

Stichting  
MUSEUM.NL

# POKEY'S MAGAZINE



57

maart 1997

# STack BBS

- Michitron Bulletin Board System (MBBS), in beheer van STichting Algemene Computer Kenners (Stack).
  - Sysops *Bo Schreurs* en *Ernest Schreurs*.
  - **24 uur** per dag bereikbaar, **7 dagen** per week, **365 dagen** per jaar.
  - **45 minuten** toegangstijd per gebruiker per dag.
  - **GEEN** verplichting om bestanden aan te leveren (**geen upload verplichting**).
  - **GEEN** beperking ten aanzien van het overhalen van bestanden (**geen download beperking**)
  - Het **STack BBS** is geheel **GRATIS!**
  - Houd wel rekening met de normale PTT tarieven, interlokaal bellen is gauw een tientje per uur.
  - **Registratie** gebeurt zo *snel mogelijk*, na opgave van uw *gehele voornaam* en *achternaam* en volledige *adresgegevens*. Alleen ter controle wordt ook uw *telefoonnummer* (spraak) gevraagd.
  - Wij hebben **Public Domain** en **Shareware** software online voor de navolgende systemen:  
**Atari** computersystemen (ST, ST(e), Mega ST(e), TT en Falcon en de Portfolio)  
**Atari 8 bit** computersystemen (XL/XE, XEGS)  
**IBM** en compatible computersystemen  
**Macintosh** computersystemen
  - Wij zijn bereikbaar met de volgende modem instellingen: **8 databits, no parity, 1 stopbit**.

Telefoonnummers	Baudrates	
073 - 6122766	300, 1200, 1200/75, 2400, 9600 V32	bps
073 - 6147837	300, 1200, 1200/75, 2400, 9600 V32, 14400	bps
073 - 6147838	1200, 2400, 9600 HST	bps

  - Wij hebben **44 CD-ROMs** online! Onder andere:  
**Shareware Tiger, Powerpak PCX Potpourri, Dr. Reference, Software Vault Games, Cica Windows, Science Fiction, So much Shareware Two, Tech Tools, PCSIG 11th Edition, Garbo, Windows GOLD, C Users Group Library, SIMTEL 20, MEGA Shareware Win/OS2, Shareware Studio, Phoenix v3, C Sources/Pictures, Shareware Bonanza 1 t/m 3, World Vision 1 t/m 3, Noight Owl's v7, Atari MEGA Archive, Atari CD Walnut Creek, Atari Bernd Lohrum Disk 1 en 2, Atari XL/XE Pooldisk**, en natuurlijk hebben wij ook de nodige erotische plaatjes online.
-

# INHOUD



## Colofon

Pokey's Magazine is een uitgave van Stichting Pokey. Stichting Pokey is een geheel onafhankelijke gebruikersgroep met als doel het omgaan met Atari XL/XE computers te stimuleren en te vereenvoudigen. Stichting Pokey staat ingeschreven bij de kamer van koophandel te Rotterdam onder nummer S 131831.

Voorzitter: Erik Stok  
Secretaris: John Maris  
Penningmeester: Fred Meijer

Men kan donateur worden van de Stichting door het elders in dit blad geplaatste aanmeldingsformulier in te vullen EN de donatie over te maken.

## Pokey's Magazine

**Redactie** : Fred Meijer  
John Maris  
Erik Stok  
**Drukker** : The printer  
**Opplage** : 50

Stichting Pokey is niet verantwoordelijk voor eventuele schade van wat voor aard dan ook, welke voortvloeit uit artikelen, listings van programma's of hardwaretoepassingen, zoals gepubliceerd in Pokey's Magazine. Ingezonden artikelen en brieven kunnen door de redactie worden ingekort. Redactie-adres: Postbus 798, 3100 AT Schiedam.

© Copyright Stichting Pokey. Geen van de artikelen, listings of andere informatie mag worden overgenomen op wat voor manier dan ook, zonder schriftelijke toestemming van Stichting Pokey.

## ALGEMEEN

Van de redactie .....	4
U schreef .....	5
Wie zijn er nog? .....	33
Het hobby-isme is terug	
Afscheidswoord John Maris .....	35

## SERIE

BBS-sen in Nederland:	
STack .....	21
Waargebeurde verhalen .....	30

## PROGRAMMEREN

Gestruct. programmeren .....	6
Een kids-controller en dan? .....	24

## HARDWARE

ST Muis aan 8-bit Atari .....	17
GTIA gedoe .....	29
Muiskabel ergenissen .....	31

Dit maal een extra dik nummer van maar liefst 36 pagina.

Bij dit nummer vindt u geen programmadiskette, maar als extraatje 10 lege diskettes.

Veel plezier met dit magazine!

Na bijna zes jaar Pokey's Magazine valt nu toch het doek voor de Stichting. Voor u ziet u het allerlaatste nummer van het Pokey's Magazine. De redenen waarom wij stoppen met de uitgave van het magazine heeft u al kunnen lezen in de brief die u vorige week ontvangen heeft. Voor diegenen die die brief om een of andere reden niet ontvangen hebben, zal ik het nog even kort samenvatten.

In de afgelopen zes jaar is het aantal donateurs teruggedaan van zo'n 250 naar een kleine 50. Dat is logisch omdat de interesse voor Atari 8-bit computers in 1990 nog redelijk groot was. De SAG (Stichting Atari Gebruikers) mocht destijds een vrij grote groep 8-biters tot haar donateursbestand rekenen. Toen de SAG er dan ook mee op hield maakte een redelijk aantal donateurs haar overstap naar de ABBUC en/of Stichting Pokey. Echter vanaf 1990 is het verder alleen maar bergafwaarts gegaan met de XL/XE computer. In het westen werd nauwelijks nog iets meer uitgebracht en de omzet van de produkten die wel werden uitgebracht vielen erg tegen. De Atari kwam in een neergaande spiraal terecht die niet te stoppen was.

Dit klinkt allemaal erg negatief, maar laten we het eens van de andere kant bekijken. Wat hebben we in die afgelopen zes jaar nog bereikt? In totaal zo'n 57 (!) magazines, twee door ons zelf georganiseerde beursdagen, import van software uit Polen, Duitsland en het stimuleren van Nederlandse produkten (mede door ANG Software), opkoping van 17 pallets Atari spullen (mede door Bo en Ernest Schreurs) en het stimuleren van Atari 8-bit activiteiten in Nederland in het algemeen. Ja, in die jaren hebben we toch bereikt, dat de Atari XL/XE langer levend werd gehouden dan (zeker voor Atari zelf) de bedoeling was. Ik vind dat we met z'n allen een enorme pluim verdiend hebben!

Maar hoe moet het nu verder? Het ligt nu aan de gebruikers om de Atari 8-bit computer levend te houden. In dit magazine wordt via advertenties en artikelen extra aandacht besteed aan de gebruikersgroepen, bbs-sen en tijdschriften die nog bestaan. Misschien zit daar iets voor u tussen.

Rest mij u nog veel plezier te wensen met dit laatste magazine. Hopelijk zien wij elkaar op de 'Pokey afscheidsdag' en zo niet, dan komen wij elkaar vast nog wel eens tegen tijdens andere Atari-gelegenheden, want ook voor ons blijft de hobby voortbestaan!

*Fred Meijer,*

Mede namens alle bestuursleden van Stichting Pokey

# U schreef . . .



De laatste brievenbus van Pokey's Magazine. De afgelopen tijd (!) hebben ons nauwelijks brieven bereikt. De paar brieven die we kregen waren van (nog) niet-donateurs. Aangezien dit de laatste brievenrubriek is, heeft het geen zin deze brieven hier te behandelen. Welke behandelen we wel? Dat leest u hieronder!

