

21 MSX2

Najaar 2001

Info Blaad

Inhoud:

Voorwoord	2
Artificiële intelligentie	3
LPE-EPR-v1	10
LCD Beeldscherm op MSX	11
MSX wint op ASM 2001	19
MSX Radio	20
Cursus PowerBasic 8	23
PLR 5.02	27
UZIX	29
The Best of Hamaraja Night 2	31
Beursverslag Bussum 2000	33



Hamaraja Night nu ook voor MSX2!



Beluister nu MSX muziek live vanaf internet

Verslaat de MSX straks Kasparov?
Artificiële Intelligentie op de MSX

Colofon:

Het MSX-Info Blad is een uitgave van V.C.L. te Landgraaf. Het verschijnt 4 keer per jaar. Een jaarabonnement kost Fl.15,— (€ 6,81). Losse nummers kosten Fl.4,50 (€ 2,04). Op beurzen geldt de speciale prijs van Fl.3,50 (€ 1,59) per nummer.

Uitgever:

M.T. Stoker
Rotterdamstraat 73
6415 AV Heerlen
Tel: (045) 572 95 09
Mobiel: (06) 44 15 85 45
Giro: 24.95.572
SNS: 92.88.51.222
Email: msxinfo@msxinfo.com

Redactie:

Gerrit van den Berg
Fokke Post
Jan-Marten van der Reest

Medewerkers aan dit nummer:

Mark Jelsma, Dennis Koller,
Sjoerd Mastijn, Jack Tresoor,
Tristan Zondag

Lay Out:

Fokke Post
Waaiershoek 36
8321 BH Urk

Drukwerk:

Copy Service Hattem
Hoopjesweg 1
8051 DB Hattem
Tel: (038) 444 85 77

Copyright © 2001

V.C.L. te Landgraaf

Niets uit deze uitgave mag worden vermenigvuldigd zonder schriftelijke toestemming van de uitgever.

Beste lezer

Helaas! We hebben ons best gedaan om dit nummer voor de beurs in Bussum bij u in de brievenbus te krijgen. Vanwege het feit dat we zo snel na de vakantie weer met een nieuw nummer uit moesten komen, konden we dit niet op tijd klaren. Wij vragen uw begrip hiervoor. We hopen dat dit u niet heeft verhinderd om de beurs te bezoeken.

Natuurlijk hebben we ter compensatie wel een compleet beursverslag opgenomen om degenen, die deze beurs niet bezocht hebben, toch een beeld te geven hoe het er op de beurs aan toe ging. Verder kunt u in dit nummer onder andere het slot vinden van de miniserie 'hoe sluit ik een LCD scherm op de MSX aan'. Tevens is een korte beschrijving opgenomen van de nieuwste *PLR*.

We zijn trots op het feit dat wij het spel *The best of Hamaraja night*, dat destijds door het Japanse *Pastel Hope* alleen voor de turboR is uitgebracht, hebben kunnen converteren naar de MSX2. Daar wij de officiële copyrights hebben om dit spel in Europa te verkopen, hebben wij besloten om dit spel in ons assortiment onder te brengen. Dit betekent dat het geleverd wordt met een fraai label in een mooie verpakking. Laat onze inzet voor nieuwe MSX software alstublieft niet onbeloond en schaf dit spel aan voor slechts vijftien gulden!

We willen u nogmaals vragen om ons te blijven voorzien van recensiemateriaal of artikelen over actuele of innovatieve MSX ontwikkelingen. Het valt beslist niet mee om tegenwoordig een MSX blad vier keer per jaar te laten uitkomen! Bent u ergens mee bezig op MSX gebied, aarzel dan niet en breng ons op de hoogte, zodat wij ons brede lezerspubliek kunnen informeren over uw nieuwe product. Draagt u ook uw steentje bij aan de promotie van de MSX computer? Alvast bedankt!

Rinus Stoker



Artificiële intelligentie op de MSX

Artificiële intelligentie of kunstmatige intelligentie fascineert mensen al vanaf het ontstaan van computers. *Mark Jelsma* legt u in de komende edities van dit blad uit wat het precies inhoudt en hoe het op de MSX toegepast kan worden.

Hoewel artificiële intelligentie (hierna afgekort tot AI) na al deze jaren nog steeds in de kinderschoenen staat worden er voordurend discussies gehouden over vragen als: "Kan een computer echt denken?" en "Wat is de exacte definitie van intelligentie?".

Het enige wat een computer kan is optellen. Dit is de basis van elke computer. Dus met een nuchtere blik kunt u met gemak de eerste vraag beantwoorden. Er kan weliswaar een computerprogramma geschreven worden, dat zeker voldoet aan een vorm van intelligentie maar waarbij de computer in feite nog steeds alleen maar kan optellen.

Redeneren

Een mens redeneert veel. Dat is dan ook een belangrijk onderwerp voor programmeurs, die eventueel een expert systeem willen maken. Wanneer u een redenerend programma wilt maken moet het wel zogenaamde syllogismen kunnen oplossen zoals:

"Mark is a programmer";
"A programmer is cool";
"Is Mark cool";

Zo'n voorbeeld laat zien dat er wel degelijk kunstmatige intelligentie aan te pas komt. Niet expliciet ingevoerde data kan door middel van redeneren worden herkend. Dit is eigenlijk vrij logisch, maar er moet wel kennis aanwezig zijn van taalkundig ontleden. Hoe moeilijk dit soms ook is en hoe lang en ingewikkeld de zinnen zijn, er moet altijd een verband in zitten.

Het principe van dit programma is een conclusie trekken uit expliciet of niet expliciet ingevoerde data. De conclusie is nog steeds beperkt tot ja, nee of onbekend. Ook moet er een 2D array aanwezig zijn dat de data bewaart. Vooraf moet overigens gezegd worden, dat vrijwel alle voorbeelden in het Engels zijn, omdat de Engelse taal vele malen simpeler is om te ontleden en te gebruiken voor artificieel intelligente doeleinden.

Het array moet gezien worden als een tabel. De bovenste rij van het array bevat de onderwerpen en onder elk onderwerp staat een aantal elementen, dat betrekking heeft op het bovenstaande onderwerp. Wanneer dit gedaan is, gaan we de eerste regel invoeren: "Mark is a programmer". Het onderwerp "Mark" wordt



er direct uitgehaald en op de eerste de beste vrije positie voor onderwerpen in het array geplaatst. Daarna wordt "is" herkend en "a" wordt er gewoon uitgehaald, omdat het niet van toepassing is voor een programma. Als laatste wordt "programmer" naar de eerste de beste vrije positie verplaatst onder het onderwerp "Mark".

Als we nu de computer vragen "Is mark a programmer" wordt onze veronderstelling bevestigd met een zelfverzekerd "Yes". Maar dit is nog niet echt redeneren. Dit is gewoon een simpel programma, dat expliciet ingevoerde data herkent. Om het nog echter te maken gaan we nog de zin invoeren: "A programmer is cool". Wederom wordt de vorige ontleding toegepast op de nieuwe zin. Maar dit keer wordt "programmer" niet een element van "Mark" maar wordt het een onderwerp met als element "cool".

Als we de computer nu vragen "Is a programmer cool" wordt dit wederom bevestigd. Maar om het echt moeilijk te maken vragen we "Is Mark cool". Er is onder het onderwerp "Mark" nergens een element te vinden dat het woord "cool" bevat. Maar "Programmer" is wel gevonden. Dus

wordt er gekeken of het onderwerp "Programmer" bestaat. Is dit niet het geval dan zoeken we het eventueel volgende onderwerp dat ook als element onder "Mark" staat. Zijn er geen elementen meer, dan zal de computer de vraag niet kunnen bevestigen.

Maar in ons geval bestaat het onderwerp "Programmer" wel degelijk. Nu hoeven we alleen nog maar te zoeken naar het element "cool" en dat staat er ook, dus kan de computer de vraag bevestigen. Om nog verder te gaan in het redeneren geef ik een voorbeeld van een Multi-level search. Het eerste level van het geheugen is de expliciet ingevoerde data (Figuur 1), bijvoorbeeld "Mark is a programmer". Bij "Mark is cool" is duidelijk, dat de computer een onderwerp verder moet zoeken, het zogenaamde tweede level. Maar wanneer de computer nog verder moet zoeken zal toch een veel ingewikkelder routine geschreven moeten worden.

Omdat het op een PC met 800 MHz al traag gaat zal op de MSX toch nog drastischer maatregelen getroffen moeten worden om het een beetje snel te krijgen. Maar daar zijn de nodige foefjes wel voor te bedenken. Als

Mark	Programmer	MSX	8 bit computer	Old computer
Programmer	Cool	8 bit computer	Old computer	Super
Sweet		Very cool		

Figuur 1 — Geheugen diagram



eerste gaan we de gegevens allemaal netjes sorteren. De elementen onder de onderwerpen sorteren we eerst. Zorg er wel voor dat de lege elementen onderaan komen vanwege de zoek routine (Figuur 1).

Dan gaan we ook netjes de onderwerpen sorteren, iets dat een klein beetje pittiger is omdat de bijbehorende elementen ook mee moeten verhuizen naar hun nieuwe gesorteerde plaats. Om je een beetje op weg te helpen zijn er in figuur 2 tot en met 5, vier verschillende sorteermethoden te zien, die allemaal gebruikt kunnen worden. Elke sorteermethode heeft zo zijn eigen voordelen maar een algoritme om te bepalen welke van de vier het best kan worden gebruikt op een bepaald moment heb ik nog niet gemaakt, hoewel het wel mogelijk is om alle vier

in het programma te gebruiken. Een algoritme om dat te bepalen is ook AI. Maar we dwalen af dus laat ik verder gaan met de sorteeralgoritmen.

De Bubble sorteer methode is meestal ontzettend traag, maar is het snelste wanneer het om relatief goed gesorteerde lijsten gaat. Ook is de Bubble methode erg begrijpelijk. Als tweede hebben we de Insertion methode, die met slechte lijsten net iets sneller is dan Bubble maar met redelijke lijsten is hij net iets langzamer. Deze is net iets moeilijker dan Bubble.

Als derde en vierde hebben we twee vrij moeilijke en vreemde methoden om te sorteren, de Shell en de Shell-Metzner sorteermethoden. Bij slechte en zeer slechte lijsten zijn ze beide

```

10 for i=1 to le-1
20 v=0
30 for k=1 to le-i
40 if ar(k+1)<ar(k) then swap ar
(1+1),ar(k) :v=1
50 next k
60 if v=0 then goto 80
70 next i
80 return

```

Figuur 2 — Bubble methode

```

10 for i=1 to le
20 z=ar(i)
30 for k=i to 2 step -1
40 if ar(k-1)<=z then goto 70
50 ar(k)=ar(k-1)
60 next k
70 ar(k)=z
80 next i
90 return

```

Figuur 3 — Insertion methode

```

10 h1=le
20 if h1<=1 then goto 100
30 h1=int(h1/2):h2=a-h1
40 v=0
50 for i=1 to h2
60 h3=i+h1
70 if ar(i)>ar(h3) then swap ar(i),
ar(h3):v=1
80 next i
90 if v=1 then goto 40 else goto 20
100 return

```

Figuur 4 — Shell methode

```

10 h1=int(le/2)
20 if h1<=0 then return
30 h2=le-h1: for i=1 to h2
40 for k=i to 1 step -h1
50 h4=k-h1
60 if ar(k)<ar(h4) then goto 90
70 swap ar(k),ar(h4)
80 next k
90 next i
100 h1=int(h1/2)
110 goto 20

```

Figuur 5 — Shell-Metzner methode

aanzienlijk sneller, vooral de laatst genoemde methode steekt er met kop en schouders bovenuit. Maar bij relatief goede lijsten zijn ze ontzettend traag. Om deze vier methoden allemaal in ons programma te integreren hebben we een beslissingsalgoritme nodig. Dat zal zeer zeker een uitdaging zijn om te maken. Noteer wel dat bij de volgende stukjes code variabele [LE] de lengte van array [AR] is, en dat array [AR] het array is dat gesorteerd moet worden.

Nu hebben we eindelijk het hele geheugen netjes gesorteerd en kunnen we beginnen met de multi-level search routine. Hieronder de geheugen lay-out van figuur 6.

