

MASTER

De Dominostem, een communicatiehulpmiddel voor spraakgehandicapten

Winthagen, F.L.C.

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 EME

DE DOMINOSTEM,
EEN COMMUNICATIEHULPMIDDEL
VOOR SPRAAKGEHANDICAPTEN.

door F.L.C. Winthagen

Verslag van het afstudeerwerk
uitgevoerd van 15-12-'88 tot 14-12-'89.
In opdracht van prof. dr. ir. J.E.W. Beneken,

Onder begeleiding van dr. ir. R.P. Waterham,
ir. W.H. Leliveld
ir. E.J.J. Bierens

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

Samenvatting.

Dit afstudeerverslag beschrijft de ontwikkeling, realisatie en evaluatie van de Dominostem, een proefmodel van een communicatiehulpmiddel voor spraakgehandicapten.

Uit de evaluatie van de Pocketstem [1] is gebleken dat er behoefte bestaat aan een communicatiehulpmiddel met een groter vocabulaire. Dit heeft geresulteerd in het ontwerp en de realisatie van de Dominostem, een hulpmiddel waarbij de zinselectie met behulp van een symbolentaal geschiedt. Het vocabulaire is hierdoor essentieel groter dan bij de Pocketstem.

In de beginfase van het project zijn de ontwerpeisen voor de Dominostem verzameld. Dit is gedaan met medewerking van therapeuten.

Aan de hand van deze ontwerpeisen zijn hard- en software voor de Dominostem ontwikkeld.

Tevens is voor de Dominostem ondersteuningsprogrammatuur ontwikkeld. Met behulp van deze ondersteuningsprogrammatuur is het mogelijk de Dominostem bij een gegeven zinnenset van de juiste data te voorzien. Hiermee is het ook mogelijk, indien de zinnenset van de Dominostem gewijzigd moet worden, de Dominostem opnieuw automatisch van de juiste data te voorzien.

De evaluatie van de Dominostem is recentelijk gestart. Omtrent de bruikbaarheid van het apparaat kan op dit moment geconcludeerd worden, dat de zinselectie met behulp van symbolentaal een bruikbare methode voor de zinselectie is.

Tijdens de evaluatie is ook gebleken dat de klank van de spraak als "hol" ervaren wordt. Dit kan verbeterd worden door in een volgend model een andere klankkast-constructie te gebruiken.

Een ander punt dat om verbetering vraagt is het volgende:

Bij het gerealiseerde model is het altijd mogelijk om te achterhalen wat de laatst uitgesproken zin is. Dit lijkt niet correct wanneer het om de privacy van de gebruiker gaat. Indien dit ook uit de evaluatie blijkt, moet er een mogelijkheid ingebouwd worden om de laatste zin te kunnen uitwissen.

Summary.

This report describes the development, realization and evaluation of the "Dominostem" (Domino-voice), an experimental model of a communication aid for the speech impaired.

The evaluation of the Pocketstem [1] showed us that there is a need for a communication aid with a larger vocabulary than the Pocketstem.

This resulted in the Dominostem, an aid in which selection of a message is realized with the use of a symbol language. As a consequence the vocabulary increased essentially with regard to the Pocketstem.

At the start of the project the design specifications for the Dominostem are collected. This is done in cooperation with therapists.

With regard to these specifications the hard- and software for the Dominostem are developed. Also support-software for the Dominostem is developed. This software makes it possible, given a set of messages, to provide the Dominostem with the correct data. This software also makes it possible to easily change the set of messages of the Dominostem.

The evaluation of the Dominostem recently started. Considering the usefulness of the Dominostem, it can be concluded that selection of a message with the use of a symbol language is a useful method to select a message. During the evaluation it turned out that the sound of the speech was found to be "hollow". This can be improved by creating more holes in the front of the Dominostem.

Another point that should be improved is the following:

in the realized experimental model it is always possible to find out which message was the last one spoken. This may turn out to be not correct, considering the user's privacy. If this turns out to be as expected, a facility has to be built in with which it is possible to erase the last spoken message.

Inhoud

Samenvatting.	1
Summary.	2
1 Inleiding.	5
2 De opzet van het Dominostem-project.	8
2.1 De symbolentaal.	8
2.2 Het eisenblad.	8
2.3 Keuze voor realisatie van de uitspraak van zinnen.	10
3 Opbouw van de Dominostem.	13
3.1 De Dominostem: uiterlijk.	13
3.2 De Dominostem: het inwendige.	16
3.3 Functionele werking van de Dominostem.	17
3.3.1 Symbolen-toetsen.	17
3.3.2 Vaste boodschappen toetsen.	18
3.3.3 De spreektoets.	18
3.3.4 De Clear en Clear-all toetsen.	19
3.3.5 De controle toets.	19
4 De hardware.	21
4.1 Uitgangspunten bij het ontwerp.	21
4.2 De systeemconfiguratie.	21
4.3 De microprocessor en de geheugens.	24
4.3.1 Motivatie van de keuze van de EPROM's.	24
4.3.2 De processor en de geheugens.	25
4.4 Het reset-circuit.	25
4.5 Het keyboard en de led-matrix.	27
4.5.1 Het keyboard en het scan-circuit.	27
4.5.2 De led-matrix.	29
4.6 Het spanningsverzorgings- en detectiecircuit.	32
4.6.1 Het spanningsverzorgingscircuit.	33
4.6.2 Het spanningsdetectiecircuit.	33
4.6.3 De accu-leeg indicatie.	36

4.6.4	De acculader.	36
4.6.5	De spanningsvoorziening van het audio-circuit.	36
4.7	Het audio-circuit.	37
4.7.1	De spraaksynthesizer.	38
4.7.2	De audioversterkers.	38
4.7.3	De referentiespanningsregelaar voor de PCF8200.	39
5	De software.	40
5.1	Uitgangspunten bij het ontwerp.	40
5.2	Opzet van de software.	40
5.3	Het hoofdprogramma.	45
5.4	De subroutines.	48
5.5	De interruptroutines.	66
5.5.1	De PCF8200 interrupt routine.	66
5.5.2	De timer-interrupt routine.	67
6	De ondersteuningssoftware.	71
6.1	De opzoektabel.	71
6.1.1	De structuur van de opzoektabel.	71
6.1.2	Het genereren van de opzoektabel.	76
6.2	Het genereren van de spraakdata.	79
6.2.1	Het genereren van de A/P-files.	79
6.2.2	Het genereren van de spraakdata.	81
7	De evaluatie.	83
7.1	Aandachtspunten voor de evaluatie.	83
7.2	De evaluatie.	84
7.3	Resultaten en conclusies.	85
8	Conclusies.	86
	Literatuurlijst.	88
	Bijlagen.	90

1 Inleiding.

De interfaculteitswerkgroep "Communicatie hulpmiddelen voor gehandicapten" is een samenwerkingsverband tussen de vakgroep Medische Elektrotechniek van de Technische Universiteit Eindhoven en het Instituut voor Perceptie Onderzoek.

Een aantal mensen binnen deze werkgroep houdt zich bezig met het toepassen van synthetische spraak in hulpmiddelen voor spraakgehandicapten. Een apparaat dat uit een project binnen deze werkgroep is voortgekomen is de Pocketstem [1].

De Pocketstem is een klein, draagbaar apparaat dat 28 van te voren ingeprogrammeerde zinnen kan reproduceren. Het bevat een membraantoetsenbord met 28 toetsen waarmee een zin geselecteerd en gereproduceerd kan worden door het indrukken van de betreffende toets. Verder is elke toets voorzien van een pictogram, dat de betekenis van de betreffende zin weergeeft, om de gebruiker te helpen onthouden welke zin onder welke toets hoort.

De primaire doelgroep van potentiële gebruikers van de Pocketstem bestaat uit mensen met zowel een spraakhandicap als één of meerdere additionele handicaps. Uit de praktijkevaluatie is gebleken dat de Pocketstem bij een deel van deze doelgroep een goed hulpmiddel is. Verder bleek ook dat de doelgroep nog vergroot kon worden en wel op twee manieren:

1. Door de Pocketstem uit te rusten met een één-toets bediening. Hierdoor kunnen ook personen die moeite hebben met de bediening van de Pocketstem met behulp van het apparaat communiceren. Van deze variant van de Pocketstem wordt verwacht dat de doelgroep essentieel vergroot wordt.
2. Door de zinselectie te laten geschieden met behulp van een symbolentaal. Hierdoor kan het vocabulaire van de Pocketstem drastisch vergroot worden voor mensen die het vocabulaire van de Pocketstem te beperkt vinden. Echter deze variant van de Pocketstem zal de doelgroep minder uitbreiden dan de één-toets bediening. Dit komt vanwege het feit dat van de gebruiker wordt geëist dat hij de symbolentaal beheerst.

Gebruikelijke symbolentalen op dit moment zijn onder andere Bliss [2] of Dominolex. De taal Bliss heeft een aantal symbolen dat zeer groot is (ordegrootte enkele honderden). Daarentegen heeft de taal Dominolex een aantal symbolen dat veel lager is, namelijk 44.

Voor de praktische toepassing van een symbolentaal in een meeneembaar communicatie hulpmiddel is het essentieel dat het aantal symbolen beperkt is (symbolen kunnen worden ingevoerd met een toetsenbord; bij een beperkt aantal symbolen is het aantal benodigde toetsen dus beperkt). Derhalve is gekozen voor de taal Dominolex.

Op dit moment wordt de taal Dominolex ook gebruikt door spraakgehandicapten. Eén manier waarop dat gebeurt is de volgende:

De gehandicapte legt blokjes voorzien van Dominolex-symbolen achter elkaar. Degene aan wie de gehandicapte iets duidelijk wil maken moet nu deze rij symbolen vertalen.

Bij een communicatie hulpmiddel waarbij de zinselectie geschiedt met behulp van Dominolex voert de spraakgehandicapte de symbolen in, door bijvoorbeeld toetsen voorzien van Dominolex-symbolen in te drukken, en het apparaat spreekt de bijbehorende zin uit. Degene aan wie de gehandicapte iets duidelijk wil maken hoeft de symbolentaal dan niet meer te kennen, hetgeen een groot voordeel is.

De variant van de Pocketstem waarbij de zinselectie geschiedt met behulp van Dominolex is gedurende mijn afstudeerperiode ontwikkeld, gerealiseerd en beperkt geëvalueerd.

Deze variant van de Pocketstem wordt de Dominostem genoemd. Het principe van de Dominostem is symbolentaal-onafhankelijk. Dit wil zeggen dat in principe elke symbolentaal geïmplementeerd kan worden op de Dominostem.

In hoofdstuk 2 worden de opzet en het eisenblad van de Dominostem beschreven. Vervolgens wordt in hoofdstuk 3 de Dominostem functioneel beschreven en wordt de opbouw van het apparaat nader toegelicht. In hoofdstuk 4 wordt de hardware van de Dominostem behandeld. Deze hardware is gebaseerd op de hardware van de Pocketstem.

In hoofdstuk 5 komt de besturingssoftware van de Dominostem aan de orde. In hoofdstuk 6 wordt de ondersteuningsprogrammatuur voor de Domino-

stem beschreven.

In hoofdstuk 7 worden de evaluatie en de resultaten daarvan beschreven.

In hoofdstuk 8 zijn de conclusies vermeld en worden aanbevelingen gedaan voor een prototype.

Dankwoord.

Ik heb binnen de interfaculteitswerkgroep IPO-EME met plezier mijn afstudeeropdracht vervuld. Bij deze wil ik dan ook iedereen hartelijk bedanken voor de prettige werksfeer en de goede begeleiding tijdens mijn afstudeerperiode. Speciale dank gaat in dit geval uit naar Ronald Waterham, Ed Bierens, René Deliege, en René Verhoeven.

Tevens wil ik prof. J. Beneken, ir. W. Leliveld en dhr. H. Ossevoort bedanken voor de prettige sfeer en de hulp bij het afstudeerwerk.

Tenslotte wil ik een woord van dank richten aan Jan van Mierlo en Yvonne van Montfoort van instituut Zonhove voor hun bijdrage aan mijn afstudeerwerk.

2 De opzet van het Dominostem-project.

De in het Pocketstem project gebruikte strategie om tot een goed ontwerp te komen is ook hier gebruikt:

Met een aantal elementaire gegevens van therapeuten en literatuur [1,14] worden de ontwerpeisen samengesteld. Aan de hand hiervan wordt de hard- en software ontworpen. Het apparaat wordt vervolgens in de praktijk geëvalueerd. In het geval dat de evaluatie een negatief eindoordeel oplevert, kunnen met behulp van de resultaten daarvan kunnen de ontwerpeisen voor een eventueel volgend model bijgesteld worden.

Bij de Dominostem is tijdens mijn afstudeerperiode bovengenoemde cyclus éénmaal doorlopen. In dit hoofdstuk wordt de manier waarop de Dominostem werd gerealiseerd besproken.

2.1 De symbolentaal.

Het doel van deze afstudeeropdracht is om na te gaan of een zinsselectie met behulp van symbolentaal een bruikbare methode is in een communicatiehulpmiddel. Van minder belang hierbij is welke symbolentaal gebruikt wordt in het hulpmiddel. Echter voor de praktische realisatie moet een keus gemaakt worden. Wij hebben gekozen voor de symbolentaal Dominolex. Een andere reden hiervoor (naast het feit dat die taal een beperkt aantal symbolen heeft) is dat deze taal dichtbij (in instituut Zonhove in Son) gebruikt wordt. Het feit dat het instituut dichtbij de TUE is, is efficiënt voor de evaluatie. In het geval van een storing is het mogelijk om snel service te verlenen. Een ander aspect is dat ik hierdoor meerdere malen zelf tijdens de evaluatie aanwezig heb kunnen zijn, wat een leerzame ervaring was.

2.2 Het eisenblad.

De primaire eisen voor een communicatie hulpmiddel voor spraakgehandicapten zijn:

1. De bediening van het hulpmiddel moet eenvoudig zijn.
2. Het aantal zinnen dat het hulpmiddel kan reproduceren moet zo groot mogelijk zijn, om de gebruiker een zo groot mogelijke vrijheid te geven.
3. Het apparaat moet overal te gebruiken zijn. Het apparaat moet niet plaatsgebonden zijn door het feit dat het bijvoorbeeld afhankelijk is van het lichtnet.
4. De spraak die geproduceerd wordt moet duidelijk verstaanbaar zijn en sociaal geaccepteerd worden.
5. Het apparaat moet betaalbaar zijn, zodat iedereen in principe het apparaat kan verkrijgen.

Voor de realisatie hadden deze eisen de volgende consequenties:

1. De voeding van de Dominostem gebeurt met behulp van accu's.
2. De produktie van de spraak in het hulpmiddel gebeurt net als bij de Pocketstem met behulp van de spraakchip PCF8200. Deze chip bleek tijdens de evaluatie van de Pocketstem spraak te produceren die aan de gestelde eisen voor een communicatiehulpmiddel voldoet.
Er is echter een concessie gedaan: aangezien de zinnenset van de Dominostem 1200 zinnen bedraagt (zie ook punt 5) zou het inspreken van de zinnen door een spreker te veel tijd kosten; derhalve zijn de zinnen gerealiseerd met behulp van difoonspraak, waarvan de kwaliteit lager ligt dan bij ingesproken zinnen.
3. Om de kosten van het hulpmiddel wat betreft materiaal te beperken is bij het ontwerp een kosten-baten analyse uitgevoerd.
4. De bediening van het apparaat geschiedt met een toetsenbord. De toetsen van het toetsenbord zijn voorzien van de symbolen van de symbolentaal. Een symbool kan door de gebruiker ingevoerd worden, door de toets die dat symbool representeert in te drukken.

5. De zinnenset van de Dominostem bedraagt 1200 zinnen, hetgeen een afweging is tussen een zo groot mogelijke vrijheid enerzijds en de kosten van de opslagruimte van de zinnen anderzijds.

Tijdens de gesprekken in de eerste fase van het werk bleek dat de bediening van de Dominostem veel belangrijke aspecten kent:

- Symbolen kunnen door de gebruiker ingevoerd worden via een toetsenbord door de betreffende toetsen in te drukken. Aangezien om een zin te selecteren in het algemeen meer dan één symbool nodig is, is het noodzakelijk dat de gebruiker een terugmelding krijgt welke toetsen er ingedrukt zijn. Ten behoeve van deze terugmelding is onder elke toets die een symbool representeert, een LED aangebracht.
- Wanneer de gebruiker een rij symbolen ingevoerd heeft en de bijbehorende zin wil laten uitspreken, moet hij dit aan de Dominostem kenbaar maken door een “spreek-toets” in te drukken.
- Om fouten bij een zinselectie te vermijden is een correctie-faciliteit noodzakelijk. Met deze correctie-faciliteit kunnen één of meerdere reeds ingevoerde symbolen uitgewist worden.
- Op de Dominostem zullen een aantal toetsen gereserveerd worden als Pocketstem-toetsen. Door een dergelijke toets in te drukken wordt een zin meteen geselecteerd en uitgesproken. Deze toetsen dienen om in bijvoorbeeld noodsituaties de aandacht te trekken.

Het eisenblad, waarin de eisen die aan de bediening gesteld worden zijn vermeld, is opgenomen in bijlage 2.

2.3 Keuze voor realisatie van de uitspraak van zinnen.

De zinnenset van 1200 zinnen die in de Dominostem aanwezig is zou voor de meeste dagelijkse situaties uitkomst moeten bieden. De zinnenset is in

grote lijnen opgedeeld in categorieën die elk betrekking hebben op een begrip.

Binnen één en dezelfde categorie komt een woord of een stuk zin vaak meerdere malen voor. Het zou daarom voor de hand liggend zijn om deze woorden of stukken zin slechts eenmaal in het geheugen op te slaan. Bij het reproduceren van een zin moeten dan de bijbehorende stukken zin of woorden achter elkaar "geplakt" worden.

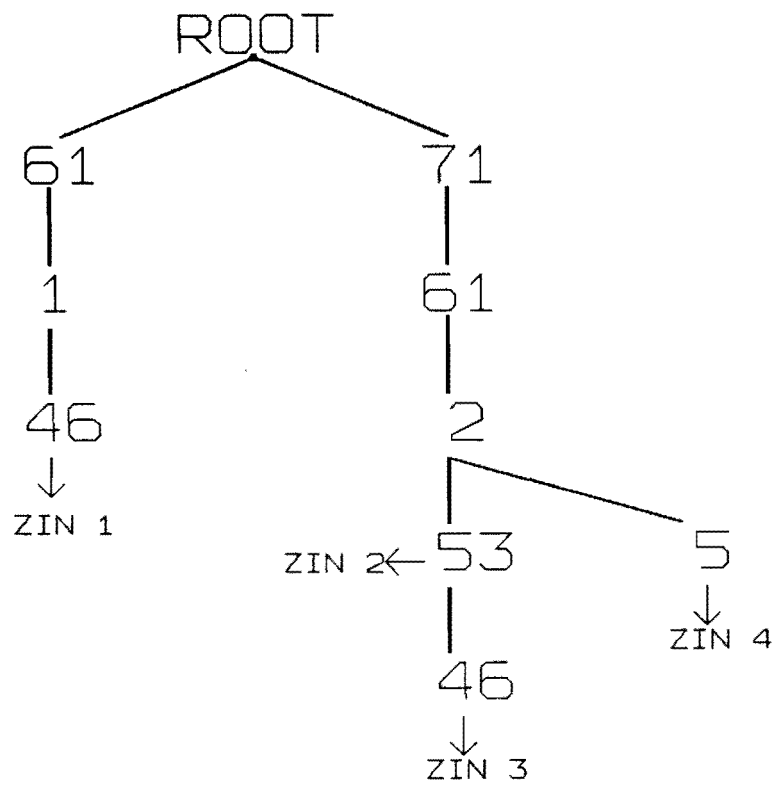
Een groot nadeel van deze methode is dat de intonatie en zinsmelodie bij dat achter elkaar plakken gecorrigeerd moet worden, daar ze anders verloren gaan.

Voor het nagaan of de zinselectie met behulp van symbolentaal succesvol is, is het terdege van belang hoe de uitspraak van een zin tot stand komt. Derhalve is voor de realisatie wat betreft uitspraak van zinnen niet gekozen voor het concateneren van woorden of stukken zin, maar voor het opslaan en uitspreken van complete zinnen.

De taalregels van een symbolentaal verschillen per symbolentaal. Daar het principe van de Dominostem onafhankelijk van de symbolentaal is worden dus geen taalregels opgeslagen in de Dominostem.

De omzetting van een rijtje ingevoerde symbolen naar een zin kan het best weergegeven worden met behulp van een boomstructuur (zie figuur 2.1 waarin een voorbeeld is gegeven van een zinnenset van 4 zinnen en waar de symbolen voor de algemeenheid aangeduid zijn door cijfers)

- zin 1 : Ik wil meer eten.
- zin 2 : Mag ik koffie?
- zin 3 : Mag ik nog wat meer koffie?
- zin 4 : Mag ik wat water?



Figuur 2.1: Voorbeeld van een omzetting van symbolen naar een zin.

3 Opbouw van de Dominostem.

Een belangrijk feit is dat de Dominostem een proefmodel is dat zich nog moet waarmaken in de praktijkevaluatie. Van dit proefmodel is namelijk maar één exemplaar. Om voor dit éne exemplaar onderdelen op maat te laten maken zou veel te duur worden. Gevolg hiervan is dat alle gebruikte onderdelen standaard verkrijgbaar zijn. Dit heeft consequenties voor het uiterlijk van het apparaat. Voor een eventueel volgend model, indien het aantal gewenste exemplaren zodanig is dat het op maat laten maken financieel haalbaar is, is het aan te bevelen om bepaalde onderdelen als het kastje en het toetsenbord op maat te laten maken.

3.1 De Dominostem: uiterlijk.

Het proefmodel van de Dominostem heeft als afmetingen $20 \times 25 \times 6,5 \text{ cm}^3$. Het kastje dat gebruikt is, is van grijs kunststof (ABS) en is standaard verkrijgbaar bij TELECOM (merk Bimbox, prijs: fl. 25,00).

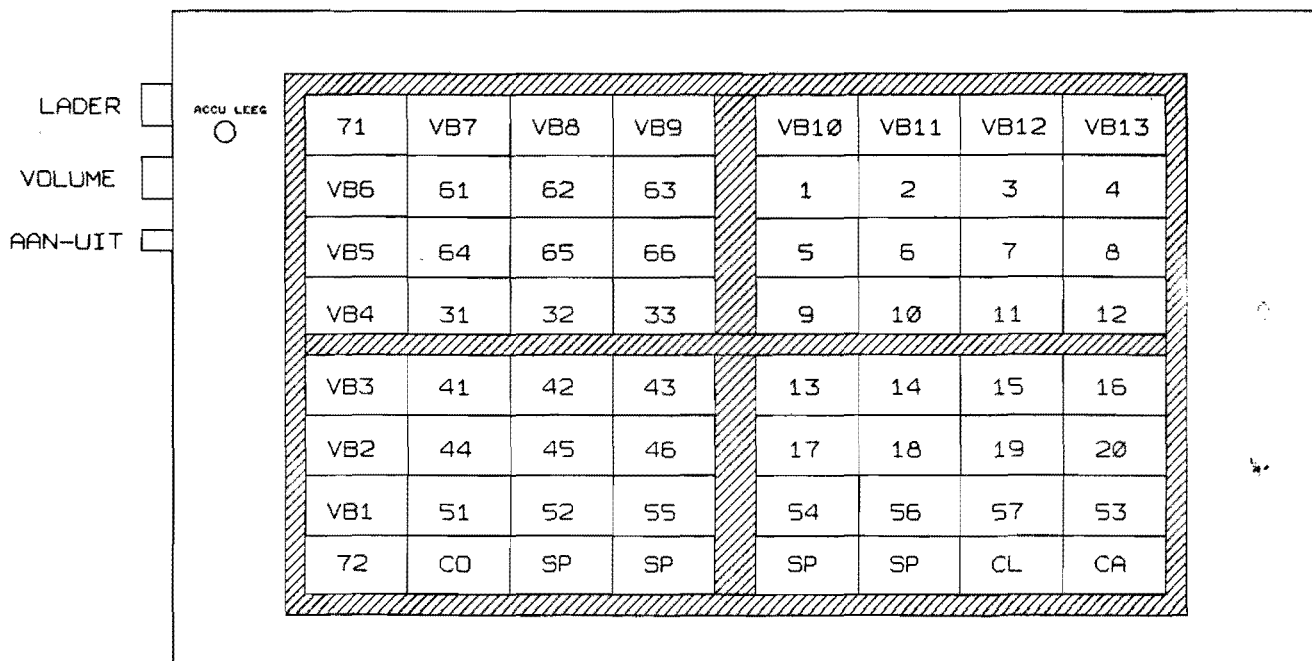
Er is gezocht naar een toetsenbord met een aantal toetsen van ≈ 50 . Dit aantal is opgebouwd uit 44 symbolen-toetsen, 2 correctie-toetsen, een spreek-toets en enkele vaste boodschappen-toetsen.