*"Aan de makers van Pokey's Magazine*

*Hierbij wil reageren op jullie oproep voor een landelijke Atari dag in Nederland. Ik hoop dat er voor zo'n dag genoeg animo is en dat jullie deze dag kunnen organiseren.*

*Tevens wil ik vragen of jullie een adres weten, waar ik de CD-speler voor de Jaguar kan kopen.*

*Met vriendelijke groet,*

*Martijn Brinkman, Leeuwarden."*

**RED:** Ja er komt toch een soort beursdag. Namelijk het 'afscheidsfeest' van Stichting Pokey. Meer informatie over deze dag kun je lezen in dit magazine.

Op de vraag, waar een Jaguar CD speler te koop is, kunnen wij een kort antwoord geven. Wij zijn op de hoogte van 1 adres die de speler verkoopt. Er hangt echter wel een prijskaartje aan van f 399,-

Het adres:

Future Zone  
West Kruiskade 48  
3014 AT Rotterdam

Er schijnt ook een Jaguar gebruikersgroep in Nederland te zijn, die de CD-spelers aanbiedt voor fl. 249,- inclusief een aantal games. Hierover werd gesproken op het bbs Becotel. Daar kun je waarschijnlijk ook meer informatie krijgen.  
Tel. 020-6326859 na 18.00 uur met een modem.

*" Hoi*

*Hierbij heb ik een artikeltje voor Pokey's Magazine (...).*

*Ik ben zelf geen donateur van Pokey meer, omdat ik mijn Atari 800XE kwijt wil, zodat ik wat geld heb voor een PC-upgrade. Sorry dat ik jullie 'verraad', hoewel ik nog steeds een 600XL heb :-)*

*Ik zal altijd wel een beetje Atari minded blijven, maar de computers zal ik niet gauw meer aanschaffen, omdat ik er toch niks mee doe en dan komt er toch maar een dikke laag stof op en dat is ook niet de bedoeling.*

*Ik wil jullie heel veel succes wensen met Pokey en geef alsjeblieft niet op, want dan is Atari Nederland pas echt dood en dan heeft de C64 toch nog gewonnen!*

*Wietze de Boer, Drachten."*

**RED:** Ik hoop dat je met het bedrag dat je Atari je oplevert een PC-tje kunt kopen, hoewel ik daar aan twijfel...

Dank voor de aanmoediging, echter wij geven het nu ook op. De groep gebruikers is zo verschrikkelijk klein geworden en de groep actieve gebruikers (de dikke laag stof?) is op een hand te tellen. Helaas de C64 heeft gewonnen...

Op de valreep kregen we bij de reserveringsbonnen een briefje van Ria van Hoorn.

*"Ik vind het jammer dat ik niet kan komen. Verder wens ik jullie al het goede toe en bedank ik jullie voor de fijne jaren met de Atari.*

*Jammer dat het gestopt wordt.*

*Met vriendelijke groeten,  
Ria van Hoorn"*

En daarmee wil ik graag deze laatste 'U schreef' mee eindigen.

# GESTRUCTUREERD PROGRAMMEREN

*Door B.F. Schreurs*

Waarschijnlijk ben ook jij een van de velen die zichzelf heeft leren programmeren. Wellicht eerst door het intypen van listings uit een boek of tijdschrift, en later door het zelf bedenken van programma's. Afhankelijk van je achtergrond breng je al dan niet structuur aan in je werkwijze en/of programma's. Wellicht heb je zelf ook wel eens ervaren, dat als je in het "wilde weg" gaat programmeren, je vroeg of laat spaghetti code overhoudt, anders gezegd, code die op dat moment misschien wel werkt, en waarvan je ook nog wel weet wat het doet en moet doen, maar dat niet onderhoudbaar is en zeker niet als je er een jaar later nog eens een blik op werpt. De kans is dan groot dat jij, als maker van het programma, niet meer weet wat het programma doet, dan wel moet doen.

Die ervaring heeft de computerindustrie natuurlijk ook al lang, en vandaar dat er richtlijnen zijn om programma's te ontwikkelen. Laat ik me beperken tot een paar aanwijzingen en hints, ook zelf door ervaring opgedaan overigens. Gebruik de tips, of niet, maar lees ze in ieder geval eens een keer door, en doe er je voordeel mee. Wellicht zit er toch iets bruikbaar tussens. De meeste opmerkingen gelden namelijk ongeacht de programmeertaal waarin en/of computer waarop je ontwikkeld.

## *HOE TE BEGINNEN*

Stel dat je ineens een briljant idee krijgt voor een programma. Voordat je het programma schrijft, onderzoek dan eerst of er al niet een

dergelijk programma op de markt is. Waarom zou je iets maken als er al een vergelijkbaar programma is, welke je al meteen kunt gebruiken. Je kunt bijvoorbeeld het zoveelste menu programma maken, maar er zijn er al zoveel! Indien het programma alleen voor eigen gebruik is, of als het alleen maar dient om meer te weten te komen over het gebruikte systeem, dan maakt dat laatste natuurlijk niet uit. Heeft het programma echte noviteiten, dan wordt het natuurlijk weer wel interessant. Zorg er dan voor om de leuke dingetjes van die andere programma's ook op te nemen in jouw programma, dat maakt jouw programma nog interessanter. Let wel, het "stelen" van ideeën van andere programmeurs, is heel gewoon. Waarom zelf het wiel uitvinden niet waar? Ook het kijken naar andere programma's brengt je zo nu en dan op ideeën. Je denkt namelijk altijd: "Dat kan ik beter". Aan jou om te bewijzen dat dat zo is.

## *DE ONTWERP FASE*

Je hebt een idee voor een programma, het programma bestaat nog niet, en wat dan? Begin dan eerst met om op papier een schematisch ontwerp van je programma te maken. Nu blijkt dat je dit het beste kunt doen vanuit de praktijk. Met andere woorden, wat krijgt de gebruiker van je programma te zien, moet deze met je programma communiceren, zijn er externe factoren die op je programma van toepassing zijn (datum, tijd, disk, printer), en moet het programma overdraagbaar zijn (moeten anderen weten hoe het programma werkt). Van belang is ook met welke hulp-

middelen je gaat werken en/of programmeertaal. Als je bijvoorbeeld een programma ontwikkeld voor een Atari 8-bit computer en je maakt gebruik van een BASIC XE cartridge, dan kunnen gebruikers je programma alleen maar draaien als ze zelf ook beschikken over een BASIC XE cartridge. Laten we beginnen met wat de gebruiker te zien krijgt. In de meeste gevallen zal dat een scherm zijn (al dan niet vergezeld van geluid). In de praktijk blijkt dat een programma door een gebruiker beoordeeld wordt op zijn uiterlijk. Het kan ook meestal niet anders, want dat is meestal het enigste wat een gebruiker te zien krijgt. Dus besteedt nogal wat aandacht aan dit onderdeel. Zo ken ik het verhaal van een bedrijf die een pakket had overgenomen dat alleen gebruik maakte van 24 x 80 koloms ASCII schermen. Er werd geen gebruik gemaakt van een grafische gebruikers schil. Uiteindelijk was dat bedrijf dan ook zeer teleurgesteld. Dat blijkt ook wel een beetje uit de opmerking van een van de Amerikaanse eigenaars: "WHAT? It doesn't have a GUI???" GUI staat voor Graphical User Interface. Een grafisch scherm ziet er namelijk altijd gelijker uit dan een gewoon ASCII scherm. Op een seminar voor programmeurs werd hierover opgemerkt: "A GUI is just like sex, it sells!". Ik bedoel maar.