U kunt goed zien dat het hier mogelijk is om een multi-level search routine te maken. Wanneer we de computer vragen of Mark ook super cool is, zal hij zelfverzekerd mijn vermoeden bevestigen. Maar hoe doen we dat nu snel genoeg op een MSX2 met 3,54 MHz terwijl het op mijn AMD 800 MHz Windows computer al traag gaat. Op de PC maakte ik ook weinig gebruik van zeer ingewikkelde algoritmen. Hoewel ik de moeilijke algoritmen nu wel gebruik, kan ik zien dat het vele malen sneller gaat, zo ook met een MSX. Eerst gaan we

echter bezig met de basis van multi-level search.

Als ik het programma vraag of Mark ook super cool is, zal er geantwoord worden met ja. Maar hoe gaat dat? Nou, als eerste wordt het onderwerp "Mark" opgezocht, is dat niet aanwezig zal er geantwoord worden met "Unknown", omdat er simpelweg geen data aanwezig is over het onderwerp "Mark". Maar het onderwerp is wel aanwezig. Er is echter geen element dat "Super cool" bevat, dus dan wordt er overgegaan naar de multi-level search routine. De routine zoekt een gelijknamig onderwerp met het eerste het beste element dat hij tegenkomt. En dat doet hij tot het einde bereikt is. En omdat "Composer" nou eenmaal het eerste element is wordt er eerst met een non-level search naar "Super cool" gezocht. Wanneer dat niet gevonden wordt zal het programma weer overslaan naar de multi-level search routine en wel te beginnen bij "Musician". Dan wordt het gelijknamige onderwerp gevonden en wordt de non-level search routine gestart onder het onderwerp. In dit geval wordt het gezochte "Super cool" gevonden en kan de vraag weer bevestigd worden. Maar het kan ook nog iets moeilijker, kijk maar naar fi-

Mark	Composer	Programmer	Musician	Smart one
Composer	Musician	Smart one	Super cool	Cool
Programmer				

Figuur 6 — Geheugen diagram



Mark	Composer	Programmer	Musician	Smart one
Composer	Musician	Smart one	Programmer	Cool
Programmer				

Figuur 7 — Geheugen diagram

guur 7 en gebruik dat als geheugen diagram.

Figuur 6 en 7 zijn bijna identiek aan elkaar, alleen is een muzikant deze keer niet super cool maar een programmeur. Als ik deze keer vraag of Mark heel misschien ook cool is zal de computer de vraag wederom bevestigen maar wel via een hele andere weg. Via deze afwisseling van routines zal de weg van figuur 8 worden gevolgd om het doel te bereiken. NLS en MLS staan respectievelijk voor non en multi-level search, NLS zoekt naar een element zoekt en MLS zoekt naar een onderwerp zoekt dat misschien het gezochte element bevat of een element dat gelijknamig is aan een onderwerp, zodat de MLS routine verder kan zoeken. NLS kan alleen maar naar het gevraagde element zoeken en MLS kan naar allerlei verschillende onderwerpen zoeken. Een onderwerp is trou-

wens een subject en een element is een object.

Hier bij figuur 8 wordt perfect weergegeven wat er allemaal moet gebeuren bij een simpele vraag. Op deze manier kunt u het verschil zien tussen het biologische menselijke brein en een computer programma dat fungeert als zogenaamde artificiële intelligentie. Onthoud wel dat er een soort [RETURN] functie in moet zitten die terugkeert naar een elementlijst wanneer er niet meer gezocht kan worden. Als bij het onderwerp "Mark" begonnen wordt met zoeken begint multi-level bij "Composer" en er wordt doorgezocht tot er niks meer te zoeken valt, dan moet het programma MLS verdergaan bij het element "Programmer" (niet verwarren met het gelijknamige onderwerp) en netjes verder zoeken. Wanneer dat gebeurt vindt u het vorige doel sneller maar dat weet het computerprogramma niet.

```
NLS[COOL]/Subject[MARK] = Not found > Boot MLS
MLS[COMPOSER]/Subject [MARK] = Found > Boot NLS
NLS[COOL]/Subject [COMPOSER] = Not found > Boot MLS
MLS[MUSICIAN]/Subject[COMPOSER] = Found > Boot NLS
NLS[COOL]/Subject[MUSICIAN] = Not found > Boot MLS
MLS[PROGRAMMER]/Subject[MUSICIAN] = Found > Boot NLS
NLS[COOL]/Subject[PROGRAMMER] = Not found > Boot MLS
MLS[SMART ONE]/Subject[PROGRAMMER] = Found > Boot NLS
NLS[COOL]/Subject[SMART ONE] = Found > Data request confirmed!
```

Figuur 8 — Zoek monitor



Zoeken

Bij het vorige voorbeeld kon goed gezien worden hoe de computer zoekt en dat er niet bepaald lang in één lijst wordt gezocht. Wanneer iemand er tijdens een langdurige tijd data instopt en opvraagt, worden de elementlijsten langer en groter. Dus het non-level zoeken duurt dan vele malen langer en dat willen we natuurlijk niet. Dit is wanneer verschillende zoekroutines aan bod komen.

Als eerste de routine voor lineair zoeken. Deze is toch wel het toonbeeld van simpelheid. De lijst waarin gezocht moet worden hoeft niet eens gesorteerd te zijn. Hier komt het voorbeeld (Figuur 9) waarbij het array [AR] de lijst is waarin gezocht moet worden, [ZO] is de gezochte string of variabele en [LE] is wederom de lengte van de lijst of array. Variabele [ID] is overigens de index of positie van het gezochte item in de lijst. Wanneer integer [ID] =-1 dan is het gezochte niet gevonden.

Deze methode is simpel en makkelijk maar wanneer er met een trage computer of een lange lijst wordt gewerkt is men genoodzaakt om een ietwat robuustere aanpak te nemen, binair zoeken (Figuur 10). Wederom gebruiken we dezelfde variabelen als bij het voorbeeld van lineair zoeken.

Dit stukje code lijkt zeer simpel,

```
10 id=-1:for i=1 to le
20 if zo$ar$(i) then id=i: return
30 next i
40 return
```

Figuur 9 — Lineair zoeken

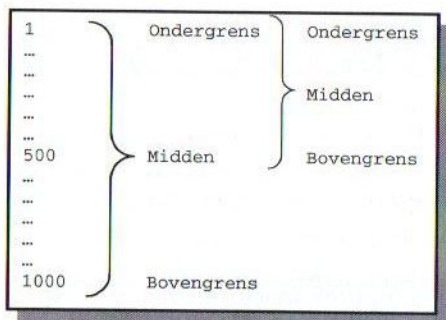
maar bedenk het maar eens. Binair zoeken is niet erg moeilijk maar het is een stuk pittiger dan lineair zoeken. Deze routine splitst de lijst in twee helften (Figuur 11) met een ondergrens (lo), een bovengrens (up) en het midden (m). Wanneer het gezochte kleiner is dan het item op index m zal het bovenste deel aan de kant worden gelegd. Ook wordt er dan een nieuwe boven- en ondergrens gemaakt. Verwar de onderkant overigens niet met de bovenkant van de lijst, dat is iets wat veel mensen doen. Dit gaat zo door totdat de hele lijst is doorzocht. Iedere keer wanneer het doel niet is gevonden wordt de lijst met factor 2 verkleind. Op deze supersnelle manier hoeft een lijst met 1.000 items maximaal slechts tien keer worden doorzocht. Met lineair zoeken moet het maximum identiek zijn aan de lengte van de lijst. Overigens moet een lijst die doorzocht wordt met een binaire zoekroutine beslist helemaal netjes gesorteerd zijn, want anders verschijnen de vreemdste resultaten.

Zoeken is niet zomaar zoeken in de artificiële intelligentie. Zoeken naar bepaalde items in lijsten hebben we nu net gehad, maar nu komt een echte vorm van zoeken die alleen maar in de artificiële intelligentie te

```
10 id=-1 lo=0: up=le+1
20 m=int((lo+up)/2)
30 if m=lo then m=0: goto 70
40 if zo$ar$(m) then goto 70
50 if zo$ar$(m) then up=m else lo=m
60 goto 20
70 id=m
80 return
```

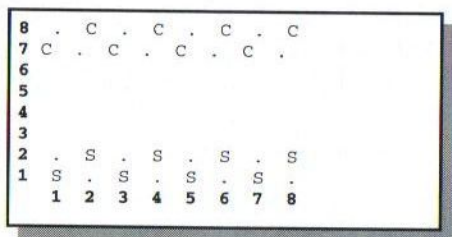
Figuur 10 — Binair zoeken





Figuur 11 — Binair zoeken

vinden is. Deze vorm wordt gebuikt om bepaalde beslissingen te nemen. Het computerprogramma dat met schaken van de Rus *Kasparov* won heeft een zeer goede en erg intelligente routine, die naar de verschillende beslissingen zoekt. Die routines moeten gebruik maken van lastige trucjes die later aan bod komen. Eerst stellen we ons een simpel damspelletje voor, zonder dammen (twee

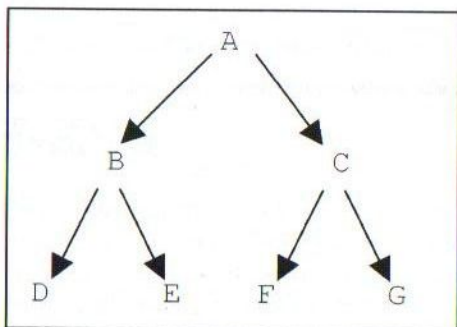


Figuur 12 — Simpel dam spel

stukken op elkaar) en slechts vier stukken in de breedte met vier erboven (Figuur 12). [C] is de computer en [S] de speler.

Als we de computer laten starten weet iedereen, dat er exact zeven verschillende zetten mogelijk zijn. Nadat elke speler een zet heeft gedaan zijn er al 49 uiteenlopende kanalen aanwezig. Nog voordat de te-

genspeler z'n tweede zet heeft gedaan zal de computer 49 x 8 zetten in overweging moeten nemen. Uiteindelijk zijn er voor de computer zo ongelofelijk veel mogelijkheden. Als ik 10^{45} mogelijkheden zeg, zal niemand je geloven maar dat is toch de keiharde werkelijkheid. Als je dan een computer hebt die 5 miljoen van deze mogelijkheden per seconde aan kan, zal hij er toch nog ongeveer 6×10^{35} jaar voor nodig hebben om alles te overwegen. Dit is natuurlijk voor een mens absurd lang. Het wordt daarom tijd om er wat aan te doen.

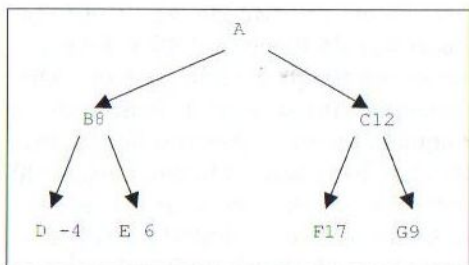


Figuur 13 — Zoekboom of boomstructuur

Zo'n ongelofelijke structuur van mogelijkheden noemen we een zoekboom of boomstructuur (Figuur 13). Dit komt doordat het aantal mogelijkheden per tak uitbreidt, net zoals de stamboom van uw familie.

De keuzes D tot en met G zijn louter en alleen maar implicaties van de direct bovenliggende zet. Op die manier kan een tak geweldig groeien en zeer onoverzichtelijk worden. Dus gaan we een juweel van een routine even onder de loep leggen. Met het





Figuur 14

zogenaamde 'Mini-Maxen' snoeien we de boom eigenlijk, maar de routine moet wel in staat zijn om een numerieke waarde aan elke spelpositie toe te kennen (Figuur 14).

De waarde kan met behulp van een

aantal factoren vast worden gesteld.

- ◆ Is het stuk een bedreiging voor de tegenspeler?
- ◆ Wordt door de zet een stuk geslagen?
- ◆ Wordt het potentiële vak bedreigt?

De computer moet een keuze maken uit een variëteit van opties waarbij het de maximale score zal of kan behalen en de mogelijkheid dat de menselijke opponent een minimale score behaalt. Daar komt dus ook direct het woord Mini-Maxen vandaan. Tot zover dit eerste deel over AI.