Echter één toetsenbord met het gewenste aantal toetsen bleek niet verkrijgbaar. Het toetsenbord moest dus samengesteld worden uit enkele kleine toetsenborden. Na een inventarisatie bleken er 2 mogelijkheden te zijn:

1. 2 toetsenborden van elk 4×4 toetsen en 2 toetsenborden van elk 3×4 toetsen, tezamen 56 toetsen.
2. 4 toetsenborden van elk 4×4 toetsen, tezamen 64 toetsen.

Uiteindelijk is gekozen voor de laatste mogelijkheid, waardoor het aantal beschikbare toetsen groter is dan het aantal benodigde toetsen. De toetsen waar nog geen functie voor bestemd is, zijn later ingevuld.

Na invullen van de functies van deze toetsen zag de lay-out van het toetsenbord uit als weergegeven in figuur 3.1.



Figuur 3.1: Bovenaanzicht van de Dominostem.

De bovenkant van de Dominostem met alle symbolen erbij, is weergegeven in bijlage 4. De symbolen (44 stuks) zijn onderverdeeld in 6 categorieën:

- categorie 1: genummerd 1 tot en met 20,
- categorie 2: genummerd 31 tot en met 33,
- categorie 3: genummerd 41 tot en met 46,
- categorie 4: genummerd 51 tot en met 57,
- categorie 5: genummerd 61 tot en met 66,
- categorie 6: genummerd 71 tot en met 72.

Alle toetsen die symbolen representeren zijn voorzien van een LED die op zal lichten wanneer de betreffende toets is ingedrukt.
De vaste boodschap toetsen bevinden zich in de bovenste rij (7 stuks) en in

de meest linkse kolom (6 stuks). In figuur 3.1 zijn ze genummerd van VB1 tot en met VB13.

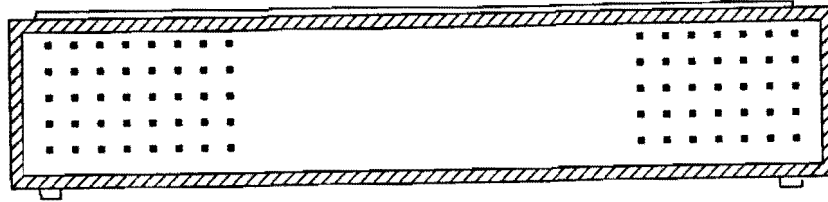
De onderste rij van het toetsenbord bevat 7 toetsen met speciale functies:

- Spreektoets (aangegeven in figuur 3.1 met SP): indien we een toetsenbord op klantenspecificatie zouden hebben, zou er gekozen zijn voor een spreekbalk (net als de spatiebalk op een computer). Om dit te benaderen hebben we gekozen voor 4 spreektoetsen met identieke functie.
- Clear toets (aangegeven in figuur 3.1 met CL): Door deze toets in te drukken wordt het laatst ingevoerde symbool uitgewist.
- Clear-all toets (aangegeven in figuur 3.1 met CA): Door deze toets in te drukken worden alle ingevoerde symbolen uitgewist.
- Controle toets (aangegeven in figuur 3.1 met CO): Nadat deze toets ingedrukt is kan de gebruiker zien in welke volgorde de symbolen ingevoerd zijn. De Dominostem laat dan namelijk in de volgorde zoals de symbolen ingevoerd zijn de LED's onder de betreffende toetsen oplichten.

Aan de linkerkant van het toetsenbord bevindt zich een gele LED die op zal lichten wanneer de accu van de Dominostem opgeladen moet worden.

Aan de linker zijkant bevinden zich de aan-uit schakelaar, een 5 standen volume regelaar en de aansluiting voor de acculader.

Verder zijn in de voorkant van de Dominostem aan de linker- en rechterkant gaatjes gemaakt voor de luidsprekers die zich hierachter bevinden (zie figuur 3.2).



Figuur 3.2: Vooraanzicht van de Dominostem.

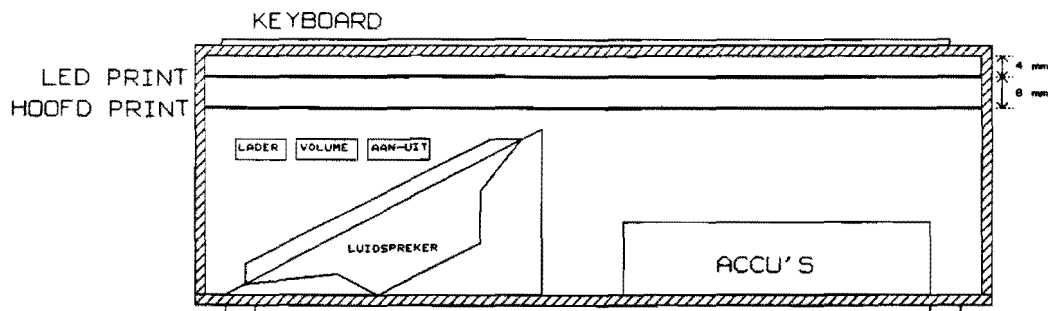
3.2 De Dominostem: het inwendige.

De elektronika van de Dominostem bevindt zich op twee printplaten elk $18 \times 23 \text{ cm}^2$. Op één printplaat zijn de LED's gemonteerd die zorgdragen voor de indicatie naar de gebruiker toe. Deze printplaat is op 4 mm van de bovenkant gemonteerd (zie figuur 3.3). Op deze printplaat worden ook de connectoren van de 4 toetsenborden aangesloten, welke op de LED print samengebundeld worden tot één connector, waarmee de toetsenborden aangesloten worden op de hoofdprint. Verder bevat de LED print een connector waarmee de LED's op de hoofdprint aangesloten worden. De hoofdprint bevat de complete schakeling: microprocessor, geheugen, audio-circuit, spanningsregeling en -detectiecircuit, keyboard en LED-sturing circuit. Verder bevat de hoofdprint 4 connectoren:

- connector voor de LED's,
- connector voor het keyboard,
- connector voor de luidsprekers en de accu's,
- connector voor de volumeregelaar, aan-uit schakelaar en de oplaadplug.

Door alle verbindingen met de hoofdprint te realiseren met behulp van deze connectoren is het eenvoudiger om de hoofdprint uit de Dominostem te verwijderen voor test, reparatie, uitwisseling van EPROM's en dergelijke.

De luidsprekers bevinden zich net als de accu's op de bodem van het kastje. De luidsprekers zijn schuin opgesteld en gemonteerd in klankkastjes van PVC die op de bodem vastgelijmd zijn. De projectie van het inwendige op het linker zijvlak is weergegeven in figuur 3.3.



Figuur 3.3: Projectie van het inwendige op het linker zijvlak.

3.3 Functionele werking van de Dominostem.

Indien de aan-uit schakelaar in de “aan” stand staat, zal de Dominostem reageren op een toetsaanslag. Deze invoeracties worden achtereenvolgens besproken.

3.3.1 Symbolen-toetsen.

Indien een toets die een symbool representeert ingedrukt wordt, zal de LED die bij de betreffende toets hoort oplichten. Meerdere symbolen kunnen achter elkaar ingevoerd worden door de betreffende toetsen achter elkaar in te drukken. Na elk ingevoerd symbool zullen de LED's van alle tot dan toe ingevoerde symbolen oplichten.

Na elk ingevoerd symbool zal de processor nagaan of de tot dan toe opgebouwde symbolenrij "geldig" is, hetgeen betekent dat aan één van de volgende twee voorwaarden is voldaan:

- de ingevoerde symbolenrij komt in z'n geheel in de tabel (zinnenset) voor
- de ingevoerde symbolenrij kan aangevuld worden, zodat na aanvulling de rij overeenkomt met een rij uit de tabel.

Als het laatste symbool de rij geldig maakt, zal de LED van dat symbool (samen met de LED's van de reeds ingevoerde symbolen) oplichten, en het laatste symbool in het geheugen opgenomen worden.

Indien het laatst ingevoerde symbool de rij ongeldig maakt, zal de LED van het betreffende symbool niet oplichten en de laatste toets niet in het geheugen opgenomen worden. Tevens zal de LED onder de controle-toets in dit geval 3 seconden knipperen ter aanduiding dat het laatste symbool ongeldig was.

Als na het invoeren van een symbool gedurende 15 seconden géén volgende toets wordt ingedrukt zullen de LED 's uitgaan; echter de processor houdt wel het ingevoerde symbool.

De reden dat na 15 seconden de LED's uitgaan is om het stroomverbruik binnen de perken te houden zodat de Dominostem zolang mogelijk met een acculading gebruikt kan worden.

3.3.2 Vaste boodschappen toetsen.

Een vaste boodschap-toets kan op een willekeurig moment ingedrukt worden. In alle gevallen zal de betreffende boodschap worden uitgesproken.

Enige restricties zijn dat de aan-uit schakelaar in de stand "aan" moet staan en dat de accu niet leeg mag zijn.

3.3.3 De spreektoets.

Door de spreektoets in te drukken geeft de gebruiker opdracht aan de Dominostem om de zin behorende bij de ingevoerde symbolenrij uit te spreken.

Er zijn hierbij 2 gevallen te onderscheiden:

1. Een correcte, in het geheugen van de Dominostem aanwezige, symbolenrij is ingevoerd. In dit geval zal de bij de combinatie behorende zin worden uitgesproken en de ingevoerde symbolenrij zal uit het geheugen gewist worden. De zin kan nogmaals ten gehore worden gebracht door de spreektoets nogmaals in te drukken.
2. De symbolenrij in het geheugen is nog niet compleet, dit wil zeggen de rij moet nog aangevuld worden om een zin te vormen uit de zinnenset van de Dominostem. In dit geval zal een "spreek-opdracht" tot gevolg hebben dat de LED onder de controle toets gedurende 3 seconden gaat knipperen, terwijl er niets wordt uitgesproken. Na de 3 seconden zal de Dominostem weer de LED's van de ingevoerde symbolen laten oplichten.

3.3.4 De Clear en Clear-all toetsen.

Na indrukken van de "clear-toets" zal het laatst ingevoerde symbool gewist worden. Tevens zullen de LED's van de resterende symbolen oplichten. Op deze wijze kunnen één of meerdere symbolen gewist worden.

Na indrukken van de "clearall-toets" zullen alle ingevoerde symbolen in het geheugen van de Dominostem gewist worden.

3.3.5 De controle toets.

Als de gebruiker de controle toets indrukt zal van de ingevoerde symbolenrij de volgorde getoond worden. Dit gebeurt door in de volgorde waarin de symbolen ingevoerd zijn, de LED's te laten oplichten. Tijdens deze procedure zal de controle LED oplichten (dit is de LED onder de controle toets), en worden symbolen, clear(all) en spreekacties (behalve vaste boodschappen) genegeerd.

Wanneer de volgorde getoond is, zal de Dominostem weer overgaan tot het laten oplichten van de LED's van alle ingevoerde symbolen.

Een andere manier om uit de "controle-toestand" te komen is door de

controle-toets nogmaals in te drukken. Hierna zal de Dominostem weer overgaan tot het laten oplichten van de LED's van alle ingevoerde symbolen, waarbij weer alle invoer-acties mogelijk zijn.

4 De hardware.

Dit hoofdstuk bevat de beschrijving van het hardware ontwerp. Allereerst worden de uitgangspunten voor het ontwerp besproken. Vervolgens komt de systeemconfiguratie aan de orde, waarna de systeemonderdelen afzonderlijk besproken worden.

4.1 Uitgangspunten bij het ontwerp.

De algemene eisen die aan het ontwerp gesteld worden zijn dezelfde als die bij de Pocketstem [1]. Derhalve is rekening gehouden met:

- Het gebruik van componenten en schakelingen die zuinig zijn wat betreft stroomverbruik.
- Het gebruik van compacte schakelingen en componenten, om zodoende het gewicht van het apparaat beperkt te houden.

Het feit dat het stroomverbruik laag gehouden dient te worden is een gevolg van de wens dat de Dominostem zo lang mogelijk gebruikt kan worden met één acculading. Dit komt er op neer dat CMOS componenten gebruikt zijn bij het ontwerp.

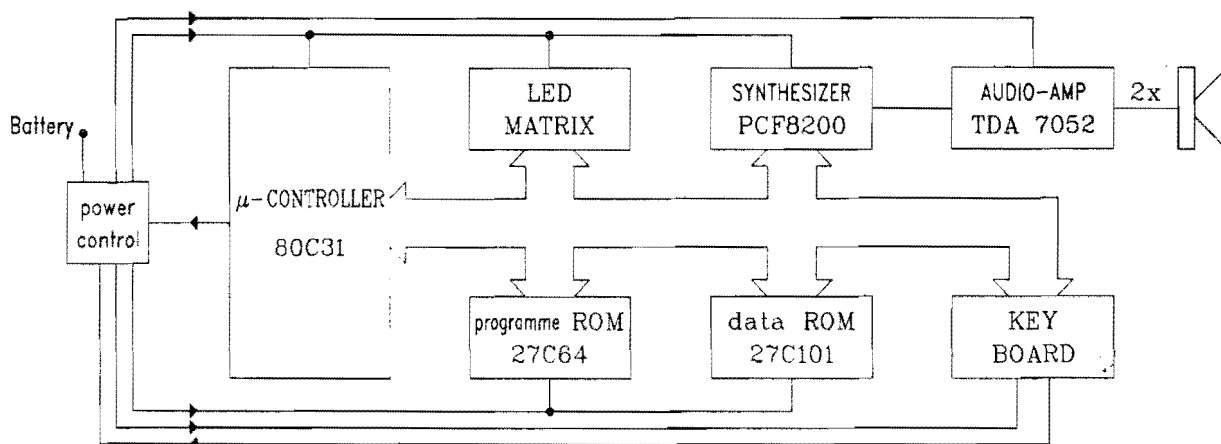
4.2 De systeemconfiguratie.

De hardware van de Dominostem is opgebouwd rond een 80C31 microprocessor [4] van Intel uit de MCS 51 serie. Het geheel is weergegeven in figuur 4.1.

Rond de microprocessor bevinden zich de volgende schakelingen:

- een keyboard tesamen met een scancircuit. Het keyboard is in matrix vorm met 8 ingangslijnen en 8 uitgangslijnen.
- een LED-matrix met 8 ingangslijnen en 16 uitgangslijnen.

- een programmageheugen (EPROM). De EPROM is 8 kbyte groot.
- een data geheugen (5 EPROM's). De EPROM's zijn tesamen 640 kbyte groot.
- een audio circuit met PCF8200 spraakchip en twee audio versterkers en twee luidsprekers. Er is gekozen voor 2 versterkers en luidsprekers (hiervoor is gekozen omdat er voldoende ruimte hiervoor aanwezig was) om meer geluidssterkte ter beschikking te hebben.
- een power-control circuit. Dit circuit verzorgt de spanning voor de schakeling, detecteert wanneer de spanning te laag wordt en schakelt stroomverbruikende delen van de schakeling af wanneer ze niet in gebruik zijn.



Figuur 4.1: De systeemconfiguratie.

De microprocessor heeft vier I/O poorten van elk acht lijnen. De indeling

van de I/O poorten zijn weergegeven in tabel 4.1.

Tabel 4.1 : Indeling van de 4 I/O poorten van de processor.

P 0.0		P 1.0	AB16, adresbusbit 16
P 0.1		P 1.1	A0, bit 0 van EPROM selector
P 0.2	data	P 1.2	A1, bit 1 van EPROM selector
P 0.3	en	P 1.3	A2, bit 2 van EPROM selector
P 0.4	adres	P 1.4	\overline{ESEL} , enable EPROM selector
P 0.5	bus	P 1.5	n.c
P 0.6		P 1.6	pconb1, bit 1 power control lijn
P 0.7		P 1.7	pconb2, bit 2 power control lijn
P 2.0		P 3.0	\overline{BATLED} , stuurt accu leeg led aan
P 2.1		P 3.1	SWPOS, inputlijn voor schakelaarstand
P 2.2		P 3.2	$\overline{INT0}$, PCF8200 interrupt lijn
P 2.3	adres	P 3.3	$\overline{INT1}$, batterij detectie interrupt lijn
P 2.4	bus	P 3.4	\overline{CEkeyb} , enable keyboard
P 2.5		P 3.5	\overline{CEpcf} , enable PCF8200
P 2.6		P 3.6	\overline{WR} , write signaal
P 2.7		P 3.7	\overline{RD} , read signaal

Met poort 0 en 2 samen kan de processor een bereik van 128 kbyte adresseren, te weten 64 kbyte programma-geheugen (aangestuurd via de PSEN-puls) en 64 kbyte data-geheugen (aangestuurd via de RD-puls).

Voor het programma-geheugen (8 kbyte) is dat ruimschoots voldoende, echter voor het data-geheugen (640 kbyte) niet.

Het data-geheugen is opgebouwd uit 5 EPROM's van elk 128 kbyte. Om binnen één EPROM te kunnen adresseren zijn 17 adreslijnen nodig. Poort 0 en 2 leveren slechts 16 adreslijnen. P 1.0 is gekozen voor adreslijn 17. Met poort 0, 2 en P 1.0 kunnen 128 kbyte geadresseerd worden.

Om één van de 5 EPROM's te selecteren zijn nog 3 extra lijnen nodig. Hiervoor worden bit 1,2 en 3 gebruikt van poort 1. Deze 3 poortlijnen sturen een 3 naar 8 line decoder, die één van de 5 EPROM's selecteert. De 3 naar 8 line decoder heeft 8 uitgangen waarvan er 5 gebruikt worden om de 5

EPROM's te selecteren. De overige 3 uitgangen worden gebruikt om de latches van de LED-matrix te selecteren.

Indien géén van deze 8 componenten geselecteerd behoeven te worden is het noodzakelijk om met een andere lijn van de processor dat aan te geven. Hiertoe is P 1.4 gebruikt. Deze lijn selecteert de 3 naar 8 line decoder, die vervolgens één van de 8 te selecteren componenten aanstuurt.

4.3 De microprocessor en de geheugens.

In deze paragraaf wordt de keus van de data EPROM's gemotiveerd en vervolgens wordt het deel van de microprocessor en de geheugens besproken.

4.3.1 Motivatie van de keuze van de EPROM's.

Het aantal zinnen dat de Dominostem moet kunnen uitspreken bedraagt ongeveer 1200.

Uit ervaringen met de Pocketstem is gebleken dat de gemiddelde hoeveelheid benodigde geheugenruimte per zin (wat betreft spraakdata) ongeveer 0.5 kbyte bedraagt. Met deze vuistregel is het snel uitgerekend dat de benodigde geheugenruimte $1200 \times 0.5 \text{ kbyte} = 600 \text{ kbyte}$ bedraagt.

Dit kan op meerder manieren gerealiseerd worden:

1. 20 IC's van elk 32 kbyte,
2. 10 IC's van elk 64 kbyte,
3. 5 IC's van elk 128 kbyte.

De verschillende mogelijkheden zijn vergeleken wat betreft kosten, ruimte en stroomverbruik. Mogelijkheid nummer 3 is het beste wat betreft ruimte en stroomverbruik, echter niet wat betreft kosten. Toch is voor 3 gekozen, daar de ontwikkelingen van prijzen op het gebied van geheugen IC's tegenwoordig zo snel gaan, dat verwacht wordt dat dit alternatief binnen niet al te lange tijd goedkoper zal zijn dan de alternatieven 1 en 2.

4.3.2 De processor en de geheugens.

Het circuit is weergegeven in bijlage 1. De data EPROM's worden geselecteerd door de 3 naar 8 line decoder. Data uit de EPROM's wordt gelezen met behulp van een MOVX instructie die gepaard gaat met een \overline{RD} puls. De \overline{RD} uitgang van de processor is derhalve doorverbonden met de \overline{OE} ingangen van de data EPROM's. De 3 naar 8 line decoder wordt gestuurd door P 1.4 (\overline{ESEL}). De EPROM gebruikt namelijk als hij geselecteerd is meer stroom (1 mA) dan wanneer hij niet geselecteerd is (1 μ A).

In de praktijk komt het erop neer dat een data EPROM alleen dan geselecteerd wordt wanneer er data wordt gelezen.

Het programma geheugen is geselecteerd wanneer de processor actief is. Bit 6 van poort 1 is hoog wanneer de processor actief is (ook tijdens IDLE), en laag wanneer de processor powerdown is. Aangezien bij een RESET puls alle poortlijnen hoog worden wordt het programma geheugen automatisch geselecteerd wanneer de processor actief wordt. De chip enable van het programma geheugen is laag actief. Derhalve is een inverter geplaatst tussen P 1.6 en de chip enable ingang.

Het programma geheugen wordt niet gestuurd door de \overline{RD} puls, maar door de \overline{PSEN} puls (Program Store ENable). Het lage orde adres (poort 0) wordt via de latch en de ALE puls ingeklokt, terwijl het hoge orde adres (poort 2) gedurende de lees cyclus aanwezig is.

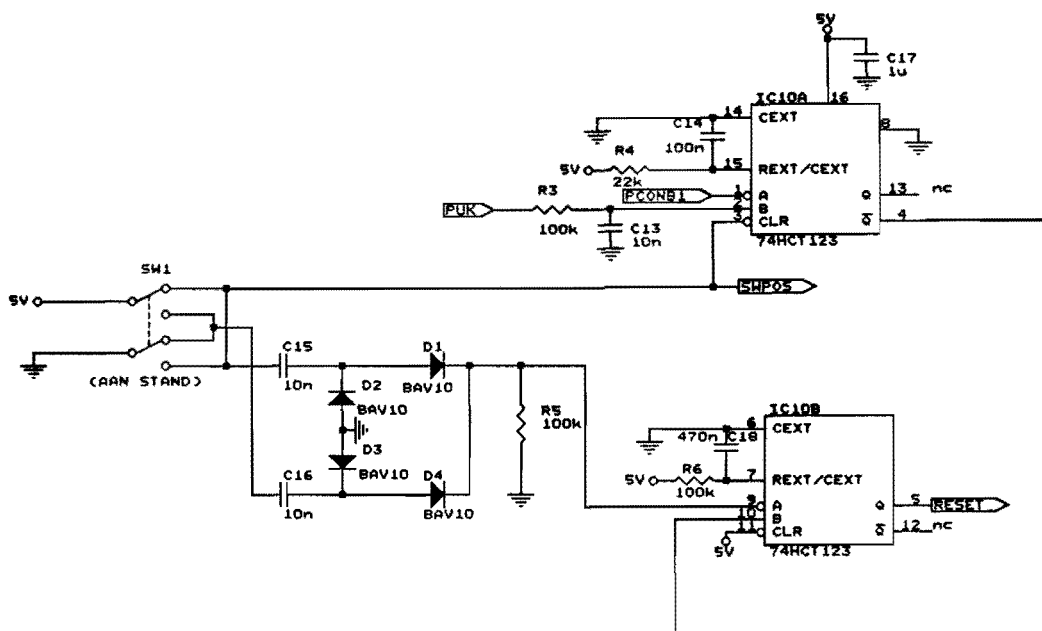
4.4 Het reset-circuit.

