

- [1] Bosch, J.G.
EEN GOEDKOPE SPRAAKHERKENNER VOOR TOEPASSING IN HULPMIDDELEN
VOOR GEHANDICAPTEN.
Vakgroep Medische Elektrotechniek (E.M.E.),
Faculteit der Elektrotechniek,
Technische Universiteit Eindhoven, 1985
Afstudeerverslag.
- [2] Traas, A.M.
NADERE ANALYSE VAN EEN SPRAAKHERKENNINGSSYSTEEM VOOR
TOEPASSING IN HULPMIDDELEN VOOR MOTORISCH GEHANDICAPTEN.
Vakgroep Medische Elektrotechniek (E.M.E.),
Faculteit der Elektrotechniek,
Technische Universiteit Eindhoven, 1986
Afstudeerverslag.
- [3] National Semiconductor Corporation,
Linear Databook
Uitgave 1982.
- [4] ADC0831, ADC0832, ADC0834 AND ADC0838 (COP431, COP432,
COP434 AND COP438) 8-BIT SERIAL I/O A/D CONVERTERS WITH
MULTIPLEXER OPTIONS.
Santa Clara, CA. (USA): National Semiconductor, 1982.
XH/093 (IM-B25M62).
- [5] SP1000 VOICE RECOGNITION/SYNTHESIS DEVICE.
Palo Alto, CA. (USA): General Instrument, Micro-electronics division,
1983.
General Instrument datasheet DS5000 1B (SP1000-8311 Rev. B.)

- [6] Ciarcia, S.A.
PLUG-IN REMOTE CONTROL SYSTEM,
Artikel uit: Radio Electronics, Vol. 51, No. 9,
September 1980, blz. 47-51.
- [7] SP1000 MANUAL.
Documentatie over de SP1000, afkomstig van
General Instrument.
1 Juni 1984.
- [8] Neutelings, P.
SPRAAKHERKENNING MET DE SP1000 SPRAAK ANALYSE CHIP.
Technische Universiteit Eindhoven en N.V. Philips Eindhoven,
maart 1986 - februari 1987.
Afstudeerverslag.

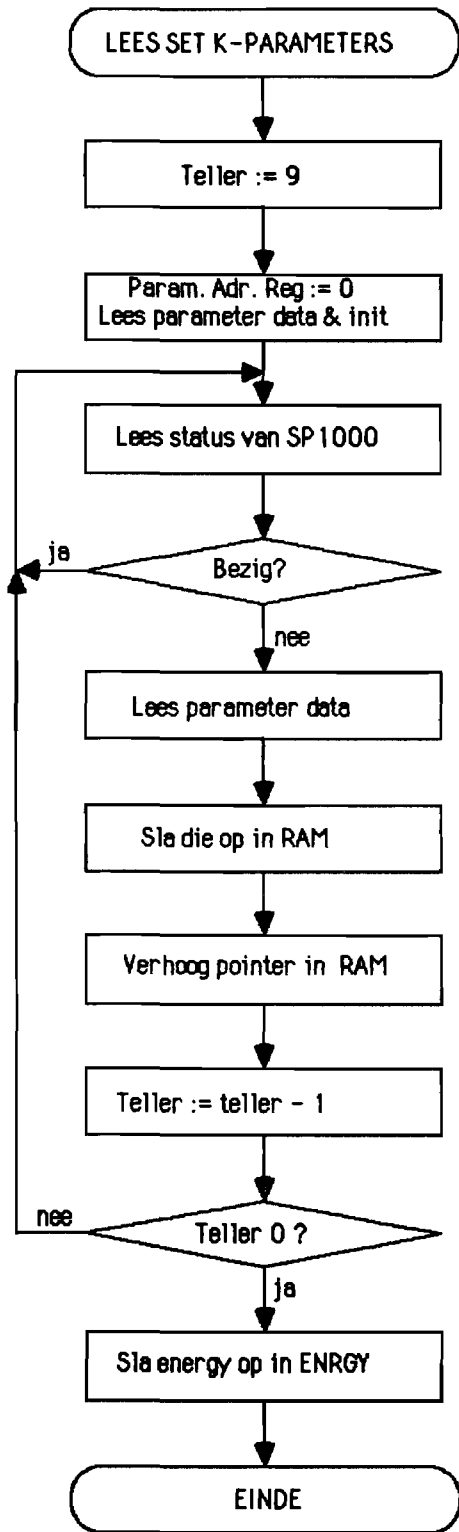
Verdere informatie omtrent Busch Timac systemen:

Busch Jaeger Elektro GmbH,
Hausleittechnik, Busch Timac X-10
Informatie brochure.

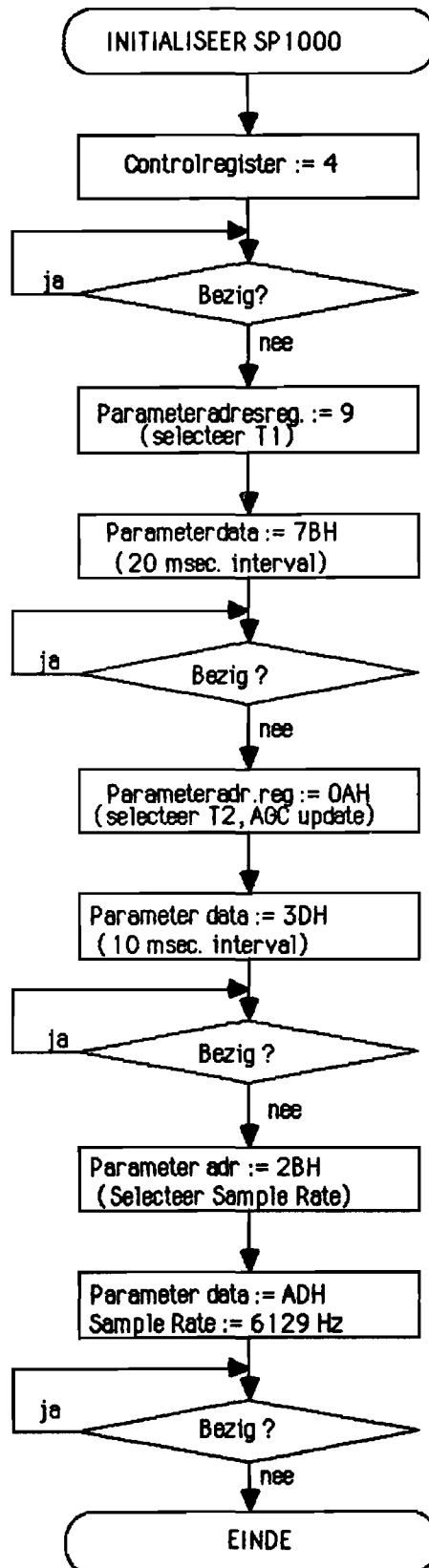
en

diverse andere artikelen ter inzage bij de vakgroep E.M.E.

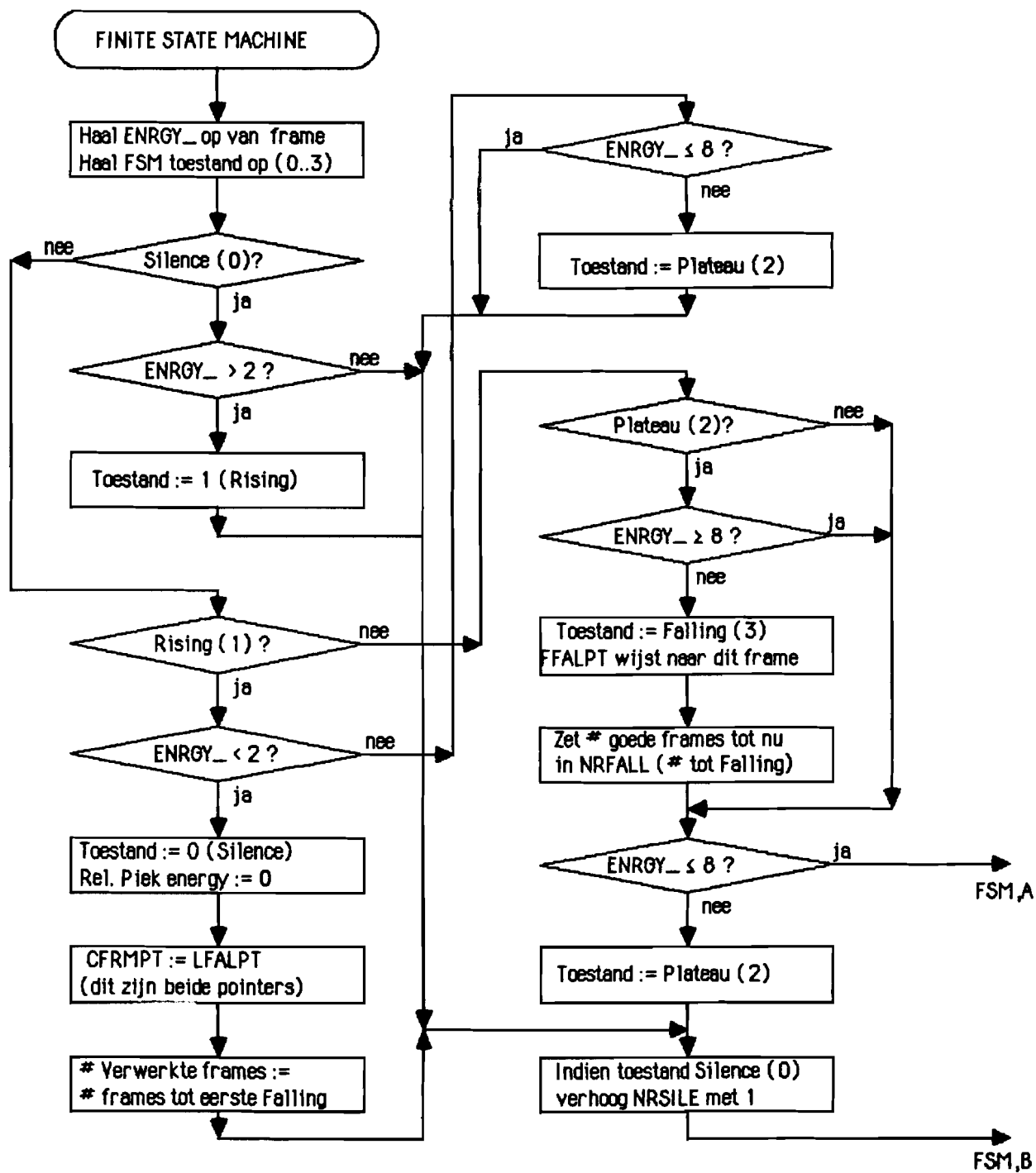
BIJLAGE 1: FLOWDIAGRAMMEN VAN DE SOFTWARE