Mark Jelsma

LPE-EPR-v1

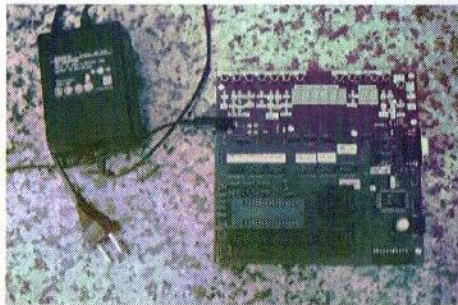
Leonardo Padial blijft aardig actief bezig met het ontwerpen van nieuwe MSX hardware. Ditmaal verrast hij ons met zijn nieuwste ontwikkeling: een EPROM/flash brander, genaamd *SPACE1* of *LPE-EPR*. Het is een apparaat om geheugen evenals andere chips, zoals PLDs en GALs, te branden of te testen. Het kan geprogrammeerd worden met behulp van een RS-232 interface en is daardoor ook op een MSX computer te gebruiken.

De specificaties zijn:

- Externe voeding.
- Werkt met 1.8V, 3.3V en 5V geheugen.
- Display voor checksums, adressen, sectoren en data.
- Keyboard aansluiting.
- RS-232 poort.

- 8 en 16 bit data.
- Brandt geheugen tot en met 128MByte/1GByte.
- 512 kB Flash geheugen voor resident programmeren of het testen van algoritmes.

De prijs zal ongeveer 365 Euro gaan bedragen (Fl. 800,-). Bij voldoende belangstelling zijn wij bereid enkele exemplaren naar Nederland te halen.



LCD Beeldscherm op MSX

- Slot -

Na het eerste deel van het vorige nummer sluiten we de beschrijving van de ombouw van een LCD scherm op de MSX af met het slot aangevuld met specifieke informatie over een kleuren LCD scherm.

In het vorige nummer heb ik uitgelegd hoe een LCD scherm op een vrij eenvoudige manier kan worden aangesloten op de MSX. Hierbij is uitgegaan van een testopstelling, die we in dit laatste deel van de uitleg gaan perfectioneren.

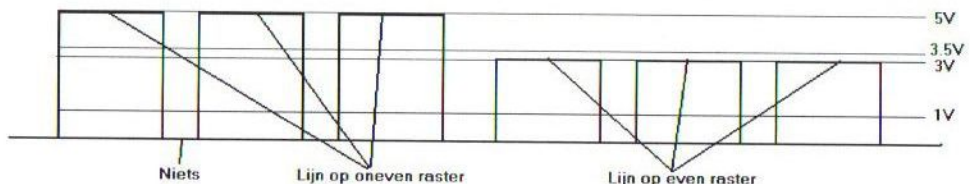
Om het aantal analoge externe componenten te beperken, is het handigste om het meeste werk door een PIC processor te laten doen. In een PIC processor zijn eenvoudig voorwaarden te programmeren, bijvoorbeeld wanneer er nou wel of geen extra sync gegenereerd moet worden. De gebruikte PIC processor is een standaard *PIC16F84*, die elke onderdelenboer wel heeft liggen.

In rust staat de PIC in een loop van twee instructies te controleren of de CSync laag wordt. Een probleem hierbij is dat de tijd tussen de neergaande flank en het verlaten van de loop niet constant is. De eerste instructie is de controle zelf en kost

$0,4\mu\text{s}$ en de tweede is een GOTO en kost $0,8\mu\text{s}$. De neergaande flank kan op elk moment komen, de PIC verlaat de loop dan ergens tussen de $0,4\mu\text{s}$ en $1,2\mu\text{s}$ na het verschijnen van de neergaande flank. Voor de signalen HSync en VSync is een spelting van $0,8\mu\text{s}$ geen probleem, maar voor het DTMG signaal is dit teveel. Bij een dot-clock van 16.27 MHz komt $0,8\mu\text{s}$ overeen met 13 pixels.

Om er voor te zorgen dat het display op het juiste moment begint te schrijven wordt het BLEO signaal van de VDP gebruikt. Het BLEO signaal geeft aan of er niets gebeurt, of er een lijn van het even raster afgedrukt wordt, of een lijn van het oneven raster, zie figuur 1.

Deze uitgang kan behalve 0 en 5 Volt een tussenwaarde hebben van ongeveer 3 Volt. Dit is dus net niet digitaal. Een comparator (LM339) maakt hier twee digitale signalen van. De ene geeft aan dat BLEO >



Figuur 1



1V en de andere geeft aan dat BLEO > 3.5V. Het staat vast dat als BLEO > 1V er een lijn geschreven moet worden. Dit signaal wordt gebruikt als DTMG signaal voor het display.

De PIC genereert HSync en VSync uit het CSync signaal van de VDP en houdt daarbij rekening met het even/oneven raster. Aan het digitale signaal uit de comparator voor BLEO > 3.5V kan de PIC zien of het huidige raster even of oneven is.

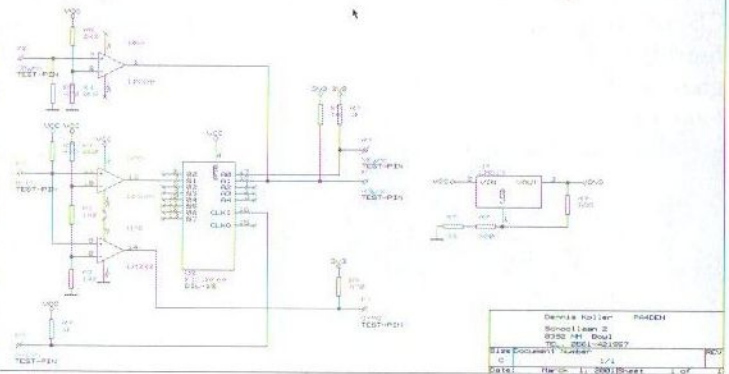
PIC software

Het programma aan het eind van dit artikel is geschreven voor een klok-snelheid van 10 MHz. De PIC kan dus geklokt worden door de DHCik uitgang van de VDP. Omdat de signalen voor het LCD display 3.3 Volt zijn en de PIC 5 Volt uitgangen heeft, wordt er softwarematig een soort open-collector uitgang gecreëerd. Dit wordt gedaan door de uitgang vast in te stellen op laag (0V), en als er een hoog gegeven moet worden, wordt de uitgang als ingang geschakeld. De pull-up weerstanden trekken het signaal dan naar de 3.3 Volt.

Voor een oneven raster moet het display op lijn 1 beginnen en voor een even raster dus op lijn 2. Dat betekent dat er voor een oneven raster op de eerste lijn geen extra sync-puls gegenereerd zou moeten

worden. Een ander probleem hierbij is dat de PIC na het detecteren van een HSync van de VDP dan eerst nog moet uitzoeken wanneer er precies een extra sync gegeven moet worden. Omdat dit extra tijd kost, ongeveer 4µs, duurt het 4µs langer voordat de extra sync gegeven wordt en valt de extra sync in de beeldinformatie. Tevens raken dan de eerste circa 20 pixels verloren. Daarom wordt er altijd een extra syncpuls gegeven direct na de originele. Daarna kan de PIC nadenken en kan op één van de eerste lijnen bovenaan nog een tweede extra syncpuls gegeven worden om op het even raster uit te komen.

Bij het label 'MAIN' staat de PIC te wachten tot het CSync signaal laag wordt. Als het laag is en vrij direct weer hoog wordt, springt de PIC naar 'CHKH'. Duurt het wat langer voordat het signaal weer hoog is, is de PIC inmiddels weer een paar instructies verder en springt hij naar 'HSYN'. Als het nóg langer duurt komt de PIC vanzelf in de VSync routine. Per raster komt de PIC 6



Figuur 2 — Het uiteindelijke schema voor de synchronisatie

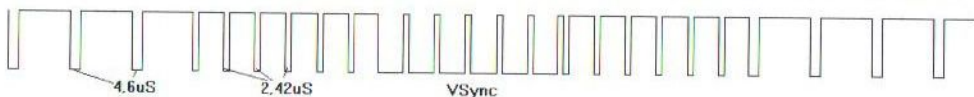


keer in dit VSync stukje. Bij de derde keer wordt de VSync uitgang laag gemaakt en bij de zesde keer weer hoog. Bij 'HSYN' wordt eerst direct de extra HSync gegeven, daarna wordt gecontroleerd of de derde HSync gegeven moet worden. De derde sync wordt gegeven op de twaalfde keer dat 'HSYN' uitgevoerd wordt én als BLEO > 3.5V. Bij het detecteren van een korte CSync puls wordt er verder niets gedaan. De PIC wacht tot het signaal weer hoog wordt en begint opnieuw, zie figuur 3.

Kleuren

De eerste maand heeft het LCD scherm eigenlijk geen fatsoenlijke kleuren weergegeven. Alle 6 datalijnen van rood, groen en blauw zaten aan elkaar, elke kleur was dus aan of uit. In totaal waren er dus al acht kleuren zichtbaar, maar bepaald niet hoe het zijn moest. Als die MSX print in BASIC terecht kwam, werd het beeld geel in plaats van bijvoorbeeld blauw. Het blijkt dat alle datalijnen geïnverteerd zijn, dus als alle bits 0 zijn, is de kleur maximaal.

Het liep op dat moment vast op fatsoenlijke A/D converters. Ik had gekozen voor het type *TDA8706A* omdat mijn scherm 3,3V nodig had. Blijkbaar is het lastig om een 3,3V A/D converter te vinden, dus als er enigszins een keuze mogelijk is heeft 5V duidelijk de voorkeur.



Figuur 3 — CSync signaal uit de VDP

Op een gegeven moment stond er op internet een adres waar de *TDA8706* te krijgen was, die heb ik dus maar gekocht. Later bleek ook deze een 5V versie te zijn. Voor 3,3V moet het de *TDA8706A* zijn. Uiteindelijk heb ik toch deze ADC's gebruikt omdat het anders waarschijnlijk eindeloos gaat duren. Afgezien van de spanningen op de data-uitgang is elke AD converter te gebruiken, die tot ca. 11 MHz kan omzetten. De hoogste frequentie die uit de VDP komt is DHCLK, 10.738 MHz. Voor de *TDA8706* kan ook een *TDA8703* genomen worden. Een andere veel voorkomende AD converter is de *TDA8708*, die meer mogelijkheden heeft maar deze zijn eigenlijk totaal overbodig. Bovendien heeft hij meer externe componenten nodig en heeft hij een grotere behuizing. Het grote voordeel is dat hij wel beter te krijgen is. Deze AD converters zijn oorspronkelijk ook bedoeld voor videosignalen, ze hebben dus in elk geval een clamp ingang.

Voor de VDP geldt dat een hogere spanning van de RGB uitgangen betekent een heldere intensiteit op het scherm. Hoe hoger de spanning op het R pootje, des te meer rood, hoe lager de spanning op G, des te minder groen, etc. Vrijwel elke AD converter geeft bij een lage spanning alle bits nul en bij een hoge spanning één. Dit past dus niet helemaal, het LCD scherm geeft juist de maximale



intensiteit bij alle bits nul... Dit is op twee manieren op te lossen, óf alle bitjes inverteren, óf het analoge signaal uit de VDP inverteren. Naast het analoge signaal van de VDP heeft de AD converter nog een clock en een clamp signaal nodig.

Clock

Elke keer bij het hoog worden van het clocksignaal zet de AD converter het analoge signaal om en wordt de digitale data op de uitgangen gezet. In principe zou deze clock met de DHCLK van de VDP gestuurd kunnen worden, omdat de VDP met DHCLK aangeeft of er een nieuwe pixel geschreven wordt. De analoge uitgang van de VDP verandert dus ook hoogstens één keer per DHCLK periode, en bij een resolutie van 256 zelfs één keer per twee DHCLK periodes. Eén keer per DHCLK het analoge signaal omzetten zou voldoende zijn, maar een voorwaarde is dat dan ook het LCD scherm dezelfde clock krijgt. Als het LCD scherm een clocksignaal krijgt wat niet in verband staat met de clock van de AD converter, ontstaat er een soort sneeuw in beeld. Dit wordt veroorzaakt doordat de AD converter bij elke clock puls even tijd nodig heeft om de nieuw omgezette data goed op de data-uitgangen te zetten. Het omzetten van een data-uitgang van 0 naar 5V of terug kost altijd even tijd. Het LCD scherm kijkt eveneens in het ritme van de clock naar zijn data-ingangen. Als de clock van de AD converter en het LCD scherm niet synchroon lopen, zal het geregeld voorkomen dat het scherm de data leest terwijl de databits nog niet

goed staan, valse data dus...