Een reset-puls op de ingang van de processor zorgt ervoor dat (ongeacht wat de processor op dat moment deed) de programcounter van de processor op adres 0 van het programmeergeheugen wordt gezet. Vanaf daar start de processor dan met het uitvoeren van instructies. Met behulp van de reset-puls kan de processor uit de powerdown toestand (dit is een toestand waarin de klok is uitgeschakeld en de processor nauwelijks nog stroom verbruikt) komen.

De reset-puls wordt door het reset circuit gegenereerd onder de volgende omstandigheden:

- Een reset puls moet gegenereerd worden wanneer de aan-uit schakelaar van positie verandert.
- een reset puls moet gegenereerd worden wanneer de processor power-down is en de aan-uit schakelaar in de stand “aan” staat en een toets wordt ingedrukt. In dat geval komt op de PUK-lijn (power up keyboard) een positieve spanningspuls.

Het reset circuit is weergegeven in figuur 4.2. Het circuit is opgebouwd rond een 74HCT123. Dit IC bestaat uit 2 monostabiele multivibratoren.



Figuur 4.2: Het reset circuit.

Bij het omschakelen van de aan-uit schakelaar ontstaat op de \bar{A} -ingang van IC10B een positieve puls. Ongeacht wat de B-ingang van IC10B op dat moment voor waarde had, zal op de Q-uitgang van IC10B een positieve puls komen. Bij het veranderen van de stand van de schakelaar komt er dus altijd een reset puls.

Wanneer de processor powerdown is, is $pconb1 = 0$. Als de Dominostem aan staat, geldt dat de \overline{CLR} -ingang van IC10A hoog is. Onder deze twee voorwaarden zal een positieve PUK puls op B-ingang van IC10A ervoor zorgen dat een negatieve puls op de \overline{Q} -uitgang van IC10A komt. Deze puls zorgt, omdat de \overline{A} -ingang van IC10B laag is, voor een positieve puls op de Q-uitgang van IC10B.

De reset-puls moet een gedefinieerde lengte hebben die minimaal 10 milliseconden bedraagt [1]. Deze lengte is vereist daar anders het interne RAM van de processor verminkt kan worden.

De tijdsduur van de reset puls hangt samen met de waarden van R en C op de Cext en Rext/Cext ingangen volgens de relatie [7]:

$$t = 0.45 * R * C$$

R is hierbij gebonden aan grenzen, terwijl voor C een vrije keus mogelijk is. R is gekozen op 100 k Ω ; C is gekozen op 470 nF. Dit resulteert in een theoretische tijdsduur van de reset puls van 20 milliseconden.

Bij meting van de tijdsduur van de reset-puls bleek dat de tijdsduur nogal aan spreiding onderhevig was. De gemeten tijdsduur lag tussen 16 en 20 milliseconden.

Bij een lagere waarde van C (330 nF) leverde de spreiding in enkele gevallen tijdsduren op die korter waren 10 milliseconden. Om risico's te vermijden is derhalve gekozen voor een condensator van 470 nF.

4.5 Het keyboard en de led-matrix.

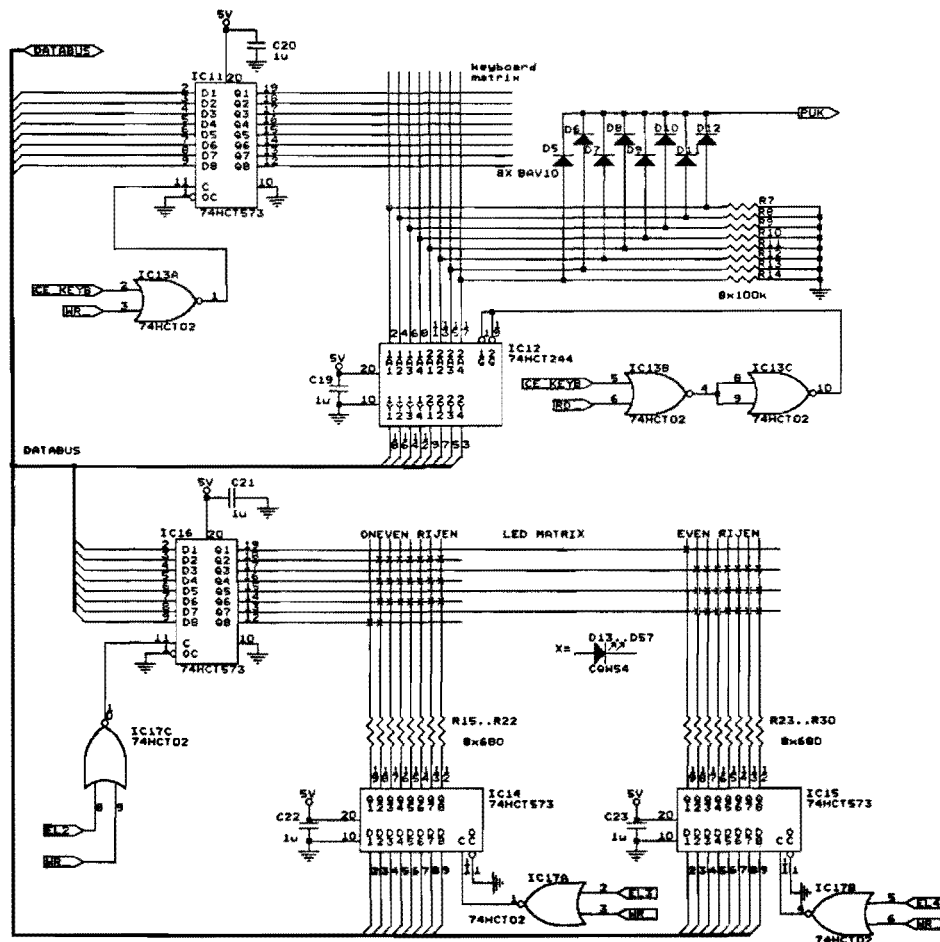
Het keyboard en de led-matrix zijn weergegeven in figuur 4.3. Achtereenvolgens worden het keyboard en het scan-circuit [9] en de led-matrix [8] besproken.

4.5.1 Het keyboard en het scan-circuit.

Voor het keyboard is, zoals eerder vermeld, gekozen voor 4 kleine membraankeyboards (elk 4x4 toetsen in matrix-vorm). Deze toetsenbordjes

zijn van het merk Brady (leverancier Cisper Electronics te De Lier). Het type nummer van de keyboards is F16 PRM 4 BK en de prijs is fl. 70,00 per stuk.

Deze keyboards hebben in elke toets een ronde uitsparing van 3 millimeter doorsnede waar het licht van een LED doorheen kan schijnen.



Figuur 4.3: Het keyboard, het scan-circuit en de led-matrix.

Data kan naar het keyboard geschreven worden via de latch 74HCT573 (zie figuur 4.3).

Het keyboard bevat twee onderdelen die gebruikt worden voor data-transport,

een latch (voor transport van data naar keyboard) en een buffer (voor transport van data naar de processor).

De latch wordt geselecteerd door 2 signalen: \overline{CEkeyb} en \overline{WR} . Indien data naar de latch geschreven wordt, gaat dat gepaard met een \overline{WR} puls. Beide signalen, \overline{WR} en \overline{CEkeyb} , zijn inputs van een twee input nor gate. De output van die nor gate is het enable signaal voor de keyboard latch.

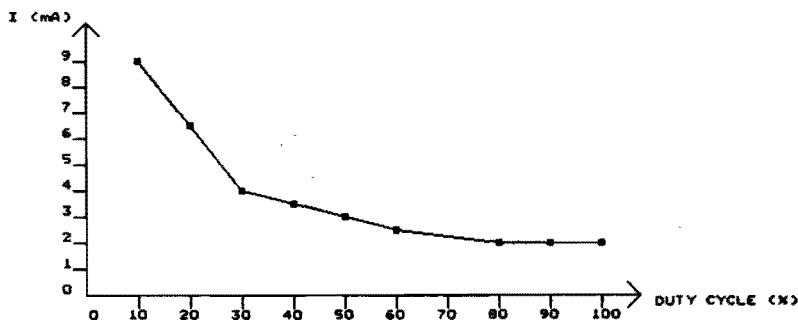
Het keyboard wordt gelezen met behulp van een buffer 74HCT244. Dit buffer zorgt dat de informatie van de uitgangslijnen van het keyboard op het juiste moment op de databus komen. Als er data gelezen wordt van het keyboard, gaat dat gepaard met een \overline{RD} puls. Deze verzorgt samen met de \overline{CEkeyb} lijn het enable signaal van de keyboard buffer.

De 8 uitgangslijnen van het keyboard zijn elk met een pull-down weerstand van 100 k Ω naar aarde verbonden. Dit is gedaan om de uitgangslijnen altijd een gedefinieerd niveau te geven, hetgeen bij CMOS noodzakelijk is.

De uitgangslijnen van het keyboard zijn elk met behulp van een diode (logical-or) verbonden met de PUK-lijn. In de situatie dat de processor powerdown is, zijn alle ingangslijnen van het keyboard hoog. Als de processor powerdown is kan met behulp van de PUK-lijn gedetecteerd worden of een toets ingedrukt is. In deze situatie zal wanneer een willekeurige toets wordt ingedrukt, op de PUK-lijn een positieve spanningspuls komen.

4.5.2 De led-matrix.

Voor een efficiënte aansturing (wat betreft stroomverbruik) van de LED's, is allereerst nagegaan wat de lichtopbrengst van een LED als functie van de duty-cycle is. Het resultaat hiervan is weergegeven in figuur 4.4.



Figuur 4.4: De stroom als functie van de duty-cycle van een LED met voor het oog gelijkblijvende lichtsterkte.

De lichtsterkte is beoordeeld met behulp van het oog. Deze subjectieve meting levert toch nuttige informatie, daar we eruit kunnen afleiden hoe de stroom door de LED optimaal gekozen kan worden, zonder dat met het oog verschil in lichtsterkte wordt waargenomen.

Een eis die uit het eisenblad (bijlage 2) naar voren komt, is dat de LED's in de Dominostem met twee verschillende lichtsterkten moeten kunnen branden.

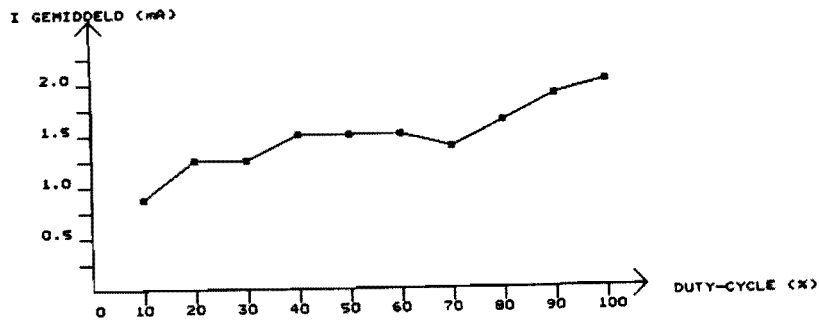
Deze twee intensiteiten kunnen gerealiseerd worden door de LED's in de éne situatie met een andere duty-cycle te laten oplichten dan in de andere situatie.

Gekozen is voor een duty-cycle van 25% bij de felle sterkte en een duty-cycle van 12.5% bij de minder felle sterkte.

Uit figuur 4.5 blijkt dat het rendement (voor het oog!) van de LED toeneemt met afnemende duty-cycle. Wat betreft gemiddelde stroomverbruik is het derhalve raadzaam om een zo laag mogelijke duty-cycle te nemen.

Toch is niet gekozen voor bijvoorbeeld 5 en 10% duty-cycles omdat bij dergelijke duty-cycles de topwaarde van de stroom hoog wordt. Deze hoge

piekstromen kunnen dan voor storingen zorgen.



Figuur 4.5: Tijdgemiddelde stroom als functie van de duty-cycle van een LED met gelijkblijvende lichtsterkte.

De led-matrix bevat 45 LED's die zijn geplaatst onder de symbolen-toetsen en de controle-toets. Bij een duty-cycle van 25% worden elke 2 milliseconden 2 rijen LED's (een even en een oneven rij) aangestuurd. Om bij deze 2 rijen niet de LED's in dezelfde kolommen te laten oplichten moeten de uitgangslijnen gescheiden blijven.

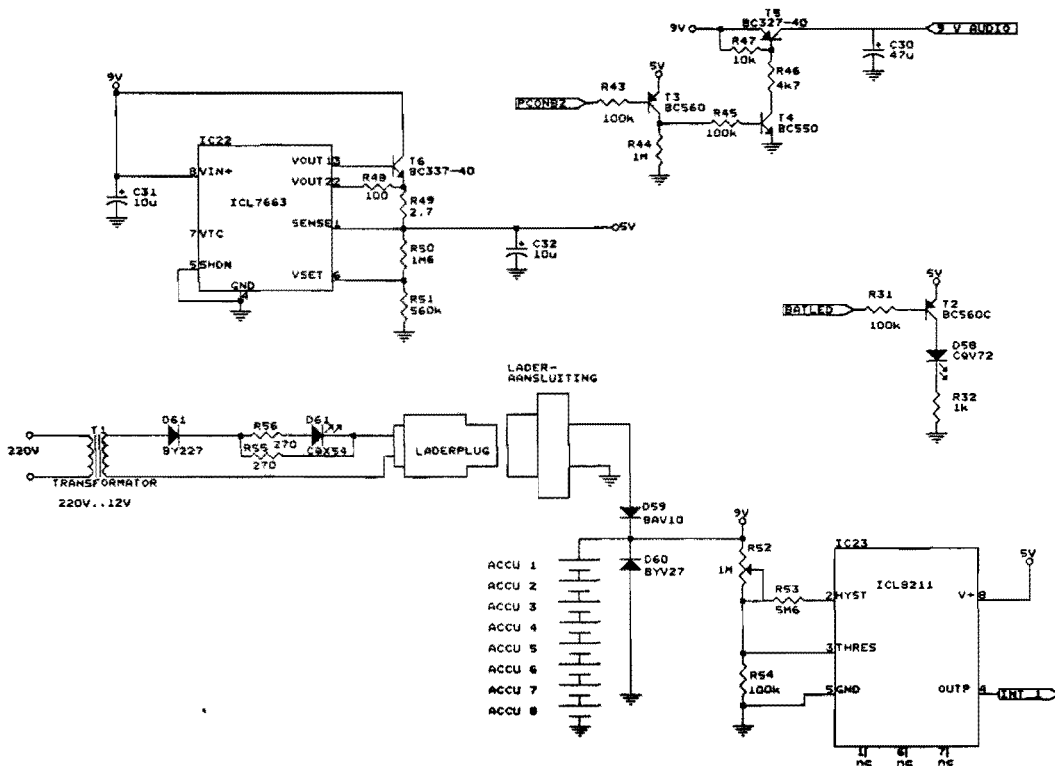
Dit is gerealiseerd door 2 latches voor de uitgangslijnen te nemen. Eén latch is verbonden met de uitgangslijnen van de even rijen; de andere is verbonden met de uitgangslijnen van de oneven rijen. De led-matrix heeft daarom 8 ingangslijnen en 16 uitgangslijnen.

Bij een duty-cycle van 12.5 % worden om de 2 milliseconden 2 rijen aangestuurd.

Het enable signaal voor elk van de 3 latches van de led-matrix wordt gegenereerd door de nor-functie van een chip-enable signaal en het \overline{WR} signaal van de processor.

4.6 Het spanningsverzorgings- en detectiecircuit.

Het spanningsverzorgings- en detectiecircuit is weergegeven in figuur 4.6 en bestaat uit verschillende delen:



Figuur 4.6: Het spanningsverzorgings- en detectiecircuit.

- het spanningsverzorgingscircuit,
- het spanningsdetectiecircuit,
- de accu leeg indicatie,
- de accu lader,
- het circuit dat het audio-gedeelte van spanning voorziet

Deze delen worden achtereenvolgens besproken.

4.6.1 Het spanningsverzorgingscircuit.

Het circuit is opgebouwd rond een ICL7663 spanningsregelaar. De uitgangsspanning van de spanningsregelaar moet 5 Volt bedragen, want de regelaar voedt alle logische IC's van de Dominostem. Volgens de specificaties [5] van de ICL7663 geldt:

$$V_{sense} = V_{set} * (1 + \frac{R_{50}}{R_{51}}) = 5V \quad (1)$$

Waarbij V_{set} 1.29 V (± 15 mV) bedraagt. Uit dit gegeven en (1) kunnen de waarden van R_{50} en R_{51} bepaald worden.

$$\frac{R_{50}}{R_{51}} = 2.88 \quad (2)$$

De waarde voor R_{50} is gekozen op 1.6 M Ω . Deze waarde is gekozen om het stroomverbruik van het spanningsverzorgingscircuit minimaal te houden. De waarde van R_{51} nu volgt uit de verhouding: 560 k Ω .

Verder kan de regelaar zodanig ingesteld worden dat de geleverde stroom begrensd wordt. Dit is gedaan om de schakeling te beschermen tegen kortsluiting. De stroombegrenzing wordt bepaald volgens de relatie [5]:

$$I_{cl} = \frac{0.7V}{R_{49}} \quad (3)$$

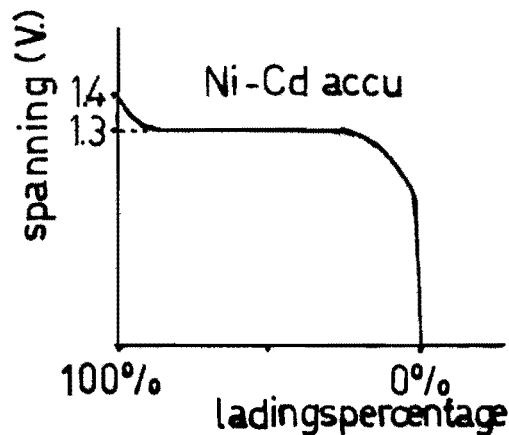
De stroombegrenzing is ingesteld op circa 250 mA met een weerstandswaarde van R_{49} van 2.7 Ω .

4.6.2 Het spanningsdetectiecircuit.

Dit circuit opgebouwd rond een ICL8211 spanningsdetector [5] detecteert wanneer de accuspanning te laag wordt en geeft dit door aan de processor.

De uitgang van de ICL8211 is een open-drain output, die rechtstreeks met een poortlijn van de processor verbonden is. De stroom die de ICL8211 kan laten wegvloeien (sink current) is 7 mA, hetgeen voldoende is om de poortlijn "omlaag te trekken". De poortlijn van de processor die gebruikt is, is een externe interrupt lijn. Derhalve kan de accuspanning altijd gedetecteerd worden.

Het is van belang het spanningsniveau van de Dominostem zodanig te houden, dat de voedingsspanning van elke component binnen de specificaties blijft. Dit is nodig om te voorkomen dat de Dominostem dingen doet die de gebruiker niet verwacht (bijvoorbeeld rare geluiden produceren). Dit zou het geval kunnen zijn wanneer de accu's te ver leeg zijn, en het spanningsdetectiecircuit nog geen actie ondernomen heeft. Om dit te voorkomen moet de schakeling triggeren op een spanningsniveau waarbij de accu's nog niet helemaal leeg zijn. Bij een spanning per accu van 1.1 V is een accu voor 90% leeg. Vanaf een spanning van 1.1 V daalt de accuspanning steil bij afnemende acculading [16] (zie figuur 4.7).



Figuur 4.7: Accuspanning als functie van de acculading.

Derhalve is besloten om het spanningsniveau waarop de schakeling moet triggeren in te stellen op 8.8 V (een spanning van 1.1 V per accu). Om oscillaties van de $\overline{INT1}$ lijn van de processor te voorkomen is een hysteresis

van 0.5 V ingebouwd. De Dominostem zal, wanneer een te lage accuspanning gedetecteerd is, pas weer normaal functioneren als de accuspanning 9.3 V bedraagt.

Deze hysteresis kan gerealiseerd worden met behulp van de hysteresis-uitgang van de ICL8211 [5]. Deze uitgang is een open-drain uitgang, die verbonden is met 5 V indien $V_{thres} > 1.15V$, en hoogohmig wanneer $V_{thres} < 1.15V$. Uitgegaan van deze gegevens kunnen de weerstanden R_{52} , R_{53} en R_{54} bepaald worden:

Uitgaande van $V_{in} > 8.8V$ geldt: $V_{thres} > 1.15V$; $V_{hyst} = 5V$; laten we nu V_{in} dalen tot 8.8 V dan zal V_{outp} gelijk aan 0 V worden bij $V_{in} = 8.8 V$:

$$\frac{V_{in}-1.15V}{R_{52}} + \frac{V_{hyst}-1.15V}{R_{53}} = \frac{1.15V}{R_{54}} \quad (4)$$

Invullen van de waarden van V_{in} en V_{hyst} levert:

$$\frac{7.65V}{R_{52}} + \frac{3.85}{R_{53}} = \frac{1.15V}{R_{54}}$$

Nu is de hysteresis pin van de ICL8211 hoogohmig geworden en levert geen bijdrage tot de stroomvergelijking. V_{outp} is 0 V geworden.

Laten we nu V_{in} weer stijgen tot 9.3 V, dan zal bij 9.3 V de V_{outp} pin weer hoogohmig worden.

$$\frac{V_{in}-1.15V}{R_{52}} = \frac{1.15V}{R_{54}} \quad (5)$$

Invullen van V_{in} levert:

$$\frac{8.15V}{R_{52}} = \frac{1.15V}{R_{54}} \quad (6)$$

Hieruit kunnen de volgende verhoudingen voor de weerstandswaarden bepaald worden:

$$\frac{R_{52}}{R_{54}} = 7.1$$

$$\frac{R_{52}}{R_{53}} = 0.13$$

De volgende waarden zijn gekozen, overeenkomstig de bovenvermelde ver-

houdingen (de waarden zijn zodanig gekozen dat het stroomverbruik van het spanningsdetectiecircuit minimaal blijft):

$$R_{52} = 710 \text{ k}\Omega;$$

$$R_{53} = 5.6 \text{ M}\Omega;$$

$$R_{54} = 100 \text{ k}\Omega.$$

Voor R_{52} is gekozen voor een potmeter van $1 \text{ M}\Omega$ om een goede fijnregeling voor het omslagpunt te krijgen.

4.6.3 De accu-leeg indicatie.

De accu-leeg indicatie wordt gestuurd door P 3.0 van de processor. Indien de processor op P 3.3 een te lage accu-spanning detecteert (P 3.3 is dan laag) zal de processor gedurende 4 seconden de \overline{BATLED} uitgang laag maken. De PNP transistor zal dan opengestuurd worden en de LED zal oplichten.

4.6.4 De acculader.

De acculader [15] is gebouwd rond een 12 V, 0.12 A trafo (Spitznagel). De 2 weerstanden samen met de LED zorgen ervoor dat zodra er stroom geleverd wordt door de lader (als de accu wordt opgeladen) de LED zal gaan branden.