Zelf kom ik een beetje uit de mainframe wereld, en daar wordt nog steeds gebruikt gemaakt van ASCII schermen, zelf heb ik daar geen moeite mee. Mij gaat het meer om wat een programma doet, en niet hoe het er uitziet. Maar een feit blijft, dat je programma er wel op beoordeeld wordt. Volgens ontwerpers wordt heden ten dage bij het ontwikkelen van programma's in een visuele programmeertaal (zoals Visual Basic), meer dan 80 procent van het programmeren besteedt aan het ontwerpen van de schermen. De rest wordt automatisch gegenereerd, of behoeft slechts kleine aanpassingen in de programma code. Was het vroeger normaal dat een gebruikers alles moest intypen, tegenwoordig mag die de waarde van velden waar een tabel achter zit, selecteren uit een veld selectie box. Dit geldt ook voor de invoer zelf, veel ge-

vensinvoer gaat ervan uit dat er van een muis gebruik gemaakt wordt. Maar daar zitten ook nadelen aan, het kost geheugen, snelheid, en het vereist een standaard wijze van programmeren. Veel software fabrikanten leveren dan ook kant en klare interfaces die je alleen aan hoeft te roepen, waarna die dan diverse GUI taken voor je uitvoeren. Deze interfaces heten API's (Application Program Interface). Mijn advies is om het gewoon allemaal in ASCII te houden, want de meeste mensen zijn uiteindelijk toch het meeste geïnteresseerd in een goed werkend programma en niet zijn sexy body. Er is zelfs een bedrijf wat een conversie programma levert wat ASCII schermen omzet naar grafische schermen, dus doe niet moeilijk, houdt het simpel!

Soms zit het tegen, en beschik je over heel weinig hulpmiddelen, en zul je zelf eerst alles moeten ontwikkelen. Probeer dan eerst na te gaan of anderen je wellicht aan goed hulpmiddelen kunnen helpen (eventueel tegen betaling). Wellicht hebben anderen hetzelfde probleem ondervonden en al iets ontwikkeld, je kunt tenslotte nooit weten.

**SCHEM LAYOUT EERST ONTWERPEN**  
Om jezelf ook een indruk te geven van hoe een en ander er uit komt te zien, kun je het beste als eerste met je scherm layout beginnen. Zet het op papier, of ontwerp het meteen op de computer. Je krijgt op deze manier al snel door of je aan 1 scherm genoeg hebt, en of er nog een en ander aan mist. Zolang er nog geen letter geprogrammeerd is, is het aanbrengen van wijzigingen op een dergelijk moment kinderspel. Hoe zet je zo'n scherm op, en wat moet er op staan. Een scherm dient meestal voor twee zaken, of je wilt de gebruiker iets mededelen, of je wilt dat deze wat vragen van jou beantwoordt. Aan de hand van deze "vragen" kan de gebruiker het programma sturen. Laten we een heel simpel voorbeeld nemen. Je wilt dat de gebruiker een datum moet ingeven, en dat jouw programma dan teruggeeft wat voor dag het is (maandag, dinsdag, etc.). Je zou dan kunnen volstaan met 1 veldje op het scherm, met een

vraagteken er voor of er achter. Je begrijpt dat een dergelijk scherm de gebruiker niks zegt. Je moet er dus een tekst voorzetten, bijvoorbeeld "Geef een datum op: ". De gebruiker weet dan wat die moet doen, maar dan nog niet hoe die het moet ingeven. Als je de tekst "(dd/mm/jjjj)" er ook nog bij vermeldt wordt het al een stuk duidelijker. Maar ja, dan weet die nog niet waarom die een datum moet ingeven. Het doel van het programma moet je dan ook altijd op het scherm vermelden. Een en ander ziet er natuurlijk nog beter uit als er ook diverse helpschermen bij zijn voor de invoer velden en/of doel van het programma. Wees voorzichtig met het gebruik van kleuren. Gebruik er niet teveel, anders begint het scherm teveel op een mozaïek te lijken.

Als je kleuren gebruikt, zorg er dan voor dat ze functioneel zijn, bijvoorbeeld een rode achtergrond indien het desbetreffende veld fout is ingevuld, of rode letters indien het een foutmelding betreft. Het gebruik van kleuren is niet echt noodzakelijk, in veel programma's worden fout ingevulde velden aangegeven door het veld te laten knippen, foutmeldingen worden bijvoorbeeld helderder weergegeven dan normale tekst. Bedenk tenslotte dat er complete cursussen worden gegeven in het verantwoord gebruik van kleuren. Als het kan houdt dan een "vaste" indeling in van je schermen door alle programma's. Bijvoorbeeld de plaats van je foutmeldingen, vaste kleuren voor bepaalde acties, de plaats waar je aangeeft welke toets welke actie tot gevolg heeft. Dit is slechts een advies, bij het ontwerpen van utilities en games gaat zoiets natuurlijk niet op.

Let ook op welke je taal je gebruikt op je scherm. Als je Nederlandse teksten gebruikt, limiteer je je programma tot de Nederlandstalige markt. Engels wordt al door meer mensen begrepen. Houdt er wel rekening mee dat er nogal wat mensen zeer kritisch kunnen zijn op dit punt. Veel Duitsers verwachten bijvoorbeeld dat het programma ook met Duitse tekst beschikbaar is, anders kunnen ze er niet mee over weg. Franstaligen hebben daar ook

een handje van. Bij professionele programma's is het zelfs een vereiste dat de tekstschermen zijn aangepast aan de landstaal waar het programma gebruikt gaat worden. Programma's op grote computers zijn meestal voorzien van een landtaal code, door het wijzigen van de code schakelt het programma automatisch over op de gewenste taal, en krijgt de gebruiker dan ook schermen te zien die hij "begrijpt". Veel spelletjes laten je tegenwoordig tijdens het installeren van het spel kiezen uit een aantal gangbare talen. Als je het scherm eenmaal ontworpen hebt, kun je de rest van het logisch ontwerp van je programma gaan maken.

#### LOGISCH ONTWERP PROGRAMMA STROOMDIAGRAMMEN

Hoewel het misschien teveel van het goede is voor simpele programma's, adviseer ik toch om eerst je programma via programma stroomdiagrammen op te zetten. Via deze diagrammen kun je relatief snel zien of er een logische denkfout in je ontwerp zit. Ook kun je aan de hand van de diagrammen snel bepalen welke onderdelen moeten worden ondergebracht in subroutines. Bij grote projecten kun je zien waar het onderdeel waar jij mee bezig bent past in het geheel. Het kan verder gebruikt worden als technische documentatie.

Een plaatje zegt meer dan duizend woorden, en dat geldt zeker voor stroomdiagrammen. Voor simpele programma's zou ik de diagrammen ook wel maken, maar dan snel en wellicht op een kladblaadje. Een programma stroom diagram geeft de gevolgen weer van allerlei acties en reacties binnen je programma. "waar moet er een vraag gesteld worden aan de gebruiker", "wat moet de computer doen na een reactie van de gebruiker", "moet er iets op het scherm getoond worden", "moet er een bestand gelezen worden", dat soort acties worden allemaal beschreven in het diagram. Daar er complete cursussen worden gegeven in deze techniek zal ik hier niet verder over uitwijden. Als er belangstelling voor is, wil ik wel eens een artikel aan wijden. Zodra je het diagram opgezet hebt, kun

je beginnen met het programma.