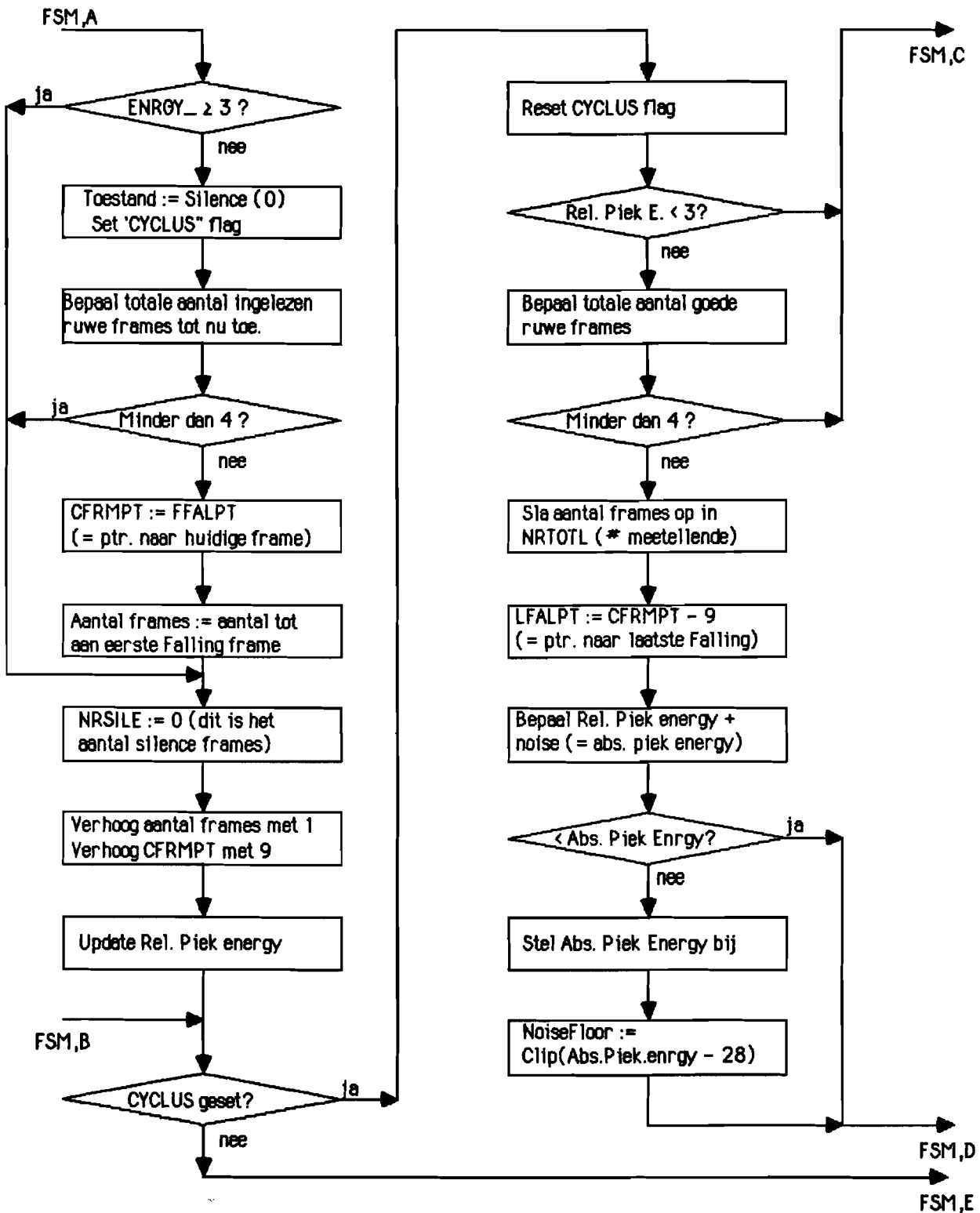
a) Inlezen van LPC parameters



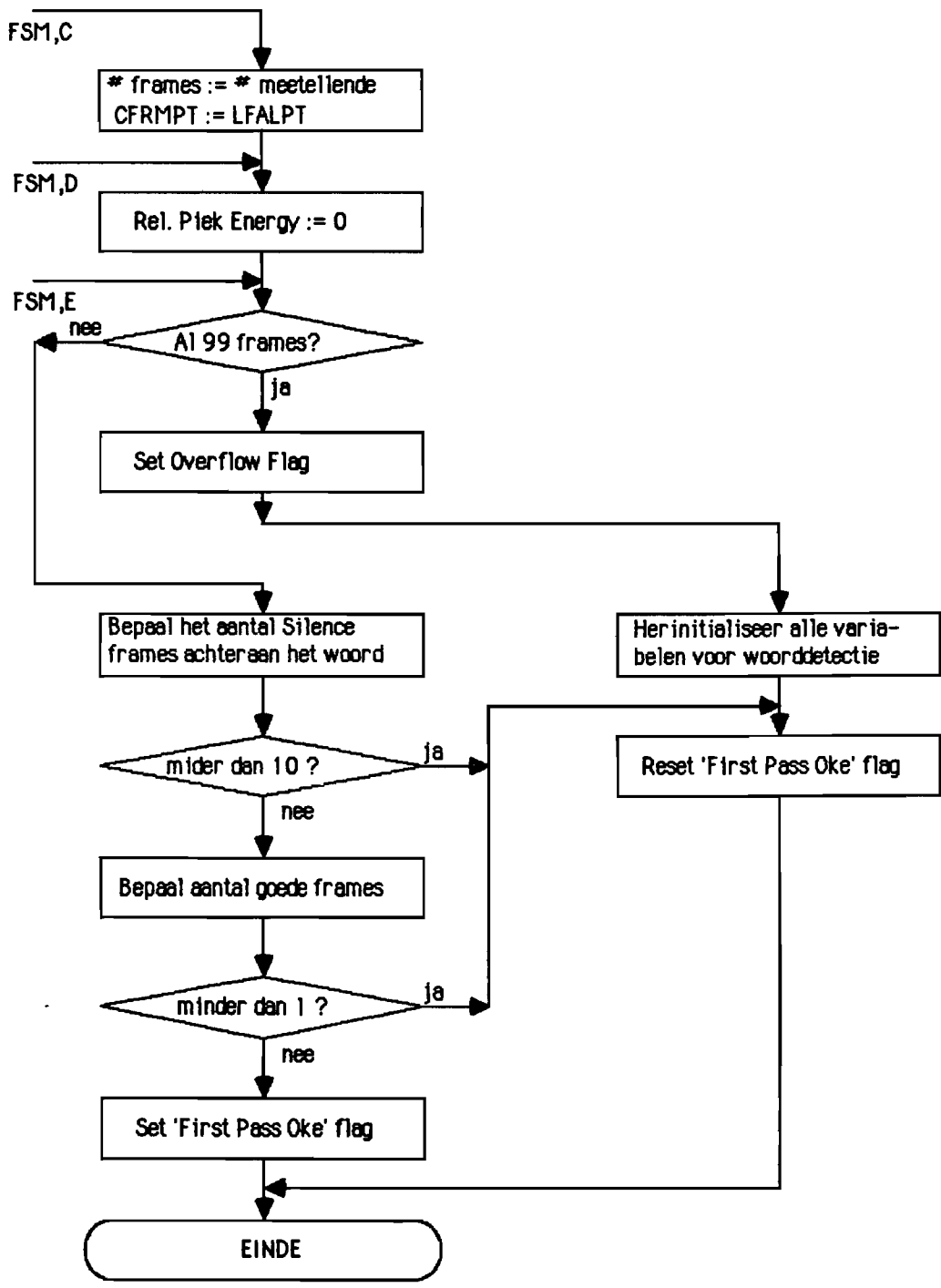
b) Initialisatie van de SP1000



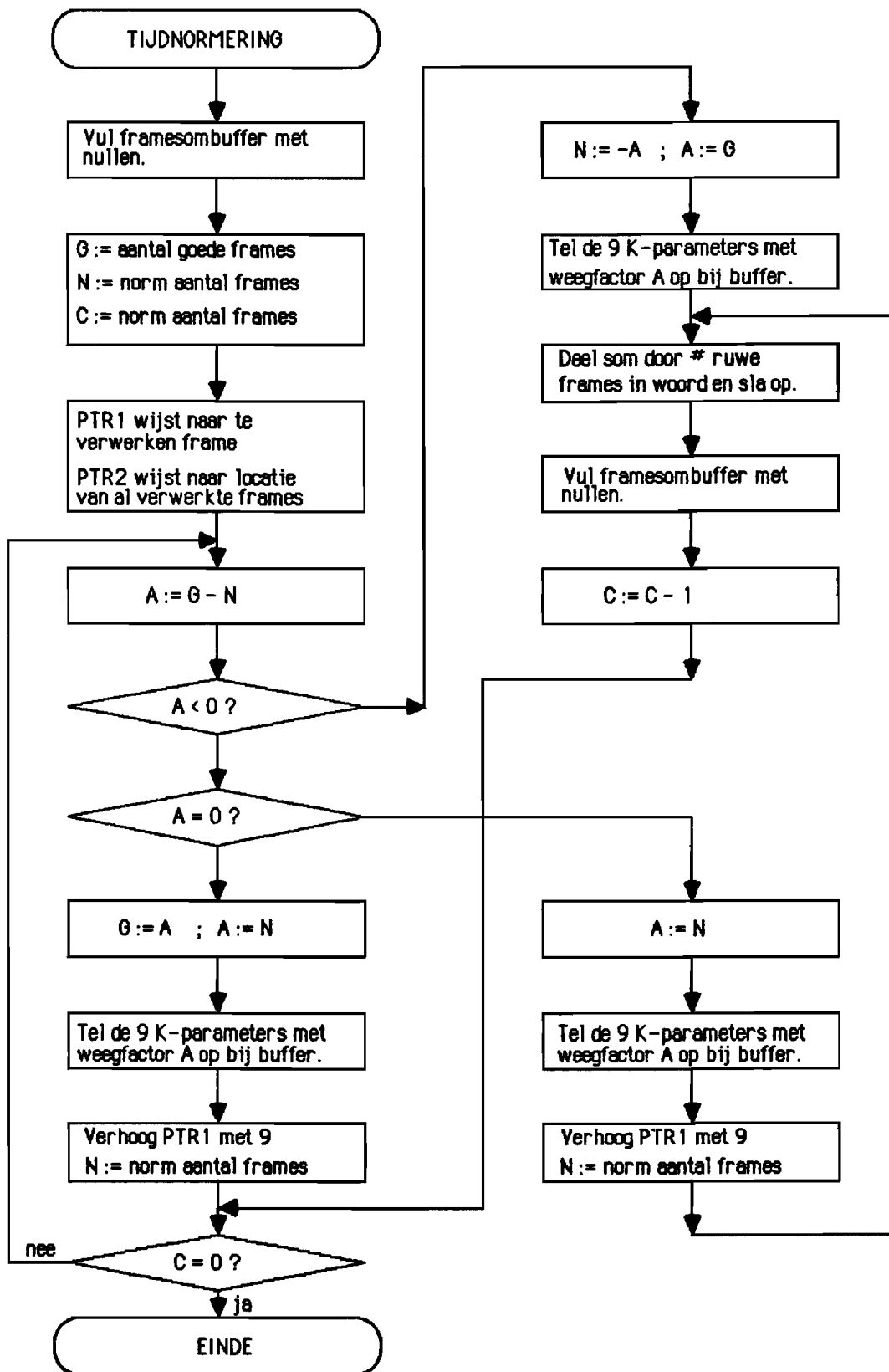
c) Finite State Machine Algoritme.....



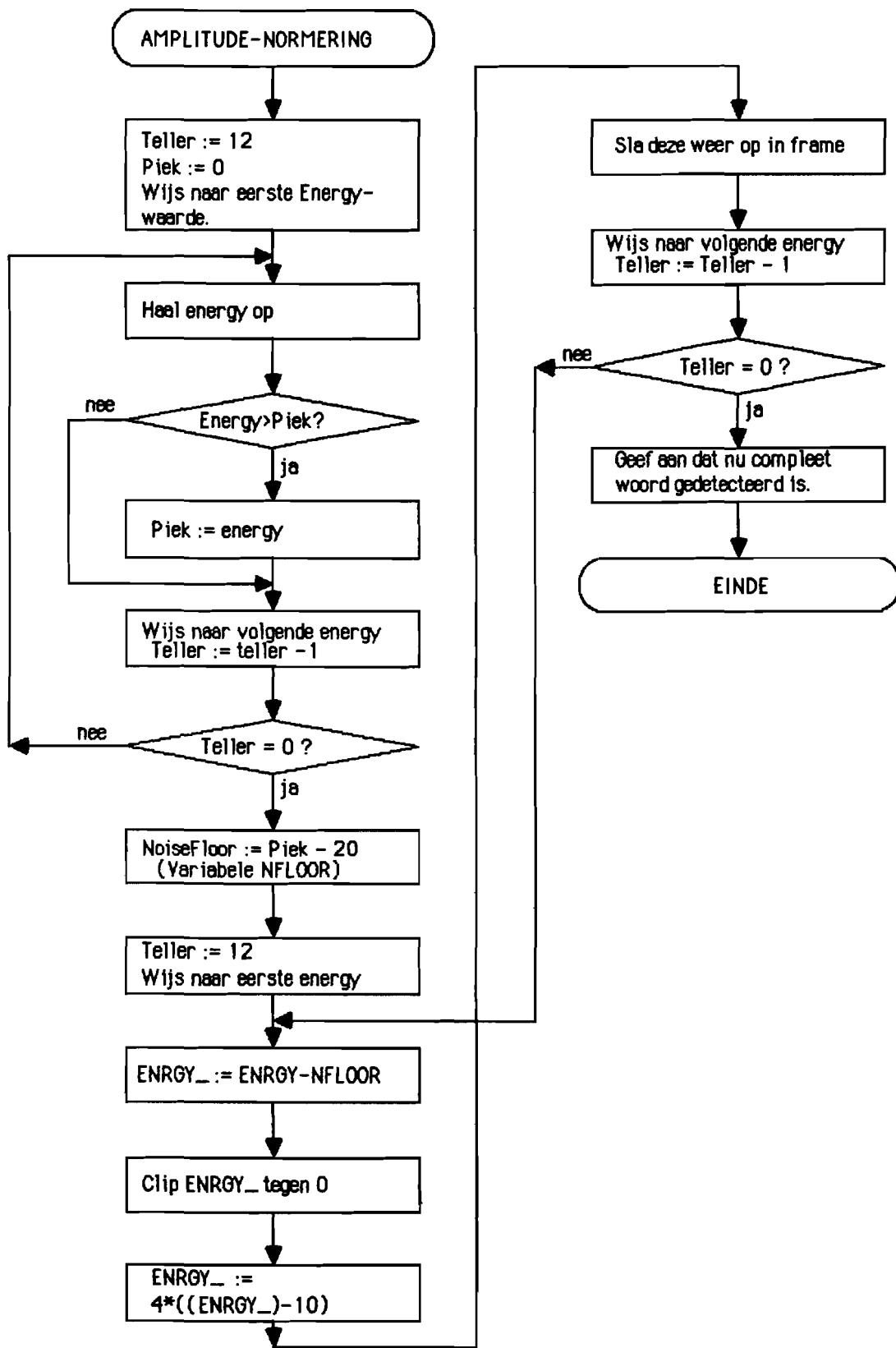
.....vervolg Finite State Machine Algoritme.....



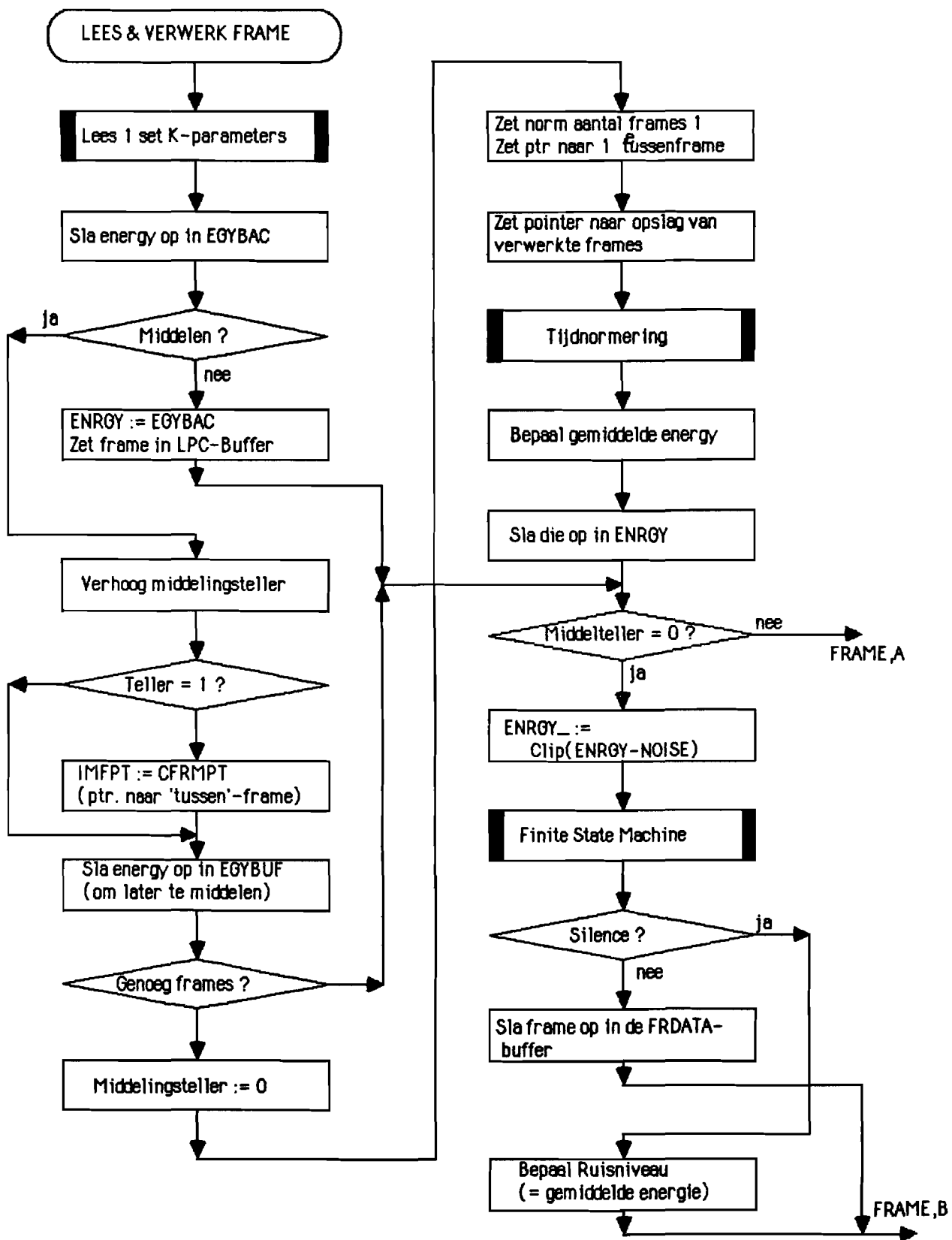
.....vervolg Finite State Machine Algoritme.