Een lastig probleem is dat de meeste LCD schermen geen lagere clock dan 14 MHz accepteren. Dit is een groot probleem voor LCD schermen van 640 bij 480 pixels. De VDP geeft 512 pixels in het ritme van DHCLK, de DHCLK is 10.738 MHz dus elke pixel duurt $1 / 10.738 \text{ MHz} = 93.12\text{ns}$. Alle 512 pixels op een rij duren dus $512 \times 93.12\text{ns} = 47,67\mu\text{s}$. Als het LCD scherm van 640 pixels een clock van 14 MHz krijgt, doet hij er $45.7\mu\text{s}$ over om alle 640 pixels te beschrijven. De VDP heeft dan nog $2\mu\text{s}$ te gaan en alle pixels in de laatste $2\mu\text{s}$ vallen dus buiten het scherm. Een LCD scherm van 800 bij 600 pixels doet er bij 14 MHz ongeveer $57,1\mu\text{s}$ over, bij schermen van 800 bij 600 en hoger raakt er dus geen beeld-informatie verloren.

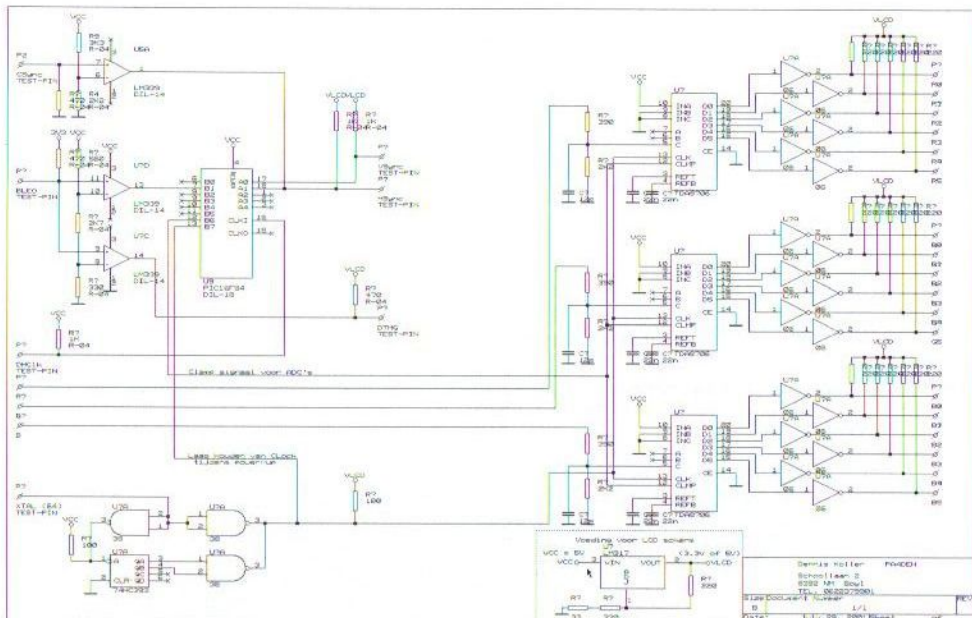
Clamp

Bij een zwart scherm worden de analoge uitgangen van de VDP ongeveer 2,2V en bij volledig wit 3,8V. Het clamp signaal wordt gebruikt om ervoor te zorgen dat de AD converter bij deze 2,2V ook constant een digitale 0 geeft.

Ik hoop hiermee voldoende informatie te hebben gegeven om zelf een LCD scherm op een MSX aan te sluiten. Indien u niet over de nodige solderervaring beschikt, raad ik u aan om dit dan door een elektronica expert uit te laten voeren. Mocht u nog vragen hebben, dan kunt u mij via het e-mailadres PA4DEN@hotmail.com bereiken.

Dennis Koller





Figuur 4 – Het definitieve schema

; VSync generator met even/odd delay

```
; A0 = U VSYNC
; A1 = HSYNC
; B1 = I BLEO
```

```
; Aantal VSYNCS 0D
; Aantal HSYNCS 0F
; Wachtlus 14
```

```
list p=16f84
#include p16f84.inc
```

```
BSF 03,5 ;Page 1
MOVLW 07F ;Poort B pull-up
MOVWF 01
MOVLW 0FF ;INI poort A
MOVWF 05
MOVLW 0FF ;INI poort B
MOVWF 06
BCF 03,5 ;Page 0
MOVLW 0FC
MOVWF 05

CLRF 0D
```

```
MAIN: BTFSC 05,1 ;Lees CSYNC ingang
GOTO MAIN ;Is hoog
```

```
BTFSC 05,1
GOTO CHKH
```



```

BTFSC 05,1
GOTO CHKH
BTFSC 05,1
GOTO CHKH
BTFSC 05,1
GOTO CHKH
BTFSC 05,1
GOTO HSYN
BTFSC 05,1
GOTO HSYN
BTFSC 05,1
GOTO HSYN

;
MOVW 0F3 ;VSYNC
MOVWF 0F ;Reset HSync counter

INCF 0D,1 ;Vertraging VSync
MOVWF 0D
SUBLW 003
BTFSS 03,2 ;3e keer?
GOTO VSY1
BSF 03,5 ;Ja, VSync laag
BCF 05,0
BCF 03,5

VSY1: MOVWF 0D
SUBLW 006 ;Lengte VSync puls
BTFSS 03,2 ;6e keer?
GOTO CHKH
BSF 03,5 ;Ja, VSync hoog
BSF 05,0
BCF 03,5
GOTO CHKH

HSYN: BSF 03,5 ;HSYNC
BCF 05,1 ;Geef extra HSync
NOP
NOP
NOP
NOP
BSF 05,1
BCF 03,5

CLRWF 0D ;Reset VSync counter
NOP

BTFSS 0F,7 ;Tijd om eventueel 3e sync te geven?
GOTO CHKH ;Al geweest
INCF 0F,1
BTFSS 03,2
GOTO CHKH ;Nog niet
BTFSC 06,1 ;Ja, het is tijd, even raster?
GOTO CHKH

BSF 03,5
BCF 05,1 ;Ja, geef 3e HSync
NOP
NOP
NOP
NOP
BSF 05,1
BCF 03,5

```




```

NOP           ;Wacht i.v.m. parasitaire C
NOP
NOP
NOP
CHKH: BTFSS  05,1      ;Einde Sync puls?
      GOTO   CHKH

      GOTO   MAIN

dw 0x0102
dw 0x0304
dw _LP_OSC & _PWRTE_OFF & _WDT_OFF & _CP_OFF
end

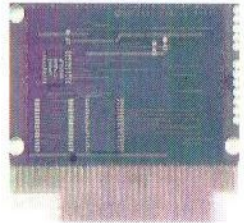
```

(Advertentie)

NIEUW!

LPE BRAM geheugenuitbreiding

Met deze nieuwe geheugenuitbreiding is het mogelijk om op een goedkope manier het RAM geheugen van de MSX uit te breiden. De *LPE-BRAM-V1* is een SRAM met een grootte van 512 kB of 512 kWord. Klok en refresh signalen zijn dus niet nodig. Het volledige geheugen is ingedeeld volgens de MSX standaard.



De BRAM is in eerste instantie gericht op het gebruik met een gewone MSX, maar kan tevens in een DMA omgeving aangesproken worden door andere (DSP) processors, zoals de *LPE Z380*. Hiervoor is echter wel een *Evolución4* slotexpander nodig.

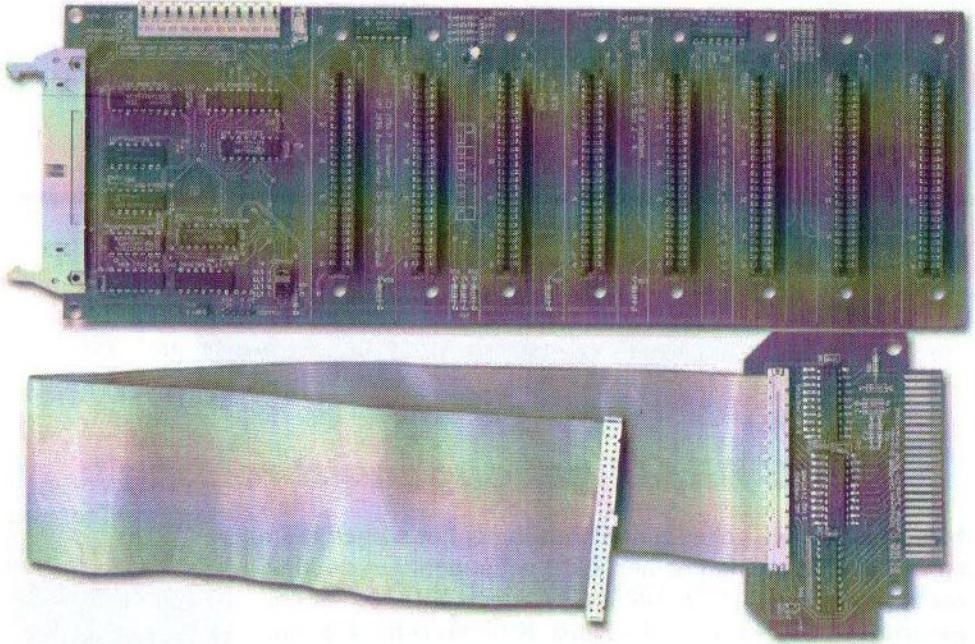
Speciaal voor gebruik met de *LPE Z380* is er de mogelijkheid om 512 kWord te plaatsen in plaats van 512 kByte, zodat er gebruik gemaakt kan worden van de *Z380* Word mode. De *LPE BRAM-V1* is via *MSX-Info Blad* verkrijgbaar voor de volgende prijzen (exclusief verzendkosten):

512 kByte Fl. 140,- (€ 63,53)
 512 kWord Fl. 200,- (€ 90,76)

Neem voor meer informatie contact op met de uitgever van dit blad. Zie de colofon voor adresgegevens.



De Sunrise 8 slotexpander nu weer verkrijgbaar!



Met 4 sub-slots voor algemeen gebruik, waaronder IDE, SCSI, SCC, FM-Pac en gamecartridges, en 4 I/O slots voor I/O-only kaarten, zoals Graphics9000, Video9000, MoonSound, en Sunrise RS232C

Nu voor slechts Fl. 299,- (€ 135,68)

Fl. 314,- (€ 142,49) inclusief verzendkosten
Verzendkosten bij rembours: Fl. 20,- (€ 9,08)

Sunrise for MSX
Postbus 61054
2506 AB Den Haag
Telefoon: (070) 360 97 07
(tussen 22:00 en 24:00 uur)
E-mail: rhiep@msx.ch
Internet: <http://www.msx.ch>



MSX wint op ASM 2001

Wie denkt dat er met een MSX1 niks meer te beginnen valt heeft het goed mis! Dat bewees de MSX groep *BandWagon* onlangs op de *Assembly2001* democompetitie, die gehouden werd in Finland, door een eerste prijs in de wacht te slepen.

Verdeeld over verschillende categorieën mochten demomakers een gooi doen naar de eerste prijs binnen hun gekozen democategorie. In de zogenaamde *Oldskool* categorie deed ook de MSX groep *Bandwagon* mee, die al vaker in de prijzen is gevallen. Zij houden zich al een paar jaar bezig met het maken van MSX1 demo's en genieten een grote populariteit onder zowel MSX-ers als andere platformgebruikers. De groep bestaat uit zes personen met ieder hun eigen specialiteit. Een *Oldskool* demo kenmerkt zich door de typische kenmerken van computers uit de jaren 80. Voorbeelden van veelgebruikte computers uit die tijd zijn de Commodore64 en natuurlijk de MSX.

BandWagon gebruikte voor hun demo *Riyadh* een MSX1 met een disk-

drive. Het is verbazingwekkend wat deze jongens uit deze machine weten te halen. Veelal bestaan hun demo's uit stilstaande achtergronden, met daaroverheen erg mooie (soms zelfs 3D) animaties, maar soms scrollen er ook hele grote animaties moeiteloos over het scherm. Aangezien de MSX1 grafisch niet tegen een MSX2 op kan is dit een enorme prestatie, dat veel van een programmeur vergt. Er moet namelijk middels vernuftige trucjes zoveel mogelijk uit de grafische chip gehaald worden. *A.K. Silvast*, die verantwoordelijk is voor het programmeerwerk en de muziek mag met recht een kunstenaar genoemd worden. In de tabel kunt u zien hoeveel punten de demo van *Bandwagon* behaalde en tevens is er te zien wie de nummers twee tot en met vijf waren. Overtuig uzelf van de kwaliteit van de *Bandwagon* demo's en neem eens een kijkje op <http://www.hut.fi/~asilvast/bw>. U zult versteld staan van wat uw MSX1 nog kan!