### *STRUCTUUR VAN HET PROGRAMMA*

Let wel, in de vorige regel stond niet "kun je beginnen met programmeren", zover zijn we nog steeds niet. We willen namelijk het programma gestructureerd op gaan zetten. Hiertoe willen we het programma gaan opdelen in subroutines. Iedere subroutine van het programma voert een specifieke taak van het programma uit. De schermafhandeling zal bijvoorbeeld zo'n subroutine zijn. Ook het initialiseren van variabelen zal uit een subroutine bestaan. Daar kan het programma stroomdiagram je prima bij helpen. Aan de hand daarvan kun je de "structuur" of "layout" van je programma opzetten. Dit hoef je in principe maar eenmaal te doen, de andere programma's die je in de toekomst ontwikkeld, kunnen namelijk allemaal gebruik maken van deze "body".

Kopiëren is een van de bezigheden die een programmeur het meeste verricht. Ik heb wel eens ergens gelezen dat 90% van de handelingen die men verricht tijdens het programmeren bestaat uit copieren van code en/of manipuleren van data gegevens. Reserveer de top van je layout voor: het volgende, doel van je programma, auteur van het programma, restricties die van toepassing zijn (of geef aan dat het Public Domain is), de creatiedatum, versienummer van het programma, en reserveer een paar regels voor commentaar inzake eventuele toekomstige wijzigingen. Plaats vervolgens programma variabelen aan het begin van je programma. Plaats voldoende commentaar bij variabelen die niet voor zichzelf spreken. Gebruik indien mogelijk, lange variabele namen. Een variabele genaamd "SV" zal niet direct door iedereen herkend worden als "STATUS-VELD". Dwingt de computeromgeving je tot korte namen, documenteer de gebruikte afkortingen dan in ieder geval. Indien mogelijk moet je gebruik maken van standaard veldbenamingen. Je kunt dan ook meteen de standaard afkortingen bepalen, het is namelijk niet al-

tijd mogelijk veldnamen altijd volledig uit te schrijven, en vaak maken hele lange veldnamen het programma niet leesbaarder. Gebruik bijvoorbeeld BEST voor BESTAND, of DIR voor DIRECTORY, of SA voor SCHERM-A. Zo kun je heel eenvoudig een veld wat op het eerste scherm wordt gebruikt, en wat verwijst naar een waarde die wordt gevuld vanuit een directory als SA-DIR-FILENAAM benoemen. Standaards maken het programmeren in die zin eenvoudiger, dat je op een gegeven moment vrij snel een variabele herkent, en ook kunt zien waar deze gebruikt wordt (het veld wat naar scherm A verwijst bijvoorbeeld). Natuurlijk kun je nooit van tevoren ALLE veldnamen al bepalen, dat is ook niet nodig, als je deze maar wel groepeert in een plaats. Ik heb wel eens in een werkgroep gezeten die er vele weken over deed om tot standaard veldbenamingen te komen. Kun je nagaan hoeveel belang sommige bedrijven hier aan hechten. Soms voorziet de fabrikant of een goed systeem (hand)boek je van standaard namen, controleer of dat ook op jouw computer van toepassing is. Na de variabelen komt de eigenlijke code van het programma.

### *STRUCTUUR VAN DE UITVOERBARE CODE*

Het uitvoerbare gedeelte van het programma gaan we opdelen in subroutines. Iedere subroutine laten we een specifieke taak uitvoeren. Dit maakt het onderhouden van het programma eenvoudiger. Ook het er uithalen van fouten ("debuggen") wordt simpeler. Als je namelijk zeker weet dat een subroutine bugvrij is en goed werkt, heb je daar geen omkijken meer naar. Zelfs niet als we het kopiëren naar een ander programma. Hoe gaan we die subroutines echter opzetten, hoeveel moeten het er zijn, en hoe moeten ze heten? Dat is allemaal afhankelijk van wat het programma moet gaan doen. Laten we ons datum programma voorbeeld weer even aanhouden. Je moet eigenlijk net als een computer gaan denken. Je zult de allereerste keer het datum veld moeten initialiseren, vervolgens de gebruiker vragen een datum in te vullen, de

datum controleren, eventueel een foutmelding verzorgen, en bij een korrekte datum de dag uitrekenen, en vervolgens het antwoord op het scherm tonen. Onafhankelijk van de programmeertaal waar je in schrijft kun je deze methode toepassen. In BASIC krijg je bijvoorbeeld iets als:

```
1000 REM ***UITVOERBARE CODE***
1010 GOSUB 2000 :REM initialisatie
1020 GOSUB 3000 :REM datum invoer
1030 GOSUB 4000 :REM datum controle
1040 IF FOUT > 0 GOTO 1020
1050 GOSUB 5000 :REM dag uitrekenen
1060 GOSUB 6000 :REM dag tonen
1070 GOTO 1020
```

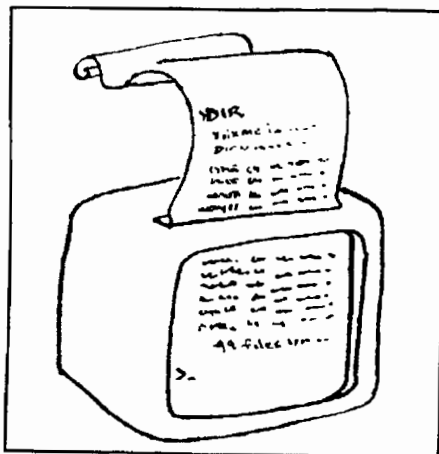
De tekst achter het REM statement is het commentaar gedeelte. Ik geef toe dit programma wel sneller zou kunnen werken als je niet zo gestructureerd werkt, maar het zal er zeker niet duidelijker op worden. Bedenk ook dat er meerdere oplossingen zijn voor een probleem, meerdere wegen leiden tenslotte naar Rome, gefit dus in je commentaar aan waarom je deze oplossing gekozen hebt. In een dergelijk simpel programma spreken een heleboel zaken wellicht nog wel voor zich, maar hoe complexer het programma, hoe minder duidelijk de verbanden kunnen worden. Zeker als het programma ook nog opgesplitst wordt in meerdere programma's. Er zijn bijvoorbeeld programma's van wel een miljoen regels code, welke onderhouden moet worden door meerdere programmeurs. Zonder standaards en zonder structuur zou dit niet mogelijk zijn.

Op computers waar geheugenlimieten een rol spelen, zal blijken dat je slechts hele kleine programma's zult kunnen schrijven als je gestructureerd programmeert, en dat ze ook niet echt snel zullen draaien. Maar in veel gevallen zal de beschreven methode wel tot tevreden resultaten leiden.

### STRUCTUUR VAN DE CODE ZELF

Een aantal belangrijke tips! Gebruik geen magische nummers! Gebruik liever veldnamen, ook als het hier een constante betreft.

Ook niet als je deze variabele maar 1 maal gebruikt! Stel je hebt een veld nodig in je programma die vasthoudt uit hoeveel sectors een diskette bestaat. En stel dat je de grootte van de disk in bytes al voorhanden hebt in het veld DISKSIZE. Dan kun je natuurlijk in je programma de volgende code opnemen: [SECTORCOUNT = DISKSIZE / 128], (de rechthoekige haakjes geven alleen het begin en eind aan van de code). Als dit slechts op 1 plaats gebeurt valt het nog wel allemaal mee, maar als je op diverse plaatsen in je programma door die 128 deelt, wordt het al wat lastiger. Want stel dat de sectorgrootte opeens niet meer 128 is, maar 256, 1024 of 2048. Als je dus had gedeeld door het veld SECTORSIZE, wat je de waarde 128 had gegeven, in plaats van het magische getal 128, had je bij een eventuele wijziging van de sectorgrootte slechts op 1 plaats de code hoeven aan te passen. Een compiler of interpreter controleert bovendien of een veldnaam bestaat, maar niet of de waarde van een variabele geldig is. Je kunt dus vele malen korrekert 128 hebben getypt, bij het schrijven van je programma, maar als je EENMAAL 127 typte, zit je toch met een aardige bug. Bovendien zegt een woord veel meer dan een getal. Het maakt de code van je programma wat groter, maar wel leesbaarder.