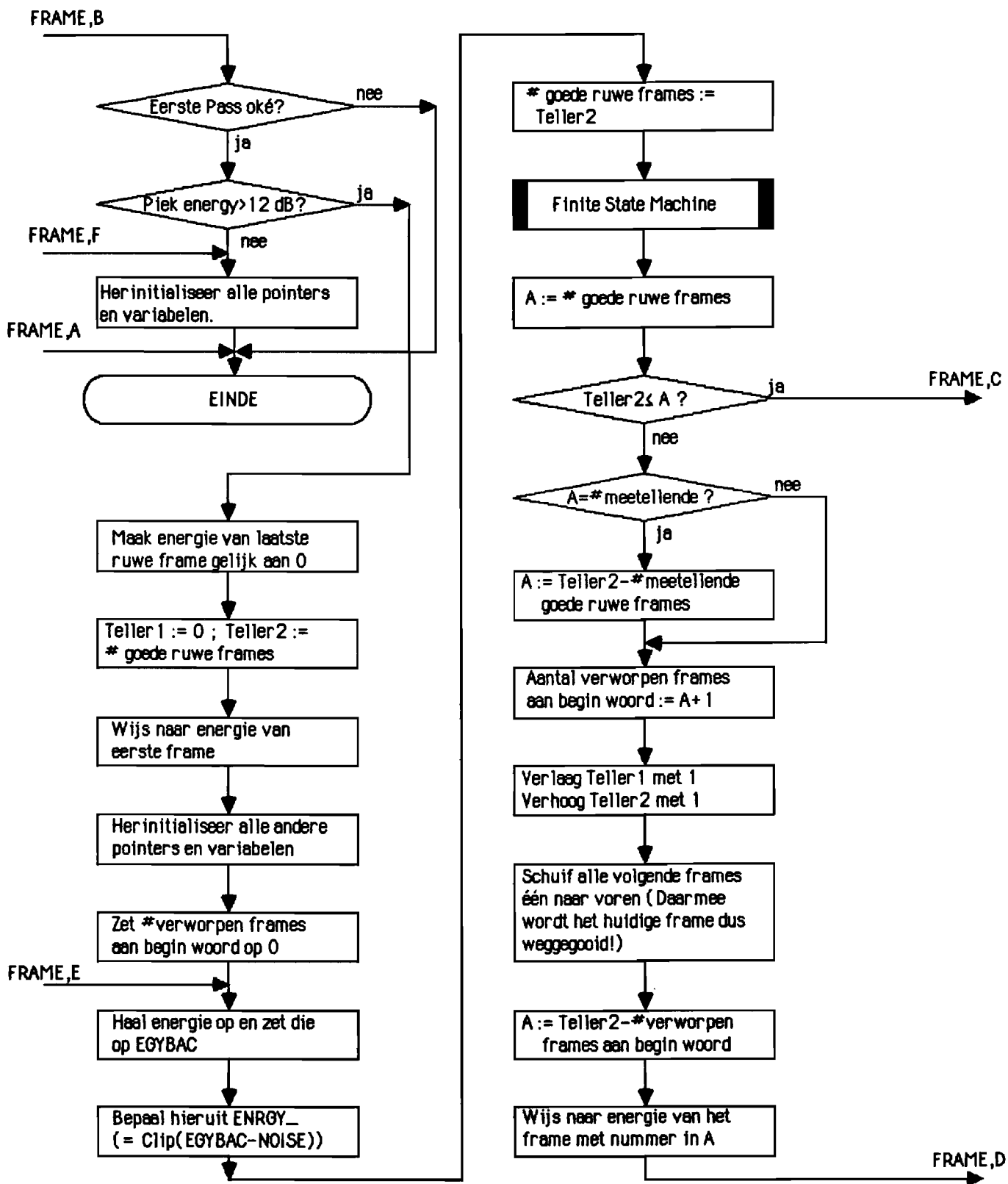
d) Tijdnormerings-algoritme



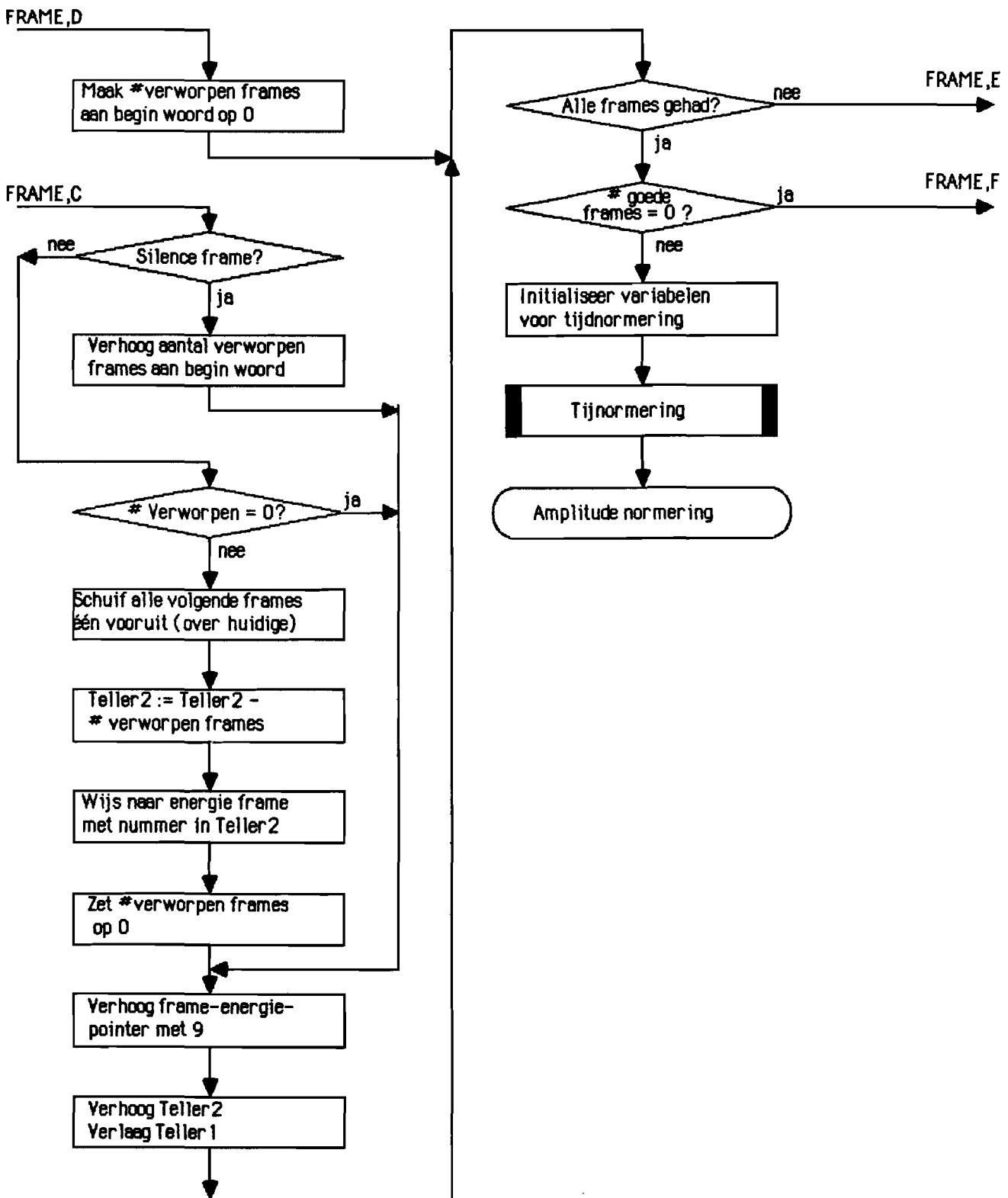
e) Amplitude-normering



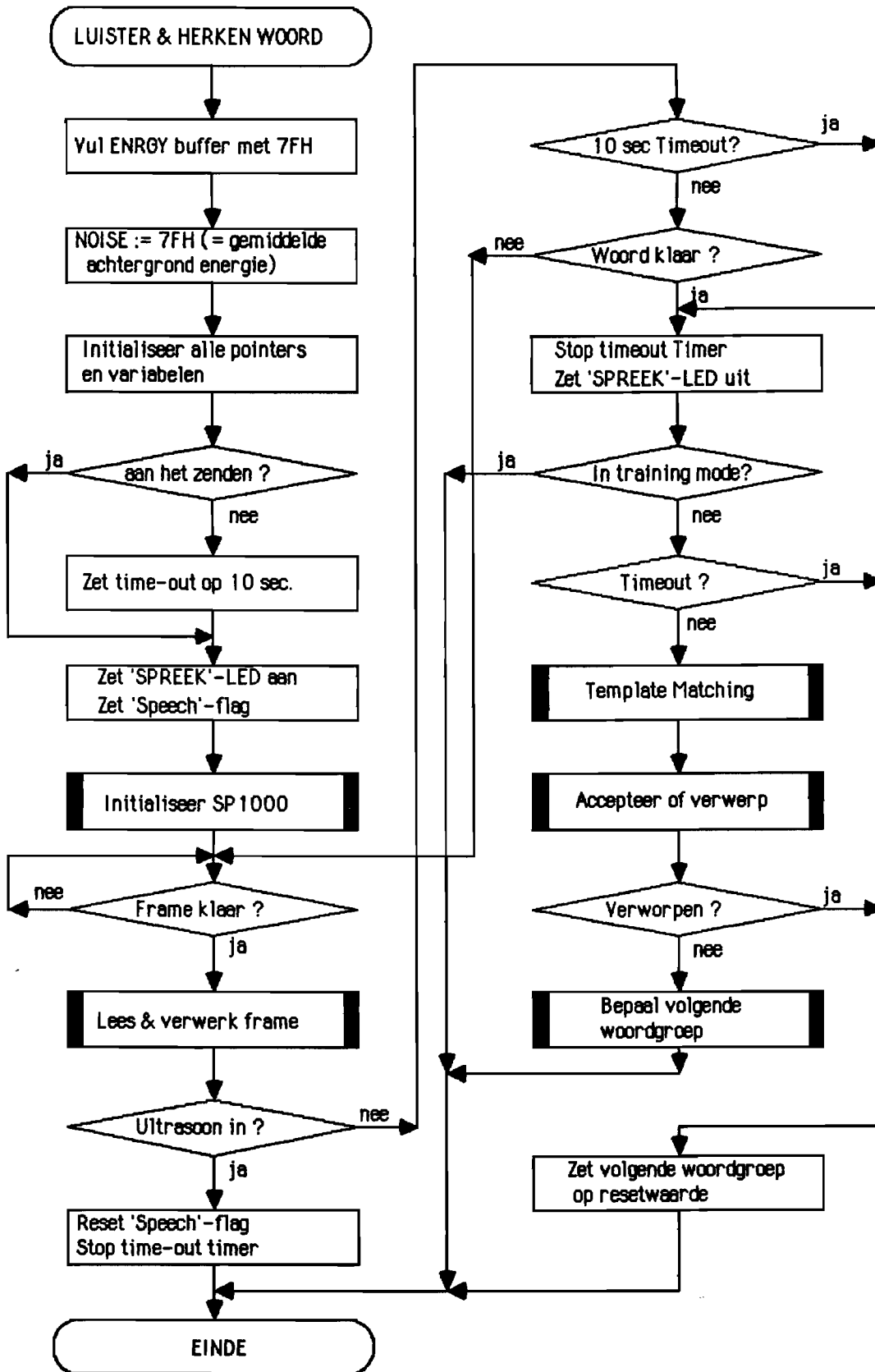
f) Lees een frame van de SP1000 en verwerk dat.....



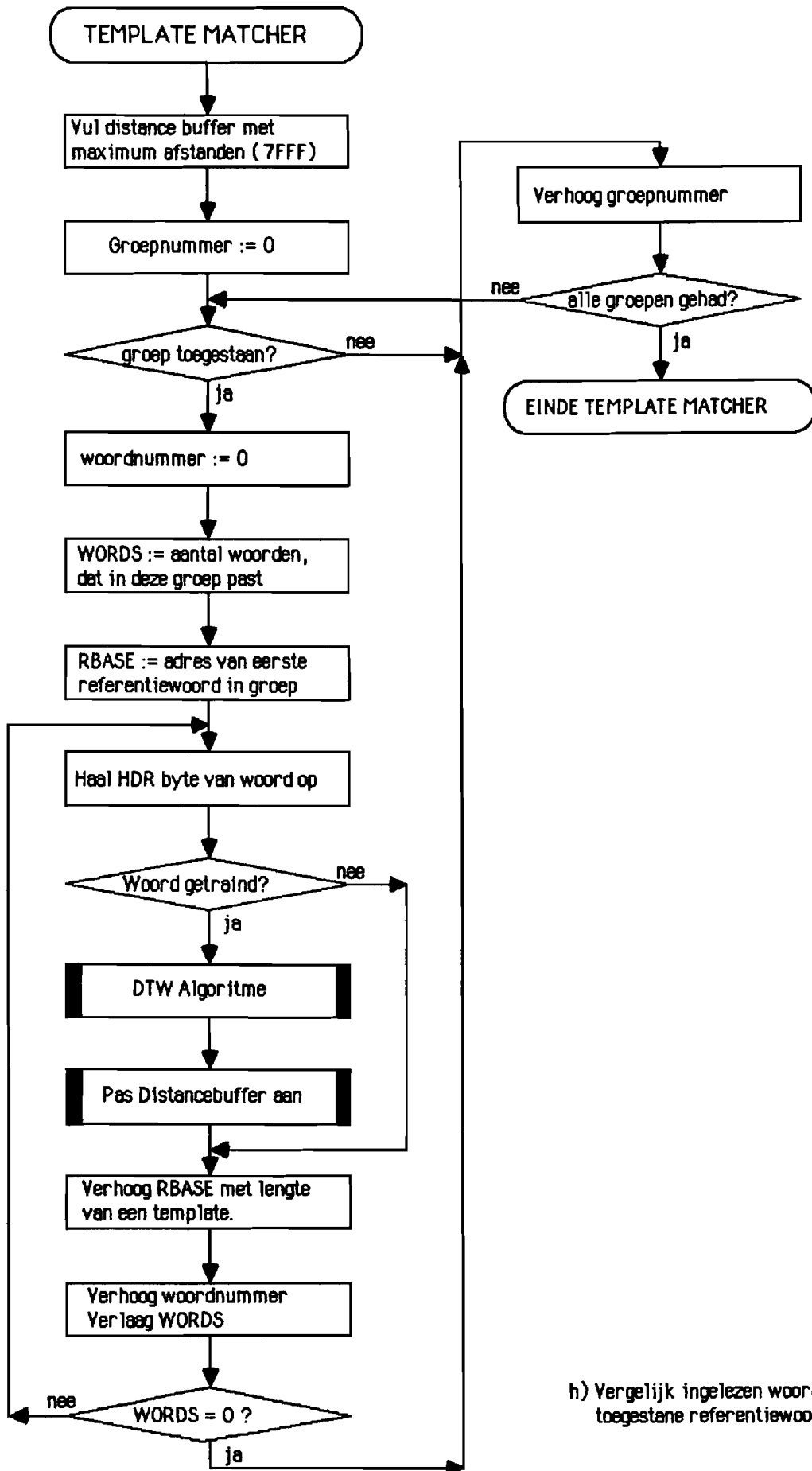
....vervolg Lees en Verwerk Frame ...