Gerrit van den Berg

Plaats	Punten	Titel	Platform	Credits
1	10225	Riyadh	MSX	Bandwagon
2	8213	Impossiblator	VIC20	pwp
3	6784	Beertime 2	C64	Dekadence
4	2774	Big Time	VIC20	CNCD
5	1812	Orivesi	C64	Extend

MSX Radio

Eén van onze vaste medewerkers is al een tijdje bezig met het opzetten van het live afspelen van MSX muziek via Internet. Nu het wat serieuze vormen heeft aangenomen willen we u zijn initiatief dan ook niet onthouden.

Via mijn huidige werkgever, de internet-afdeling van het Nederlands Omroepproductiebedrijf, kwam ik in aanraking met allerlei facetten van audio en video in combinatie met Internet. Het jargon hiervoor is "Streaming Media". Vroeger was dat alleen voor bedrijven met een stevige portemonnee weggelegd, maar tegenwoordig kan iedereen vrij eenvoudig audio en/of video uitzenden via het Internet. Aldus werd het idee geboren: een MSX Internet radiostation beginnen!

Eerst moest het technische deel geregeld worden. Als hosting service heb ik *Live365* gekozen. Dit is een bedrijf dat ruimte aanbiedt op hun servers. Ze maken gebruik van het *Shoutcast* protocol, een manier om MP3 bestanden te "streamen" over het Internet. Streamen betekent het live uitzenden van beeld en/of geluid via het Internet. Zo'n stream kan niet zondermeer bewaard worden voor later gebruik. Het is dus iets anders dan het downloaden van bestanden om ze later af te spelen.

Het gratis pakket van *Live365* omvat twee manieren om uit te zenden. De eerste is live vanaf de eigen PC, zodat men volledige controle heeft over wat er wanneer wordt uitgezonden. Door mijn MSX op de PC aan te

sluiten is het dus ook mogelijk om live van mijn MSX uit te zenden! Het nadeel hiervan is dat je 24 uur per dag een internetverbinding moet hebben en de kans is aanwezig dat de stream onderbroken wordt zodat één en ander niet soepel aankomt bij de luisteraars. De andere manier is door MP3 bestanden te plaatsen op de servers van *Live365*, waarna er een playlist aangelegd kan worden. Men krijgt 365 MB ruimte, wat bij een snelheid van 56kbps neerkomt op meer dan 15 uur muziek.

Met een programma genaamd *Loader365* kan ik simpelweg MP3 bestanden selecteren en aangeven wat de doel-bitrate is. Ik kan kiezen tussen 28k8, 33k6, 56k en ISDN en hoger. Voorlopig is *MSX Radio* alleen te ontvangen voor mensen met ISDN, kabelmodem of ADSL verbindingen, zodat de geluidskwaliteit nog heel aanzienlijk is. Het is mogelijk dat ik in een later stadium besluit ook een station voor lagere bitrates op te zetten om het publiek zo groot mogelijk te maken.

Wanneer luisteraars verbinding maken met een *Live365* station wordt een player-venster geopend. In dit venster staan de laatste drie gespeelde nummers, alsook bedieningsknoppen en aanklikbare 'Koop dit num-





mer' icoontjes die een zoekopdracht uit kunnen voeren bij een grote online CD zaak. Natuurlijk is ook reclame en links naar de pagina's van het huidige station aanwezig. Het is trouwens wel zo dat men commercials laat horen tussen de liedjes, die afgespeeld worden. *Live365* moet natuurlijk ook geld verdienen.

En de auteursrechten, hoe zit het daarmee? Men heeft hier bij *Live365* natuurlijk over nagedacht. Op dit moment is het nog zo dat de Amerikaanse wetgever vindt dat particulieren, die via dergelijke zenders uitzenden geen auteursrechten hoeven af te dragen, omdat deze zelf licentiegelden en royalty's afdragen aan de verantwoordelijke instanties. *Live365* heeft verder een aantal regels waaraan de uitzender moet voldoen, zodat men geen problemen krijgt met de *RIAA*. Het komt erop

neer dat het, gezien de huidige wetgeving, legaal voor mij is om MSX muziek uit te zenden op internet. Meer informatie hierover kan nagelezen worden op hun website.

Nadat het technische gedeelte in orde was, kon ik verder gaan met het leuke werk: muziek uit mijn verzameling selecteren voor de playlist. Op dit moment bestaat die uit voornamelijk spelmuziek uit MSX games. Zowel originele soundtracks, door mij opgenomen van mijn MSX, als gearrangeerde versies zoals die door bijvoorbeeld *Konami* op CD zijn uitgebracht. Onder andere deze spellen komen voorbij: *Snatcher*, *SD-Snatcher*, *Metal Gear 1* en *2*, *Aleste*, *Dragon Slayer*, *Ys*, *Fray*, *Xak*, *Gradius*, *Parodius*, *Rune Master*, *Mr. Ghost*, *Space Manbow* en vele andere. Maar ik ben ook bezig om muziek te verzamelen gemaakt door MSX gebruikers (voor zover mogelijk met toestemming). Ik heb al muziek ontvangen van *Wolf* (*Maarten van Strien*) en *TeddyWareZ*. Hierbij wil ik dan ook een oproep plaatsen om mij te voorzien van muziekbestanden (het maakt niet uit in welk formaat) als je ze uitgezonden wil hebben op *MSX Radio!* Stuur een e-mail naar t.zondag01@chello.nl indien je een bijdrage wilt leveren.

Het opnemen en bewerken van muziek van m'n eigen MSX is een redelijk complex werkje wat behoorlijk wat tijd in beslag neemt. Als MSX heb ik gekozen voor een Sony HB-G900P, omdat die een regelbare stereo tulp geluidsuitgang heeft met een heel acceptabele signaal/

ruisverhuiding. Mijn Philips MSX2 heeft een kwalitatief veel mindere uitgang die nogal wat storing bevat, zodat digitaliseren een stuk lastiger wordt. Ook laten Philips MSX2's het signaal van cartridges (FM-PAC, SCC of Muziekmodule) veel te hard door, waardoor de PSG overstemd wordt. Verder gebruik ik een 4-kanaals mengpaneel om het geluidsniveau te regelen en een snelle PC met een kwalitatief goede *AWE64 Gold* geluidskaart. Uiteraard probeer ik de kwaliteit zo hoog mogelijk te houden door zoveel mogelijk in digitale CD kwaliteit te werken (44kHz, 16 bits). Na de muziek opgenomen te hebben moet het nog digitaal nabewerkt worden. Daarvoor gebruik ik *Cool Edit Professional*. Met deze software kan ik de door de MSX veroorzaakte ruis wegnemen en eventueel galm of echo toevoegen en het eindresultaat in MP3 formaat wegschrijven.

De vereisten om MSX Radio te kunnen beluisteren zijn: een 200 MHz PC of sneller, Windows 95 of hoger, een internetverbinding met een snelheid van minimaal 64kbps en *Internet Explorer* of *Netscape Navigator* versie 4 of hoger.

MSX Radio website: <http://www.live365.com/stations/266034>

Live365 juridische informatie: <http://www.live365.com/info/legal.html>

Tristan Zondag

Verklarende woordenlijst:

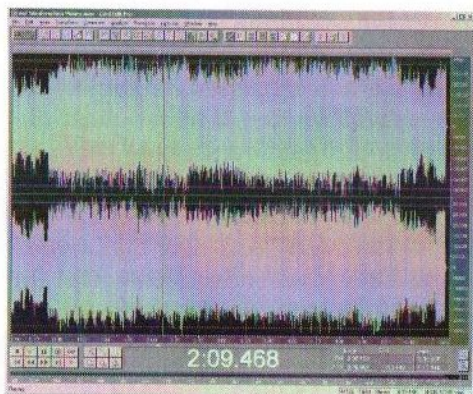
streaming media: Uitzenden van beeld en/of geluid via het Internet TCP/IP protocol in plaats van via klassieke media als kabel, ether of satelliet.

MP3: Populair bestandsformaat waarin digitale muziek met een zeer grote compressiefactor wordt opgeslagen.

shoutcast: Manier om MP3 bestanden uit te zenden (te 'broadcasten') via het TCP/IP protocol.

bitrate: Hoeveelheid (kilo)bits per seconde (kbps).

RIAA: *Recording Industry Association of America*.



Cursus PowerBasic

- deel acht -

Ondanks de titel is al enkele afleveringen achter elkaar geen aandacht besteed aan *PowerBasic*. Dit komt omdat het volgens *Sjoerd* leuker is om de *Graphics9000* direct aan te spreken.

Het heeft niets met de *Graphics9000* te maken, maar er vielen mij een paar dingen op tijdens het programmeren. De gegeven voorbeelden maken veel gebruik van strings en om die op te bouwen worden ze vaak bij elkaar 'opgeteld,' zoals $A\$=A\$+B\$$. Listing 1 geeft een 'out of string space' foutmelding, terwijl de string echt niet langer wordt dan 100 tekens. Als na het runnen met `?fre(")` het vrije string geheugen wordt opvraagd, komt naar voren dat slechts de helft gebruikt is en het geheugen dus nog maar half vol is. Het blijkt dat de hele string verplaatst wordt als er geconcateneerd moet worden. Dit maakt ook nog eens dat één en ander erg traag gaat. Het is daarom beter om eerst met `STRING$` of `SPACE$` de lengte te bepalen en daarna met `MID$` de string op te bouwen (zie bijvoorbeeld regel 60 van listing 2). Dan hoeft niet steeds de hele string gekopieerd te worden en wordt het string geheugen ook een iets minder groot zootje, zodat *BASIC* minder snel een 'garbage collection' hoeft te houden. Qua snelheid stelt het niet zoveel voor. Het gaat iets sneller, maar er wordt wel minder geheugen verspild.

Pattern mode 2

We zullen kort wat aandacht besteden aan P2, de tweede patternmode. Deze mode heeft, in tegenstelling tot P1, geen twee lagen, maar wel een resolutie van 512 x 212 beeldpunten. Dat wil zeggen: 64 x 26.5 patterns. Maar natuurlijk is de imagespace veel groter. Deze is 1024 x 512 beeldpunten groot en de display area (van 64 x 26.5 patterns) kan met de scrollregisters ingesteld worden.

Hoe de image space eruit ziet, wordt bepaald door de pattern name table. Deze is 16 kB groot: 64 rijen van 128 pattern names lang. Er worden natuurlijk twee bytes per positie gebruikt zodat er meer dan 256 patterns mogelijk zijn. In theorie is er dus ruimte voor 65.536 patterns, maar zoveel geheugen zit er nu ook weer niet in de *Graphics9000*. *Yamaha* beweert dat er plaats is voor 15.359 patterns. Dit is eigenaardig, omdat er dan een heel stuk VRAM ongebruikt is. Maar goed, voor de sprites geldt iets dergelijks. Er zijn er maar 125 tegelijkertijd in beeld mogelijk, terwijl 128 een stuk logischer lijkt. Er is ook plaats voor 128 sprites.

Listing 1

```
10 CLEAR 150: FOR I=1 TO 100: A$=A$+" " : NEXT
```

Zoals gebruikelijk staat in de pattern generator table hoe de verschillende patterns er uit zien. Dit heeft dezelfde indeling als het van MSX2 bekende SCREEN 7. De sprites zijn even breed als een stuk scherm van 16 x 16 beeldpunten en dus niet zoals in SCREEN 7 veel te breed. Verder hebben sprites dezelfde mogelijkheden als in pattern mode 1.

Lijnen tekenen

Speciaal voor de meneer die mij gemailld heeft over deze 'cursus' zal ik de voorbeeld listing 2 in z'n geheel opnemen. We gaan de lijnen in P2 tekenen, want ik kon niet zo snel een spectaculair voorbeeld van de P2-mode vinden. Dit combineren scheelt natuurlijk ook weer kostbare paginaruimte.