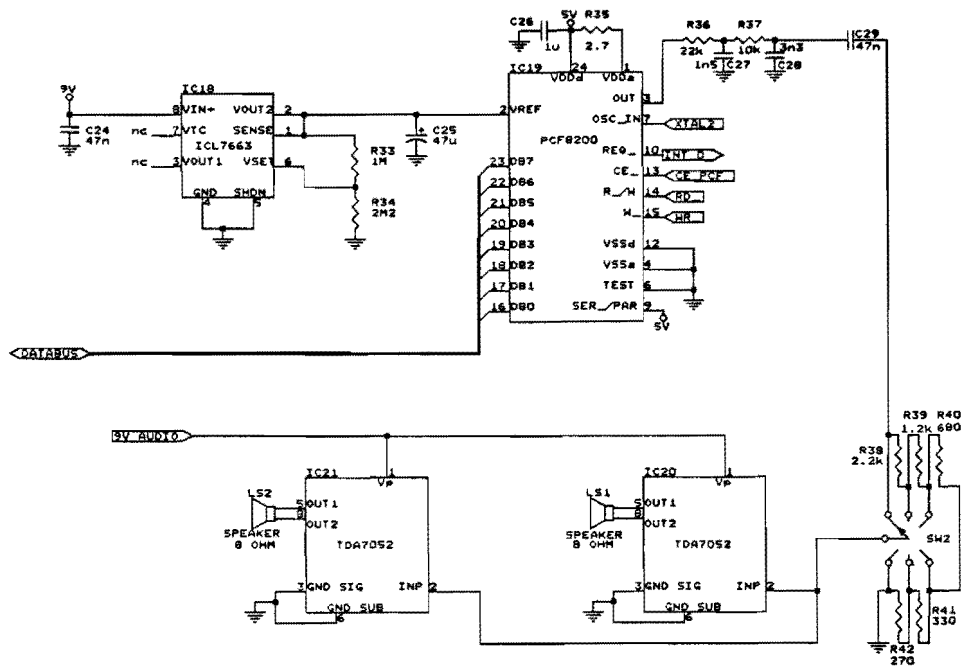
4.6.5 De spanningsvoorziening van het audio-circuit.

De spanningsverzorging van het audio-deel wordt verzorgd door poortlijn 1.7 van de processor, pconb2. Met behulp van dit logische signaal (0 of 5 V) worden de audioversterkers al dan niet van spanning (9 V) voorzien. Dit gebeurt met de 3 transistoren T3, T4 en T5. Het audio-deel wordt pas van spanning voorzien wanneer een zin uitgesproken moet worden.

4.7 Het audio-circuit.

Het audio-circuit is weergegeven in figuur 4.8. Het bestaat uit de volgende onderdelen:

- de PCF8200 spraakchip
- de audioversterkers
- de referentiespanningsregelaar voor de PCF8200



Figuur 4.8: Het audio-circuit.

De 3 onderdelen zullen achtereenvolgens worden besproken.

4.7.1 De spraaksynthesizer.

De Dominostem is gerealiseerd met een PCF8200 spraakchip. De PCF8200 is een CMOS device en kan permanent onder spanning gehouden worden (standby stroom is ongeveer $2 \mu\text{A}$).

De PCF8200 heeft twee gescheiden voedingsaansluitingen voor het digitale en voor het analoge deel van de chip, ter voorkoming van hoogfrequente storing in het audio-signaal. V_{dd_a} is afgeleid van V_{dd_d} met behulp van een R-C schakeling.

4.7.2 De audioversterkers.

Het analoge signaal van de PCF8200 wordt allereerst gefilterd met behulp van de twee R-C trappen aan de analoge uitgang van de PCF8200. Hierachter bevindt zich een condensator die zorgt voor een capacatieve koppeling aan de audio-versterkers [6]. Dit is gedaan omdat de PCF8200 de analoge uitgang verbindt met de referentiespanningspin indien er niet meer gesproken wordt. Om deze gelijkspanning niet op de ingangen van de audio-versterkers te laten is de condensator geplaatst.

Het volume kan ingesteld worden met een 6-standen (laagste stand is volume 0) volume-schakelaar. De weerstanden bij deze schakelaar zijn zodanig gekozen dat bij elke stap hoger of lager de ingangsspanning van de audio-versterkers een faktor 2 hoger respectievelijk lager wordt. Dit levert een vermogensverhouding van 6 dB tussen de verschillende standen. Deze verhouding is gekozen boven een 3 dB verhouding om bij 6 standen toch een redelijk verschil te hebben tussen hoogste en laagste geluidsniveau.

De audio-versterkers zijn van het type TDA7052. Het maximale audio vermogen bedraagt per versterker 1.2 W. Hier is gekozen voor 2 versterkers en 2 luidsprekers om meer geluidsvolume te kunnen produceren.

4.7.3 De referentiespanningsregelaar voor de PCF8200.

De referentiespanning van de PCF8200 is gerealiseerd met een ICL7663 spanningsregelaar. Deze regelaar voorziet de PCF8200 van een spanning van 1.88 V. Aangezien de stroom die de regelaar moet leveren klein is ($5 \mu\text{A}$) hoeft geen transistor toegepast te worden zoals bij de 5 V spanningsregelaar. De weerstandswaarden worden volgens formule (7) (zie [5]) berekend:

$$V_{ref} = 1.88V = V_{set} * (1 + \frac{R_{33}}{R_{34}}) \quad (7)$$

Hieruit volgt voor de verhouding van R_{33} en R_{34} :

$$\frac{R_{33}}{R_{34}} = 0.46$$

De waarde van R_{33} is gekozen op $1 \text{ M}\Omega$, om zodoende het stroomverbruik van de spanningsregelaar minimaal te houden. De waarde van R_{34} volgt nu uit de verhouding: $2.2 \text{ M}\Omega$.

5 De software.

Dit hoofdstuk bevat de beschrijving van de software. Achtereenvolgens worden de volgende onderwerpen behandeld:

- de uitgangspunten bij het ontwerp van de software,
- de opzet van de software,
- het hoofdprogramma,
- de gebruikte subroutines,
- de interrupt routines.

5.1 Uitgangspunten bij het ontwerp.

De software van de Dominostem moet de volgende functies realiseren:

- het uitlezen van de stand van de schakelaar,
- het scannen van het keyboard en interpreteren van een toetsindruk,
- het uitschakelen van de Dominostem,
- het ondernemen van actie wanneer de accuspanning te laag wordt,
- het verzorgen van de LED-indicatie naar de gebruiker toe,
- de audio versterker van spanning voorzien en de PCF8200 van data voorzien.

5.2 Opzet van de software.

Alvorens het besturingsprogramma te beschrijven is het zinvol de indeling van het data-geheugen en het interne RAM van de processor te beschrijven.

Het data-geheugen is 640 kbyte groot en bestaat uit 5 EPROM's. Dit

data-geheugen wordt hier als één geheel beschouwd. De indeling ervan is als volgt (zie figuur 5.1):

adres	data
0	opzoektabel
16 k	adrestabel blok 0..18
24 k	spraakdata blok 0..18
640 k	

Figuur 5.1: Memory map van het data-geheugen.

Het eerste blok (de opzoektabel) is de structuur waarmee een ingevoerde rij symbolen vertaald kan worden naar een zin.

De spraakdata (de data van 1200 zinnen) is opgedeeld in 19 blokken van elk 64 zinnen. De maximale lengte van een blok spraakdata bij de standaard PCF8200 datastructuur is 64 kbyte [6]. Dit is vanwege het feit dat 2 bytes per adres gereserveerd zijn. Aangezien voor de toepassing van de Dominostem de hoeveelheid spraakdata ± 640 kbyte groot is, is het onmogelijk om de spraakdata in één blok te plaatsen. Derhalve worden de 1200 zinnen opgesplitst in blokken van 64 zinnen waarvan blokken spraakdata gegenereerd worden. Er is gekozen voor 64 zinnen per blok omdat bij dit aantal zinnen een blok (bij normale zinlengte) gemiddeld 32 kbyte groot zal zijn. Ook in een ongunstig geval zal bij dit aantal zinnen per blok de grens van 64 kbyte niet overschreden worden.

De indeling van een spraakdatablok is terug te vinden in [6].

Het beginadres van elk spraakdatablok van 64 zinnen is opgeslagen in de adrestabel.

In de processor is 128 byte intern RAM aanwezig. De registers en de betreffende functies zijn weergegeven in tabel 5.1:

Tabel 5.1: De indeling van het RAM.

Register(s)	Functie
0-5	Algemeen gebruik
6	Boodschappenbufferpointer (out)
7	Boodschappenbufferpointer (in)
8-10	Algemeen gebruik
11	Bits 16-19 van spraakdatapointer
12-13	Spraakdatapointer H.O.-L.O.
14-15	Eindadres spraakdata H.O.-L.O.
16-17	Algemeen gebruik
18	Pointer LED rijen
19	Controle-bufferpointer
20	Terugspringadressenpointer
21-22	Registers voor meting tijdsintervallen van 1 en 15 sec.
23-25	Algemeen gebruik
26-31	Boodschappenbuffer
32	Device Status Word
33	Key Status Word
34	Timer Status Word
35-39	Spraakdatabuffer [6]
40	Pitch-start buffer [6]
41-52	Controle-buffer
53-76	Terugspringadressenbuffer
77	Testplaats of RAM intact is t.b.v. initialisatie
78	Lengte blok in opzoektabel
79	Command byte voor PCF8200 [6]
80-87	LEDrijen buffer
88-127	Stack

De volgende in deze tabel voorkomende begrippen behoeven enige toelichting:

- **Boodschappenbuffer:** dit is een buffer waarin nieuwe spreekacties kunnen worden opgeslagen terwijl de Dominostem nog een vorige zin uitspreekt.
- **Controle-buffer:** in de controle-toestand (zie hoofdstuk 3) zullen de LED's oplichten in de volgorde waarin de symbolen ingevoerd zijn. Om dit te kunnen bewerkstelligen zijn de symbolen in de ingevoerde volgorde opgeslagen in het controle-buffer. De controle-buffer-pointer wijst naar het laatst ingevoerde symbool in het controle-buffer.
- **Terugspringadressenbuffer:** in hoofdstuk 2 is reeds kort aangestipt dat het opzoeken van de zin bij een ingevoerde symbolenrij gebeurt met een boomstructuur (hier wordt in detail op ingegaan in hoofdstuk 6). Elke keer wanneer een symbool ingevoerd wordt, zal de Dominostem een knooppunt verder naar beneden gaan in de boomstructuur. Bij elke clear-actie echter moet de Dominostem een knooppunt terug naar boven in de boomstructuur. Om dit te kunnen bewerkstelligen moeten de plaatsen van de knooppunten die gepasseerd zijn, onthouden worden. Deze plaatsen staan in het terugspringadressenbuffer. De terugspringadressenpointer wijst naar het laatst in de buffer geplaatste terugspringadres.
- **LEDrijen buffer:** 15 seconden na het invoeren van een (geldig) symbool zullen de LED's van de ingevoerde symbolen oplichten. De informatie welke LED's wel, en welke LED's niet oplichten in deze situatie is opgeslagen in dit buffer. Het bitpatroon van elk register in dit buffer geeft aan welke LED's in de bijbehorende rij van de LED-matrix aan zijn, en welke uit zijn.

In de RAM plaatsen bevinden zich 3 registers, te weten DSW (device status word), KSW (key status word) en TSW (timer status word) die elk vlaggen bevatten om de status van de Dominostem aan te geven. Zie voor de indeling van elk van de 3 de figuren 5.2 tot en met 5.4:

bit	naam	omschrijving
0	cdisp	geeft controle-toestand aan
1	disp	geeft aan of display-tijd voorbij is
2	fout	geeft aan dat gebruiker fout gemaakt heeft
3	speane	geeft aan of nog zinnen moeten worden uitgesproken
4	speaking	geeft aan dat apparaat spreekt
5	key	geeft toetsactie aan
6	devoff	geeft aan dat apparaat uit staat
7	powfail	geeft aan dat accu leeg is

Figuur 5.2 Het device status word.

bit	naam	omschrijving
0	vb	vaste-boodschap actie
1	sb	symbool actie
2	cl	clear-actie
3	ca	clear-all-actie
4	spr	spreek-actie
5	co	controle-actie
6	out	geeft aan of complete initialisatie nodig is
7	tkspv	voorkomt dat dezelfde zin 2 keer gesproken wordt

Figuur 5.3 Het key status word.

bit	naam	omschrijving
0	even	geeft aan of een even rij aangestuurd wordt
1	coled	geeft aan of de controle led aan is
2	corkey	geeft aan of $\overline{CE}keyb$ gecorrigeerd moet worden
3	-	
4	-	
5	-	
6	-	
7	-	

Figuur 5.4 Het timer status word.

De namen van de poortlijnen in het hardware-schema en in de software zijn niet altijd gelijk. Derhalve zijn ze in figuur 5.5 naast elkaar gezet:

poortlijn	hardware-naam	software-naam
P 1.4	\overline{ESEL}	endale
P 3.0	\overline{BATLED}	nled
P 3.1	SWPOS	switch
P 3.3	$\overline{INT1}$	pcheckb
P 3.4	\overline{CEkeyb}	keyen
P 3.5	\overline{CEpcf}	pcfen

Figuur 5.5: Namen van de poortlijnen.

De software bestaat uit 2 gedeelten, te weten:

- de declaraties, in de file pdec.h.
- het eigenlijke programma, in de file pcf.d.s.

In de file pdec.h staan de declaraties. Dit zijn de definities van de poortlijnen en van het DSW, KSW en het TSW.

In de file pcf.d.s staat het hoofdprogramma en de bijbehorende subroutines.

5.3 Het hoofdprogramma.

De Dominostem moet beginnen met werken als de aan-uit schakelaar van stand verandert, en als een toets wordt ingedrukt (wanneer de Dominostem standby staat). Dan worden de volgende acties ondernomen:

Allereerst wordt nagegaan in welke stand de aan-uit schakelaar na de verandering (of de toetsaanslag) staat.

Staat de schakelaar in de stand “uit”, dan wordt het keyboard inactief gemaakt en gaat de processor powerdown.

Staat de schakelaar in de stand “aan”, dan wordt gecontroleerd of de accuspanning voldoende is. Is dit niet het geval, dan zal de accu-leeg-LED 4 seconden gaan branden en vervolgens het keyboard inactief gemaakt worden en de processor powerdown gaan. Is de accuspanning wel voldoende,

dan zal het toetsenbord gescand worden.

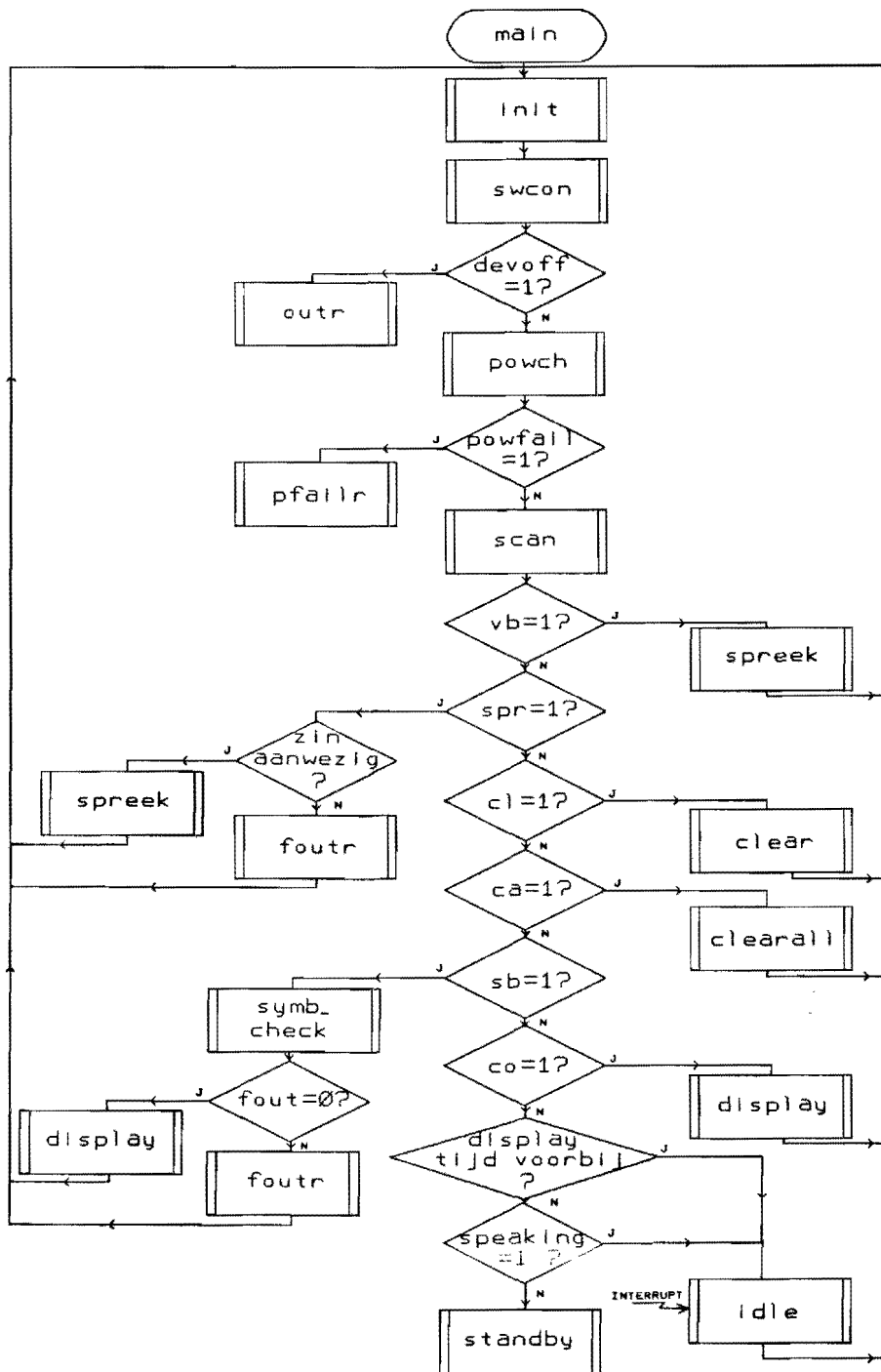
Wordt een toets gevonden die ingedrukt is, dan wordt uitgerekend welke toets en welk soort toets het is (vaste boodschap, symbool, clear, clear-all, spreek of controle). Afhankelijk van welk soort toets wordt de bijbehorende actie ondernomen:

- Bij een symbool-toets wordt gekeken of het zojuist ingevoerde symbool geldig is. Bij een geldig symbool wordt overgegaan tot het display-en van de tot dan toe ingevoerde symbolen. Bij een ongeldig symbool wordt een foutmelding gegeven.
- Bij een spreek-toets wordt eerst gecontroleerd of de ingevoerde symbolencombinatie een zin uit de zinnenset voorstelt, of dat de rij symbolen nog aangevuld moet worden. In het eerste geval wordt de betreffende zin uitgesproken. In het tweede geval vindt een fout-indicatie plaats.
- Bij een vaste-boodschap-toets wordt deze boodschap uitgesproken.
- Bij een clear-toets wordt het laatst ingevoerde symbool uitgewist.
- bij een clear-all-toets worden alle door de gebruiker ingevoerde symbolen uitgewist.
- Bij een controle-toets wordt de volgorde van de symbolen getoond.

Is er geen toets ingedrukt, dan moet de Dominostem:

- “standby” gaan (de processor is powerdown en het keyboard is actief) wanneer de display-tijd voorbij is en de Dominostem klaar is met spreken.
- “idle” gaan (de processor is idle) wanneer de display-tijd niet voorbij of de Dominostem nog spreekt.

De flowchart van het hoofdprogramma is weergegeven in figuur 5.6.



Figuur 5.6: Flowchart van het hoofdprogramma.

5.4 De subroutines.

De in het hoofdprogramma voorkomende subroutines worden in deze paragraaf nader toegelicht. De 2 interrupt routines worden in hoofdstuk 5.5 besproken. In het hoofdprogramma komen de volgende subroutines voor:

1. INIT; initialisatie,
2. SWCON; controleert de stand van de schakelaar,
3. OUTR; schakelt de Dominostem uit,
4. PFAILR; geeft een accu-leeg melding,
5. POWCH; controleert de accuspanning,
6. SCAN; scant het keyboard,
7. SPREEK; spreekt een boodschap uit,
8. FOUTR; geeft een foutmelding,
9. CLEAR; wist het laatste symbool uit,
10. CLEAR-ALL; wist alle symbolen uit,
11. DISPLAY; zorgt dat de LED's oplichten,
12. IDLE; zet processor idle,
13. STANDBY; maakt keyboard actief en zet processor powerdown.
14. SYMB-CHECK; gaat na of een ingevoerd symbool geldig is.

ad 1. De routine INIT initialiseert die delen van het RAM, die geïntialiseerd moeten worden. De routine INIT wordt na elke RESET-puls aangeroepen. De stackpointer wordt na een RESET hardware-matig geïntialiseerd op een waarde van 7. Aangezien deze plaats en de daarop volgende gereserveerd zijn voor andere doeleinden (zie tabel 5.1), moet de stackpointer na elke RESET op de juiste waarde (88) gezet worden.

Vervolgens vergelijkt de processor de testplaats (RAM 77) met a5 (hexadecimaal getal). Zijn beide gelijk, dan gaat de processor ervan uit dat het RAM intakt is. Zijn beide niet gelijk, dan worden een aantal registers geïntialiseerd.

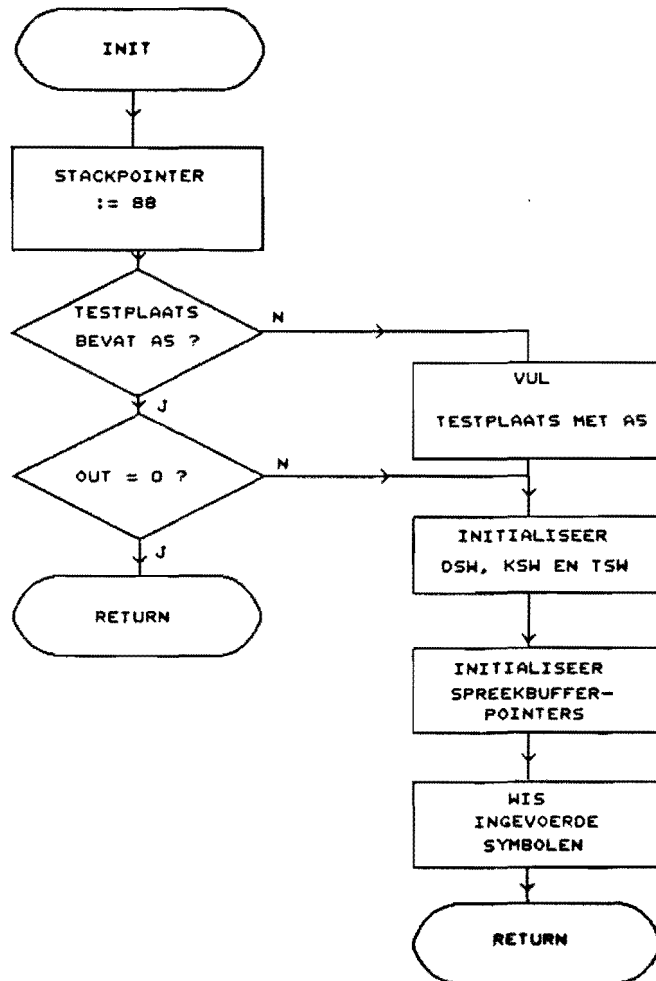
De inhoud van de testplaats wordt vergeleken met a5 (bitpatroon 10100101), omdat de kans dat dit regelmatige bitpatroon na een storing nog aanwezig is, uiterst gering is.

De bovengenoemde registers worden tevens geïntialiseerd wanneer de Dominostem een power failure heeft gehad, of wanneer het apparaat met de aan-uit schakelaar is uitgeschakeld. De flowchart van de routine INIT is weergegeven in figuur 5.7.