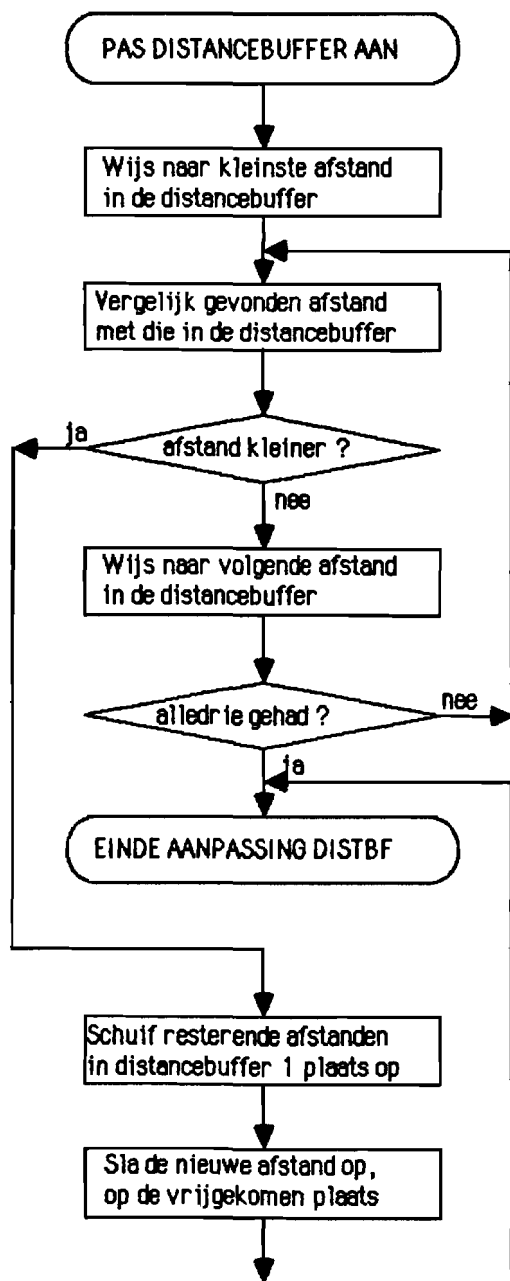
.....vervolg Lees en Verwerk Frame.



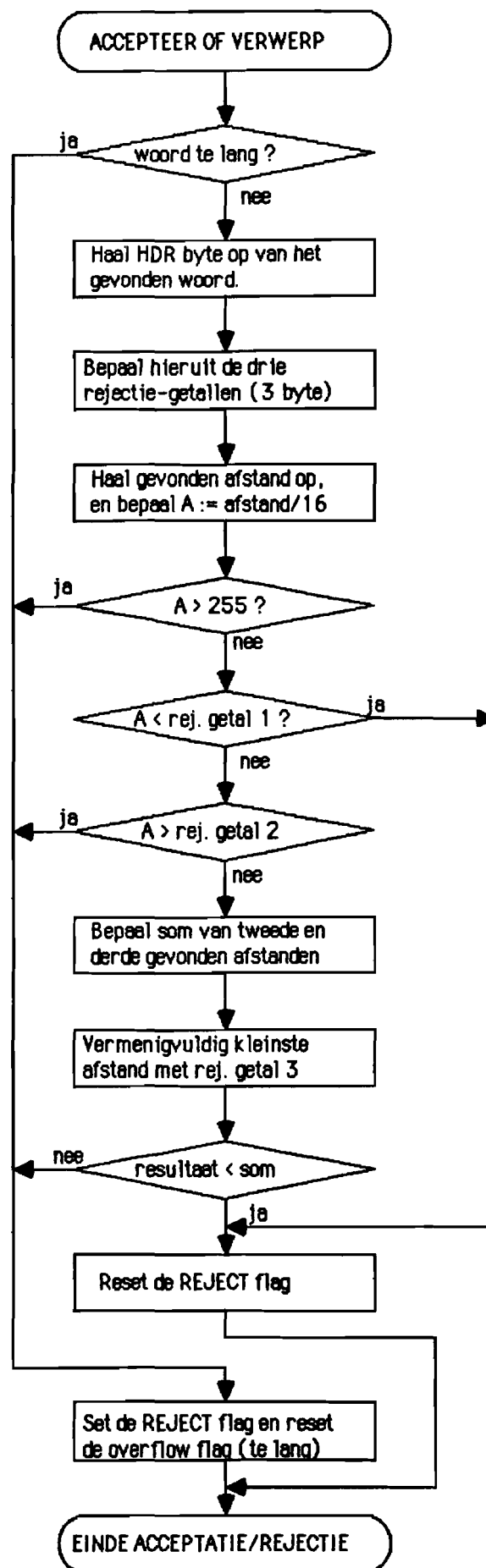
g) Luister naar spraak en probeer woord te detecteren en te herkennen



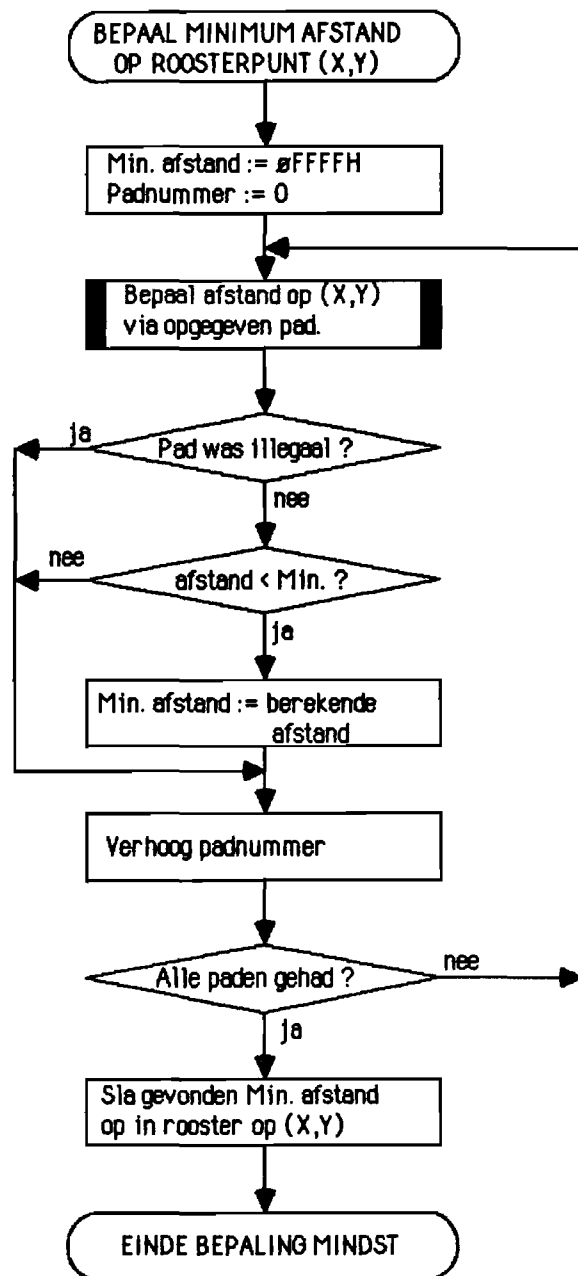
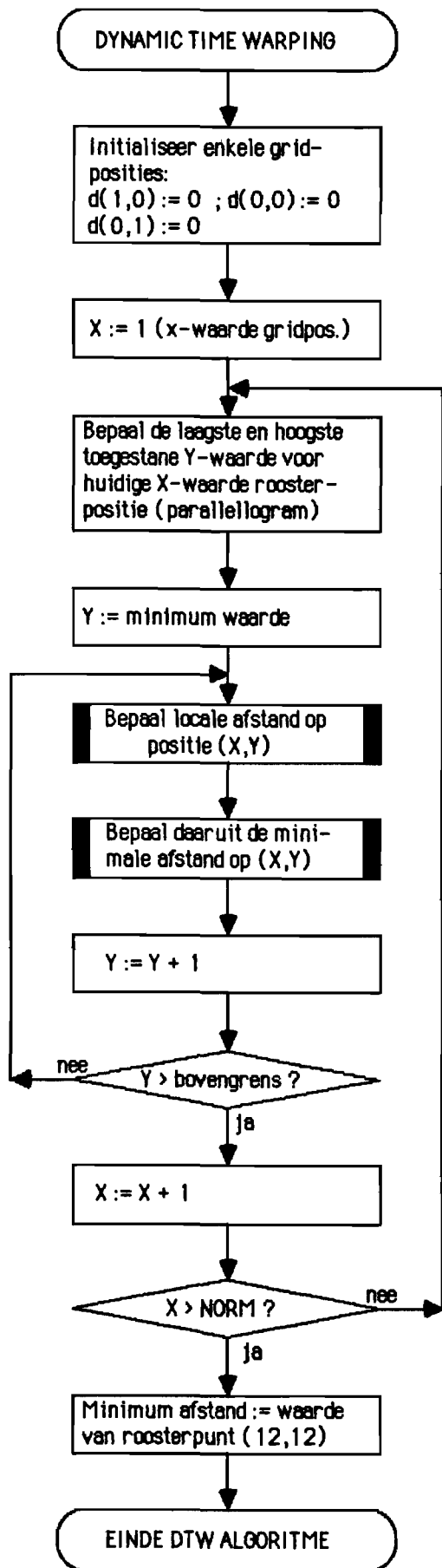
h) Vergelijk ingelezen woord met alle toegestane referentiewoorden.



1) Verwerk gevonden afstand in distance-buffer

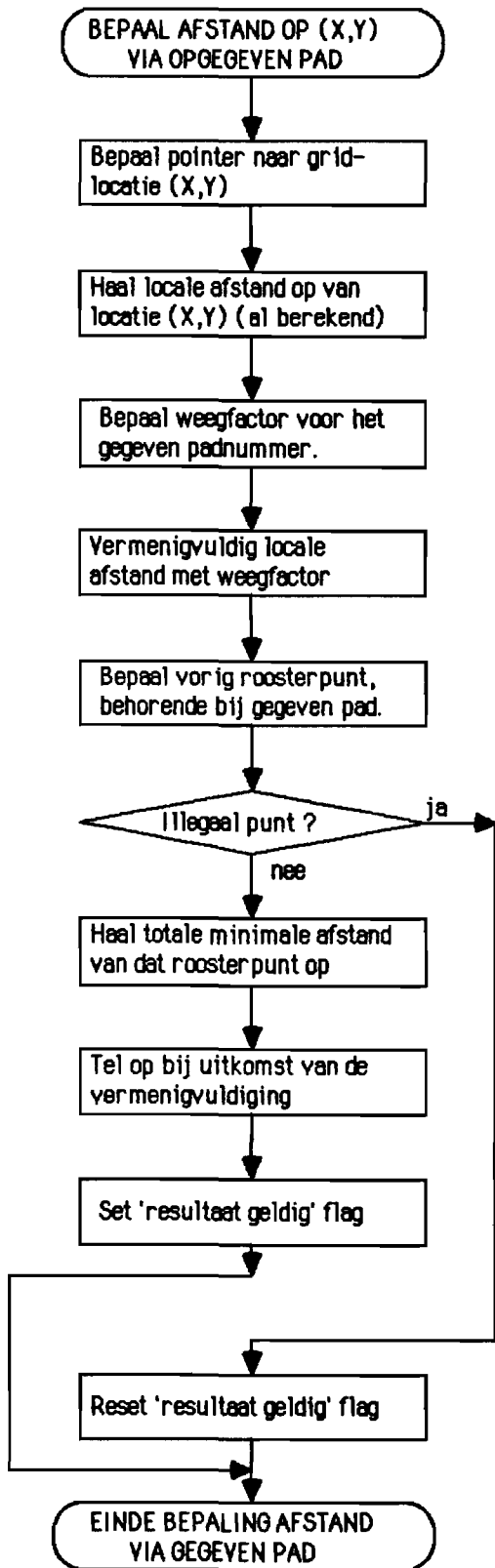


j) Test of gevonden minimale afstand binnen bepaalde grenzen ligt, en verwerp woord, indien niet.

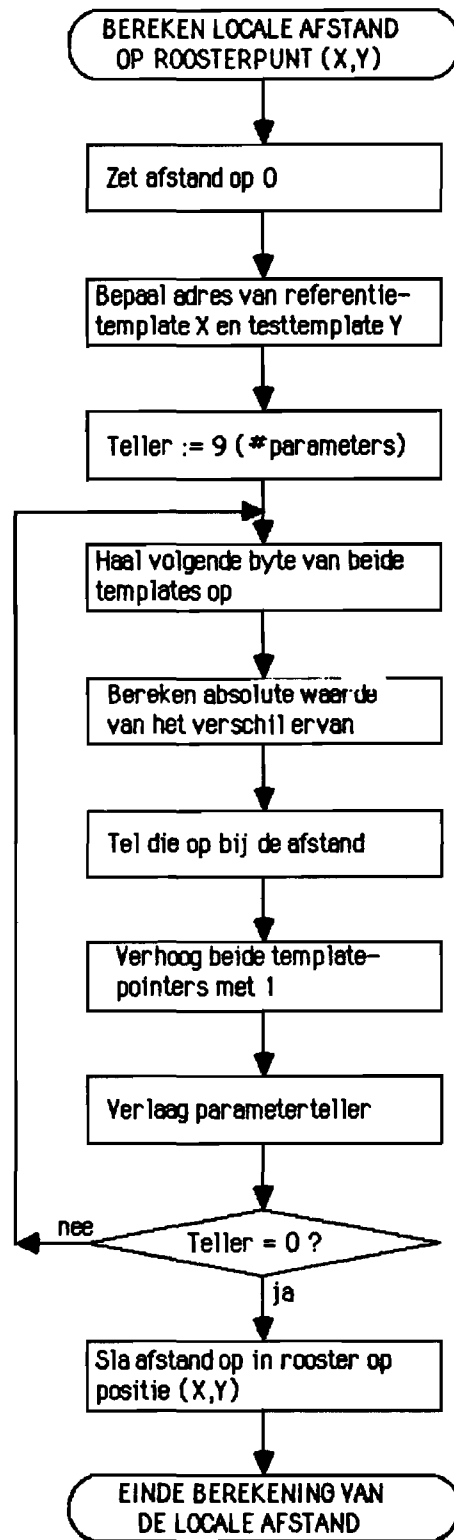


1) Bepaal minimum afstand van (1,1) tot (X,Y)