Vooruit, ik leg de USR-routines ook nog even snel uit. `USR(0)` kopieert het eerste scherm van de V99x8 naar de Graphics9000. `USR1(CHR$(X)+CHR$(Y))` plaatst Y en alle volgende tekens van de string in registers vanaf X. "xabcd" stuurt dus a naar register x, b naar x+1, c naar x+2, enz. `USR2(CHR$(X)+CHR$(Y))` schrijft Y en alle volgende tekens van de string naar poort nummer X. "xabcd" stuurt dus "abcd" naar poort x. `USR3(CHR$(X)+CHR$(Y))` plaatst na Y en elk volgend teken een X in poort 0 (VRAM write port). Als de string "abcdef" gebruikt wordt, wordt er dus het volgende in het VRAM geschreven: axbxcx-dxexfx.

In regel 40 worden de display mode

registers (registers 6 en 7) zo ingesteld dat P2 gekozen wordt. Voor de liefhebbers van 60Hz in plaats van 50Hz kan de `CHR$(8)` in deze regel vervangen worden door `CHR$(0)`. In regel 60 worden vijf kleuren ingesteld. Port 1 is de palette port. Het juiste kleurnummer waar vanaf de kleuren ingesteld worden is in regel 40 al in het juiste register geplaatst (register 14, hier moet kleurnummer*4 in). De pattern name table wordt zo gevuld dat er, net zoals SCREEN 2 op de MSX1, een bitmap mode gesimuleerd wordt. Probeer voor de grap listing 3 maar eens. De tabel wordt dus gewoon olopend gevuld. Hiertoe worden in regel 80 vier strings gevuld. In een lus wordt dan in regel 90 het juiste adres (vanaf 7C000H dus) ingesteld. Elke regel is in P2 256 bytes, dus dat is handig met de adressen. In regel 100 wordt de juiste string naar het VRAM geschreven met `USR3`. In regel 120 wordt de pattern generator table, ons bitmap scherm als het ware, gewist ter grootte van een scherm (512 x 212 beeldpunten). Dit wordt gedaan door 212 strings van 256 nullen naar het geheugen te schrijven vanaf adres 0, het begin van de pattern generator tabel.

Dan volgt dus het lijntekenen. In regel 140 worden een paar argumenten in een string gezet die niet veranderen. Dit zijn de logische operatie, de write mask en de kleur. Met de kleur is nog iets eigenaardigs aan de hand: het is mogelijk om meerdere kleuren tegelijk te gebruiken. Zoals het nu is ontstaat er een verticaal streepjes



patroon. Bevalt dit niet dan kunnen de waarden &h34 vervangen worden. Wordt hier vier keer hetzelfde ingevuld, dan wordt dus een egale kleur verkregen (bijvoorbeeld &h22,&h22).

In regel 150 worden dan 512 lijnen getekend. De lijnen lopen van (X1, Y1) tot (X2, Y2). De subroutine vanaf regel 200 berekent de juiste coördinaten. Het is namelijk niet zo dat zomaar het begin- en eindpunt opgegeven kan worden. Het startpunt kan wel direct in de registers geschreven worden. Het eindpunt wordt opgege-

ven door de hoogte en de breedte op te geven van de rechthoek (X1, Y1) - (X2, Y2). In register 44 moet dan nog aangegeven worden of de rechthoek breder is dan dat hij hoog is, of andersom. Dit wordt uitgezocht in regel 200. De richting vanuit het beginpunt moet ook opgegeven worden. Dit gebeurt in regels 210-220. In regel 230 wordt tenslotte het begin van de string met daarin de parameters van het lijn commando gemaakt. Hier moeten dus nog wel de logische operatie, de writemask, de kleur en het commando achter geplakt wor-

Listing 2

```

10 CLEAR2000: DEFINT A-Z: GOSUB9160: RESTORE
20 DEF FNP$(X)=CHR$(X MOD 256)+CHR$(X\256): DEF FNC$(X,Y)=FNP$(X)+FNP$(Y)
30 ' stel patternmode P2 in
40 A$=USR1(CHR$(6)+"Y"+CHR$(8)+CHR$(194)+STRING$(15,0)): OUT&H67,0
50 ' kleuren instellen
60 A$=STRING$(16,1): FOR I=2 TO 16: READ A: MID$(A$, I)=CHR$(A): NEXT A: A$=USR2(A$)
70 ' pattern name table vullen
80 DIM B$(3): FOR I=0 TO 3: FOR J=0 TO 63: B$(I)=B$(I)+CHR$(J+I*64): NEXT J: NEXT I
90 FOR I=0 TO 26: A$=USR1(CHR$(0)+CHR$(0)+CHR$(&HC0+I)+CHR$(7))
100 A$=USR3(CHR$(I\4)+B$(IMOD4)): NEXT I
110 ' pattern generator table wissen
120 A$=USR1(STRING$(4,0)): FOR I=0 TO 211: A$=USR3(STRING$(129,0)): NEXT I
130 ' 512 lijnen tekenen
140 B$=CHR$(12)+CHR$(255)+CHR$(255)+CHR$(&H12)+CHR$(&H34)+" "+CHR$(176)
150 FOR I=0 TO 511: X1=I: Y1=0: X2=511-I: Y2=211: GOSUB200: A$=USR1(A$+B$)
160 NEXT I: END
170 ' kleurdata
180 DATA 0,0,0,31,0,0,0,31,0,0,0,31,31,31,31
190 ' Bereken lijntje
200 MJ=ABS(X1-X2): MI=ABS(Y1-Y2): IF MI>MJ THEN SWAP MI, MJ: A=1 ELSE A=0
210 IF X1>X2 THEN A=A+4
220 IF Y1>Y2 THEN A=A+8
230 A$="$"+FNC$(X1,Y1)+FNC$(MJ,MI)+CHR$(A): RETURN
9150 ' Laad ML
9160 RESTORE 9200: A=&HF975
9170 DEF USR=A: GOSUB 9190: DEF USR1=A: GOSUB 9190: DEF USR2=A
9180 GOSUB 9190: RESTORE 9250: DEF USR3=A
9190 READ A$: POKE A, VAL("&H"+A$): A=A+1: IF A$="C9" THEN RETURN ELSE 9190
9200 DATA F3, AF, D3, 64, D3, 63, D3, 63, D3, 63, D3, 99, D3, 99, D3, 99, 3E, 8E, D3
9210 DATA 99, 01, D4, 00, DB, 98, D3, 60, 10, FA, 0D, 20, F7, FB, C9, 3A, 63, F6, FE
9220 DATA 03, C0, 2A, F8, F7, 46, 05, C8, 23, 5E, 23, 56, EB, 7E, 23, D3, 64, 0E, 63
9230 DATA ED, B3, C9, 3A, 63, F6, FE, 03, 0C, 2A, F8, F7, 46, 05, C8, 23, 5E, 23, 56
9240 DATA EB, 7E, 23, C6, 60, 4F, ED, B3, C9
9250 DATA 3A, 63, F6, FE, 03, C0, 2A, F8, F7, 46, 05, C8, 23, 5E, 23, 56, EB, 7E, 23
9260 DATA 0E, 60, ED, A3, D3, 60, 20, FA, C9

```

den. Met een 'v9990 application manual' er naast, moet het toch ook in assembly wel lukken.

In de regels vanaf 9150 wordt de machinecode voor de USR-functies geladen, vanaf adres F975H. Dit is het begin van de PLAY-wachtrij en

wordt dus overschreven als het PLAY-commando gebruikt wordt. Deze machinecode is functioneel hetzelfde als die uit *MSX-Info Blad 17*, aangevuld met die uit *MSX-Info Blad 19*.

Sjoerd Mastijn

Listing 3

```
10 SCREEN 2
15 DEF FN R(X)=RND(1)*X
20 FORI=0TO15:FORJ=0TO23:VPOKE6160+I+J*32,VPEEK(6144+I+J*32):NEXT:NEXT
30 LINE-(FNR(128),FNR(192)),FNR(16):GOTO 30
```

Oproep

Een MSX-groep uit Nederland is momenteel druk bezig om de programmacode van essentiële MSX programma's te verzamelen. Dit betekent dat er veel contact gezocht moet worden met de makers en er veel onderhandeld moet worden. Veel programmamakers zijn echter huiverig om hun programmacode af te staan, omdat ze bang zijn dat andere mensen in hun code gaan zitten snuffelen waar zij enorm veel tijd in gestoken hebben.

Veel programmeurs hebben de MSX inmiddels verlaten en hebben andere interesses gekregen. Niettemin zijn er nog veel MSX-ers die gebruik willen maken van deze programma's om ze bijvoorbeeld aan te passen of om ze te gebruiken om nieuwe software te schrijven. Een voorbeeld hiervan is het tekstverwerkingsprogramma *TED*. Het is onze taak om deze programmeurs te overtuigen, dat er nog steeds interesse is voor hun software, maar dat ze er helaas geen

geld meer mee kunnen verdienen.

We hebben al de medewerking gekregen van *Ramon van der Winkel* van het *MSX Software Team (MST)*. Hij heeft destijds al zijn code aan een persoon met de naam *Jochum* of *Joachim* gegeven. Wie kent deze jongen en wat is zijn achternaam en eventueel e-mailadres?

Tevens heeft een spelontwerper zijn medewerking toegezegd. Het publiceren van de code kan een tijdje duren, omdat we het op een correcte wijze willen organiseren. Ik wil u daarom dan ook vragen om actief mee te werken aan deze databank van MSX programmacode, zodat wij de nog resterende MSX programmeurs kunnen helpen en hun een nieuwe start kunnen aanbieden. U kunt contact opnemen met mij via e-mail op h.otten@chello.nl. Alvast bedankt voor uw hulp!

Hans Otten



PLR 5.02

Zoals de vorige keer beloofd is, plaatsen wij nu een wat uitgebreidere uitleg over de nieuwe functies van *PLR 5.02*. *Rinus* en *Jack* leggen u de nieuwste mogelijkheden uit.

Velen van u zullen ongetwijfeld weten, dat met de komst van *Norton Commander* op de PC het werken onder MS-DOS een stuk aangenamer werd. Met dit programma was het namelijk mogelijk om vrij eenvoudig bestanden te bekijken, te verplaatsen of te wissen. Sinds geruime tijd is het Japanse programma *MultiMente* voor de MSX beschikbaar, dat het werken onder MSX-DOS2 veraangenaamt.

Piet Loeve van *MSX Club Rijnmond* maakt nog steeds programma's voor MSX en hij is al geruime tijd bezig om een dergelijk programma voor de MSX te maken, genaamd *PLR*. Alhoewel *PLR* grafisch gezien niet zo mooi is als *MultiMente* zijn er wel veel gebruiksmogelijkheden aanwezig. Onlangs bracht *Piet* weer een update uit en die maakt het programma ineens een stuk interessanter. In het programma zelf is een handleiding aanwezig en daarmee kan zelfs een beginner goed uit te voeten. We beperken ons in deze bespreking dan ook tot de nieuwe of verbeterde opties in deze versie.

Het grote voordeel van *PLR* ten opzichte van *MultiMente* is, dat eerstgenoemde ook onder MSX-DOS1 kan functioneren. Hebt u dus niet de beschikking over DOS2 dan is dit zeker een programma dat niet in uw assortiment mag ontbreken. *PLR2* is gemaakt voor MSX-DOS2 en als u dit programma op een harde schijf zet, dan zal het prima werken. Wanneer u in



MSX-DOS1 *PLR2* opstart, dan schakelt het programma automatisch over naar *PLR1*. De enigste manier om onder DOS1 met subdirs te werken is door met de clusternummers te schakelen. Ook het zogenaamde path wordt niet met de BDOS2 functie bepaald, maar wel met de BDOS2 functie geschakeld. Wat echter nog niet is opgelost is dat wanneer u onder DOS2 *PLR1* opstart u in *PLR 5.00* terecht komt. Er wordt aan gewerkt om dat euvel te verhelpen.

Tijdens het testen van dit programma kwam naar voren, dat waar versie 5.00 de mogelijkheid biedt om met de toetscombinatie [ESC] + [F5] de functie DEL aan of uit te zetten, dit in versie 5.02 echter de SORT functie aan of uit zet. Beide functies

behoren eigenlijk door een dergelijk programma te worden ondersteund. Wellicht wordt dit later nog toegevoegd.

In de vorige versies wilde het nog wel gebeuren, dat in de grafische of tekstmode de computer bleef hangen. Dit is ook aangepast. De verschillende viewers en dan met name de foutafhandeling zijn sterk verbeterd. *PLR* ondersteunt nu de meest voorkomende grafische bestanden, zoals PIC, STP, SCC, SC5, SC7, S12. Ook worden GE5, GE7 en SC& ondersteund mits in dezelfde directory de losse paletbestanden PL5 en PL7 aanwezig zijn. In het INSTEL.PLR bestand kan het één en ander worden ingesteld en dat werkt nu ook goed.