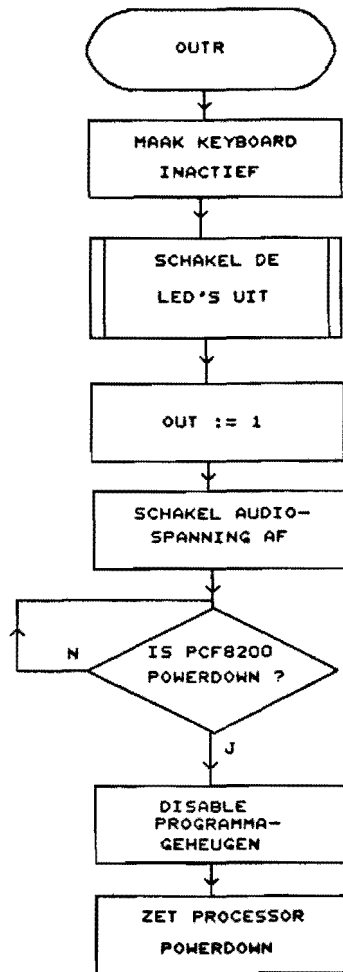
ad 2. De routine SWCON controleert de stand van de aan-uit schakelaar en geeft dit door aan het Device Status Word. Dit gebeurt op dezelfde wijze als bij de Pocketstem [9].

ad 3. De routine OUTR maakt het keyboard inactief door alle ingangslijnen van het keyboard laag te maken (nullen naar keyboard sturen). Deze routine schakelt de audiospanning uit en schakelt de LED's uit. Vervolgens wordt nagegaan of de PCF8200 powerdown is. Indien dit niet het geval is, wordt gewacht tot hij powerdown is.

Vervolgens wordt de processor powerdown geschakeld en het programma geheugen gedisableed. De processor wordt weer actief met behulp van een RESET. De flowchart van de routine OUTR is weergegeven in figuur 5.8.



Figuur 5.7: Flowchart van de routine INIT.



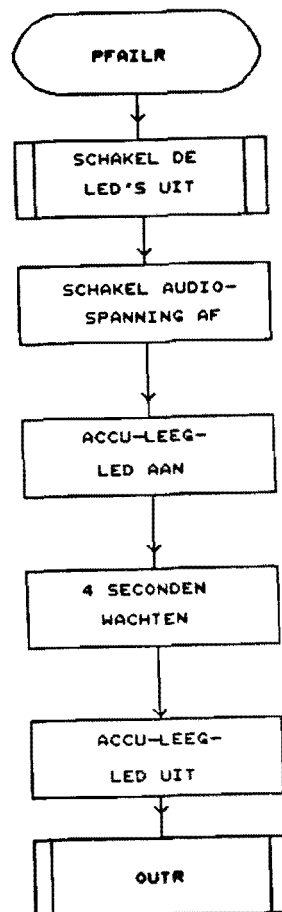
Figuur 5.8: Flowchart van de routine OUTR.

ad 4. De routine PFAILR treedt in werking als de accuspanning te laag is. In deze routine worden eerst de LED's van de symbolen uitgeschakeld. Vervolgens wordt de audiospanning uitgeschakeld. Hierna licht de acculeeg-LED 4 seconden op. Tenslotte wordt OTR aangeroepen, die de Dominostem uitschakelt.

De routine PFAILR wordt bereikt:

- Via het hoofdprogramma (als het bit powfail geset is).
- Via een interrupt wanneer de Dominostem spreekt.

De flowchart van de routine PFAILR is weergegeven in figuur 5.9.



Figuur 5.9: Flowchart van de routine PFAILR.

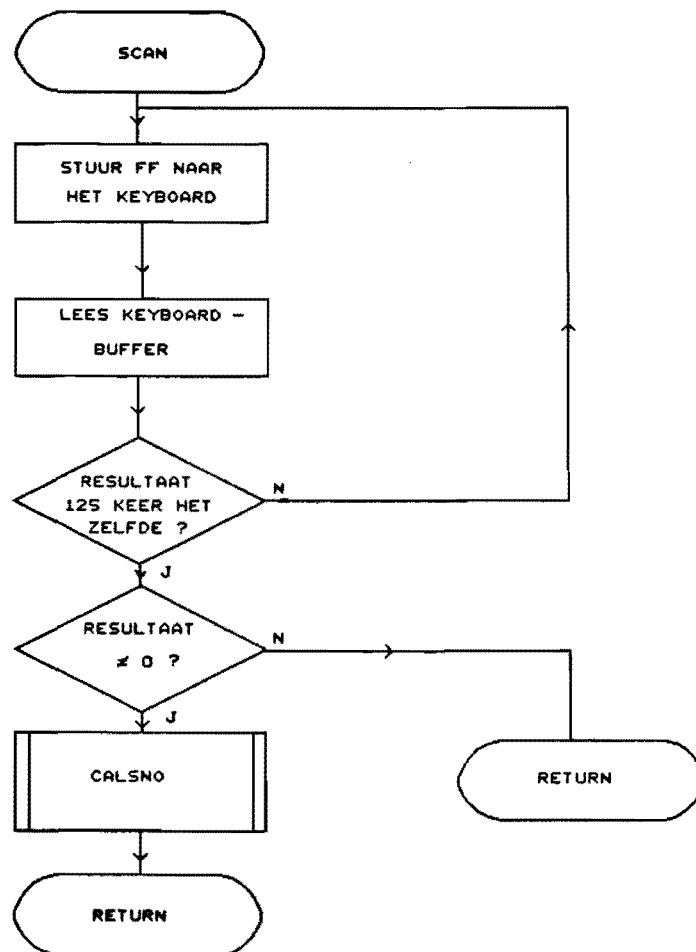
ad 5. De routine POWCH controleert of de voedingsspanning hoog genoeg is. Dit gebeurt door poortlijn 3.3 te testen. Deze lijn is met de uitgang van het spanningsdetectie-circuit verbonden. Het controleren gebeurt op dezelfde wijze als bij de Pocketstem [9].

ad 6. De routine SCAN gaat na of er een toets is ingedrukt en zoja, welke dat is.

Allereerst wordt nagegaan of er een toets is ingedrukt. Dit wordt gedaan door alle ingangslijnen hoog te maken. Als er een toets is ingedrukt, dan zal één der uitgangslijnen hoog worden en vindt de processor een waarde ongelijk aan nul bij het uitlezen van de waarde van het keyboard buffer. Om dender te ondervangen wordt gewacht tot 125 keer hetzelfde resultaat gelezen wordt uit het buffer. 125 keer hetzelfde resultaat betekent dat het resultaat ongeveer 5 milliseconden stabiel moet zijn. Daar de duur van dender ongeveer 10 milliseconden (en de frequentie van de dender grofweg een paar kilohertz bedraagt) bedraagt, kan dender hierdoor ondervangen worden.

Als het resultaat stabiel is kan worden bekeken welke toets is ingedrukt. Dit wordt gedaan door elke ingangslijn afzonderlijk hoog te maken en te kijken tot het buffer een waarde ongelijk aan nul bevat. Op dat moment kan berekend worden welke de ingedrukte toets is.

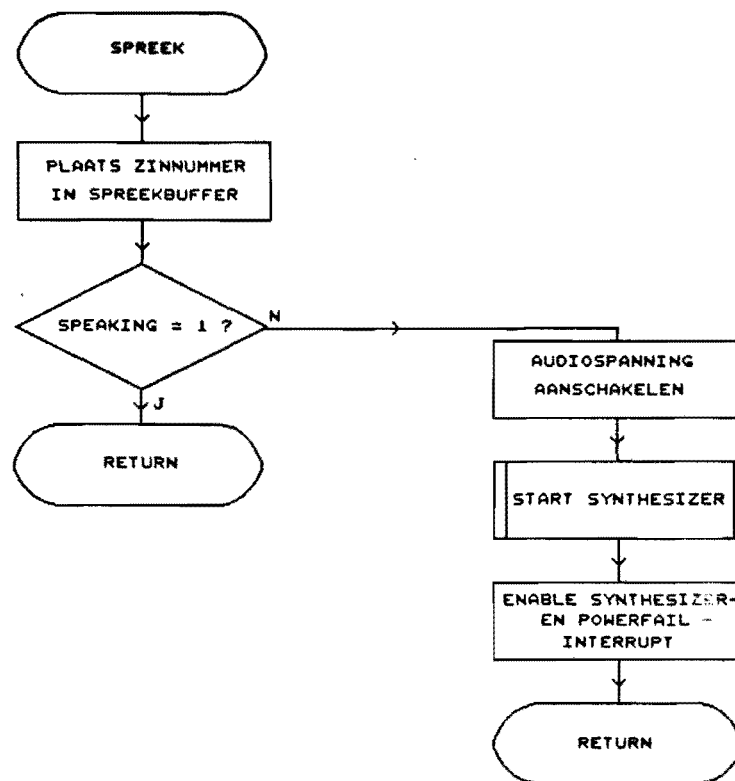
Vervolgens kan bepaald worden (in de routine CALSNO) welk soort toets (symbool, spreek, enzovoort) het is en welk nummer binnen dit soort de ingedrukte toets heeft. Dit laatste geldt alleen voor symbolen en vaste boodschappen. De flowchart van de routine SCAN is weergegeven in figuur 5.10.



Figuur 5.10: Flowchart van de routine SCAN.

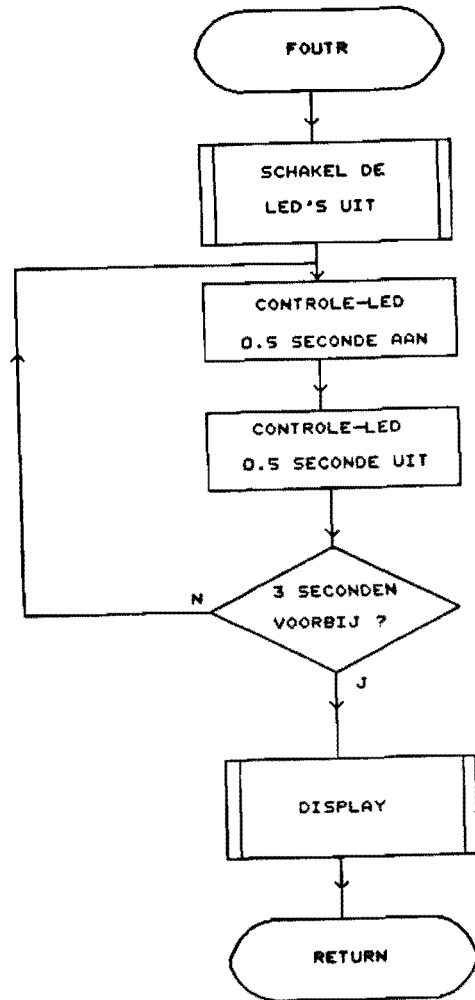
ad 7. De routine SPREEK plaatst het boodschapnummer in het spreekbuffer (indien deze niet vol is). Vervolgens wordt, indien de Dominostem niet spreekt, de synthesizer opgestart [6]. De verdere afhandeling van data-transport tussen processor en synthesizer vindt in de interrupt routine plaats.

Indien de Dominostem wel spreekt, wordt de synthesizer in de interrupt routine opgestart wanneer de andere zin(nen) uitgesproken is (zijn). De flowchart van de routine SPREEK is weergegeven in figuur 5.11.



Figuur 5.11: Flowchart van de routine SPREEK.

ad 8. De routine FOUTR laat gedurende 3 seconden de LED onder de controle toets knipperen. Na deze 3 seconden lichten de LED's van de ingevoerde symbolen weer op. De flowchart van FOUTR is weergegeven in figuur 5.12.

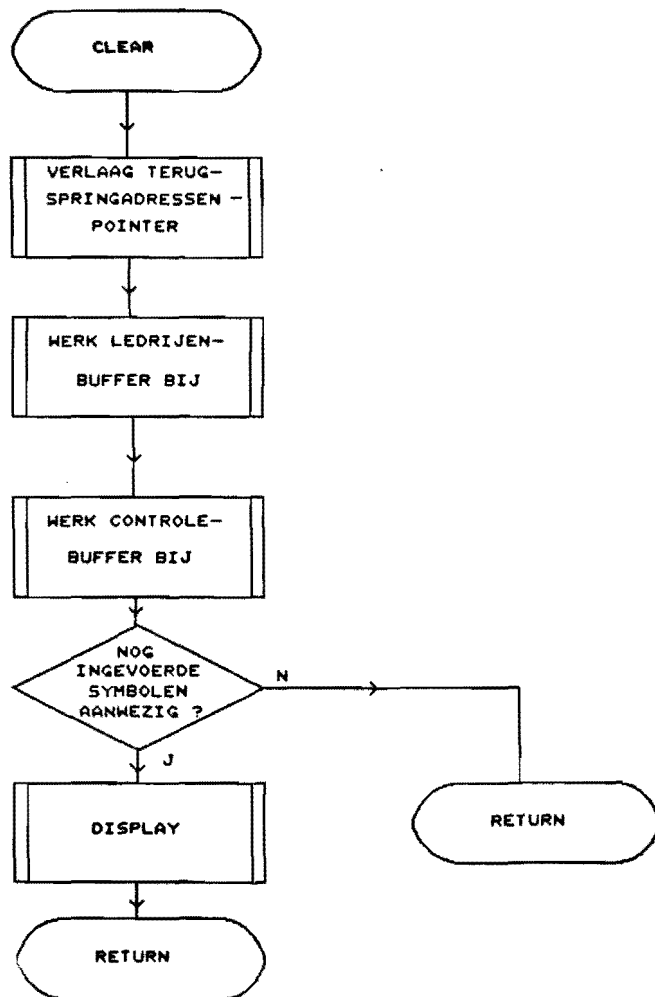


Figuur 5.12: Flowchart van de routine FOUTR.

ad 9. De subroutine CLEAR wist het laatst door de gebruiker ingevoerde symbool uit. Hiertoe moet de Dominostem één knooppunt terug in de boom (zie ook hoofdstuk 6). Het adres van dit knooppunt in de opzoektabel is het bovenste in de terugspringadressenbuffer. De terugspringadressenpointer moet dus met één verlaagd worden.

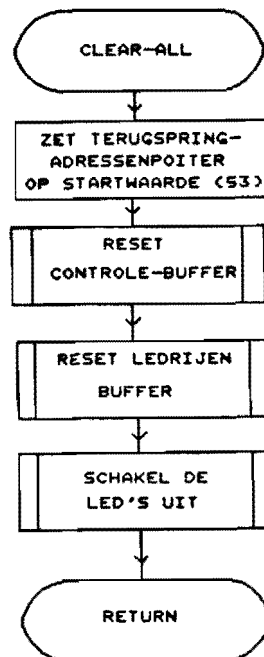
Verder hoeft de LED van het zojuist gewiste symbool niet meer op te lichten. Hiertoe moeten het controle-buffer en het LEDrijen-buffer bijgewerkt worden. Na deze acties wordt de routine display aangeroepen. Deze zorgt ervoor dat de LED's van de resterende symbolen oplichten.

De flowchart van de routine CLEAR is weergegeven in figuur 5.13.



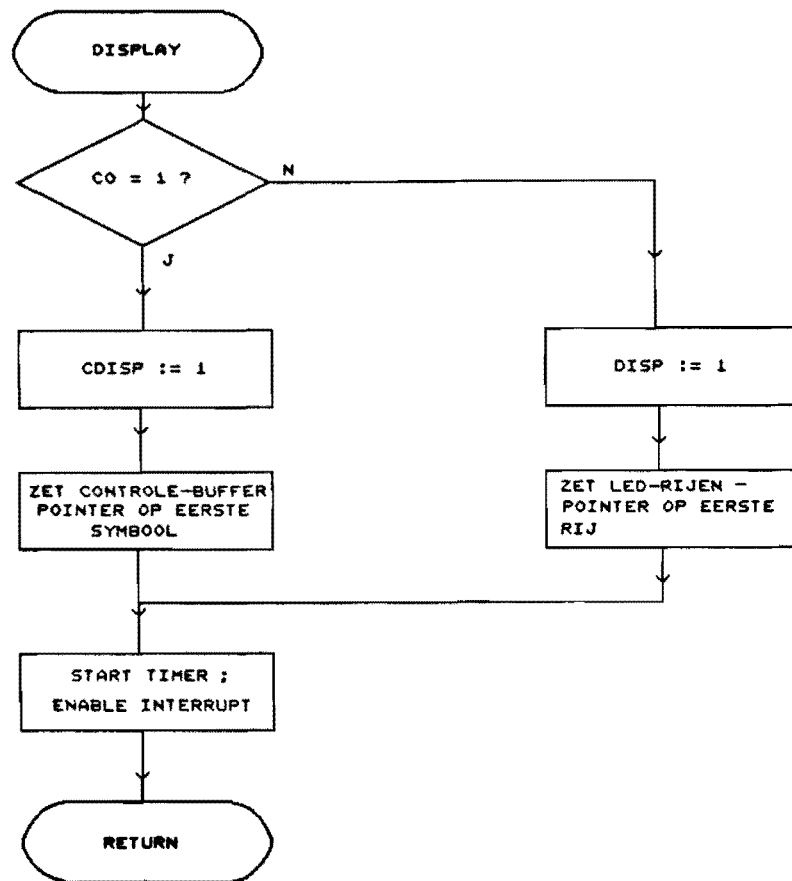
Figuur 5.13: Flowchart van de routine CLEAR.

ad 10. De routine CLEAR-ALL wist alle door de gebruiker ingevoerde symbolen uit. De Dominostem moet dan teruggaan naar het begin van de boomstructuur (opzoektabel). Hiertoe wordt de terugspringadressenpointer op de startwaarde gezet (RAM 53). Verder hoeven er geen LED's meer op te lichten, daar alle symbolen uitgewist zijn. De display-buffer en controle-display buffer worden gereset. De flowchart van de routine is weergegeven in figuur 5.14.



Figuur 5.14: Flowchart van de routine CLEAR-ALL.

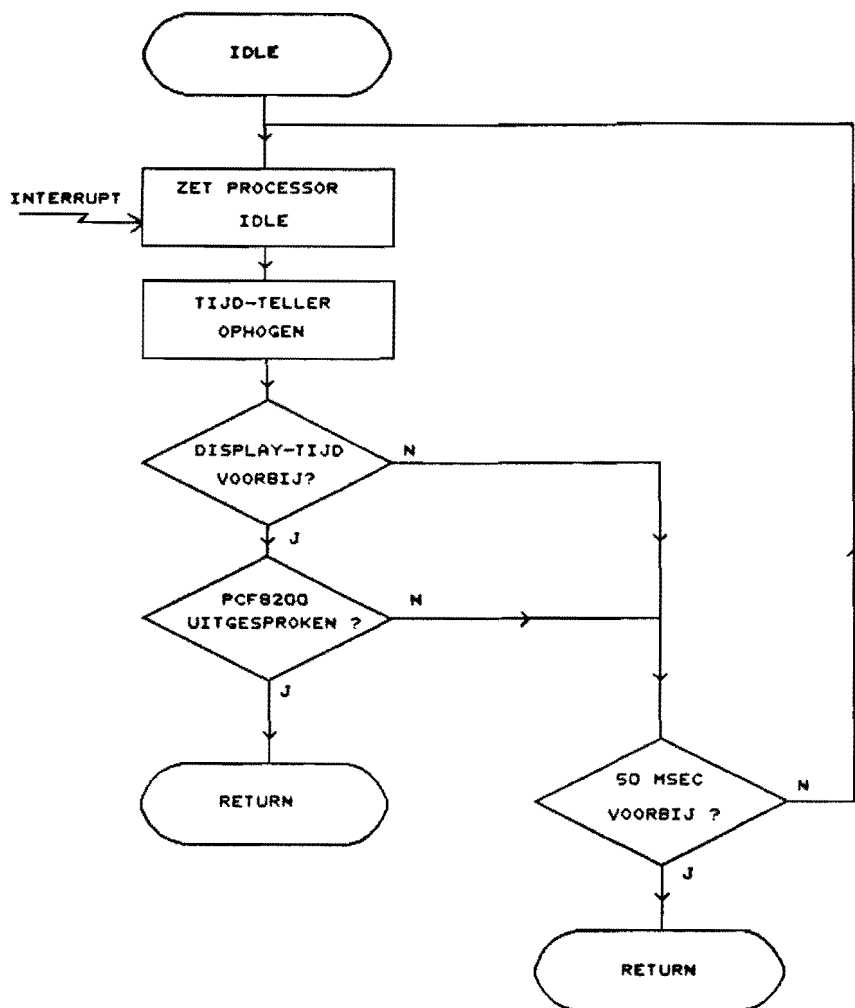
ad 11. De routine DISPLAY zorgt ervoor dat het aansturen van de LED's opgestart wordt. Ten eerste wordt nagegaan welke indicatie moet plaatsvinden (controle of de LED's van de ingevoerde symbolen oplichten) en wordt het betreffende bit geset. Bij controle wordt het bit "cdisp" geset, terwijl bij "de LED's van de ingevoerde symbolen laten oplichten" het bit "disp" geset wordt. Verder wordt de betreffende pointer (LED-rijen- of controle-buffer-pointer) op de startwaarde gezet. Tenslotte wordt de timer geladen met een tijdsinterval van 2 milliseconden en wordt de timer-interrupt ge-enabled. De flowchart van de routine DISPLAY is weergegeven in figuur 5.15.



Figuur 5.15: Flowchart van de routine DISPLAY.

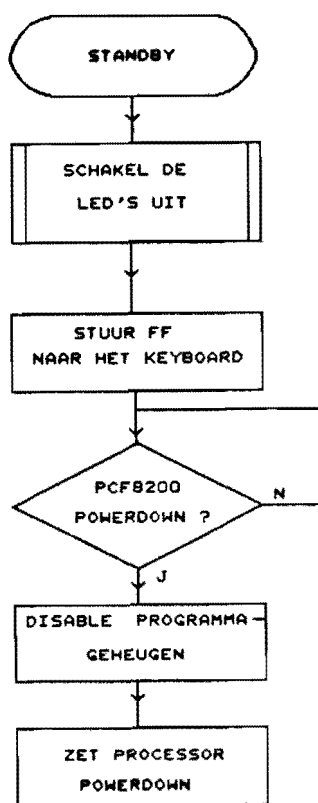
ad 12. In de routine IDLE wordt de processor in de idle-mode gezet. Dit is een toestand waarin de processor niet actief is. In de tijd dat de processor idle is wacht hij als het ware op een interrupt van de PCF8200, of van de (interne) timer. Als een interrupt komt, handelt de betreffende interrupt routine de interrupt af. Na de afhandeling van elke interrupt wordt een teller opgehoogd. Wanneer de inhoud van deze teller overeenkomt met een tijdsinterval dat kleiner dan of gelijk is aan 50 milliseconden, gaat de processor weer idle. Wanneer de waarde overeenkomt met een tijdsinterval dat groter is dan 50 milliseconden, wordt de teller op nul gezet en gaat het programma terug naar het hoofdprogramma, om het toetsenbord te scannen.

De flowchart van de routine IDLE is weergegeven in figuur 5.16.



Figuur 5.16: Flowchart van de routine IDLE.

ad 13. In de routine STANDBY worden eerst de LED's uitgeschakeld. Vervolgens wordt het keyboard actief gemaakt (er wordt FF naar het keyboard gestuurd). Hierna wordt de spanning van de audioversterkers uitgeschakeld. Als de PCF8200 nog niet powerdown is, wordt gewacht tot hij powerdown is. Vervolgens wordt het programma-geheugen ge-disabled en de processor in de powerdown mode gezet. De flowchart van de routine STANDBY is weergegeven in figuur 5.17.