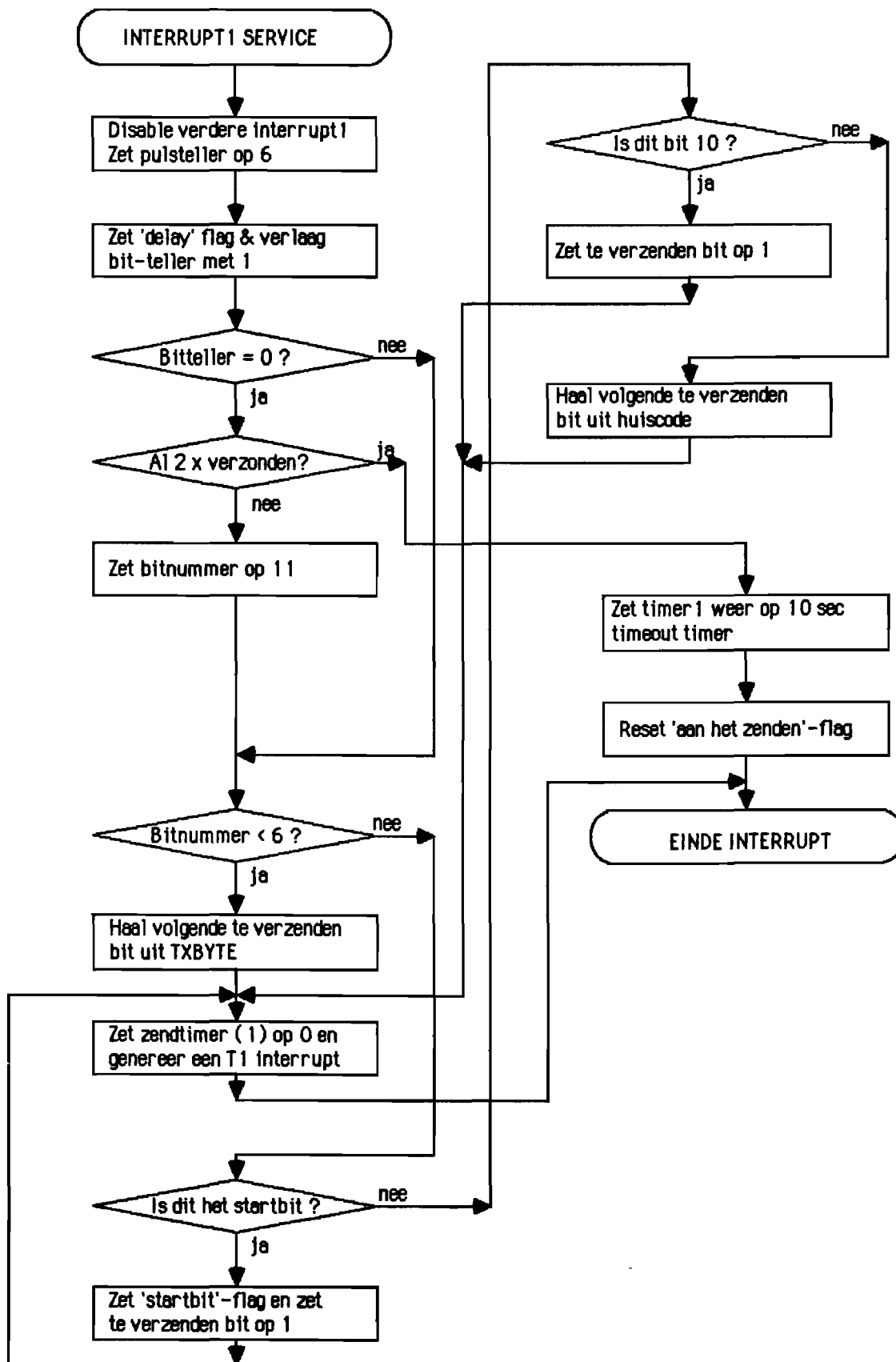
k) Bepaal afstand tussen gegeven test- en referentietemplate



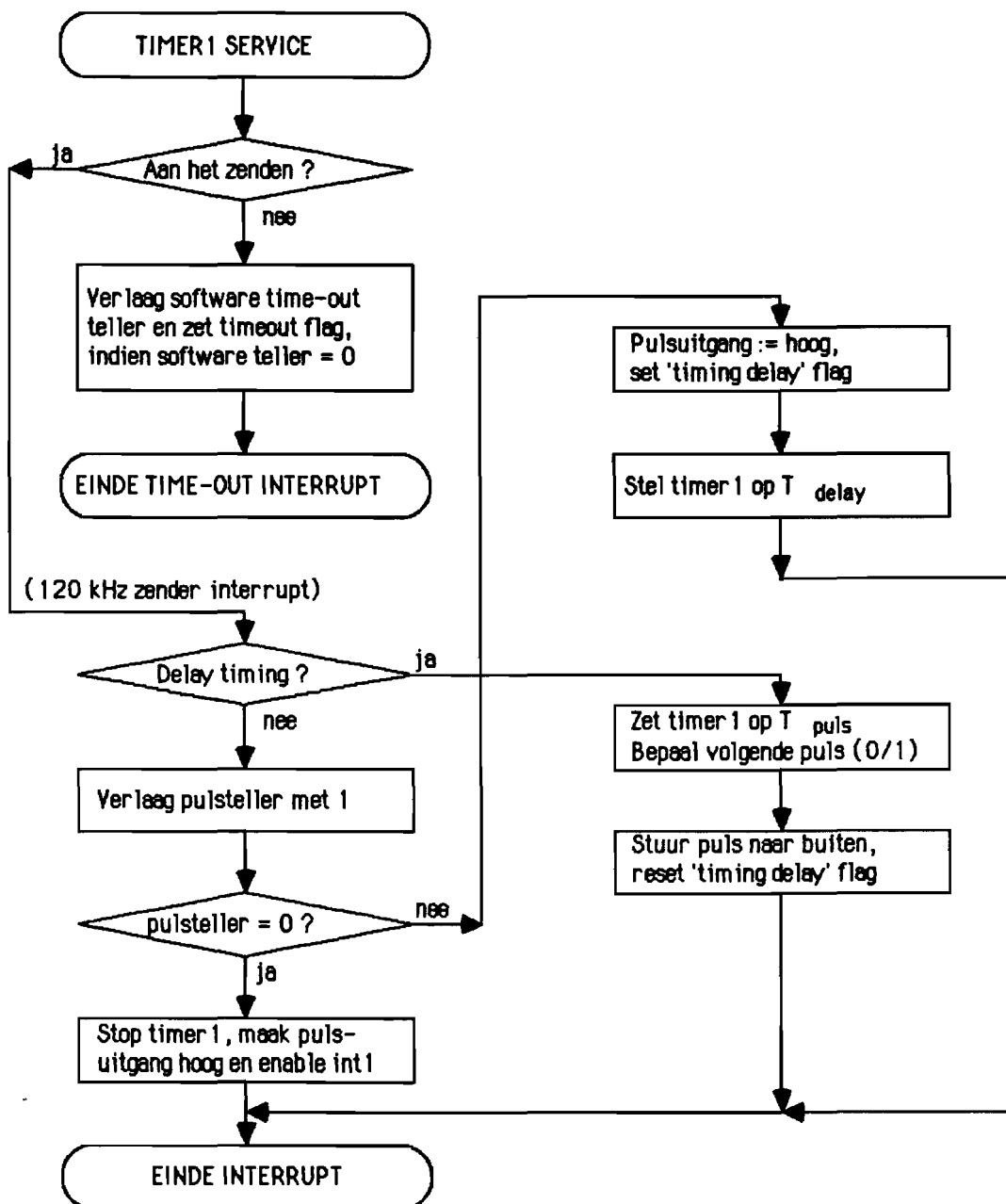
m) Bepaal afstand op (X,Y) via opgegeven pad



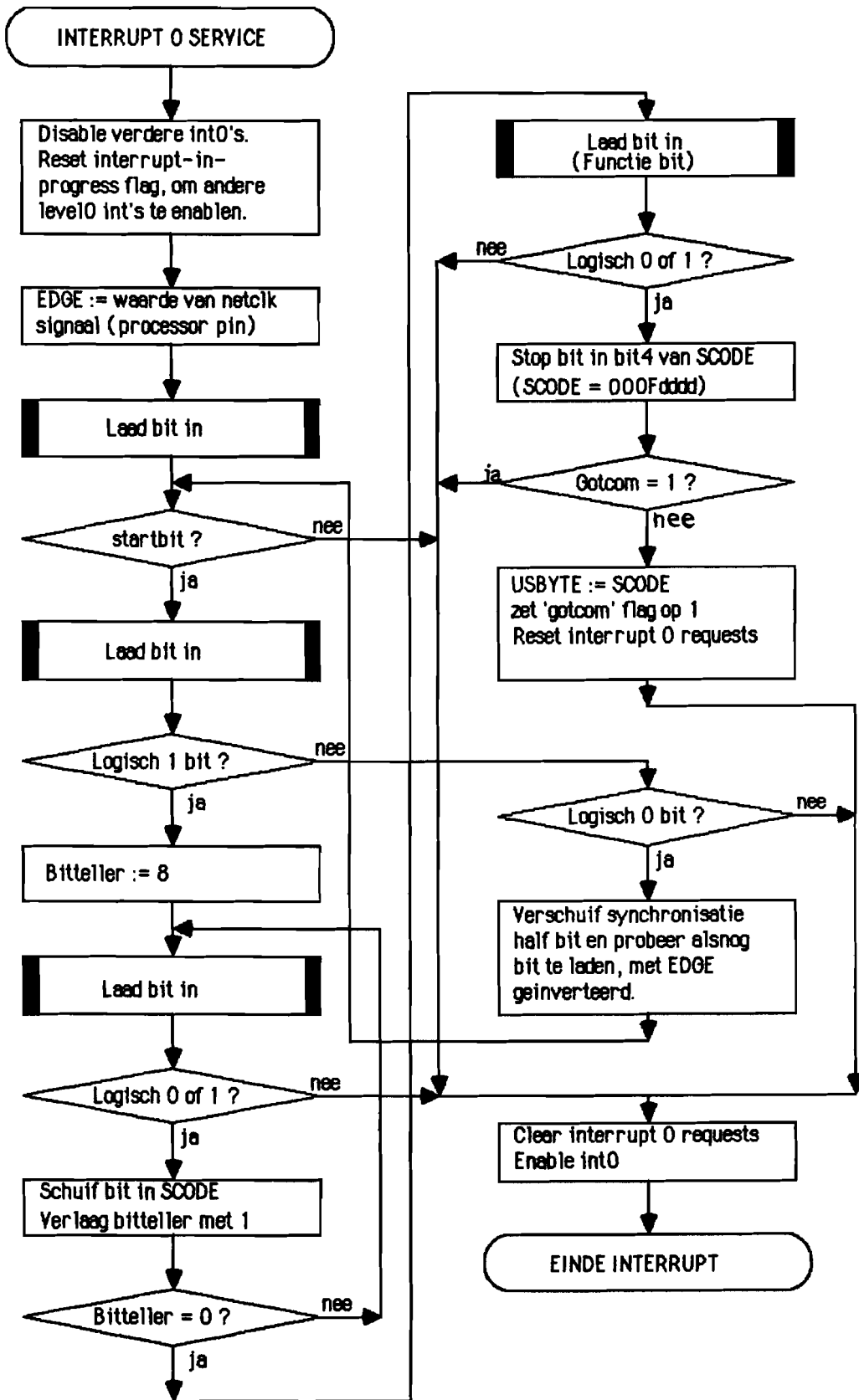
n) Bepaal afstand tussen testtemplate Y en referentietemplate X



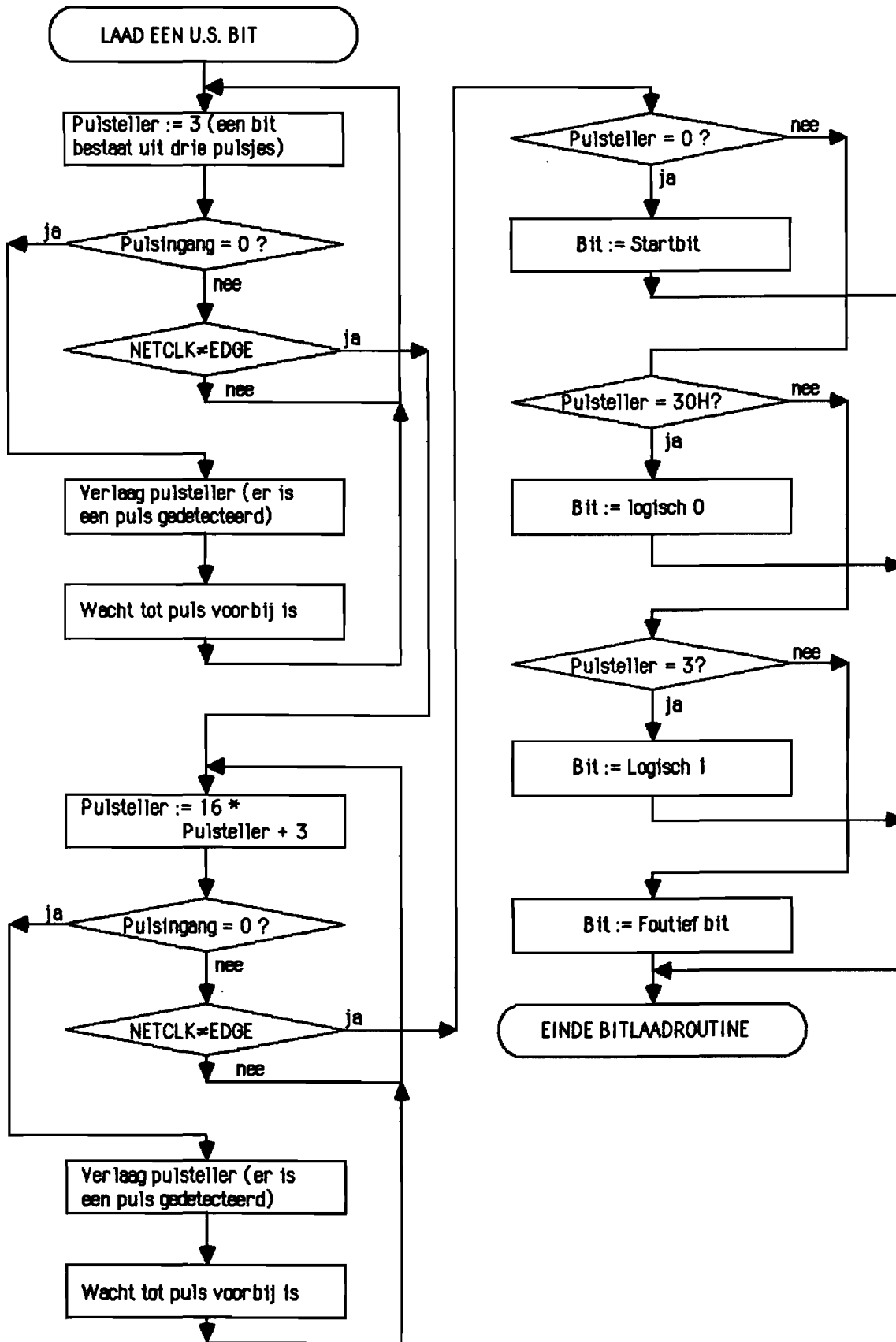
o) Verzend een besturingscode over het lichtnet (2x)