# ATARI XL/XE INFO OP INTERNET? JAWEL!



## ATARI 8-BITMAGAZINE

# FM

Hier vindt u al het laatste nieuws over uw favoriete Atari computer. Helemaal gratis, geen abonnementen, geen verplichtingen. Echt van iedereen en voor iedereen (met internet-toegang welteverstaan).

Deze pagina is verdeeld in 5 onderdelen:

- Atari nieuws
- Atari beurzen en gebruikersdagen
- Atari reviews
- Atari op het internet
- Atari shop

Nieuwsgierig? Pak uw favoriete browser en surf naar het Atari 8-bit Magazine!

## Het 8-bit adres:

# [www.meijer.demon.nl](http://www.meijer.demon.nl)

Voorals je gebruik maakt van een compiler zou ik dit doen, daar een compiler toch compacte code genereert. Ook al gebruik je een variabele maar eenmaal, gebruik dan toch nog liever een veldnaam. Zo zal een variabele genaamd PI, jou ook wellicht meer zegen dan alleen de waarde 3,14. Wil je absoluut een getal gebruiken, ga er dan vanuit dat het gebruikte getal alleen op die lokatie gebruikt wordt, bijvoorbeeld:  
[MOVE 19 TO EEUW : IF JAAR < 50 THEN MOVE 20 TO EEUW].

Je begrijpt dat je de voorgaande code niet kunt gebruiken bij geboortedatum, wanneer je het eeuwgedeelte automatisch wilt laten bepalen. Voorkom dat je buiten subroutines springt. Deze werkwijze maakt een programma erg onoverzichtelijk, en zeer moeilijk te debuggen. Bij veel bedrijven is dit ongeoorloofd "jumpen" zelfs verboden. Er zijn geniale programmeurs die "selfmodifying code" gebruiken, dat is code die tijdens het draaien van het programma, bestaande instructies in het programma, vervangt door compleet andere instructies. Zo zou in ons voorbeeld, afhankelijk van de foutwaarde in regel 1040 geen foutcontrole plaatsvinden, maar een REM statement geplaatst worden, een instructie die dus niets doet, of men zet er een GOTO 1020 statement neer, als FOUT een waarde is die groter is dan 0 (dit allemaal terwijl het programma draait!). Dit soort dingen is zeer moeilijk te debuggen. NIET DOEN dus, tenzij je jezelf tot die genialen rekent. Overigens kan het bovenstaande voorbeeld nog gestructureerder gemaakt worden. De vraagstelling [IF FOUT > 0] is op zich nog wel duidelijk, maar iets als [IF FOUT-GEVONDEN] is toch duidelijker. In sommige programmeertalen is dat mogelijk, in de meeste BASICs helaas niet.

### SUBROUTINES

Subroutines worden origineel gebruikt voor code die herhaalde malen moet worden uitgevoerd. De routine wordt vanuit een bepaald deel van de code aangeroepen, en uitgevoerd, waarna men weer terugkeert in de code op de lokatie NA het aanroepen van de subroutine.

De subroutine kan al dan niet parameter gestuurd werken, wat wil zeggen dat verschillende waarden, van velden die de subroutine gebruikt, kunnen zorgen voor een heel andere uitkomst. De uitkomst wordt dan via een resultaatveld teruggegeven. Wanneer we gestructureerd programmeren, maakt het ons eigenlijk niet uit hoe vaak een subroutine wordt aangeroepen. Logisch denken bepaald wanneer er iets in een subroutine wordt ondergebracht. Ons datum voorbeeld had in theorie uit 1 lang stuk code kunnen bestaan, met diverse [IF...THEN....] regels. Ook dit had gewerkt. Op een aantal belangrijke plaatsen had ook nog een [GOSUB...] kunnen staan. Maar in veel grotere programma's had een en ander al snel tot spaghetti code kunnen leiden. De gestructureerde manier van werken biedt hier voordelen. Wat en hoeveel moet er in een subroutine? Daar is niet echt een richtlijn voor te geven. Je kunt iedere aktie die moet worden uitgevoerd in een subroutine onderbrengen. Maar als je teveel subroutines maakt, bevordert dit de overzichtelijkheid weer juist niet. Je moet proberen een goede balans te vinden. Een goede analyse van het probleem kan hierbij helpen. In principe kun je zeggen dat de gedane analyse terug te vinden is in de gebruikte subroutines in de code! Probeer bij de analyse eens te denken wat de computer allemaal moet doen, en hoe de gebruiker moet reageren om tot het door jouw gewenste resultaat te komen. Schrijf deze stappen op, en ga na of je niet bepaalde controles, vragen, opdrachten, en dergelijke onder kan brengen in subroutines. Een programma wat hetzelfde doet als een subroutine (een specifieke opdracht uitvoeren), noemt men een subprogramma. Een programma kan bestaan uit meerdere subprogramma's, en een subprogramma kan weer bestaan uit een of meerdere subroutines. Een File Selector Box zou zo'n subprogramma kunnen zijn.

### SPRONGOPDRACHTEN

Een veel gebruikte functie is de spronginstructie. In niet zo gestructureerde programma's zul je zien dat er nogal gauw van hot

naar haar gesprongen wordt. Soms wordt zelfs een variabele gebruikt om te springen, naar een andere lokatie in het programma, bijvoorbeeld [ON WAARDE GOTO 100, 200, 300]. Zo op het oog zegt dit ons niet veel. Beperk het springen naar andere lokaties binnen een subroutine. Spring NOOIT uit een subroutine, ook niet via truuks. Dit werkt erg verwarrend, misschien niet voor jou, maar wel voor degene die later naar jouw programma kijken. Als je afhankelijk van een waarde wil springen, gebruik dan liever iets als

```
[IF WAARDE = KORTING-5 GOTO 100 :
IF WAARDE = KORTING-10 GOTO 200 :
IF WAARDE = KORTING-20 GOTO 300].
```

Het is iets meer tikwerk, maar wel meteen duidelijk waarom er naar een bepaalde lokatie wordt gesprongen. Er zijn bedrijven die een sprong instructie verbieden. Dit heet "GOTO LESS" programmeren. Ik ben daar niet zo'n voorstander van, omdat dit de programma's naar mijn idee weer minder leesbaar maken, daar men dan terug moet grijpen op ingewikkelder programma constructies. Nogmaals, zolang je voor jezelf programmeert bepaal jij wat wel en niet kan.