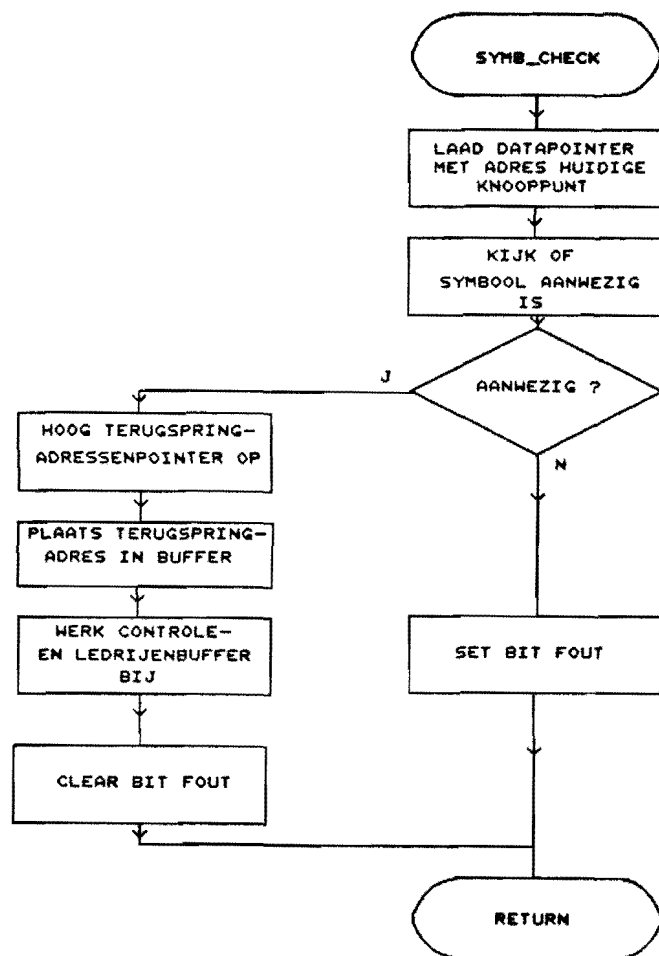
Figuur 5.17: Flowchart van de routine STANDBY.

ad 14. De routine SYMB-CHECK gaat na of het ingevoerde symbool geldig is. Hiertoe wordt het adres van het huidige knooppunt in de boomstructuur (de terugspringadressenpointer wijst naar een register, waar het adres van dit knooppunt staat) in de data-pointer geladen. In het knooppunt wordt gezocht of het ingevoerde symbool aanwezig is.

Indien dit het geval is, dan wordt de terugspringadressenpointer opgehoogd (één knooppunt verder in de boomstructuur gaan) en wordt het adres van het volgende knooppunt in het terugspringadressenbuffer gezet. Het bit "fout" wordt nul gemaakt. Vanaf nu moet de LED onder het zojuist ingevoerde symbool oplichten (zowel bij controle als bij de LED's van de ingevoerde symbolen oplichten). Hiertoe moeten het controle-buffer en het LEDrijen-buffer bijgewerkt worden.

Indien dit niet het geval is, wordt het bit "fout" geset.

De flowchart van de routine SYMB-CHECK is weergegeven in figuur 5.18.



Figuur 5.18: Flowchart van de routine SYMB-CHECK.

5.5 De interruptroutines.

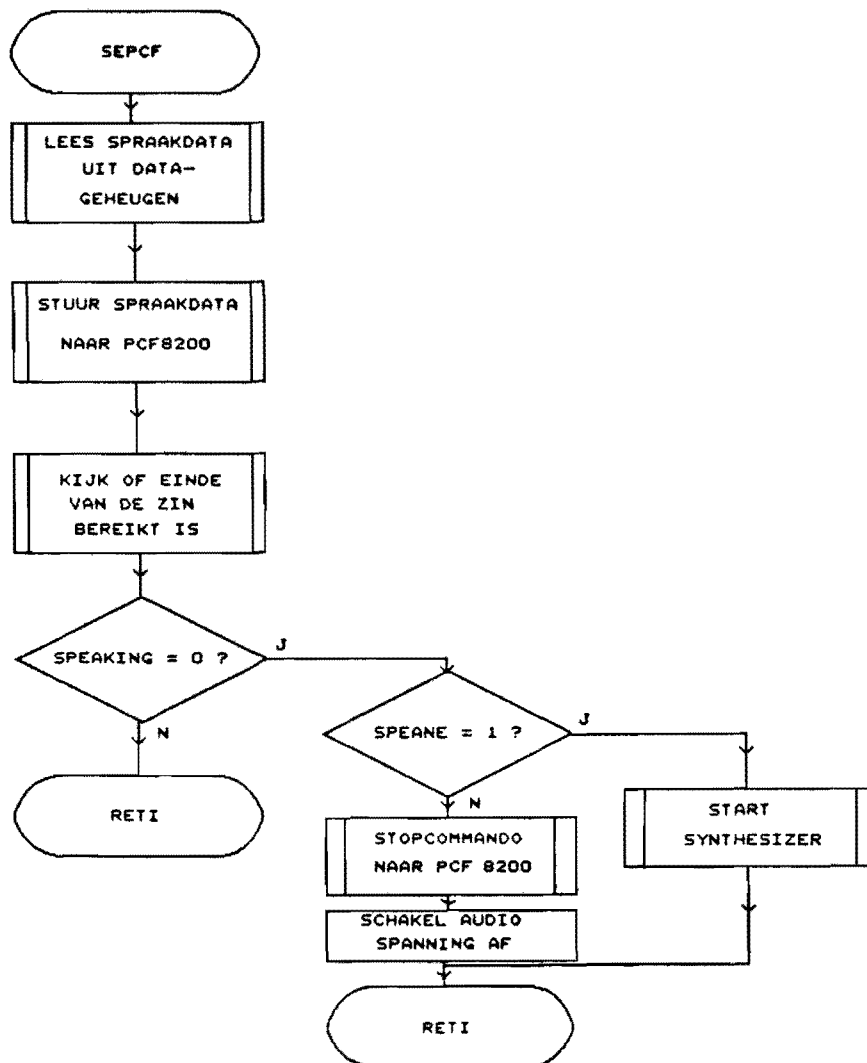
In deze paragraaf worden de interrupt routines van de timer en de PCF8200 besproken.

5.5.1 De PCF8200 interrupt routine.

Nadat in de routine SPREEK de synthesizer opgestart is [6], vindt het verdere data-transport naar de PCF8200 in de interrupt routine SEPCF plaats. Deze interrupt routine leest de spraakdata uit het data-geheugen. Vervolgens wordt deze data naar de PCF8200 gestuurd. Hierna wordt nagegaan of het einde van de zin bereikt is. Als het einde bereikt is (dan geldt $speaking=0$) kunnen zich twee situaties voordoen:

- Er moet nog een zin worden uitgesproken (dan geldt $speane=1$). Dan wordt de PCF8200 weer opgestart.
- Er hoeft geen zin meer uitgesproken te worden (dan geldt $speane=0$). Dan wordt de PCF8200 gestopt, de audiospanning wordt uitgeschakeld, en de routine keert terug.

De flowchart van de interrupt routine is weergegeven in figuur 5.19.



Figuur 5.19: Flowchart van de routine SEPCF.

5.5.2 De timer-interrupt routine.

De timer-interrupt routine (TIR) zorgt ervoor dat de LED's aangestuurd worden. Aangezien er 2 soorten indicatie mogelijk zijn, wordt er eerst nagegaan welke indicatie (controle of alle LED's van de ingevoerde sym-

bolen oplichten) er moet plaatsvinden. Afhankelijk van de waarde van het bit `cdisp` zal de betreffende routine aangeroepen worden. In het geval dat `cdisp=1`, zal de routine "conin" aangeroepen worden. Deze routine verzorgt de controle-indicatie. In het andere geval, `cdisp=0`, zal de routine "ledin" aangeroepen worden. Deze routine verzorgt de indicatie waarbij de LED's van de ingevoerde symbolen oplichten.

De routine CONIN verzorgt de controle-indicatie. Hierbij worden elke 2 milliseconden 2 rijen LED's aangestuurd. Bij deze indicatie (felle sterkte) worden de LED's aangestuurd met een duty-cycle van 25% (zie figuur 5.20a).

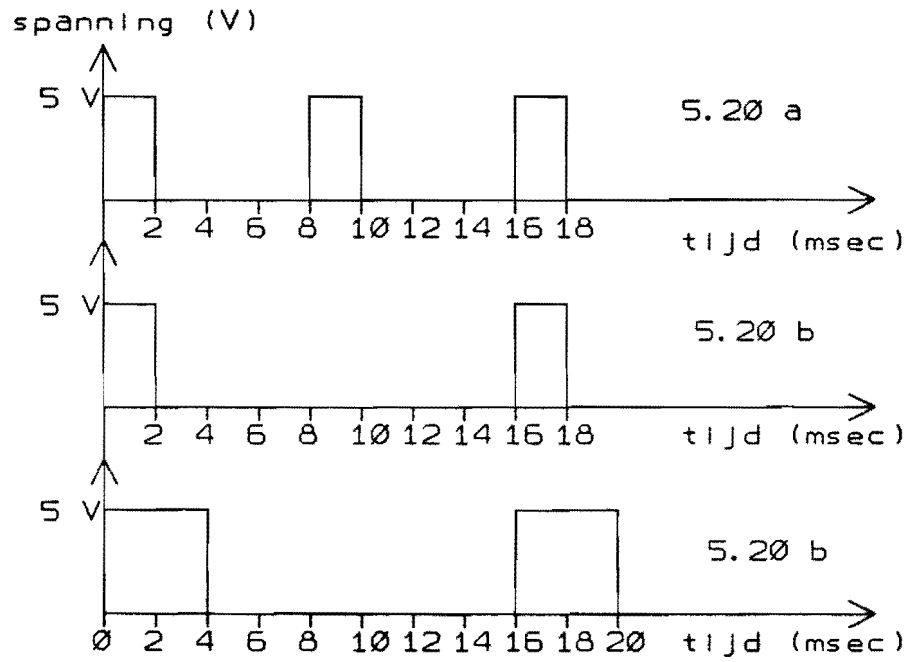
Wanneer de Dominostem de volgorde van de symbolen (conin) heeft laten zien, gaat het programma over tot het laten oplichten van de LED's van de ingevoerde symbolen (ledin).

De routine LEDIN realiseert twee functies, te weten:

- één seconde na het invoeren van het laatste geldige symbool, zal de LED van dit symbool fel oplichten, terwijl de overige LED's op halve sterkte oplichten (zie eisenblad, bijlage 2).
- na één seconde zullen alle LED's van de ingevoerde symbolen oplichten op halve sterkte. Dit laatste duurt 15 seconden.

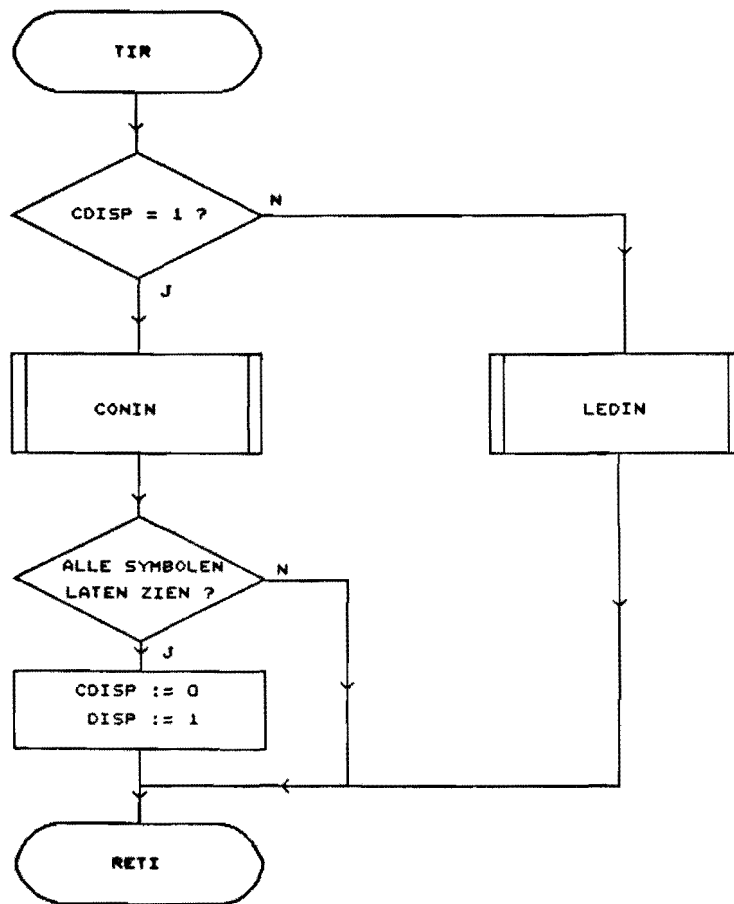
De eerste seconde na het invoeren van een symbool zal de LED van het laatste symbool met een duty-cycle van 25% oplichten, terwijl de overige LED's met een duty-cycle van 12.5% oplichten.

Na één seconde lichten de LED's met een duty-cycle van 12.5% op (zie figuur 5.20b, waar verticaal de spanning op de ingangslijn van de LED van het laatste symbool, en horizontaal de tijd uitgezet is).



Figuur 5.20: De duty-cycle bij verschillende aansturingen.

De flowchart van TIR is weergegeven in figuur 5.21.



Figuur 5.21: Flowchart van de routine TIR.

6 De ondersteuningssoftware.

In dit hoofdstuk wordt de ondersteuningssoftware voor de Dominostem besproken. Deze software kan verdeeld worden in 2 onderdelen:

- het genereren van de opzoektabel. De opzoektabel is een structuur waarmee het besturingsprogramma van de Dominostem bij een ingevoerde symbolenrij het bijbehorende zinnummer kan opzoeken.
- het genereren van de spraakdata van alle zinnen uit de zinnenset.

Deze twee zaken worden achtereenvolgens besproken in dit hoofdstuk.

6.1 De opzoektabel.

De zinnenset van de Dominostem bestaat uit circa 1200 zinnen. Elke zin kan geselecteerd worden met behulp van een unieke symbolenrij.

De Dominostem moet bij, een door de gebruiker ingevoerde symbolenrij, een omzetting maken naar een zin, die vervolgens uitgesproken wordt. Deze omzetting geschiedt met behulp van de opzoektabel. Op de VAX is een programma gemaakt dat de opzoektabel genereert voor een aangeboden zinnenset.

Allereerst wordt de structuur van de opzoektabel besproken. Vervolgens wordt het algoritme besproken waarmee de opzoektabel gegenereerd wordt.

6.1.1 De structuur van de opzoektabel.

De opzoektabel heeft een boomstructuur. Voor deze structuur is gekozen, daar het in een dergelijke structuur mogelijk is op een overzichtelijke wijze een symbolenrij op te zoeken.

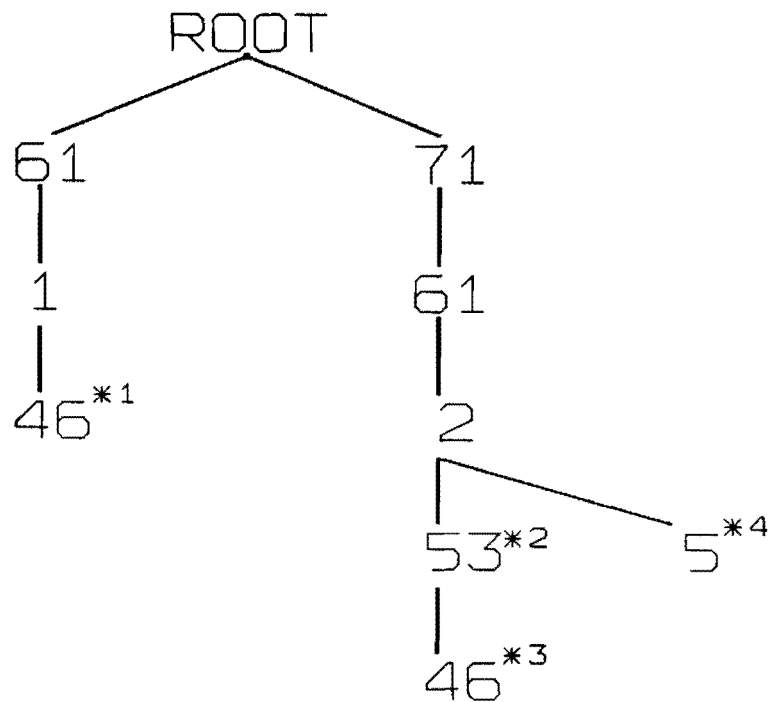
Om het principe van de opzoektabel te verduidelijken is voor de onderstaande 4 zinnen (met bijbehorende symbolenrijen) de boomstructuur gegeven in figuur 6.1. (zie pagina 12).

61 - 1 - 46 (zin 1)

71 - 61 - 2 - 53 (zin 2)

71 - 61 - 2 - 53 - 46 (zin 3)

71 - 61 - 2 - 5 (zin 4)



Figuur 6.1: De boomstructuur.

Hierbij worden de volgende definities gehanteerd:

- Een knooppunt is een plaats in de boom die aan één van de twee volgende voorwaarden voldoet:
 1. op die plaats staat één getal,
 2. op die plaats staat één getal tesamen met een ster en nog een getal.
- “Root” is eveneens een knooppunt.
- Een tak is een verbindingslijn tussen 2 knooppunten.
- Een pad is een verzameling takken van de root naar een bijbehorend knooppunt.

- Indien bij een knooppunt een ster staat betekent dit dat de symbolencombinatie behorend bij het betreffende pad een zin uit de zinnenset vormt. Het zinnummer behorend bij deze symbolencombinatie staat rechts van de ster bij het betreffende knooppunt.

In deze boom kunnen de 4 zinnen uit de zinnenset opgezocht worden. Voor zin 3 bijvoorbeeld gaat dat als volgt:

Vanuit root kijk je of er een knooppunt (bereikbaar met een tak vanuit root) is met symbool 71. Als dat bestaat, ga dan naar dat knooppunt. Als dat niet het geval is, dan komt de zin niet in de zinnenset voor.

Vanuit dit knooppunt (met symbool 71) kijk je of er een volgend knooppunt bestaat met symbool 61. Zoja, ga dan naar dat knooppunt toe. Indien dit niet zo is, dan komt de zin niet in de zinnenset voor. Voor de laatste 3 symbolen (2, 53 en 46) uit de combinatie kunnen dezelfde handelingen gedaan worden. Uiteindelijk kom je dan aan in het knooppunt gemarkeerd met een sterretje en een 3. Dit betekent dat de symbolenrij bij dit pad van de boom een zin gevormd heeft, en wel zin 3.

In deze opzoektabel kunnen aan een willekeurig knooppunt (K) een aantal gegevens toegekend worden:

- de symbolen behorende bij de knooppunten volgend op het knooppunt K.
- een zinnummer (ongelijk aan 0), indien de symbolenrij behorende bij K een zin gevormd heeft.
- een zinnummer (gelijk aan 0), indien de symbolenrij behorende bij K geen zin gevormd heeft.

De boomstructuur wordt tijdens het opzoeken altijd doorlopen van knooppunt naar knooppunt. Bij een boomstructuur is dat eenvoudig, daar onderscheid gemaakt kan worden in richtingen. Wanneer de boomstructuur in een ééndimensionale ruimte (EPROM) opgeslagen wordt, is meer informatie nodig om te weten waar een volgend knooppunt zich bevindt. Hiertoe wordt ook opgeslagen op welk adres een volgend knooppunt zich bevindt. De bovengenoemde drie items tesamen met enkele hierna te bespreken gegevens worden opgeslagen in zogeheten blokken. Elk knooppunt in de

boom is vergelijkbaar met een blok in de EPROM. In elk blok staan naast de drie bovengenoemde items, de volgende zaken:

- De lengte van het blok: deze is bepaald als het aantal getallen in het blok. Het besturingsprogramma van de Dominostem heeft dit nodig om te weten waar een ander knooppunt begint.
- De adressen in het geheugen (EPROM) waar deze knooppunten zich bevinden.

Om dit geheel wat inzichtelijker te maken wordt het voorbeeld van figuur 6.1 uitgewerkt. Het blok horende bij het knooppunt root ziet er als volgt uit:

(lengte)	6	0	(zinnummer)
(symbool)	61	xx	(adres)
(symbool)	71	yy	(adres)

De lengte (6) is het aantal getallen in het blok. Het zinnummer is 0 vanwege het feit dat geen zin gevormd is in root. De mogelijke symbolen vanuit root zijn 61 en 71. De adressen achter elk symbool (xx en yy) zijn de adressen waar het betreffende blok (knooppunt) staat.

De complete opzoektabel voor de zinnenset van figuur 6.1 is weergegeven in tabel 6.1. Hoe deze tabel gegenereerd wordt, wordt in de volgende paragraaf besproken.

Tabel 6.1: De opzoektabel.

epromadres	data	
0	6	0
2	61	6
4	71	16
6	4	0
8	1	10
10	4	0
12	46	14
14	2	1
16	4	0
18	61	20
20	4	0
22	2	24
24	6	0
26	5	30
28	53	32
30	2	4
32	4	2
34	46	36
36	2	3

De afscheidingen tussen blokken (hier gegeven door horizontale lijnen) zijn er in werkelijkheid niet. Waar een volgend blok begint, moet berekend worden door het besturingsprogramma van de Dominostem.

De blokken die 2 getallen groot zijn, zijn de eindpunten in de boom.

Het opzoeken van de combinatie 71-61-2-53-46 in deze tabel gaat als volgt: We starten in het eerste blok op adres 0. Het adres van het knooppunt 71 is 16. Op adres 16 begint dus het blok behorende bij het symbool 71. Het adres van het knooppunt met symbolenrij 71-61 is 20. Op adres 20 begint dus het blok horende bij symbolenrij 71-61. Het adres van het knooppunt met symbolenrij 71-61-2 is 24. Op adres 24 begint het blok horende bij de symbolenrij 71-61-2. In dit blok kan het adres van het blok met symbolenrij 71-61-2-53 opgezocht worden, namelijk 32. Uit het blok dat start op adres 32 kan het adres van het blok met symbolenrij 71-61-2-53-46 opgezocht

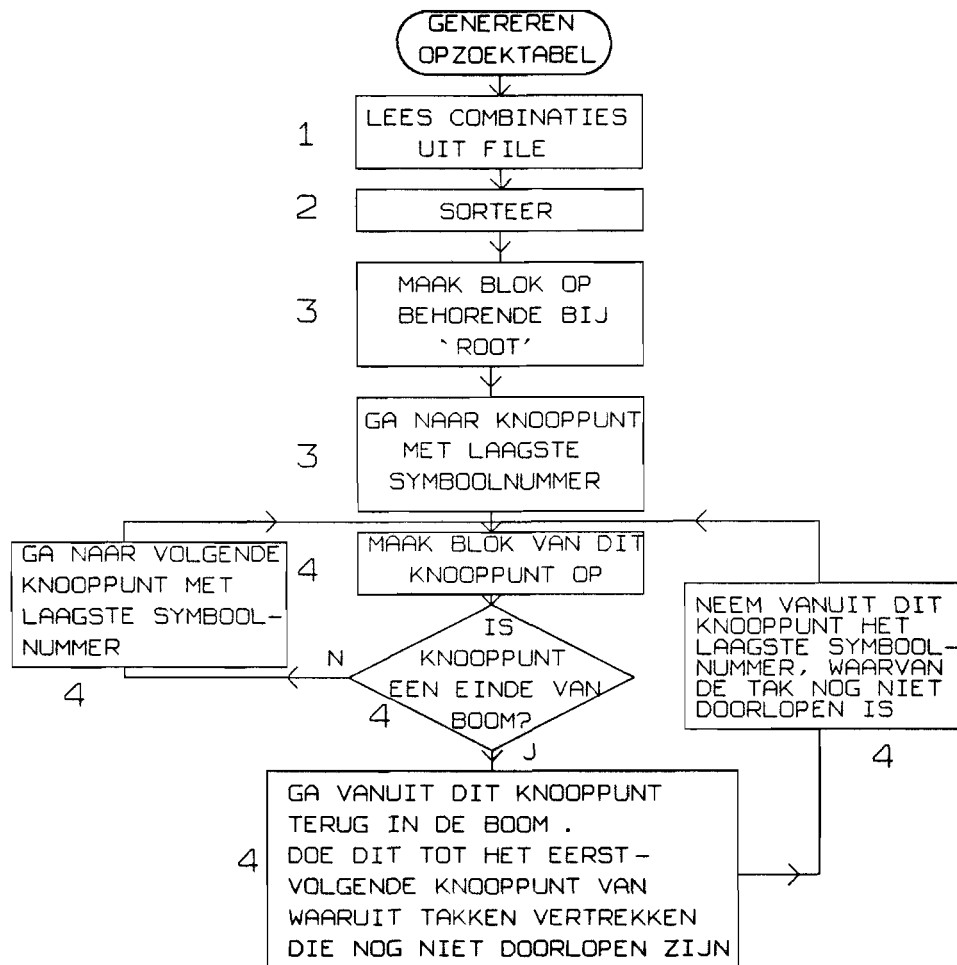
worden, namelijk 36. Uit het blok dat start op adres 36 blijkt dat het gevraagde zinnummer 3 is.