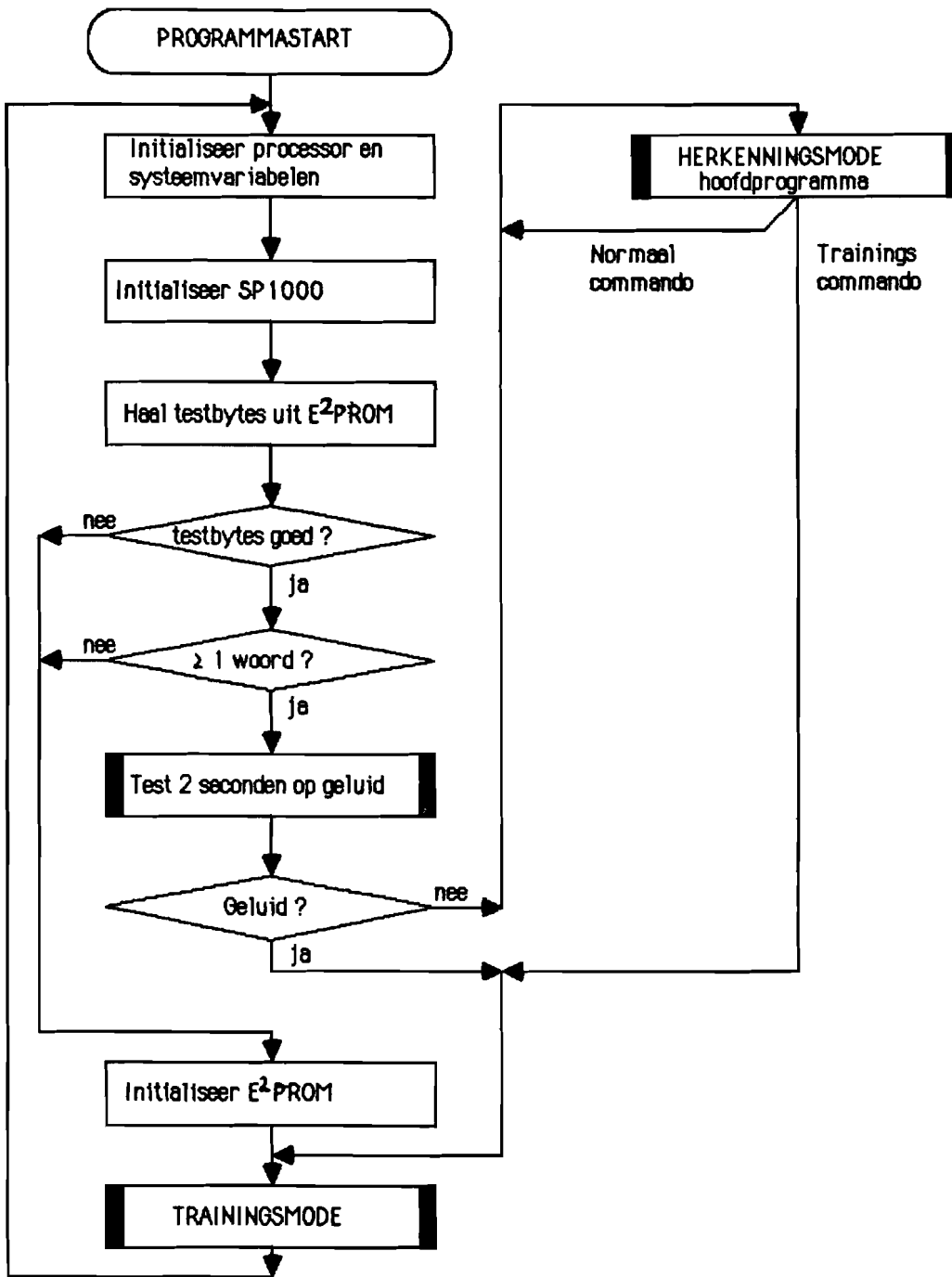
p) Veroorzaak Time-out na 10 seconden, of verzend één bit van een besturingscode over het lichtnet



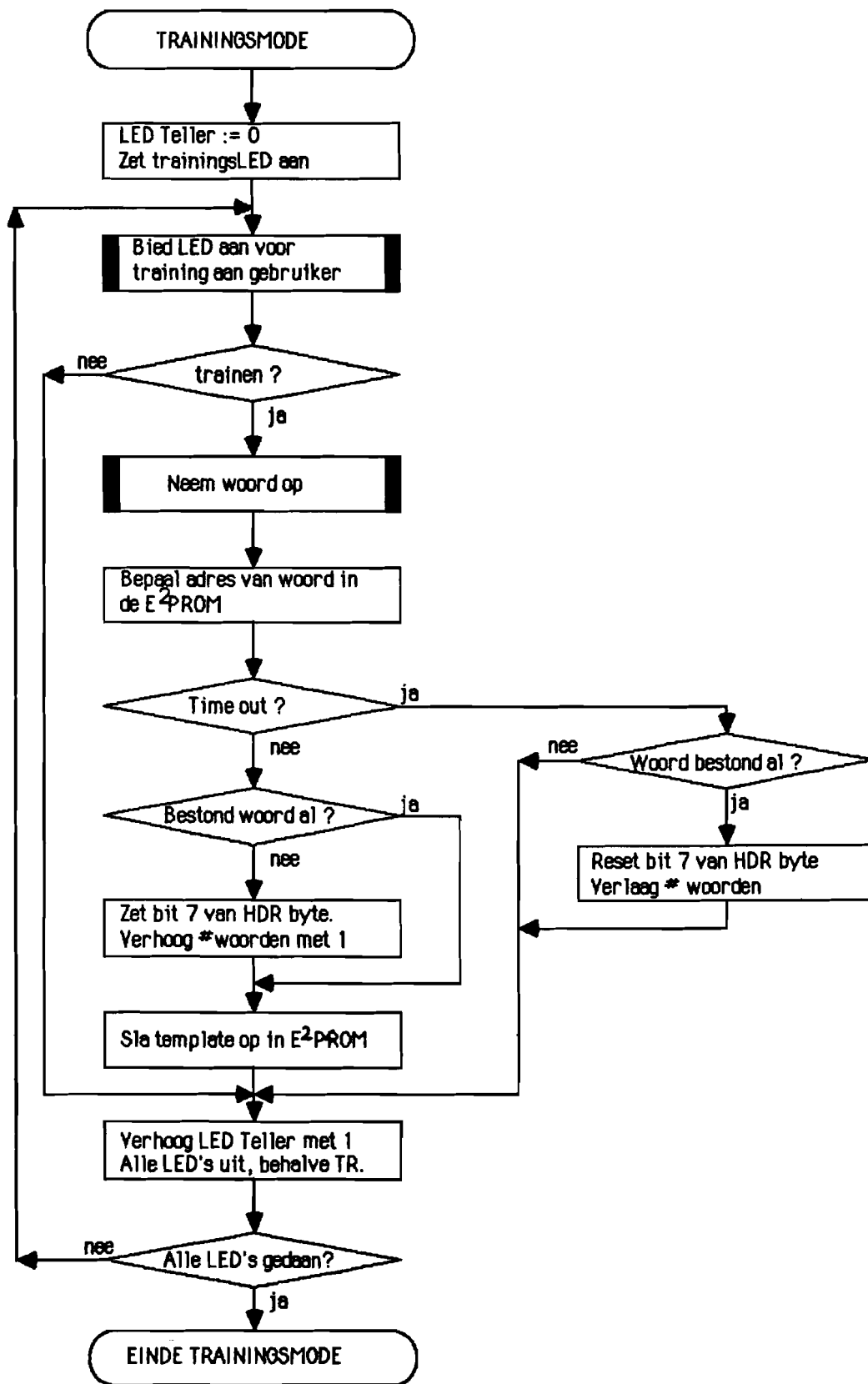
q) Leed een van de netbesturingspost komend serieel commando in



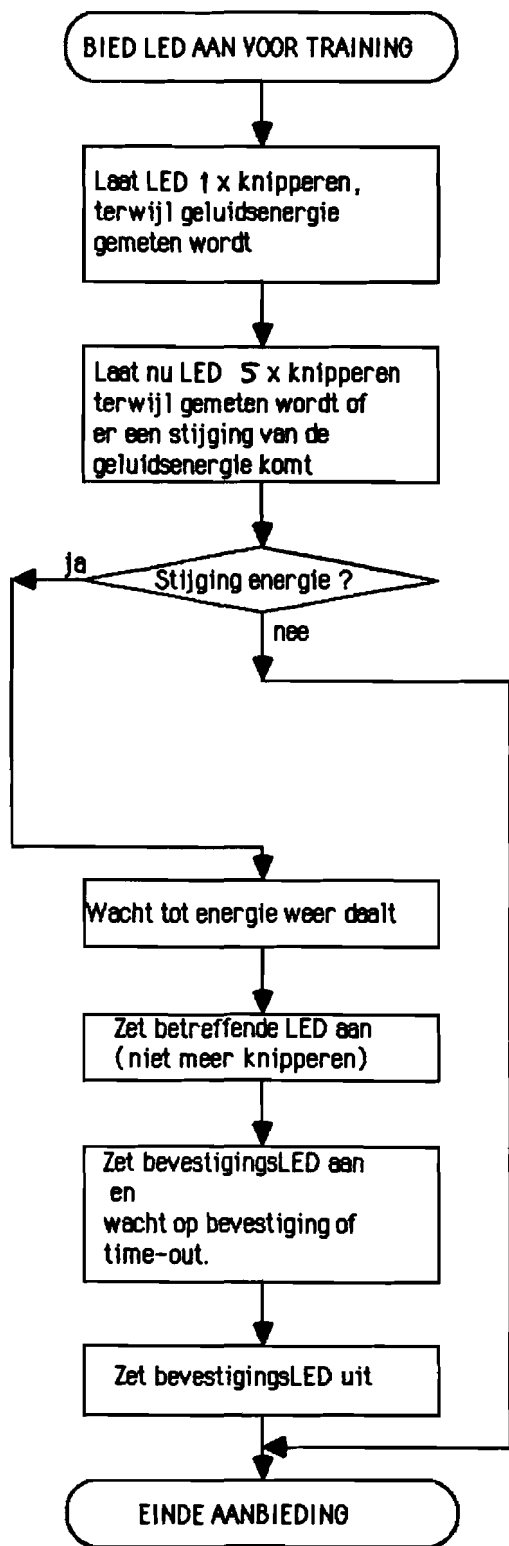
r) Laad één bit in van een seriële binnenkomend commando van de netbesturingspost



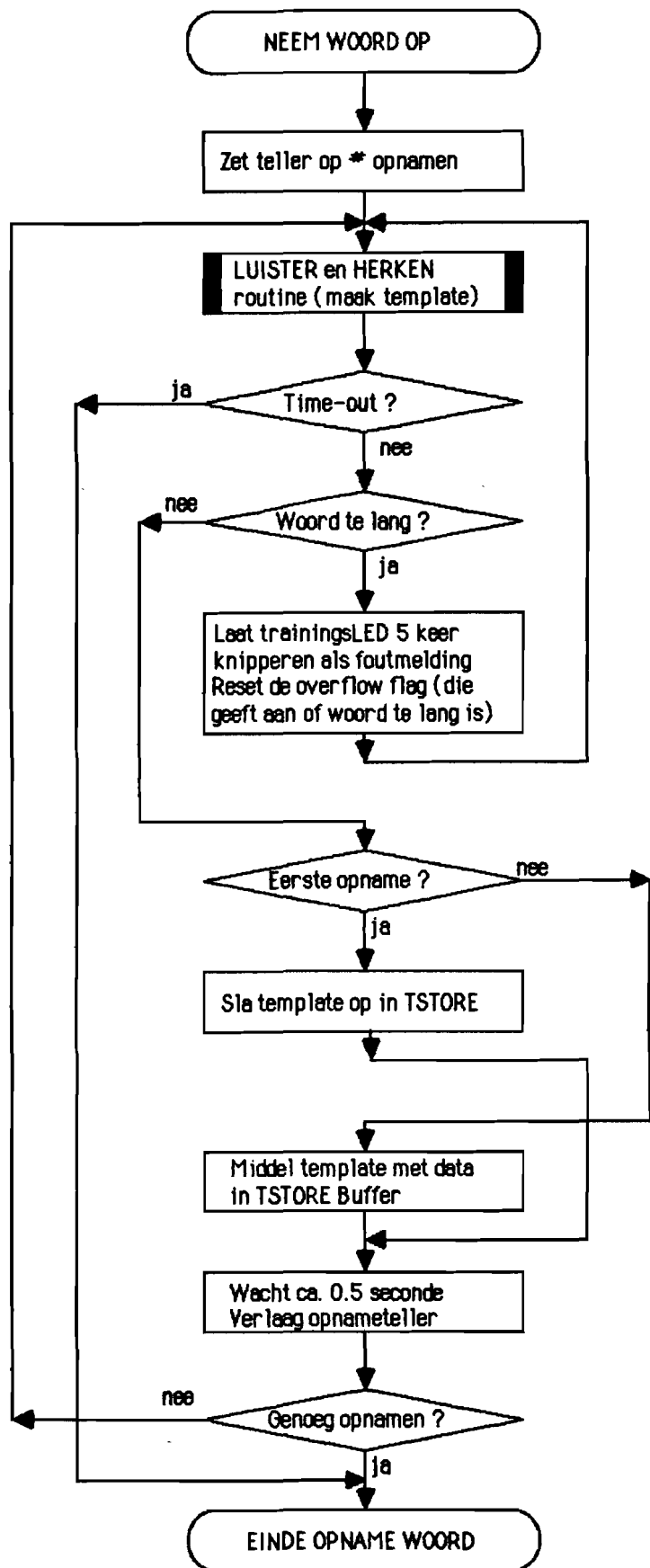
s) Het hoofdprogramma



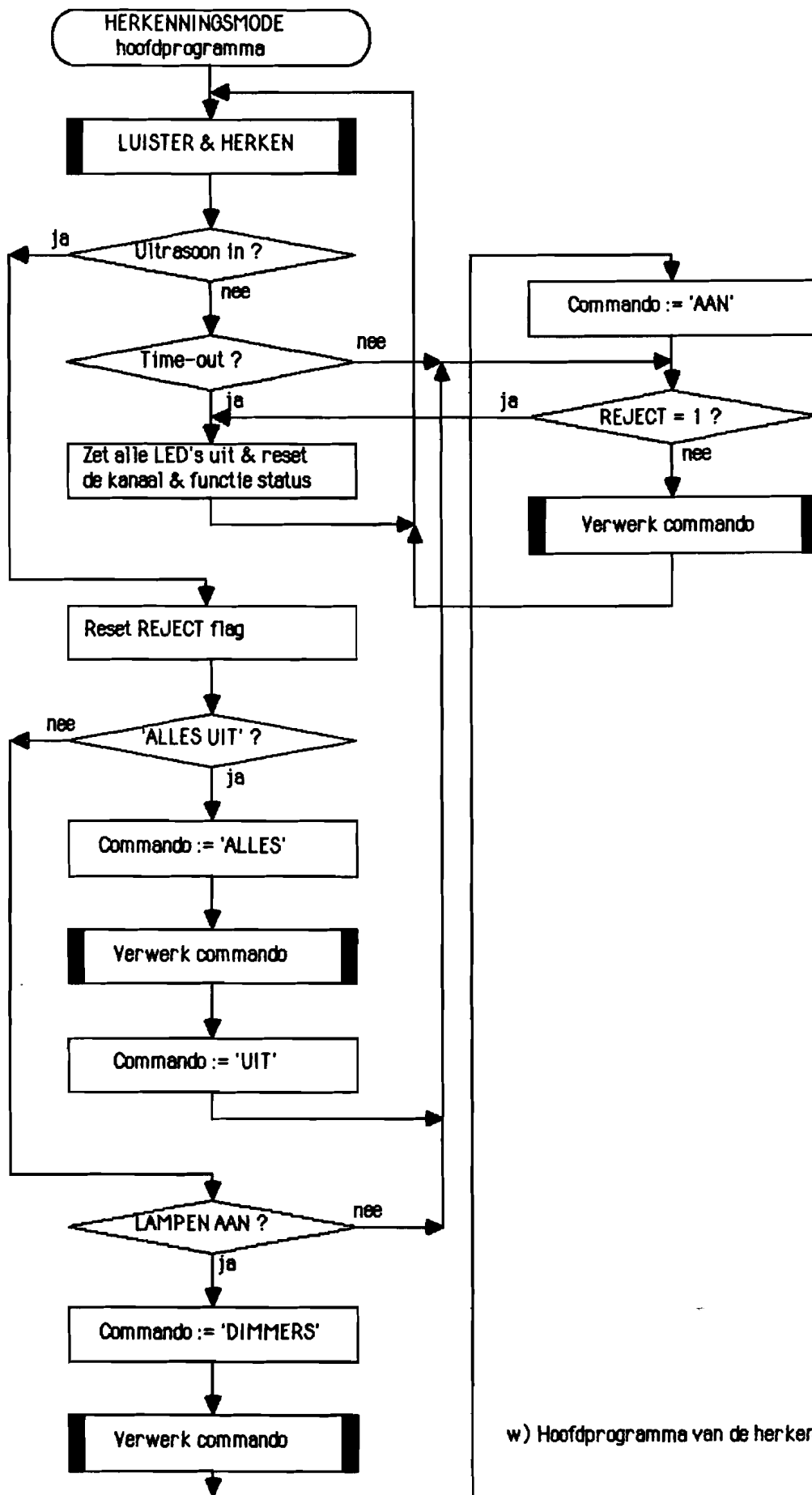
t) Hoofdroutine van het trainingsprogramma