### **GEBRUIK VAN STANDAARD SYSTEEM ROUTINES**

Probeer zoveel mogelijk gebruik te maken van de standards die de omwikkeldomgeving je biedt. Daarvoor zijn een aantal redenen, als je bestaande systeemroutines vervangt door eigen ontwikkelde routines, en een andere programmeur komt in aanraking met je programma, moet deze eerst uit gaan zoeken hoe die routines van jou werken. Als je een systeemroutine gebruikt, weet de andere programmeur automatisch al hoe de vork in de steel zit, omdat deze ook gebruik maakt van de al aanwezige systeemprogramma's. Een andere reden is dat als de leverancier van jouw bedrijfssysteem (operating system), besluit om een en ander aan interne codering aan te passen, jijzelf daar weinig hinder van ondervindt, omdat de wijze van aanroepen (vectors, systeemadressen of parameter pas-

sing) van de routine toch niet wijzigt. Bij meer complexe operating systemen is dat nog belangrijker omdat dan jouw programma's lijken op door andere programmeurs ontwikkelde programmeur, waarmee de gebruiker dan weer vertrouwd is. Vooral bij Macintosh systemen is deze standaard wijze van programmeren heel populair. Een gebruiker hoeft daardoor de leer cyclus van een programma dan maar slechts eenmaal te doorlopen, om te "weten" hoe alle andere programma's werken. Die grafische popup schermen zijn ook gewoon standaard routines die men aan kan roepen. Bij het gebruik van standaard routines moet je overigens wel goed opletten welke routines je precies moet gebruiken, daar soms door de leverancier geadviseerd wordt bepaalde routines niet (meer) te gebruiken omdat deze vervangen zijn door andere systeemroutines. Houd je je niet aan dit advies kan het gebeuren dat je programma bijvoorbeeld op een bepaalde computer wel werkt, maar op een ander niet (omdat daar misschien wel een ouder operating systeem op zit).

### **BESTANDEN OF TABELLEN**

Er zijn diverse soorten bestanden, sequentiële, relatieve en geïndexeerde bestanden. Een bestand kan bovendien ook werken met gecomprimeerde gegevens. Afhankelijk van het computersysteem en/of programmeertaal, wordt een of meerdere van deze bestandsvormen ondersteund. Een middelgrote computer met een krachtige programmeertaal zal alle bovenstaande vormen ondersteunen, terwijl bijvoorbeeld een 8bit computer slechts sequentiële ongecomprimeerde bestanden ondersteunt. Als men bij zo'n 8bit dan met indexwaarden wil gaan werken, moet men toch al gauw terug gaan vallen op tabellen. Die tabellen kunnen dan natuurlijk wel van tevoren gevuld worden door een sequentieel bestand te lezen, en aan het eind van het programma deze tabel weer op te slaan in een sequentieel bestand. In dergelijke gevallen zou ik niet het operating systeem "herschrijven" om toch die andere bestandsvormen te emuleren. Een ander kan dan namelijk je

programma alleen maar gebruiken als die beschikt over hetzelfde operating systeem.

gelijke "overflow" situaties te testen.

Er zijn firma's die aparte "systeem" routines aanleveren, zoals bijvoorbeeld het pakket Btrieve, wat indexering ondersteund voor de PC, maar ook dan weer dient een "eindgebruiker" ook te beschikken over deze routines. Kortom, probeer op een kleine computer geen grote computer te emuleren. Meestal gaat het namelijk op die kleine systemen om relatief geringe bestands-groottes, en die passen makkelijk in een tabel. En voor tabelbeheer zijn de meeste programmeertalen uitstekend toegerust. Gebruik wel een standaardwijze voor het doorlopen en zoeken in tabellen. Ook hier geldt dat je moet proberen te voorkomen om magische getallen te gebruiken. Vraag dus niet [IF INDEX > 20 GOTO 5000: KORTING = PERCENTAGE (INDEX)],

Tabellen zijn ook een handige manier om vooraf berekende waarden in op te slaan. Zo zou je om 2 tot de macht 4, niet moeten gaan uitrekenen  $2 * 2 * 2 * 2$ , maar zou je gewoon de 4e tabelwaarde op moeten halen uit de 2e macht tabel, die de waarden: 2, 4, 8, 16, 32, enzovoorts heeft. Dat spaart je heel wat rekenwerk, en je programma is (dus) sneller. Wil je 2 tot de macht 0 ook ondersteunen, zul je 1 bij de tabelpositie op moeten tellen, voordat je de waarde ophaalt uit de 2e macht tabel.

maar [IF INDEX > INDEX-MAX GOTO 5000 : KORTING = PERCENTAGE (INDEX)].

### DEBUGGING, TRAPS, MONITORS

Het is makkelijk als je programmeer omgeving is voorzien van een debugger. Dit hulpmiddel helpt je bij het zoeken naar fouten. Je hoeft alleen maar aan je compiler op te geven dat de code die gegeneerd moet worden, voorzien moet worden van debug informatie. Deze informatie gebruikt de debugger om te kunnen werken met je programma. Deze debuggers zijn vrij complexe programma's met ontelbare commando's. Het leren omgaan met een goede debugger vereist ongeveer net zoveel tijd als het leren van een nieuwe programmeertaal. Als je over deze voorziening beschikt, kun je de rest van dit hoofdstuk overslaan. Je bent er nog, dus dat betekent dat je zelf een soort debugging faciliteit in je programma moet inbouwen. Je kunt dan werken met [PRINT...] of [IF...THEN...] opdrachten. Je krijgt dan wel bijvoorbeeld wel iedere keer de inhoud van de opgegeven variabelen te zien, zodra deze instructie "gepasseerd" wordt, maar het is in ieder geval een manier om fouten te kunnen traceren. Er zijn compilers die de mogelijkheid geven om conditioneel code in je uitvoerbare programma op te laten nemen. Zo kun je in de programmeertaal C opgeven: [IF DEBUG {printf("percentage = ", %PERCENTAGE)}]. In de rest van je source kun je meer van deze [IF DEBUG...] instructies plaatsen. Indien je nu aan het begin van je programma [DEBUG = FALSE] zet, zullen de [IF DEBUG...] instructies niet opgenomen worden in je

Ikzelf gebruik altijd een Nxx veld voor indexen, gevolgd door een logische naam, tenzij het een algemene index betreft. Je kunt in mijn programma's dus de veldnamen N01-MAAND, N02-DAGNAAM, of N98 en N99 tegenkomen. Aan die laatste 2 kun je dus niet echt zien, wat het precieze gebruik is van de index, maar dat blijkt wel uit het programma. Het doorlopen van tabellen dient altijd op een zelfde wijze te geschieden, je kunt zelf bepalen welke wijze voor jou het makkelijkste is. Maar werk deze wijze goed uit, zorg ervoor dat de code foutloos werkt, en Kopieer deze werkwijze door naar andere programma's om zo een standaard doorloop routine ervan te maken. Uiteindelijk voorkomt dit het maken van fouten. Je zult verbaasd zijn hoe vaak er vergeten wordt om het laatste veld uit een tabel te testen, alleen omdat de [IF...GOTO : A=B(N99)] vraagstelling niet goed was uitgewerkt. Of er wordt wel een tabel benaderd met een indexwaarde die wijst naar een veld buiten de tabel. Gelukkig kun je sommige compilers instrueren om op der-