Ook was er eerst sprake van het twee keer moeten verwijderen van bestanden alvorens ze uit beeld verdwenen. Wanneer echter met *MultiMente* bestanden werden verwijderd konden ook deze bestanden in *PLR* nog gezien worden. Sommige mensen vinden dit wel handig, want dan kan met het commando UNDEL de oude situatie weer teruggezet worden. Dit is echter niet de bedoeling en is in de nieuwste versie dan ook opgelost. Het verwijderen of hernoemen van de subdirs op dezelfde schijf is uitgebreid met de controle van links van dezelfde subdir.

Iets wat met *MultiMente* ook goed kan, is het in- en uitpakken van bestanden met LHEXT en/of PMEXT en andere crunchprogramma's. Ook *PLR* kan hier goed mee overweg. Wij

hebben het uitgetest met ARJ bestanden en ook die werden keurig uitgepakt. Iets wat *MultiMente* echter niet kan en *PLR* wel is het uitpakken van PMA bestanden rechtstreeks van Cd-rom naar harde schijf. Dat betekent dus, dat de bestanden nu niet meer eerst naar de harde schijf moeten worden verplaatst alvorens ze kunnen worden uitgepakt.

Er zijn nog wel wat opties die (nog) niet in *PLR* zitten, zoals een screen-saver en het tonen van de tree en history. Ook is er nog geen mogelijkheid om snel naar de root te springen. Hopelijk wordt dit in een volgende update ook meegenomen. Dit programma verdient het echter wel om eens naar gekeken te worden. Schaf het daarom via *MSX-Info Blad* aan en geef eventuele wensen en/of fouten aan ons door, zodat wij er wat mee kunnen doen. Uw naam wordt genoteerd in ons archief en bij een update voorzien wij u hier gratis van. Voor (bestel) informatie kan contact worden met de uitgever van dit blad. Zie voor het adres of telefoonnummer de colofon.

Rinus Stoker
Jack Tresoor

Maker: Piet Loeve
Club Rijnmond Rotterdam

Soort: Norton Commander kloon

Systeem: MSX2 met 128 kB

Optioneel: harddisk

Prijs: Fl. 5,- (€ 2,27)



UZIX

De laatste jaren is *Linux*, een gratis besturingssysteem voor PC's, enorm in populariteit toegenomen. *Linux* is een UNIX variant, een besturingssysteem wat van oorsprong werd gebruikt op mainframes en supercomputers. Vooral multiprocessing, internetapplicaties en multi-user eigenschappen zijn sterke punten van UNIX. Sinds kort is er ook voor MSX een heus UNIX besturingssysteem: *UZIX*.

UZIX staat voor "UNIX implementation for MSX" en is gebaseerd op *UZI*, geschreven door Douglas Braun. *UZIX 1.0* was de eerste *UZIX* versie en is geport naar MS-DOS / MSX door Archi Schekochikhin en Adriano Rodrigues da Cunha. De MS-DOS versie is nooit verder gekomen dan het bèta stadium en werd voornamelijk onderhouden door Archi Schekochikhin. Adriano Cunha heeft het grootste deel van de MSX versie geschreven.

Door gebruik te maken van een memory mapper kan *UZIX* multiprocessing aan. Dat wil zeggen dat het meerdere gelijktijdige processen (programma's) ondersteunt, die in de achtergrond blijven werken. Ook zijn multi-user toepassingen mogelijk, zodat meerdere gebruikers tegelijk kunnen inloggen op een MSX die *UZIX* draait. Dit alles werkt al op een 3.58MHz MSX, maar het gaat uiteraard soepeler als men beschikt over 7MHz of een turbo R, liefst met harddisk en veel geheugen.

UZIX ondersteunt reeds een TCP/IP verbinding via een RS232 interface die verbonden is aan een PC of modem. Zo is het mogelijk om, eventueel

gebruik makend van een PC met een internetverbinding, met de MSX het Internet op te gaan. Voorlopig is het aantal applicaties nog beperkt: telnet, ping, finger, FTP, POP mail client, DNS client (nslookup), webbrowser (*Fudebrowser*), netstat, SLIP driver, PPP driver en een dial-up driver. Met telnet kan een verbinding gemaakt worden met een andere UNIX machine, ook via Internet. FTP is voor het downloaden van bestanden, en *Fudebrowser* is bijna een echte webbrowser. Dit programma maakt gebruik van een "proxy" op de PC die opgehaalde HTML pagina's vertaalt naar een formaat wat de MSX makkelijker kan weergeven. Zelfs de MSX muis wordt ondersteund om de browser te bedienen.

De maker is kort geleden begonnen aan *UZIX 2.0*. Deze versie wordt (uiteraard) veel beter dan de eerste. Dit komt voornamelijk door de grotere kernel, zodat er meer functies ingebouwd kunnen worden en er meer geheugen voor applicaties beschikbaar is zodat die krachtiger kunnen worden. Versie 1.0 was nog zowel voor MSX1 als voor MSX2 en hoger geschikt, maar de opvolger zal alleen op MSX2 en hoger werken.



De volgende hardware wordt door versie 1.0 ondersteund:

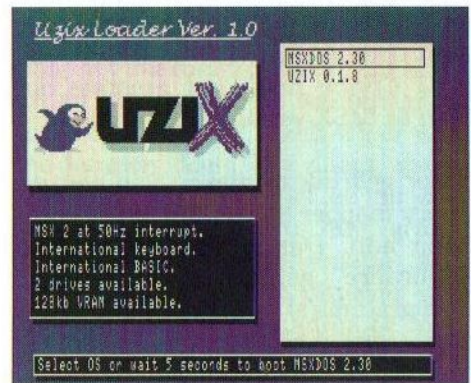
- MSX1 of hoger
- Memory mapper tot en met 4MB
- Standaard MSX RS232 interface met bios (bijvoorbeeld Sony)
- *MegaSCSI* of *Sunrise* IDE hard-disk interfaces (NIET Gouda/Novaxis/MK!)

Versie 2.0 zal de *Sunrise* RS232 en *ACCNET* interfaces aan de lijst toevoegen. *UZIX* werkt zowel op hard-disk als op floppy. Voor harddiskgebruik moet wel een hele partitie opgeofferd worden, want *UZIX* gebruikt zijn eigen filesysteem dat niet compatible is met MSX-DOS. Het is dus ook niet zondermeer mogelijk om MSX diskettes in te lezen met *UZIX*. Er zijn wel bijgeleverde utiliteiten waarmee bestanden van gewone diskettes naar *UZIX* diskettes gekopieerd kunnen worden.

Ondanks het nog in ontwikkeling zijn van deze software is *UZIX* een mooi stukje werk. Iedereen met een klein beetje UNIX kennis kan ermee omgaan. Er zijn nog weinig toepas-

singen, maar de mogelijkheid om met de MSX het Internet op te gaan is uniek. Ook wordt het in de toekomst mogelijk om andere UNIX programma's op *UZIX* te draaien. Er wordt gewerkt aan MSX-DOS emulatie zodat standaard tools als PMARC en PMEXT gebruikt kunnen worden. Ik denk dat dit een hele goede ontwikkeling is voor de MSX en raad een ieder aan om gewoon eens een diskimage te downloaden en het te proberen! In een volgend *MSX-Info* Blad willen we een uitgebreide beschrijving plaatsen hoe de MSX het Internet op kan. Kijk voor meer informatie op <http://uzix.msx.org>.

Tristan Zondag



Te koop

Philips NMS 8280 met muis, video software en kabels!
Vraagprijs: Fl. 222,25 (€ 100,85)

Telefoon: (045) 572 95 09 of (06) 44 15 85 45
Rinus Stoker



The Best of Hamaraja Night 2

~ Review ~

The best of Hamaraja Night 2 werd al eerder besproken in *MSX-Info* blad 14. Maar omdat het spel door *Mark Jelsma* nu zodanig is aangepast verdient het opnieuw een plaatsje in ons blad.

De oorspronkelijke maker van *Hamaraja Night*, *Pastel Hope*, had dit spel speciaal gemaakt voor turboR gebruikers. Dit was met name voor de snelheid en omdat er gebruik werd gemaakt van PCM-samples. Wij vonden echter dat de overige MSX-ers, die niet over zo'n machine beschikken ook het recht hebben om dit toch wel hele leuke spelletje te kunnen spelen. Daarom hebben we contact opgenomen met *Mark Jelsma* met de vraag of hij dit spel zou kunnen en willen aanpassen voor de MSX2. *Mark* had hier wel horen naar en we zijn dan ook trots u de review van *Hamaraja Night 2* te kunnen presenteren!

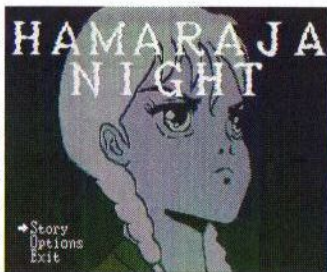
Geheugen

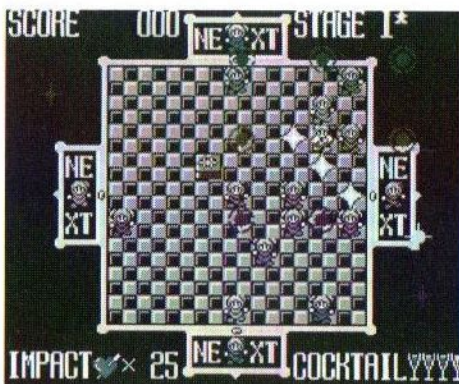
Om het geheugen weer een beetje op te frissen eerst nog even een korte uitleg van het spel. De hoofdrolspeler is een jongeman, die zich op een dansvloer bevindt. Elke keer als er op de spatiebalk wordt gedrukt, komen van alle vier de zijden personages van het andere geslacht op hem af. Met elke volgende druk op de spatiebalk of joystickequivalent komen ze een stapje dichterbij. Het is nu de bedoeling om minimaal drie

dames met dezelfde kleur jurk op een rij te krijgen. Dit kan zowel horizontaal als verticaal. Wanneer nu de [ENTER]-toets wordt ingedrukt verdwijnen deze dames. Een stage wordt pas uitgespeeld wanneer een bepaalde hoeveelheid dames van de dansvloer is verwijderd. Verder spelen er nog wel meer elementen een rol maar zie hiervoor de uitgebreide bespreking in *MSX-Info* Blad 14.

Verschillen

Op het eerste gezicht is er weinig verschil te zien met betrekking tot het oorspronkelijke spel. Het speelveld is namelijk grafisch identiek aan de turboR versie. Alles gaat natuurlijk wel wat trager. Toch is het niet zo, dat je je hier nu gelijk aan gaat ergeren, want het gaat namelijk nog snel genoeg. Eigenlijk nog sneller dan mag worden verwacht van een spel dat geheel in BASIC is geschreven. Ook zijn uiteraard de diverse PCM-samples in de nieuwe versie niet meer aanwezig. De kreten van het mannetje zullen voor de meeste mensen denk ik niet zo'n gemis zijn, want verstaan deed men het toch niet.





MoonSound!

Het oorspronkelijke spel werd muzikaal alleen door de FM-PAC ondersteund met af en toe dan een sample. De nieuwe versie is dusdanig aangepast zodat deze nu ook geschikt is voor MSX-Stereo (FM-PAC + Music Module) en zelfs voor de MoonSound! De muziek was al goed met alleen de FM-PAC, maar nu is deze nog vele malen verbeterd voor de gelukkige MoonSound bezitters. Dit heft het ontbreken van de PCM-samples ruimschoots op.

Conclusie

Bij het schrijven van deze recensie is er nog een aantal kleine bugs in het spel aanwezig, dat waarschijnlijk en hopelijk voor de beurs in Bussum is opgelost. Het is namelijk onze bedoeling om het spel opnieuw uit te brengen. Het spel zal Fl. 15,- (€ 6,81) gaan kosten. Hier krijgt men een kwaliteitsdiskette voorzien van een kleurenlabel en handleiding voor terug. *The best of Hamaraja Night 2* blijft een leuk en verslavend spelletje, ook al gaat het nu wat trager op een MSX2. Zelfs voor degenen die al beschikken over de turboR versie

kan het door de ondersteuning van de MoonSound zelfs de moeite waard zijn om deze nieuwe MSX2 versie aan te schaffen. Voor de prijs hoeft je het in ieder geval niet te laten, want voor een beetje goede muziekdiskette moet al ongeveer hetzelfde worden neergegeld. Neem voor bestelinfo contact op met de uitgever van dit blad.