6.1.2 Het genereren van de opzoektabel.

In de file “dominostem.tex” staat de complete zinnenset voor de Dominostem in de volgende structuur:

```
! symbolenrij 1
tekst zin 1
! symbolenrij 2
tekst zin 2
!....
```

De opzoektabel wordt nu als volgt gegenereerd (zie ook figuur 6.2, waar de flowchart van het algoritme is weergegeven). N.B.: De nummers in de flowchart komen overeen met de nummers van de handelingen in het algoritme.



Figuur 6.2: Het algoritme voor het genereren van de opzoektabel.

1. De symbolencombinaties worden uit de file "dominostem.tex" gelezen. De teksten van de zinnen zijn hier niet relevant. De zinnen krijgen een oplopend zinnummer in de volgorde zoals ze in de file staan (de teksten worden in dezelfde volgorde gelezen, genummerd en omgezet in spraakdata).
2. De symbolencombinaties worden gesorteerd op lexicografische index.

3. Er wordt nagegaan welke takken er vanuit "root" vertrekken en welke symbolen daarbij horen. Met behulp van deze symbolen wordt het eerste blok van de tabel behorende bij "root" opgemaakt:

- De lengte van het blok is $2 \times$ aantal takken (dat vertrekt vanuit root) + 2.
- zinnummer = 0
- De symbolen behorende bij de takken vanuit "root", elk gevolgd door het adres van het volgende blok. Deze adressen echter zijn op dit moment nog niet bekend, want de betreffende blokken moeten nog opgemaakt worden. De adressen worden voorlopig allemaal nul gemaakt.

Zet het adres van dit blok ("root") op een stack. Op deze stack worden de blokken gezet, waar nog beginadressen ingevuld moeten worden. Ga nu naar het knooppunt dat hoort bij het laagste symboolnummer.

4. Ga na of de symbolencombinatie behorende bij dit knooppunt een zin voorstelt. Geef het zinnummer de betreffende waarde.

Ga na welke takken er vanuit dit knooppunt vertrekken en welke symbolen bij elk van die takken horen. Maak het blok van dit knooppunt op, waarbij de beginadressen van de volgende blokken eveneens voorlopig nul gemaakt worden. Zet het blok op de stack, indien het beginadressen bevat die nul zijn.

Het beginadres van het nieuwe blok (dat zojuist opgemaakt is) is het beginadres van het vorige blok + de lengte van het vorige blok. Het beginadres van het nieuwe blok moet nu nog in een voorgaand blok ingevuld worden. Neem hiertoe het blok dat het laatst op de stack is gekomen. In dat blok moet het beginadres ingevuld worden op de plaats van het laagste symboolnummer, waarvan het beginadres nog nul is.

Indien dit blok (waar het beginadres zojuist ingevuld is) nog beginadressen bevat die nul zijn, kan het op de stack blijven. Indien het blok na de laatste actie compleet geworden is, moet het van de stack af worden gehaald.

Nu gaan we weer terug naar het zojuist opgemaakte blok. Er zijn 2 mogelijkheden: Dit knooppunt is een einde van de boom (er vertrekken geen takken vanuit dit knooppunt), of niet. Afhankelijk van het feit of het knooppunt wel of geen einde is, zijn de volgende bewerkingen nodig:

***Indien dit knooppunt geen einde van de boom is, zijn de acties:**

Neem uit dit blok het laagste symboolnummer en ga naar het betreffende knooppunt. Herhaal vervolgens de handelingen vanaf punt 4.

***Indien dit knooppunt wel een einde is (er gaan vanuit dit knooppunt geen takken meer verder), zijn de acties:**

Neem het laatst op de stack gekomen blok. Ga naar dit blok en neem het laagste symboolnummer waarvan het beginadres nog niet ingevuld is. Ga naar het knooppunt in de boom dat bij dit laagste symbool hoort. Ga vervolgens naar punt 4. Herhaal 4 net zolang tot je weer terug bent bij "root". Alle paden zijn nu afgewerkt. De boom is nu doorlopen in oplopende lexicografische index.

6.2 Het genereren van de spraakdata.

In afwijking van de Pocketstem [1] is bij de Dominostem niet gekozen voor het inspreken van de zinnen, omdat dit in deze fase van het onderzoek veel te bewerkelijk zou zijn voor het gewenste aantal zinnen (± 1200). Derhalve is de spraakdata gegenereerd met een tekst-naar-spraak systeem op de IPO-VAX (difoonspraak). De ondersteuningssoftware voorziet ook in het (grotendeels) automatisch genereren van de spraakdata voor de zinnenset.

6.2.1 Het genereren van de A/P-files.

Om de spraakdata te genereren (uit tekst) is het noodzakelijk om eerst van elke zin een A/P-file te maken. In een A/P-file staat de spraak in, volgens het bron-filter model [10], geparametriseerde vorm opgeslagen. Het genereren van de A/P-files uit de tekst van de zinnen gebeurt in een commando-file, make.com. Deze file bevat de volgende commando's (de nummering is slechts ten behoeve van de uitleg):

```

1. $ deassign sys$input
2. $ eve dominostem.tex
3. $ run zinnen
4. $ assign grf.prt sys$input
5. $ run grf
6. $ deassign sys$input
7. $ eve/com=eve.ini zinnen.tmp
8. $ del zinnen.tmp;*
9. $ ds/read text=ds.inp
10.$ exit

```

In regel 1 wordt de input aan het toetsenbord toegekend. Regel 2 roept de file `dominostem.tex` aan om daar eventueel wijzigingen in aan te brengen. In regel 3 wordt het programma “zinnen” aangeroepen. Dit programma leest uit de file “`dominostem.tex`” alleen de teksten van de zinnen en plaatst die in een aparte file: “`zinnen.tex`”.

In regel 4 wordt de input voor het programma “grf” [11], een grafeem-foneem omzetter, toegekend aan de file “`grf.prt`”.

In regel 5 wordt het programma “grf” aangeroepen, dat aan elke zin uit de file “`zinnen.tex`” de fonetische representatie toevoegt en beide plaatst in een file “`zinnen.tmp`”. Echter het programma “grf” is nog niet helemaal perfect wat betreft intonatie. Dit is met name het geval bij woorden die niet aan de standaard-regels van de grafeem-foneem omzetting voldoen. In dergelijke situaties is een correctie van de fonetische representatie noodzakelijk (en ook mogelijk). Om te voorkomen dat bij het opnieuw runnen van het programma “grf” de handgecorrigeerde wijzigingen teniet gedaan worden, worden in regel 7 de handgecorrigeerde file en de door “grf” gemaakte file onder elkaar gezet. Met de editor kunnen nu de gewenste delen uit de door “grf” gemaakte file aan de handgecorrigeerde file toegevoegd worden. Het resultaat hiervan staat in de file “`zinnen.fon`”. De structuur daarvan is als volgt:

```
Fonetische representatie zin 1
! Tekst zin 1
Fonetische representatie zin 2
! Tekst zin 2
.....
```

In regel 9 wordt het programma “DS” [13] (diphone speech) aangeroepen. Dit programma maakt uit de fonetische representatie van een zin een A/P-file. Het resultaat hiervan is dat van elke zin een aparte A/P-file is gemaakt. De namen van die A/P-files zijn zin., waarbij het nummer van de zin aangeeft.

6.2.2 Het genereren van de spraakdata.

Het programma “PCFL” [12] heeft als input een verzameling A/P-files. De output van het programma bestaat uit 2 zaken, te weten:

- Een file “prom.pcf”. Deze file bevat het blok spraakdata van de opgegeven zinnen. De structuur van de file “prom.pcf” is overeenkomstig de structuur besproken in [6].
- Een verzameling files “zin*.pcf”. Dit zijn files in de PCF8200 structuur die de spraakdata van één zin (zin*) bevatten.

De laatste verzameling (zin*.pcf) is voor ons doel niet nodig.

Met behulp van het programma “PCFL” wordt elke keer van 64 A/P-files een blok spraakdata gemaakt. Na het genereren van een blok spraakdata “prom.pcf” is het zinvol om elk blok spraakdata een nieuwe naam (met nummer) te geven: prom*.pcf, waarbij * het nummer aangeeft. Na afloop hiervan zijn er (in het geval van 1200 zinnen) 19 files prom0.pcf tot en met prom18.pcf.

Het programma “PCFL” geeft tijdens het maken van elke file “prom.pcf” aan hoe groot (aantal bytes) de betreffende file is. Dit moet voor elke file onthouden worden om naderhand een adrestabel (zie figuur 5.1) te maken. Deze adrestabel bevat de beginadressen van de 19 blokken spraakdata.

De blokken spraakdata worden achter elkaar (zonder tussenruimte) geprogrammeerd. Derhalve kan aan de hand van de lengte van elk blok en het

beginadres van dat blok, het beginadres van het volgende blok berekend worden.

De files "prom*.pcf" bevatten alleen ASCII-waarden. Om deze files in een EPROM te kunnen programmeren moeten ze aan de eisen, die de 8031-assembler stelt, voldoen. Deze omzetting van een file "prom*.pcf" naar een file "prompcf*.dat" (welke een vorm heeft die geschikt is voor de 8031-assembler) gebeurt met behulp van het programma "prom.exe".

7 De evaluatie.

In dit hoofdstuk wordt de evaluatie en de conclusies, die hieruit getrokken kunnen worden, beschreven. Allereerst komen de punten aan de orde waar tijdens de evaluatie op gelet wordt. Vervolgens wordt nader ingegaan op de evaluatie. Hierbij wordt een beeld gegeven van de wijze waarop de evaluatie plaatsvindt. Tenslotte worden bevindingen, opgedaan tijdens de evaluatie, besproken en worden conclusies getrokken uit de bevindingen.

7.1 Aandachtspunten voor de evaluatie.

Tijdens het ontwerp zijn een aantal punten naar voren gekomen, waar de evaluatie uitsluitel over moest geven. Deze punten hebben betrekking op de volgende zaken (de evaluatielijsten, waar deze zaken tot in detail uitgewerkt zijn, zijn weergegeven in bijlage 3):

Indicatie:

- Is de display-tijd (15 seconden) lang genoeg?
- Is de indicatie goed zichtbaar?
- Is het tempo, waarin de volgorde getoond wordt, goed?

Bediening:

- Is het formaat van de toetsen goed?
- Is het prettig om meteen terugmelding te krijgen of een zojuist ingevoerd symbool “geldig” is?
- Is voor een goede bediening een afdekplaat of iets dergelijks noodzakelijk?

Spraak:

- Is het geluid goed verstaanbaar?
- Wordt de spraak geaccepteerd?

Naast deze specifieke vragen is er natuurlijk één grote vraag: Is symbolentaal een goede methode voor de zinselectie in een communicatiehulpmiddel voor spraakgehandicapten?

7.2 De evaluatie.

De evaluatie heeft de volgende opzet:

Door therapeuten is een potentiële gebruiker gezocht, die een aantal vaardigheden moet bezitten:

- hij (zij) moet kennis van de symbolentaal Dominolex hebben,
- hij (zij) moet voldoende motorische vaardigheden hebben om de Dominostem te kunnen bedienen.

Met betrekking tot de tweede vaardigheid kan vermeld worden dat het niet eenvoudig is om van te voren met zekerheid te zeggen of een gebruiker in staat is om het apparaat te bedienen.

Vervolgens wordt de gebruiker in een aantal oefensessies bekend gemaakt met de Dominostem. Wanneer de gebruiker na een tijdje het apparaat goed kan bedienen, kan de gebruiker het apparaat continu ter beschikking krijgen.

Bij de eerste proefpersoon bleek dat bediening zonder afdekplaat (een doorzichtige plaat, die over de Dominostem gelegd wordt, met boven elke toets een gat) niet mogelijk was.

Nadat de afdekplaat gemaakt was, bleek dat ook met de afdekplaat de bediening dermate moeizaam ging, dat de proefpersoon vrij snel gedemotiveerd was.

De oorzaak van de moeizame bediening ligt in de beperkte motorische vaardigheden van de proefpersoon. Een één-toets-bedienbare Dominostem zou misschien wel bruikbaar zijn voor deze gebruiker. Alvorens over te gaan tot het ontwerpen van een één-toets-bedienbare Dominostem (waarvan de technieken al eerder toegepast zijn [8]), wordt eerst nagegaan of het principe van zinselectie met behulp van een symbolentaal bruikbaar is.

De tweede proefpersoon kon al na één oefensessie overweg met de Dominostem (zonder afdekplaat). Deze proefpersoon kan vrij snel zinnen samenstellen met de Dominostem.

7.3 Resultaten en conclusies.

Deze evaluatie is uitgevoerd met twee proefpersonen, waarvan één proefpersoon het apparaat niet kon bedienen. Statistisch gezien zou het beter zijn om de evaluatie uit te voeren met meer proefpersonen. In het kader van dit project was dit echter niet mogelijk.

Op een aantal van de vragen, gesteld in hoofdstuk 7.1, zijn bij beide proefpersonen éénsluitende meningen gebleken.

Tijdens de evaluatie zijn nog enkele punten naar voren gekomen:

- De helderheid van de LED's werd als te gering ervaren.
- De display-tijd (15 seconden) bleek lang genoeg.
- De terugmelding of een symbool "geldig" is, werd als handig ervaren.
- De spraak van de Dominostem klinkt niet altijd natuurlijk genoeg.
- Het geluidsvolume bleek toereikend.

Wat betreft de overige punten kan het volgende opgemerkt worden:

- Het tempo waarin de volgorde getoond wordt, bleek bij de eerste proefpersoon te hoog te zijn. Bij de tweede proefpersoon bleek het tempo goed te zijn.
- Bij de eerste gebruiker bleek bediening niet mogelijk. Bediening zou misschien wel mogelijk zijn als de toetsen groter waren. Bij de tweede proefpersoon was het formaat van de toetsen goed.

Om de helderheid van de LED's te vergroten, zou de stroom door de LED's groter gekozen kunnen worden.

Om de spraak natuurlijker te maken, zou overwogen kunnen worden om alle zinnen in te spreken [1].

Indien het tempo, waarin de volgorde getoond wordt, door elke gebruiker anders gewenst wordt, is het misschien zinvol om dit tempo door de gebruiker instelbaar te maken.

Om uitsluitel te krijgen op de vragen waar nog geen éénsluitend resultaat op is verkregen, wordt nog verder geëvalueerd.

Voorlopig kan geconcludeerd worden dat symbolentaal een bruikbare methode is voor de zinselectie in een communicatiehulpmiddel.

8 Conclusies.

In dit hoofdstuk zullen de conclusies vermeld worden, die getrokken kunnen worden met betrekking tot de in de voorgaande hoofdstukken behandelde onderwerpen. Tevens zullen enige algemene conclusies besproken worden.

Als hoofdconclusie kan vermeld worden dat de zinselectie met behulp van symbolentaal een bruikbare selectiemethode is. Kanttekening hierbij is dat de gebruiker over voldoende motorische vaardigheden moet beschikken. Hij moet meerdere toetsen in kunnen drukken om een zin te selecteren. Verder moet de gebruiker de symbolentaal beheersen.

De Dominostem heeft één keer geweigerd. Toen bleek dat de accuhouders (die van flexibel plastic zijn) geen contact maakten met enkele accu's. Dit is verholpen door elke accu met een draadje vast te binden in de accuhouder. Bij een eventueel prototype kan hiermee rekening gehouden worden bij de keus van de accuhouders.

De software van de Dominostem voldoet goed, behalve op één punt wat uit het oogpunt van privacy van de gebruiker niet optimaal lijkt (de evaluatie moet hierover uitsluitel geven). De Dominostem blijft namelijk altijd onthouden wat de laatst uitgesproken zin is. Dus ook als het apparaat uit en vervolgens weer aangezet wordt, zal bij een druk op de spreektoets de laatst uitgesproken zin herhaald worden. Een therapeut zou bijvoorbeeld 's morgens kunnen nagaan wat de laatst uitgesproken zin van de vorige avond was.

Dit punt zou in een volgend ontwerp op de volgende manier verbeterd kunnen worden: Wanneer de clear-all toets één keer ingedrukt wordt, worden alle ingevoerde symbolen gewist. Wanneer de clear-all toets voor de tweede keer wordt ingedrukt, moet de vorige zin uitgewist worden.

De ondersteuningssoftware van de Dominostem werkt naar behoren.

Een punt wat echter nog niet helemaal perfect werkt, is de grafeem-foneem omzetter. Een aantal woorden dat niet aan de standaard-regels van de omzetter voldoet wordt omgezet in een niet correct fonetische representatie, die met de hand gecorrigeerd moet worden.

Uit de evaluatie kan de conclusie getrokken worden dat de spraak, die uit de Dominostem komt over het algemeen "hol" klinkt. Dit komt waarschijnlijk door het feit dat er te weinig gaatjes in de voorkant van de Dominostem zijn, waardoor de luidsprekers als het ware opgesloten zitten in het kastje. Bij een eventueel prototype kan daar rekening mee gehouden worden, door een andere constructie voor de luidsprekerklankkastjes te kiezen.

Er is bij het apparaat geen afdekplaat bijgeleverd. Dit is bewust gedaan, daar voor elke gebruiker, indien die een afdekplaat nodig heeft, weer individuele aanpassingen aan de plaat gemaakt moeten worden. Het maken van de afdekplaat is dan ook overgelaten aan de technische dienst van het instituut.

Literatuurlijst.

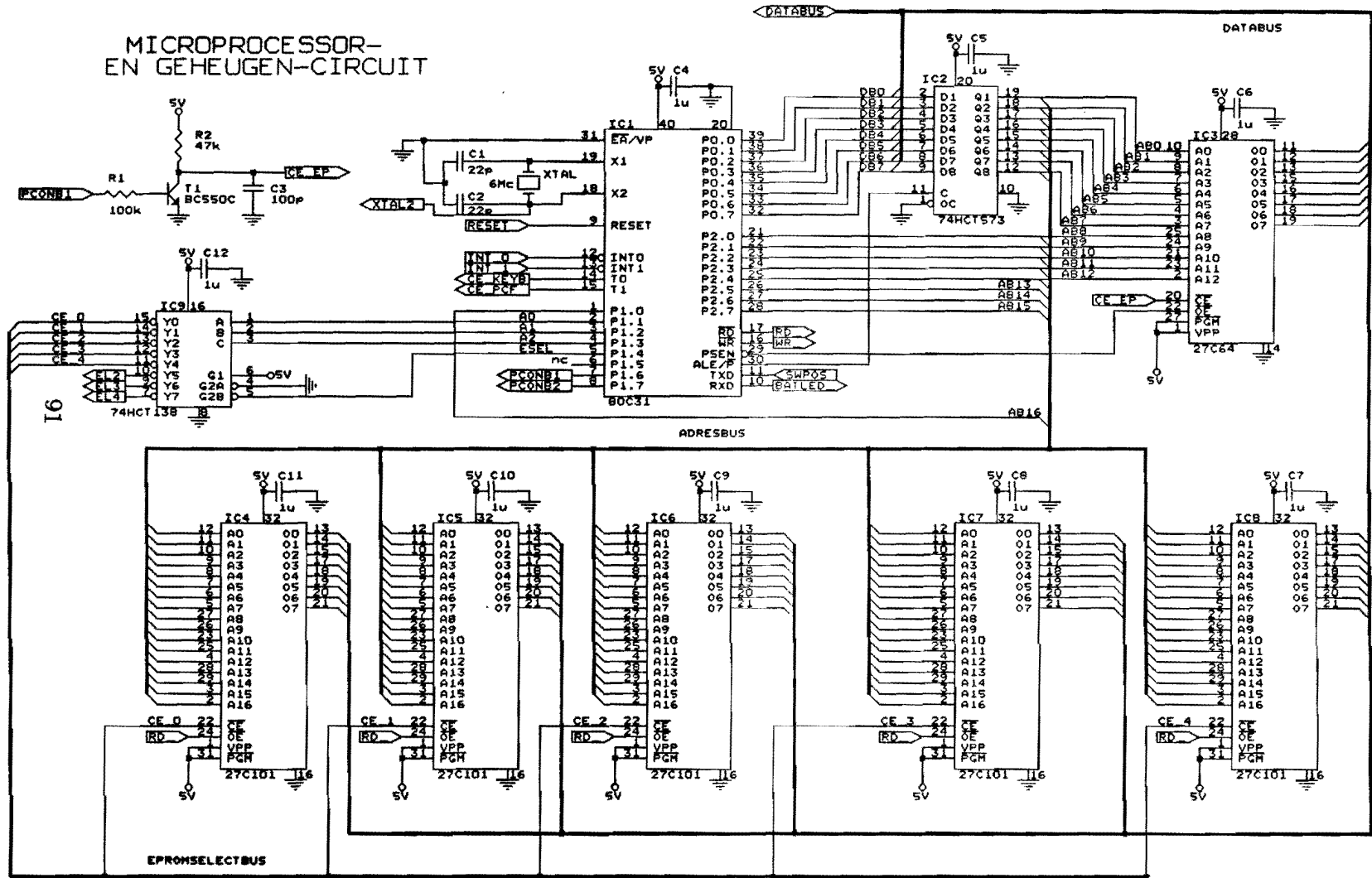
1. *Waterham, R.P.*: The "Pocketstem": an easy-to-use speech communication aid for the vocally handicapped; Proefschrift Technische Universiteit Eindhoven, Eindhoven, 1989.
2. *Bliss, C.K.*: "Semantography", Semantography-Blissymbolics Publication, 2 Vicar St., Coogee Sydney, Australia, 1965.
3. *Mierlo, J.M.A. v. en Doremaele, J.A.C.B. v.*: "Dominolan": communicatologie, Tijdstroom, Lochem, 1986.
4. *Intel Corporation*: Micro-controller Handbook, User's manual, 1985.
5. *Maxim Integrated Products*: Maxim Power Supply circuits, Databoek, 1986.
6. *Have, ten, M.en Zelle, H.W.*: Application report for the PCF8200 Formant Speech Synthesizer, Rapport nr. EDP8807, Philips 1988.
7. *Philips data handboek*: High speed CMOS, PC54/74HC/HCT/HCU logic family, Boek IC06N, Philips, 1985.
8. *Bierens, E.J.J.*: Eén-toets bediening voor de Pocketstem, IPO-rapport 709, Eindhoven, 1989.
9. *Waterham, R.P. en Verhoeven, M.W.C.*: Technische beschrijving van de Pocketstem, IPO-rapport 606, Eindhoven, 1987.
10. *Vogten, L.L.M.*: Syllabus van het college Spraaktechnologie, IPO-rapport 649, Eindhoven, 1988.
11. *Deliege, R.J.H.*: The "Tiepstem": an experimental Dutch keyboard-to-speech system for the speech impaired; Proefschrift Technische Universiteit Eindhoven, Eindhoven, 1989.