uiteindelijke code, terwijl het met [DEBUG = TRUE], wel wordt opgenomen in je uiteindelijke code. Met deze DEBUG switch kun je dus eigenlijk de "debugger" aan en uit zetten. Een zelfde soort functionaliteit kun je ook wel proberen te emuleren in BASIC, door instructies als: [IF DEBUG = 1 THEN PRINT "percentage = " PERCENTAGE] alleen moet je dan wel bedenken dat dit altijd in je code staat. Zorg er ook voor dat er velden in je programma staan die de programmanaam en versienummer bevatten, en eventueel een datum. Doe dit niet in een commentaar regel. Als je namelijk een object programma bekijkt, en je weet niet zeker of dit de laatste versie is, kun je aan de hand van het zoeken naar programmanaam in je object code, en versienummer zien of het de meest recente is. Natuurlijk is het allemaal nog fraaier als je een fraai pop up scherm hebt, waarin al deze gegevens staan vermeldt. Zorg ervoor dat bij het uiteindelijk vrijgeven van je programma en/of source code alle debug voorzieningen zijn verwijderd. Onnodige code vertraagd het programma tenslotte, en gestructureerd programmeren drukt toch de snelheid al een beetje. TRAPS zijn moeilijker te emuleren, of misschien wel helemaal niet. Het mooie van een geavanceerde debugger is dat je een TRAP kunt zetten op het wijzigen van de inhoud van een variabele, een stuk geheugen, register, en nog meer van dat fraais. Je kunt er zelf complete instructies ingeven, zoals [IF PERCENTAGE > 10], het programma stopt dan als de waarde van PERCENTAGE groter wordt dan 10. Je kunt wel proberen die TRAP te emuleren door iets als volgt te gebruiken [IF WAARDE <> WAARDE-OUDE THEN PRINT...] waarbij WAARDE-OUDE vooraf gelijk is gemaakt aan WAARDE, maar waar moet je deze instructie in je programma zetten? Als je dat van tevoren weet, weet je waarschijnlijk ook waar de fout zit. Vooral bij hele grote programma's zal het je duidelijk zijn, dat je nooit van tevoren weet wanneer de inhoud van WAARDE verandert. En als je instructie het wel merkt, welke instructie heeft de inhoud dan verandert? Je ziet je eigen "debugger" maken is nog niet zo eenvoudig.

Een MONITOR zorgt ervoor dat een programma op instructie nivo kan worden gevolgd. Het is meer een soort achteraf debuggen van al reeds aangemaakte code. Je kunt wel de code weer disassembleren, en er weer "leesbare" programma instructies van maken. Je kunt geheugenlocaties bekijken en registergegevens uitvragen, maar op zich is het nogal allemaal beperkt. Het fraaie is wel dat je bestaande instructies kunt veranderen (waardoor het een beetje op selfmodifying code lijkt). Een monitor is het laatste hulpmiddel waar je wel op terug moet grijpen als je die andere hulpmiddelen mist. Het is ook de enigste manier om te kunnen "volgen" wat een programma doet als je alleen maar beschikt over de object code en de source code mist.

### DOCUMENTATIE

Een ondergeschoven kind bij de meeste programmeurs onder ons is de documentatie van een programma. Ik heb het hier dan niet over de interne documentatie, maar de werkelijke gebruikersdocumentatie. Wat heeft dat nu met gestructureerd programmeren te maken zul je wellicht zeggen. Welnu, tijdens het schrijven van deze documentatie zul je namelijk aan de gebruiker van jouw programma duidelijk moeten maken hoe jouw programma precies werkt. Het komt vaak voor dat je tijdens het schrijven van deze documentatie er achter komt, dat er nog iets aan jouw programma mist. Iets waar je wellicht niet aan gedacht had tijdens het ontwerpen van je programma. Ook komt het dan regelmatig voor dat je vergeten was om op een specifieke uitzondering te testen. Het is jou wellicht ook wel eens overkomen dat als je met een programmeer probleem zat, en je er met een "collega" programmeer over ging praten, je doordat je je probleem precies moest omschrijven, je al vertellende, je daardoor duidelijk werd waar het probleem in je programma zat. Die andere programmeer hoefde je niet eens te vertellen wat er fout was (waarschijnlijk wist die niet eens waar je het over had!).

Nu snijdt documentatie aan 2 kanten, het

maakt een aantal zaken voor de gebruiker duidelijker, en ook voor jezelf. Als je namelijk een jaar later hetzelfde programma opakt, om er wijzigingen in aan te brengen of om fouten op te lossen, dan kun je je documentatie doorlezen om je weer snel vertrouwd te maken met je programma. Ook kun je door deze wijze van werken, het programma "overgeven" aan een andere programmeur, waardoor deze je programma op een redelijk snelle manier "begrijpt". Vooral als je hem dan ook nog van de nodige programma stroomdiagrammen voorziet. Overigens heb ik ook wel eens meegemaakt dat het documenteren van een programma twee tot driemaal zo lang duurde in uren, dan dat het gekost had om het programma zelf te schrijven, inclusief ontwerp, schrijven, debuggen en alles. Het is dan de vraag of in een dergelijk geval de balans niet te veel heeft doorgeslagen naar de andere kant.

#### **VOORDELEN VAN EEN STANDAARD WERKWIJZE**

Waarom is een standaard manier van werken nu zo belangrijk? Je zult wellicht gemerkt hebben dat er op heel wat plaatsen in dit verhaal opgemerkt wordt "als je dit programma overdraagt aan een ander...". En daar gaat het nu allemaal om, je doet het niet voor jezelf (alhoewel daar ook voordelen aan kleven), maar omdat je wil dat anderen snel je programma begrijpen, en dus niet eerst uit moeten gaan zoeken hoe jouw werkwijze is. Dit komt nog meer naar voren, wanneer je programmatuur moet gaan ontwikkelen in teamverband, en met meerdere programmeurs een specifiek project probeert te verwezenlijken. Door een standaard wijze van werken kun je programmatuur uitwisselen, of het project opdelen in meerdere stukken. Aan het uiteindelijke resultaat kun je niet zien of het door 1 of meerdere programmeurs ontwikkeld is. Projecten kunnen sneller ontwikkeld worden, omdat het uitvallen van 1 programmeur geen grote gevolgen heeft op het klaarkomen van het project. De taken die door de uitgevallen programmeur moesten worden gedaan, kunnen door een

ander lid van het team worden uitgevoerd.

Maar ook voor jezelf kleven er wel degelijk voordelen aan. Het ontwikkelen van vervolprogramma's gaat veel en veel sneller, omdat na het kopiëren van een standaard programma slechts minimale aanpassingen hoeven plaats te vinden om er een compleet nieuw programma van te maken. Het is ook makkelijker om additionele functies aan reeds bestaande programma's te bouwen.

#### **CONCLUSIE**

Al uit het voorgaande mag je onderhand duidelijk geworden zijn dat een standaard wijze van programmeren toch eigenlijk helemaal niet zo eenvoudig is. Het kost tijd, veel tijd. Vooral bij simpele programma's moet je je natuurlijk wel afvragen, is dat in dit geval nu wel allemaal nodig. Soms wel, soms niet. Jij bent degene die dat het beste kan bepalen. Soms zul je alleen maar een programma stroomdiagrammetje maken, om je gedachten te ordenen, meer is er soms ook niet nodig. Maar nu weet je in ieder geval hoe je moet handelen bij het werken aan grote projecten.

Tijd is onze grootste vijand, tijdslimieten van projecten, is de grootste schuldlige die ons er toe dwingt om de eisen aan onze programmatuur (ontwerp, standaards, documentatie), geweld aan te doen. We nemen genoegen met een mindere kwaliteit, omdat het snel af moet. Later betalen we daar wellicht de prijs voor (soms met zijn allen), maar dat is van later zorg. Zo is het probleem van de onadreseerbare sector nul voor de 8bit Atari XL/XE je wellicht bekend, hierdoor is sector 720 officieel niet adresseerbaar bij single density diskettes. Een foutje van een programmeur onder tijdsdruk, en miljoenen sectors over de gehele wereld zijn "verspild". Probeer de gestructureerde manier van werken in ieder geval eens een tijdje uit, misschien zie jij er de voordelen ook wel van in. En anders heb je er in ieder geval over nagedacht, en heb je goede redenen om het juist NIET te gebruiken.