Jan-Marten van der Reest

Auteur:	Pastel Hope
Conversie:	Mark Jelsma
Soort:	spel
Genre:	puzzel
Systeemeisen:	MSX 2 met 128 kB
Optioneel:	FM-Stereo / Moon-Sound
Medium:	1 * DD-diskette
Prijs:	Fl. 15,- (€ 6,81)



Beursverslag Bussum 2001

Helaas konden we dit nummer niet voor deze beurs afkrijgen. Hopelijk heeft dat geen gevolgen gehad voor het bezoekersaantal van deze beurs. Degenen die het gemist hebben kunnen in onderstaand verslag alsnog een impressie van deze dag krijgen.

De dertiende computerdag werd ook dit jaar in Bussum georganiseerd in het *Sociaal-Cultureel Centrum UITWIJK* aan de Dr. Schaepmanlaan. Dit centrum is gevestigd in een oud schoolgebouw, dat uit twee verdiepingen bestaat. Op de tweede verdieping waren, verdeeld over drie klassen en de gang, in totaal 13 MSX clubs aanwezig.

De organisatie, die evenals vorige jaren uit *Laurens Holst* en *Jaap Hoogendijk* bestond, realiseerde zich, dat de beurs in de traditionele opzet niet echt aantrekkelijk meer is voor de gemiddelde MSX-er. Men besloot daarom om naast het gewone vertrouwde beursgedeelte ook enkele demonstraties, een spelcompetitie en een MSX Lagerhuis in te plannen.

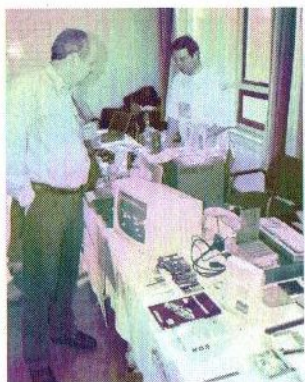
Voor veel mensen blijft het gewone beursgedeelte toch het meest interessante gedeelte van deze (MSX) computerdag. Helaas waren er niet zoveel MSX groepen aanwezig. Ook de aanwezige nieuwe software viel wat tegen in vergelijking met vorige jaren. Eigenlijk was je met een tien tot vijftien minuten wel uitgekeken. Het onderhouden van de sociale contacten tussen groepen en individuen onderling was dan ook waarschijnlijk de belangrijkste bezigheid van de ongeveer 100 bezoekers, die de

moeite genomen hadden om te kijken hoe MSX er momenteel voor staat in Nederland.

“Was er dan echt niets te zien?” zo vraagt u zich af na het lezen van deze inleidende woorden. Nu, gelukkig was er toch wel het nodige te zien en te koop. Afgezien van oudere producten, zoals de verkoop van de software van *DeltaSoft* en van *TeddyWareZ*, presenteerde *R. SoftCorp 2000* een netwerk adapter, waarmee MSX-en met elkaar een netwerk kunnen vormen. Het project is nog in ontwikkeling, maar belooft best een interessante ontwikkeling te worden. Of deze netwerkkaart kan concurreren met de eveneens nog nieuw te ontwikkelen netwerkkaart van *Sunrise*, die gebaseerd zal worden rond de nieuwe eZ80 chip van Zilog, is nog de vraag.

Op de stand van *R. SoftCorp 2000* lagen overigens een hele hoop blauwe





staafjes, die mij deden denken aan het spel *Mikado*. — Of het ook die functie had is mij echter niet duidelijk. Positief dat er nog mensen

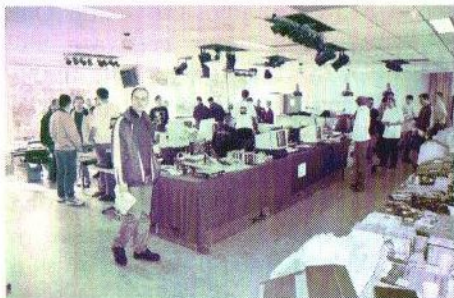
zo met hun MSX omgaan, dat ze tijd en energie vrijmaken om dergelijke zaken te ontwikkelen. We hopen u op de hoogte te houden van deze ontwikkeling.

Natuurlijk was *Sunrise* ook weer aanwezig en men had een nieuwe lading MoonSound muziekuitbreidingen gemaakt, die er mede dankzij de nieuwe PCB lay-out erg fraai uitzien. Ook kon men hier terecht voor de spellen van het *Gamesabonnement*, die ondanks de al soms vrij gedateerde releasedata toch nog erg gewild blijken te zijn.

Helaas kon ook *MSX NBNO* geen nieuw nummer van *XSW Magazine* voor de beurs uitbrengen. Of het niet uitkomen van de MSX bladen voor deze beurs van invloed is geweest op het bezoekersaantal, is natuurlijk niet met zekerheid te zeggen, maar het zal zeker een rol gespeeld hebben. Naast de gebruikelijke *NV-magazines* uit Japan en reeds eerder uitgebrachte software, zoals *Final Bout* en *Gradius Legends* van *Imanok* had *MSX NBNO* niet veel nieuws te bieden.

MSX Club West-Friesland was zoals vanouds weer van de partij en degenen, die nog iets zochten op hardwaregebied hadden grote kans om dit op deze stand te vinden. Naast hen was een gezamenlijke stand ingeruimd voor *MSX.org* en *MSX2.com*, die de laatste tijd aardig aan de weg timmeren wat betreft de MSX promotie op Internet. Men kon achter een stuk of wat computers plaats nemen om meer te weten te komen over de sites, die door hen onderhouden worden. Volgens het beursprogramma schijnt men ook nog een PX-V60 MSX 1 met laserdiskspeler meegenomen te hebben, maar deze heb ik niet kunnen ontdekken.

Ook *MSX vereniging "De Amsterdammer"* had een stand en promootte zijn club en verkocht diverse software met onder andere utils voor *Dynamic Publisher*. *The New Image* was present en toonde hun nieuwe MSX projecten, waaronder een cross-assembler, waarmee op de PC programma's voor de MSX gemaakt kunnen worden. Ook *TeddyWareZ* is met een assembler bezig onder de naam *Chaos Assembler* en zij toonden de nieuwste versie 3.0. Hopelijk dekt de vlag de lading in dit geval niet en kan er netjes en ordelijk met





Al met al mogen we toch wel stellen, dat het ondanks de weinige nieuwe soft- en hardware toch een geslaagde beurs was. Het is toch ook zeker belangrijk om de onderlinge contacten weer eens op te poetsen en hieruit komen vaak de mooiste samenwerkingsverbanden voort, die het doel hebben om MSX in stand te houden. Veel werd er overigens gepraat over de MSX

dit programma gewerkt worden. Mocht iemand zich geroepen voelen om dit programma voor ons te testen, dan kan er contact opgenomen worden met ons via de telefoonnummers die in de colofon vermeld staan.

Denyu Land beurs in Japan, die op dezelfde dag gehouden werd en vermoedelijk het nodige nieuws zou brengen aangaande de plannen om een nieuwe MSX compatible te lanceren.

In de kamer waar de stand van onze eigen *Vereniging voor computergebruikers Landgraaf*, het overkoepelende orgaan van het *MSX-Info Blad* en *Totally Chaos Team*, was gevestigd, was *S.G.G.A.* in de weer om zijn overvloedige MSX spullen te verkopen. Wij hadden zelf de primeur met de release van het spel *The best of Hamaraja Night 2*, dat door *Mark Jelsma* is geconverteerd naar de MSX2. Helaas is het spel qua functionaliteit nog niet gelijk aan het origineel, maar maken de nieuwe spelstructuur, de prachtige verpakking en vooral de Moon-Sound ondersteuning het spel zeker de moeite waard. Ook verkochten wij de nieuwste versie van *PLR* en onze oude soft- en hardware en tevens demonstreerden wij de nieuwe *LPE Z380* cartridge en *EV4* slotexpander van *Leonardo Padial*.

De Bussum beurs was en is nog steeds een goed voorbeeld dat er ook in Nederland nog steeds aan MSX promotie wordt gedaan en dat is natuurlijk prijzenswaardig. Hopelijk komt er ook volgend jaar weer een beurs in Bussum en kunnen we dan rekenen op een wat hoger bezoekersaantal.

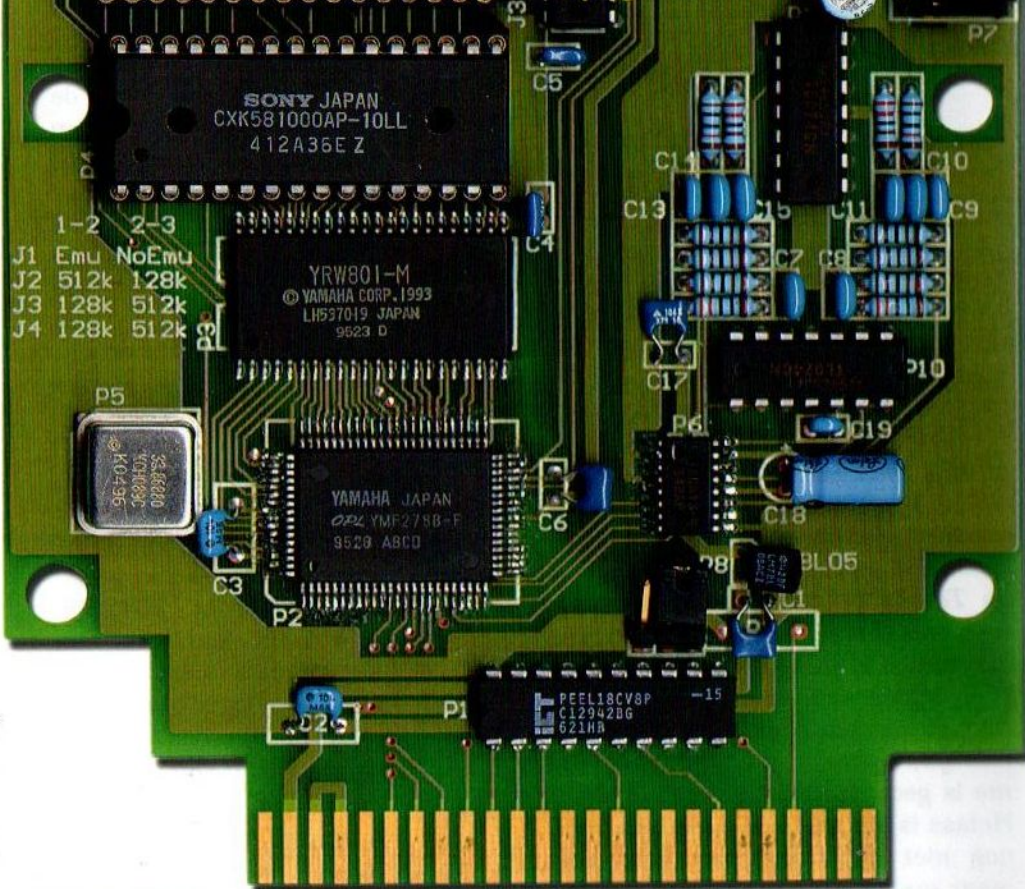
Gerrit van den Berg



DE MOONSOUND IS WEER VERKRIJGBAAR!

MSX MOONSOUND 2.0
(P) Sunrise Swiss 1995

Deze inmiddels beroemde geluidskaart met OPLL4 chip beschikt over 18 FM en 24 Wave kanalen en wordt geleverd inclusief behuizing, software en AUDIO kabel. Er is een beperkte oplage dus wees er snel bij!



Prijzen:

MoonSound 128 kB	HfI. 379,- (€ 171,98)
MoonSound 512 kB	HfI. 439,- (€ 199,21)
MoonSound 640 kB	HfI. 489,- (€ 221,90)
MoonSound 1024 kB	HfI. 579,- (€ 262,74)
(alle prijzen excl. HfI. 18,50 (€ 8,39) verzendkosten)	
Rembours kost HfI. 33,50 (€ 15,20)	

Sunrise for MSX

Postbus 61054

2506 AB Den Haag

Telefoon: (070) 360 97 07

(tussen 22:00 uur en 24:00 uur)

E-mail: thiep@msx.ch

Internet: <http://www.msx.ch>