12. *Vogten, L.L.M.*: LVS: Speech processing programs on IPO-VAX 11/780, IPO-handleiding 67, Eindhoven, 1985.
13. *Rijnsoever, P.A.*: From text to speech: User manual for Diphone Speech program DS, IPO-handleiding 88, Eindhoven, 1988.
14. *Winthagen, F.L.C.*: Uitbreiding van de Pocketstem en ondersteuningsprogrammatuur, Stagerapport, Eindhoven, 1988.
15. *Deliege, R.J.H.*: Technische beschrijving van de Tiepstem, IPO rapport 548, Eindhoven, 1986.
16. *Philips Designer's Guide*: High speed CMOS, Eindhoven, 1986.

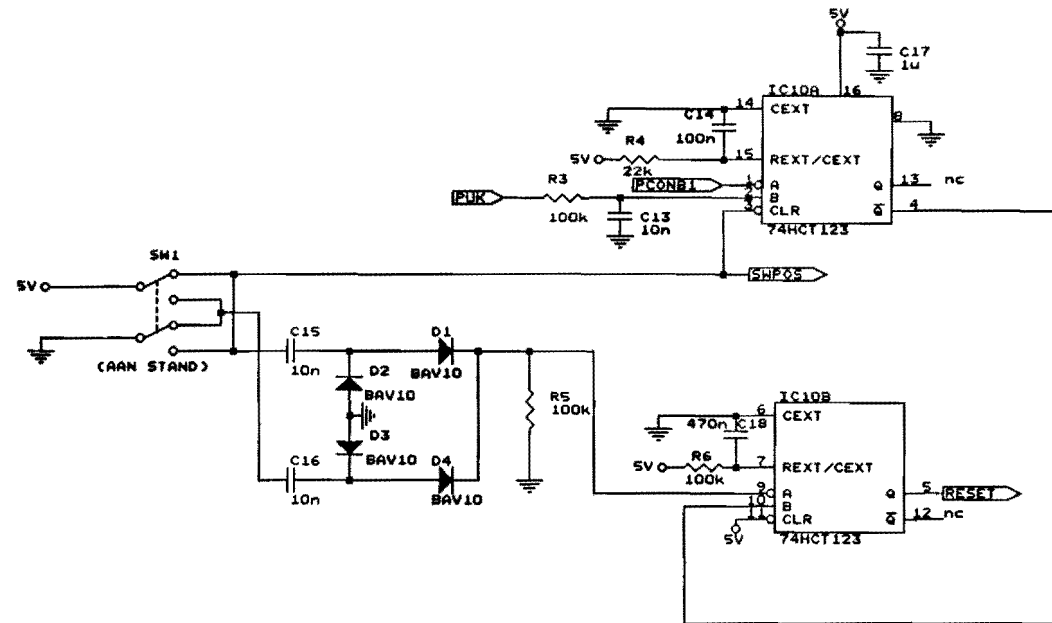
Bijlage 1.

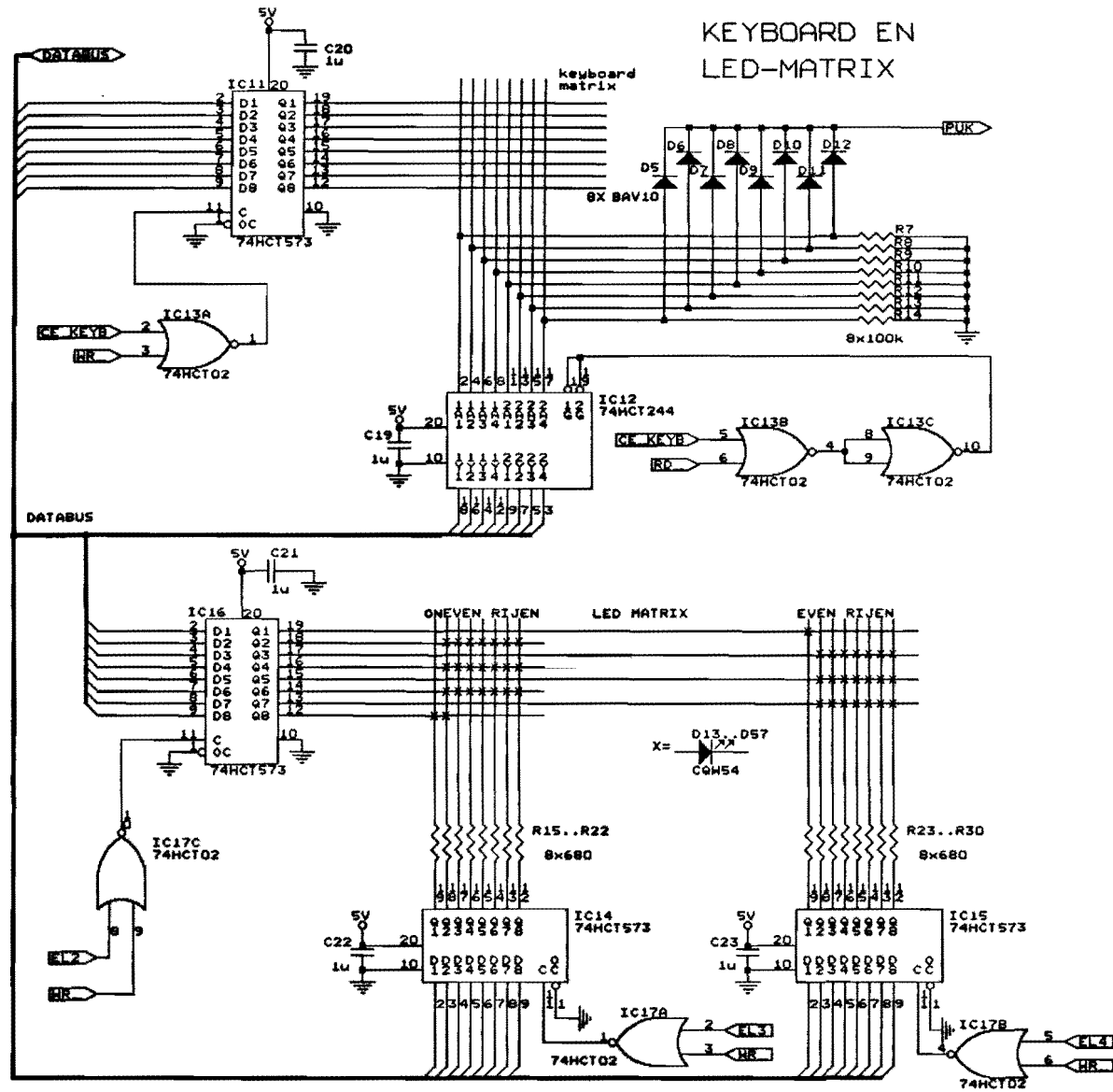
De Hardware.

MICROPROCESSOR- EN GEHEUGEN-CIRCUIT



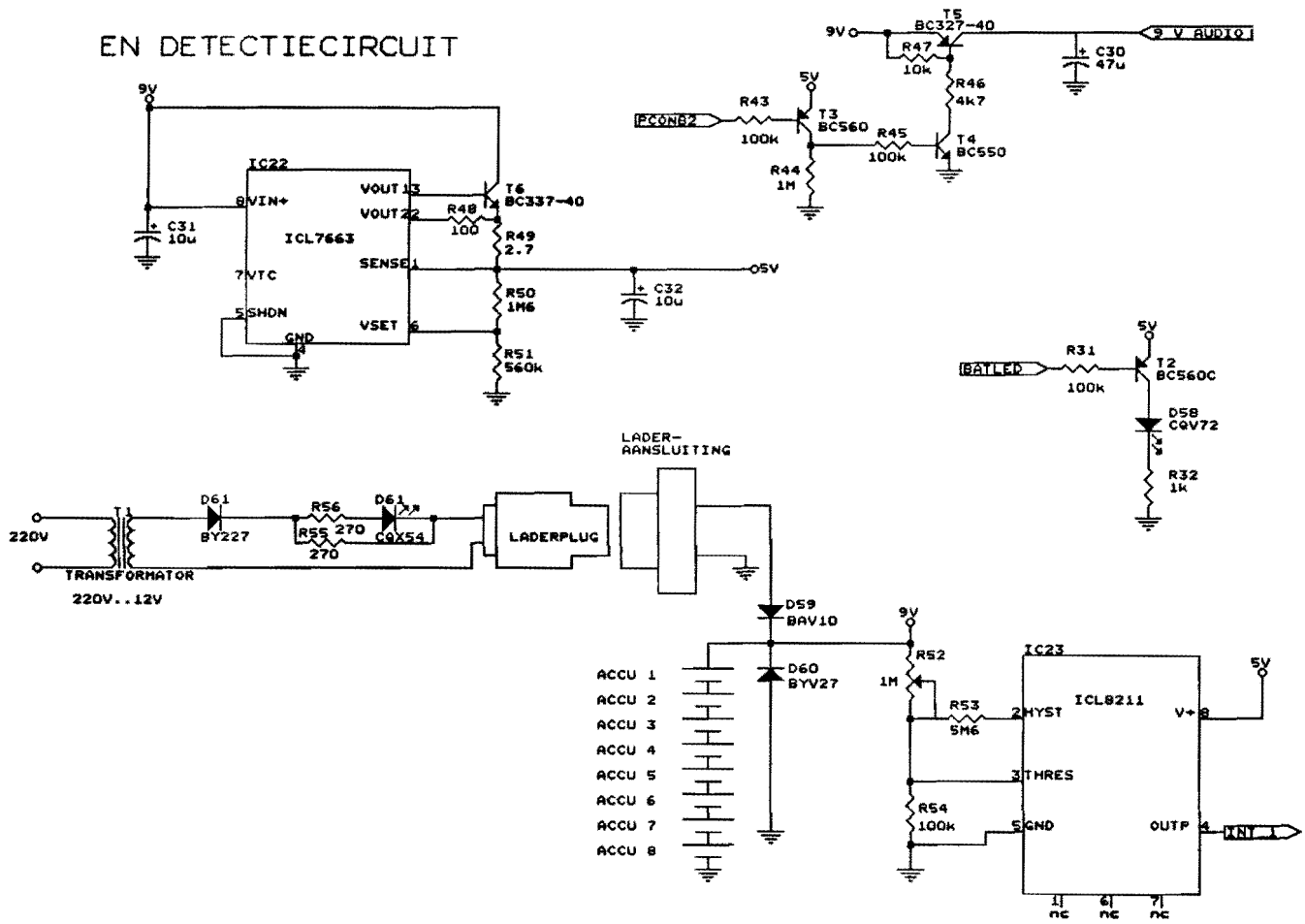
RESET-CIRCUIT



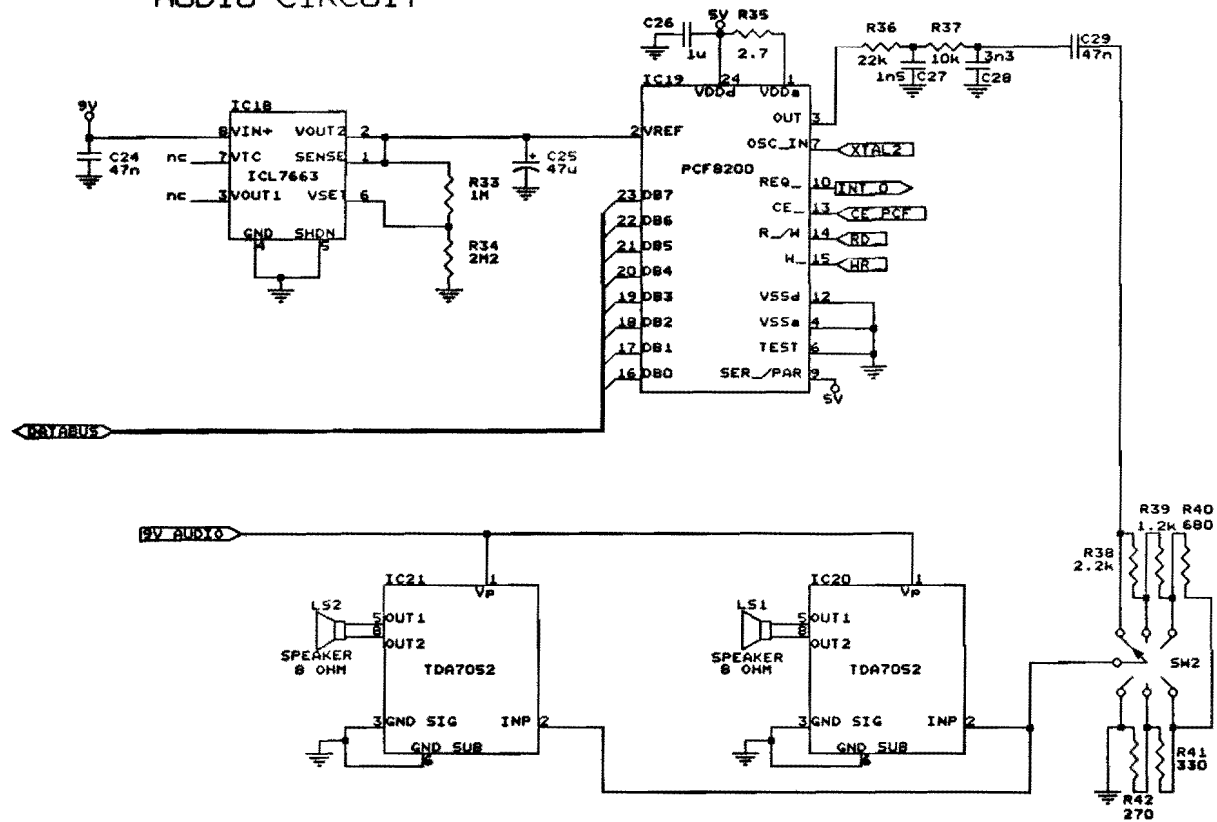


SPANNINGSVERZORGINGS- EN DETECTIECIRCUIT

94



AUDIO-CIRCUIT



Bijlage 2.

Het Eisenblad.

Eisenblad.

1. Funktietoetsen.

- Clear toets: veegt laatst ingetoetste symbool uit en zet de "cursor" één positie terug (eventueel tot het begin).
- Clear all toets: veegt alle ingetoetste symbolen uit en zet de "cursor" terug aan het begin.
- Spreektoets: deze wordt ingedrukt na een aantal symbolen om de betreffende zin uit te spreken of om de nog in het geheugen aanwezige zin te herhalen. N.B: De spreektoets beslaat meerdere toetsen.
- Contrôle toets: laat de volgorde van de ingetoetste symbolen zien. Alle ingedrukte symbolen verlicht op halve sterkte, en de volgorde wordt aangegeven door de LED's in de volgorde feller te laten branden.
- Vaste boodschappen: een aantal toetsen hebben een vaste boodschap, die op elk willekeurig tijdstip opgeroepen kan worden.

2. Indicatie.

- Elke toets bevat een LED, behalve de toetsen "clear", "clear all", "spreek" en "vaste boodschappen".
- Er wordt aangegeven welke toetsen ingedrukt zijn door de betreffende LED's op halve sterkte (kontinu) te laten branden.
- De laatst ingedrukte toets 1 seconde na indrukken fel laten branden; daarna weer op halve sterkte laten branden.
- Na indrukken van de "spreek" toets de LED's uitschakelen (symbolen wel onthouden).
- Bij niet bekende symbolencombinatie LED onder "contrôle" toets laten knipperen.
- Indien 10 à 20 seconden geen toets ingedrukt wordt gaan de LED's uit. De symbolen worden wel onthouden.

- Wanneer na het uitspreken van een zin een van de toetsen “clear”, “clear all” of “contrôle” wordt ingedrukt, worden de LED’s van de reeds ingedrukte toetsen weer geactiveerd.
- Wanneer na het uitspreken van een zin een symbool wordt ingevoerd, wordt de in het geheugen aanwezige zin gewist en wordt het nieuwe symbool opgeslagen waarmee een nieuwe zin kan worden gemaakt.

3. Algemeen.

- Apparaat moet 1200 zinnen uitspreken.
- Een aantal achter elkaar ingetoetste symbolen moet onthouden worden.
- Het maximale aantal symbolen per zin wordt globaal gesteld op 15 à 20.
- Volumeregeling geschiedt met een vijfstandenschuifschakelaar aan de zijkant van het apparaat.
- Transportschakelaar (schakelt het toetsenbord uit) bevindt zich aan de zijkant van het apparaat, evenals de oplaadplugaansluiting.

Bijlage 3.

De Evaluatielijsten.

1.	Technische bruikbaarheid.	JA	NEE
1.1	Heeft het apparaat een storing gehad? Zoja, welke:	0	0
1.2	Mogelijke oorzaak hiervan:.....		
1.3	Gaan de batterijen lang genoeg mee bij normaal dagelijks gebruik?	0	0
2.	Praktische bruikbaarheid:		
2.1	Bediening.	JA	NEE
2.1.1	Kunt U het keyboard makkelijk bedienen?	0	0
	Zo nee, waardoor komt dat?		
	- de afstand van de toetsen t.o.v elkaar?	0	
	- formaat van de toetsen?	0	
	- de aanslag?	0	
	-	0	
2.1.2	Heeft U een overlay voor het keyboard nodig?	0	0
2.1.3	Drukt U wel eens de verkeerde toets in?	0	0
	Zo ja is dat in verband met:		
	- de aanslag?	0	
	- de indicatie van de reeds ingevoerde symbolen?	0	
	- het niet aanwezig zijn van de betreffende combinatie in de Dominostem?	0	

Zoja welke correctie mogelijkheid gebruikt
U dan meestal?

een symbool uitvegen 0
alle symbolen uitvegen 0

2.1.4 Kunt U het apparaat zelf opladen? 0 0

2.1.5 Kunt U de aan/uit schakelaar zelf bedienen? 0 0

2.2 Spraakkwaliteit:

2.2.1 Hoe vindt U de spraakkwaliteit/
verstaanbaarheid? GOED MATIG SLECHT GEPROBEERD NIET

- in rustige ruimte 0 0 0 0

- in licht rumoerige ruimte 0 0 0 0

- in drukke ruimte 0 0 0 0

- buiten 0 0 0 0

2.3 Geluidssterkte: JA NEE

2.3.1 Verandert U de geluidssterkte wel eens? 0 0

2.3.2 Is de geluidssterkte toereikend? 0 0

2.3.3 Kunt U voldoende aandacht trekken
met dit apparaat? 0 0

2.3.4 Hoe reageert men? SNEL LANGZAAM NIET
0 0 0

2.4 Is de gebruiksaanwijzing duidelijk? JA NEE
0 0

3. Persoonlijke bruikbaarheid:

3.1 Hoeveel zinnen werden met het apparaat per dag gemiddeld gesproken?

- max 5 per dag 0
- tussen 5 en 20 per dag 0
- tussen 20 en 40 per dag 0
- meer dan 40 per dag 0

	JA	NEE
3.2 Gebruikt U het apparaat ook buiten Zonhoven?	0	0

Zo ja, waar heeft U het apparaat gebruikt?

3.3 Met wie wordt gecommuniceerd m.b.v de Dominostem?

- verpleging 0
- therapeuten 0
- familie 0
- mede revalidanten 0
- anderen 0
- 0
- 0

3.4 Hoe verliep communicatie via dit apparaat?

- In beginperiode:.....
.....
.....
- Na gewenningsperiode:.....
.....
.....

- Hoe lang duurde het voor U het apparaat kon bedienen?

.....

	JA	NEE
3.5 Zijn Uw communicatie mogelijkheden vergroot door dit apparaat?	0	0
3.6 Kunt U voldoende snel communiceren met dit apparaat?	0	0
4. Boodschappenset:	JA	NEE
4.1 Zijn de vaste boodschappen naar wens?	0	0
4.2 Ontbreken er naar Uw mening Dominolex zinnen?	0	0
4.3 Vindt U het prettig dat de Dominostem al bij het invoeren van een symbool aangeeft of een symbool toegestaan is of niet?	0	0
4.4 Vindt U het vervelend dat de Dominostem, indien geen symbolen ingevoerd zijn, bij een spreekactie ALTIJD de vorige zin herhaalt?	0	0
4.5 Vindt U het voldoende dat U tijdens het uitspreken van een boodschap al 3 volgende spreekopdrachten kunt invoeren?	0	0

Zo nee, hoe veel zou U dan willen?.....

5. Indicatie:	JA	NEE
---------------	----	-----

- 5.1 Zijn de 15 seconden dat de lampjes van de toetsen blijven oplichten lang genoeg? 0 0
- 5.2 Lichten de lampjes fel genoeg op? 0 0
- 5.3 Ervaart U de controle mogelijkheid als handig? 0 0

Zo nee, waarom niet en wat zou er aan veranderd moeten ?

.....

.....

.....

- 5.4 Is de snelheid waarmee de volgorde getoond wordt te volgen? 0 0

6. ALGEMENE VRAGEN:

- 6.1 Hoe vindt U de Dominostem?
- 6.2 Zou U het apparaat op meer plaatsen en in meer situaties willen gebruiken dan U nu de gelegenheid heeft gehad?

- | | | | |
|-----|---|----|-----|
| | | JA | NEE |
| 6.3 | Vindt U het apparaat gebruikersvriendelijk? | 0 | 0 |

Zo nee, waarom niet?.....

.....

.....

.....

- 6.4 Zijn er nog speciale wensen m.b.t de Dominostem?

6.5 Algemene opmerkingen/suggesties m.b.t.

- Technische bruikbaarheid:
.....
.....
- Praktische bruikbaarheid:
.....
.....
- Persoonlijke bruikbaarheid:
.....
.....
- Boodschappenset:
.....
.....
- Indicatie:
.....
.....

In te vullen door therapeut(e).

Persoonsgegevens gebruiker:

1. Medische diagnose (korte omschrijving anamnese toegespitst op de communicatieproblematiek):

2. Motoriek (korte omschrijving restfuncties, m.n. betreffende bovenste extremiteiten en oog-hand coördinatie):

3. Sensoriek (korte omschrijving van zintuigelijke restfuncties):

4. Mobiliteit (manier van verplaatsen, lichaamshouding bij communicatie):

5. Taal- en spraakvermogen (taalbegrip en taalproductie, waaronder spraak):

6. Communicatievermogen (wijze van communiceren voor het hulpmiddel werd gebruikt):

ALGEMENE VRAGEN:

1. Bij hoeveel patienten heeft U de Dominostem geïntroduceerd?

2. Hoeveel daarvan hebben de Dominostem zinvol gebruikt?

	JA	NEE
3. Heeft het apparaat een storing gehad? Zoja, welke en wat was volgens U de oorzaak?	0	0

4. Had de gebruiker een overlay nodig?	0	0
--	---	---

5. Is het toetsenbord gemakkelijk (evt. met overlay) te bedienen?	0	0
---	---	---

6. Hoeveel tijd kost de introductie naar de patient?

7. Zijn er boodschappen niet in de Dominostem die wel gewenst zijn?	0	0
---	---	---

Zoja, welke?

8. Hoe verloopt de communicatie via dit apparaat?

- hoeveel hangt hierbij af van de wil tot communiceren?

- hoeveel hangt hierbij af van de omgeving?

9. Hoe reageert de gebruiker op dit hulpmiddel?

10. Hoe reageert de omgeving?

11. Hoe vindt U als therapeut(e) dit hulpmiddel?

12. Hoe vindt U de verstaanbaarheid, spraakwaliteit
en het volume?

13. Is de gebruiksaanwijzing duidelijk?

14. Is de snelheid van communiceren voor de gebruiker en zijn omgeving voldoende?

15. Hoe is Uw mening over de nu ontstane mogelijkheid van spraakuitvoer?

16. Vindt U het apparaat gebruikersvriendelijk?

	JA	NEE
-wat betreft bediening	0	0
-wat betreft indicatie	0	0
-in het algemeen	0	0

17. Op welke punten zou U het apparaat graag verbeterd zien?

18. Ziet U door de mogelijkheid van spraakuitvoer veranderingen in het gedrag van de spreker?

Bijlage 4.

Layout van het toetsenbord.