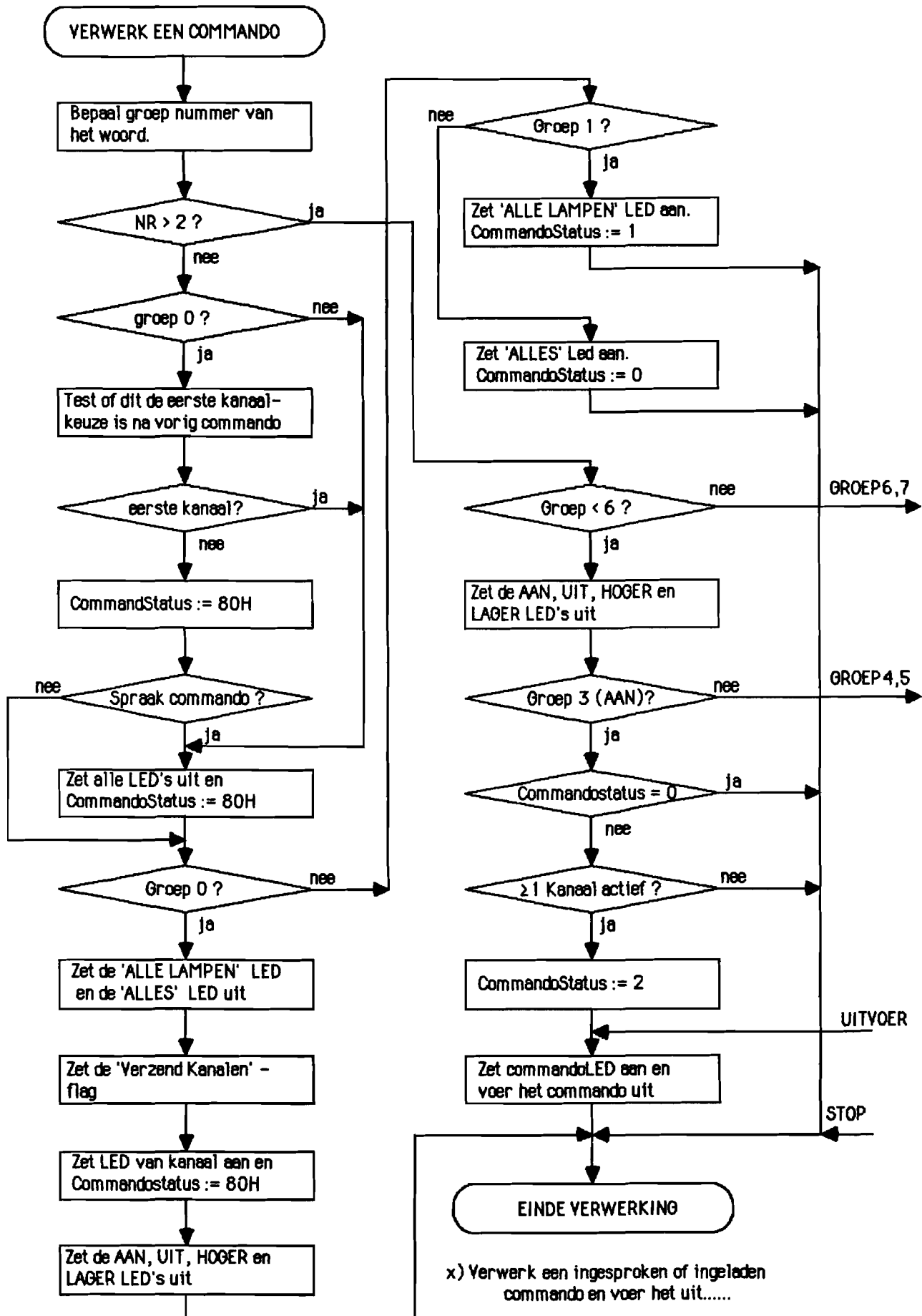
u) Bied een kanaal of functie aan voor training en test of de gebruiker dit woord wil trainen



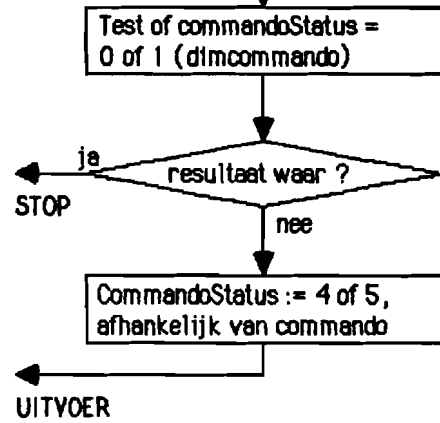
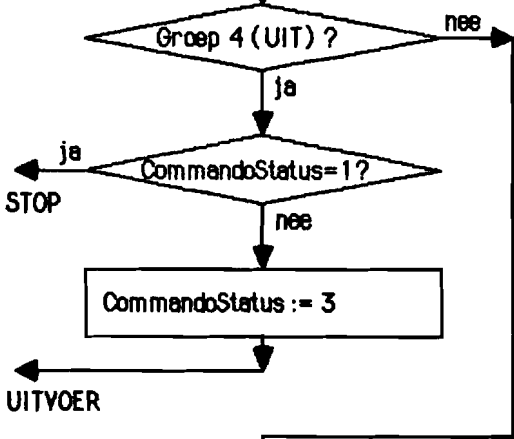
v) Neem een woord op (middel 2 opnames)



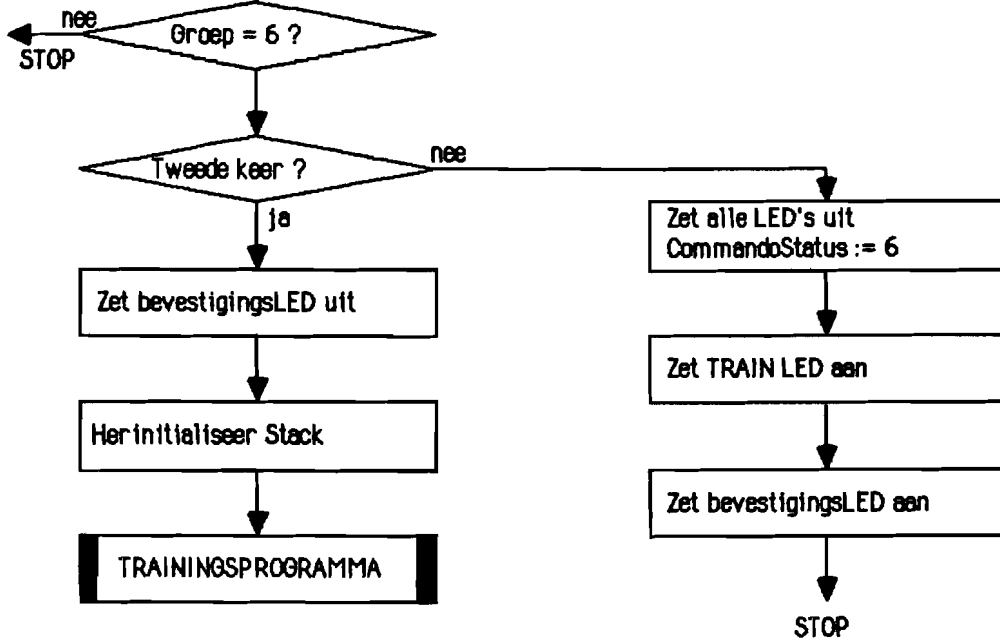
w) Hoofdprogramma van de herkenningmode



GROEP 4,5



GROEP 6,7



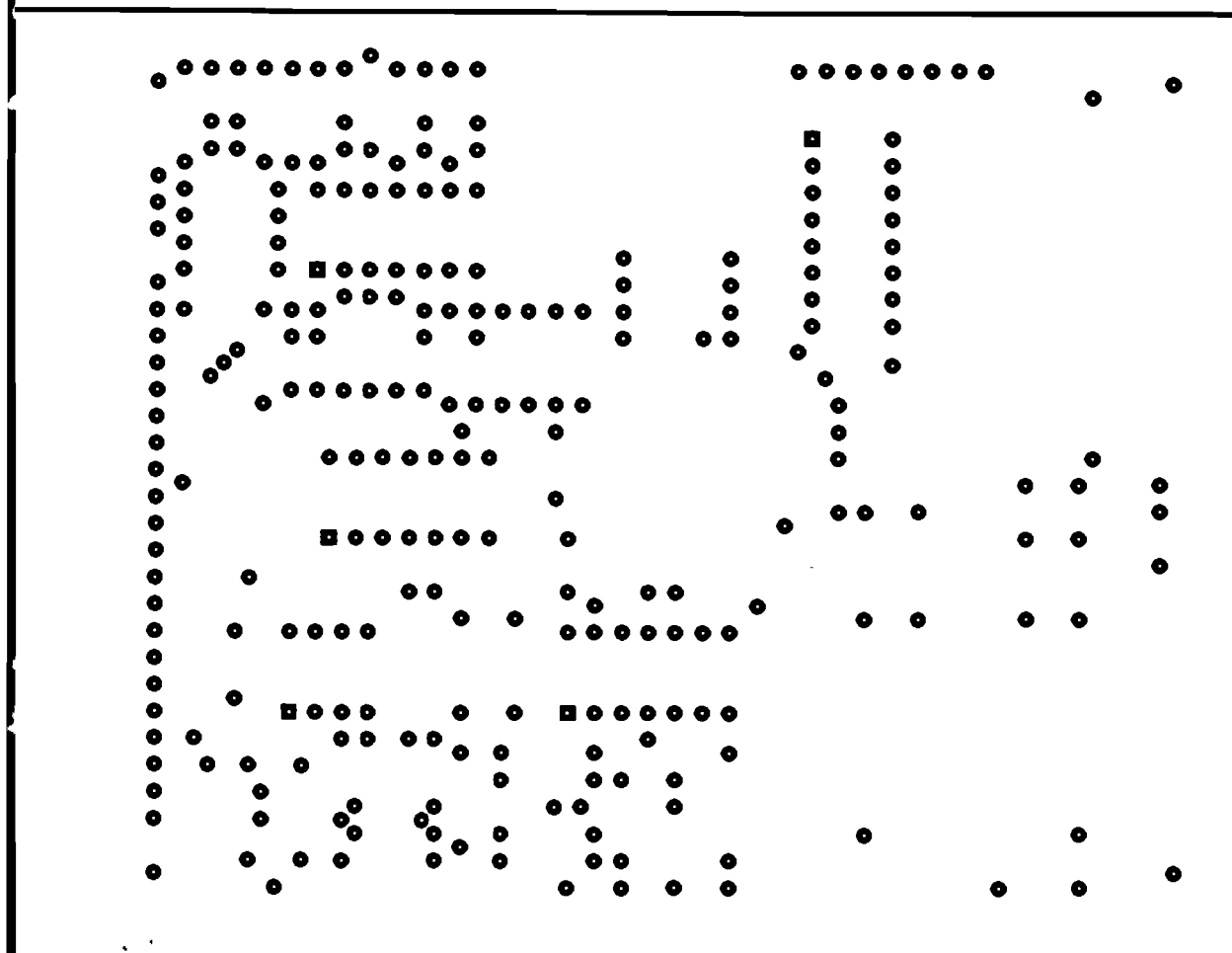
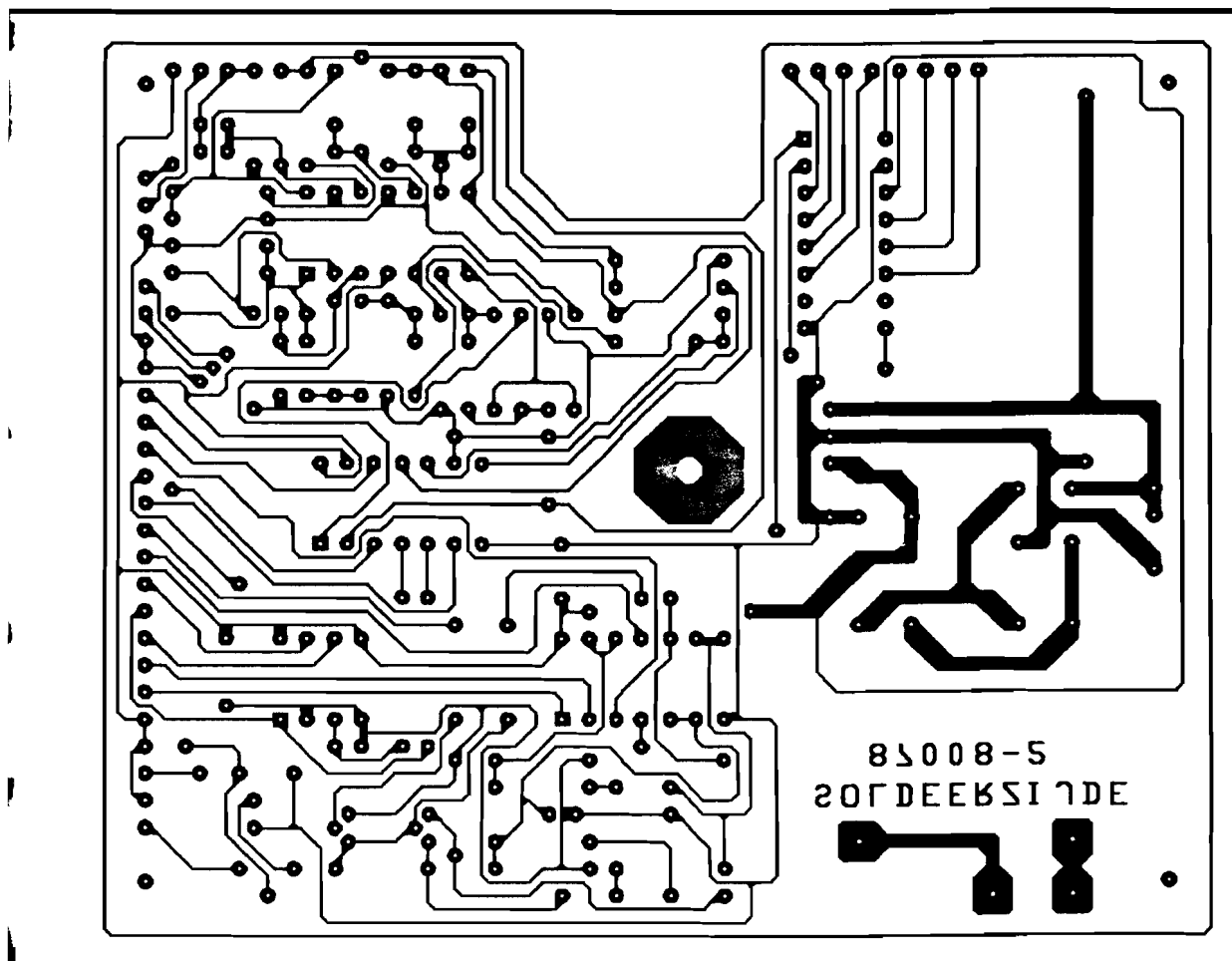
.....vervolg Verwerking Comando.

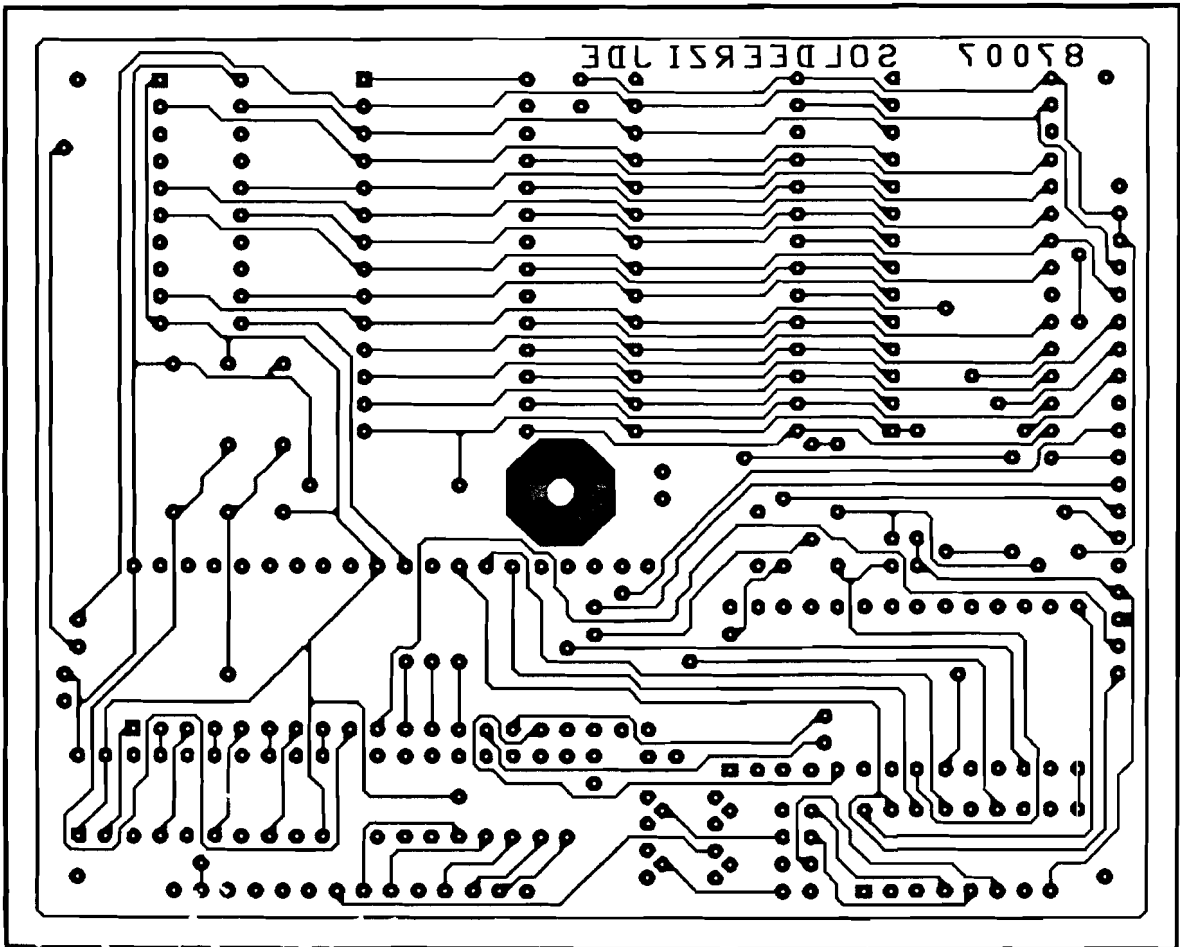
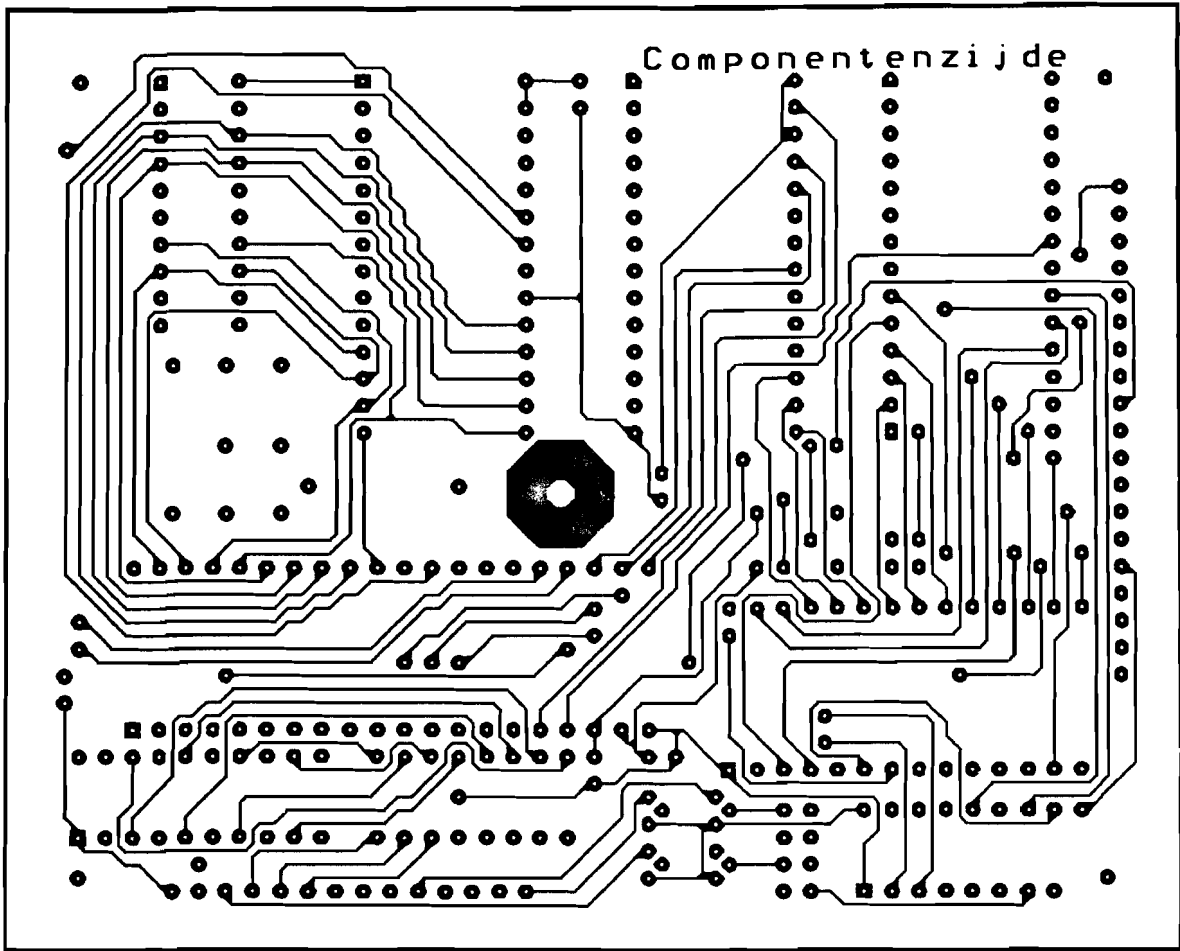
BIJLAGE 3

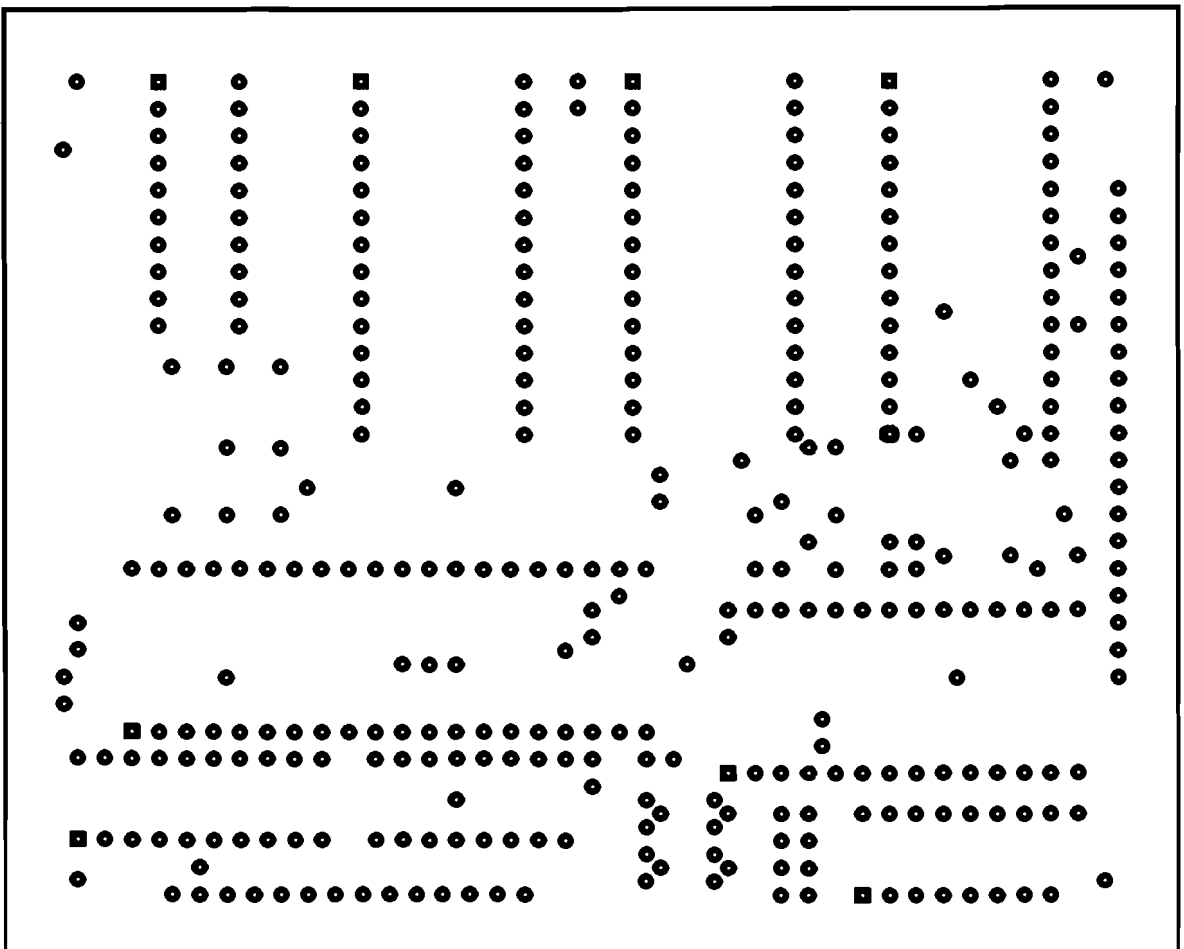
ONDERDELENLIJST

<u>IC's</u>	<u>Condensatoren</u>	<u>Weerstanden</u>
7805	1 x 18 pF	8 x 82 Ω
LM324	2 x 22 pF	1 x 1 KΩ
4066	1 x 39 pF	1 x 2K2
4039	1 x 200 pF	4 x 4K7
40373	1 x 1 nF	1 x 5K6
74HCT139	1 x 1.2 nF	4 x 6K8
74HCT165	1 x 2.2 nF	1 x 8K2
74HCT244	1 x 3.3 nF	5 x 10 KΩ
ADC0831	4 x 10 nF	1 x 11 KΩ
SP1000	1 x 22 nF	5 x 22 KΩ
28C64	7 x 100 nF	2 x 27 KΩ
27C64	1 x 10 μF	1 x 39 KΩ
64C64	1 x 100 μF / 6.3V	2 x 56 KΩ
80C31	1 x 470 μF / 25V	1 x 68 KΩ
		1 x 91 KΩ
		6 x 100 KΩ
		1 x 110 KΩ
		1 x 120 KΩ
		2 x 1 MΩ
		1 x 4K7 Instel
<u>Diversen</u>		
X-tal 12 MHz		
X-tal 7.16 MHz		
Printtrafo type 307-2 (7.5V)		
Electret microfoon	4 x BC 416	
Microfoon plug	2 x BC414	
Busch Timac X-10 systeem	1 x Bruggelijkrichter B250C800	
Print 87008 (analoog deel)	25 x Rode LED, high efficiency	
Print 87007 (digitaal deel)	1 x BZX 79C 9V1	
Print 86025 (LEDpaneel)		
	<u>Diversen Halfgeleiders</u>	

BIJLAGE 4. PLOT VAN DE PRINTLAYOUT







**BILAGE 5. FREQUENTIE-KARAKTERISTIEK VAN DE
ANALOGE INANGSTRAP**

G	V _i (dBV)	200 Hz	500 Hz	1000 Hz	1500 Hz	2000 Hz	2500 Hz	3000 Hz	4000 Hz	8000 Hz
0	-15	-27	-12.8	-6.7	-3.1	-2.1	-2.4	-3.5	-6.2	-16.1
1	-20	-26.8	-12.4	-6	-2.6	-1.4	-1.6	-2.5	-5.4	-15.5
2	-27	-27.8	-13.4	-7	-3.2	-2.1	-2.3	-3.2	-6.2	-16.1
3	-33	-28.4	-13.6	-7.1	-3.7	-2.4	-2.6	-3.3	-6.4	-16.7
4	-39	-31.4	-15.5	-9.2	-5.7	-4.7	-5.1	-6.4	-9.7	-22.6
5	-44	-30.4	-14.4	-8.2	-4.9	-3.8	-4	-5.2	-9.2	-22.1
6	-48	-28.4	-13	-6.4	-2.9	-2.1	-2.5	-3.9	-7.4	-20.4
7	-54	-27.3	-13.1	-6.5	-3.3	-2.2	-2.5	-3.8	-6.2	-20.4

a) Gemeten uitgangsspanningen als functie van de frequentie, de digitaal AGC code en de ingangsspanning in dBV.

G	200 Hz	500 Hz	1000 Hz	1500 Hz	2000 Hz	2500 Hz	3000 Hz	4000 Hz	8000 Hz
0	-12	2.2	8.3	11.9	12.9	12.6	11.5	8.8	-1.1
1	-6.8	7.6	14	17.4	18.6	18.4	17.5	14.6	4.5
2	-0.8	13.6	20	23.8	24.9	24.7	23.8	20.8	10.9
3	4.6	19.4	25.9	29.3	30.6	30.4	29.7	26.6	16.3
4	7.6	23.5	29.8	33.3	34.3	33.9	32.6	29.3	16.4
5	13.6	29.6	35.8	39.1	40.2	40	38.8	34.8	21.9
6	19.6	35	41.6	45.1	45.9	45.5	44.1	40.6	27.6
7	26.7	40.9	47.5	50.7	51.8	51.5	50.2	47.8	33.6

b) Versterkingsfactor als functie van frequentie en digitale AGC code in dB.