*Reacties? email op: schreurs@horizon.nl*

# ST MUIS aan 8-bit ATARI

*Door Mathy van Nisselroy*

De laatste tijd kom je in dit magazine verhalen tegen, over hoe je vanalles aan je 8 bit Atari kunt hangen. Bijvoorbeeld "XL aan SCART" (als je een STEREO uitbreiding hebt, kun je pinnen 2 en 6 van de SCART stekker aansluiten aan respectievelijk de rechter en linker geluidsuitgang. Bij "Handy-Man" hebben ze trouwens tegenwoordig SCART kabels (met alle pinnen aangesloten) voor fl. 9,95. Knip die door midden en je hebt twee SCART-stekkers met kabel.), CD-ROM voor de XL, enz.

Maar je kunt ook SEGA joypads, de Amiga muis of de ST muis op je computer aansluiten. Voor de SEGA maken ze joystickverlengkabels die je ook prima op onze computer gebruiken kunt. Over de joypads moet trouwens nog gezegd worden, dat waarschijnlijk niet alle extra functies (zoals auto-fire) werken.

Maar dit verhaal gaat over de ST muis. Al jaren was ik opzoek naar een ST muis. Niet omdat er ontzettend veel software is die gebruik kan maken van de muis, want die is er amper, maar gewoon omdat ik gek ben op zo'n soort hebbedingetjes. Toen ik er dus op de laatste JHV van de ABBUC een zag liggen voor maar DM10 (ruim fl. 11.00), was de koop gauw gesloten. En, zo werd mij verteld, hij was al aangepast, zodat je de rechter muis knop kunt gebruiken. Dat bleek achteraf niet het geval te zijn.

Waarom aanpassen? Bij de ST muis is de rechter muisknop aangesloten op joystick pin nummer 9. Op de 8 bit Atari wordt pin 9 (en ook pin 5) gebruikt als analoge ingang d.w.z.

dat, als je er een (regelbare) weerstand, zoals bv. een paddle, op aansluit, de computer meet hoe groot die weerstand is. De muisknoppen zijn gewone schakelaartjes. Voor de rechter muisknop moet dus een oplossing gevonden worden. De eenvoudigste oplossing is deze: Je hebt een kruisschroevendraaiertje en een weerstand nodig (tussen 4700 en 18000 Ohm).

- Draai de muis om.
- Om te voorkomen dat de muisbal onverwacht uit de behuizing valt en we op de knieën door de kamer moeten, verwijderen we de bal tijdelijk. Aan de onderkant van de muis zie je een gat met daarin de bal. Eromheen zit een plaatje dat de bal op zijn plaats houdt. Is dit plaatje rond, dan moet je het verdraaien, is het een beetje hoekig, dan schuif je het naar onder. Draai de muis even om en leg het balletje even aan de kant. Pas op dat ie niet alsnog ervandoor gaat.
- In de buurt van de kabel vind je twee schroefjes. Verwijder deze.
- Draai muis weer om en verwijder de bovenkant. Dit gaat het gemakkelijkst als je 'm bij de muisknoppen voorzichtig omhoog trekt. Als het wat stroef gaat kun je even met de muiskabel wiebelen.
- Rechtsboven vind je de connector waarmee de muiskabel aan het printplaatje van de muis zit. Zoek het witte en het rode draadje. Daar waar deze in de connector verdwijnen duw je de pootjes van de weerstand in de connector. Controleer even of de aanpassing werkt, door de volgende regels in BASIC in te typen;

```
10 ? PADDLE(0):GOTO 10
RUN
```

Of, als je de muis in de rechter poort hebt geplugd;

```
10 ? PADDLE(2):GOTO 10
RUN
```

Als je de rechter muisknop NIET indrukt, moet de waarde kleiner zijn dan 9. Druk je de knop in, dan is de waarde 228, net zoals hij altijd was VOOR de aanpassing. (Op de printplaat zitten twee van die gele, bijna vierkante dingetjes met een zwarte rand. Dat zijn de schakelaartjes die je normaal met de muisknoppen bedient. Je kunt die gewoon met een vinger induwen.)

Hopelijk krijg je de juiste waardes uit ('n weerstand van 4700 Ohm geeft bij mij een waarde van 3, een weerstand van 18.000 Ohm geeft een waarde van 8). Naar het schijnt, zijn er verschillende versies van de muis, maar deze aanpassing zou zoals hierboven beschreven op de meeste muizen moeten werken. En zo niet, dan is de weerstand er snel weer uit getrokken. Het witte draadje gaat bij mij naar pin 9 en het rooie draadje gaat bij mij naar pin 7. Als je naar de stekker kijkt, ziet die er zo uit:

```
o o o o o
  o o o o
```

Die als volgt genummerd zijn (denk er aan dat we naar de stekker kijken en niet naar de joystickpoorten aan de computer, zoals in de meeste boeken.):

```
5 4 3 2 1
  9 8 7 6
```

Het kan zijn dat bij sommige muizen de kleuren in een andere volgorde zijn gebruikt. Mijn muis komt uit Taiwan en is een "rev.c" volgens het printplaatje, dat zelf het nummer CO70903 heeft.

- Als alles goed is gegaan, trekken we de weerstand er even uit om de lengte van de pootjes wat in te korten, zodat we de muis straks ook weer dicht kunnen maken.

- Nu kunnen we de bovenkant weer op de muis zetten. Eerst de kant die het verst van de muiskabel is, dan de andere kant. Je moet even kijken of de kabel goed in de uitsparing zit, anders gaat de bovenkant er moeilijk op.  
- Dan schroef je de schroefjes weer vast en doe je muisknop en plaatje weer op hun plaats.  
- En klaar is Klara. (Ja, je mag je chocomel nu opdrinken!)

Markus Witte zegt in ABBUC magazine 34, dat de aangepaste muis ook nog gewoon op de ST werkt. Ik heb alleen XE computers, dus heb dat niet kunnen uitproberen. De linker muisknop wordt trouwens met STRIG(0) of STRIG(1) uitgelezen, net zoals de vuurknop op een joystick.

### Mijn muis is ziek!

Een tijdje geleden stond er op een van de ABBUC magazine's een demoversie van BOMBDOWN. BOMBDOWN is te vergelijken met Minesweeper voor de PC. Dat zal niet iedereen veel zeggen, dus zal ik even kort iets over het spel vertellen. Het "speelveld" is onderverdeeld in hokjes (zoals een dambord). In sommige van die hokjes zit een bom. Jij moet zoeken waar. Aan het begin van ieder niveau zie je een aantal hokjes met daarin een cijfer. Dit cijfer geeft aan hoeveel bommen er rond om dat hokje zitten. Met wat geluk en een stevige dosis denkwerk kun je er meestal wel achterkomen waar de bommen zitten. Helaas is het bij BOMBDOWN niet altijd gemakkelijk om met de joystick te werken. En toen ik dus op het scherm zag, dat je ook met een ST of Amiga muis kunt werken, heb ik mijn ST muis er maar eens bij gepakt.

Poeh, dat gaat een stuk gemakkelijker. Maar... wat ziet mijn oog (excuus, wat zien mijn beide ogen), hij gaat alleen maar naar links en naar rechts. Niet naar onderen dus. Nu was me dat ook al bij een programmaatje van John Maris (Pokey mag. 17, Maart 1994) opgevallen, maar dat was nog niet af (alleen om even met de muis te kunnen spelen), dus ik dacht: "Het zal wel aan het programma



liggen". Maar toen het bij Operation Blood (de joystick-versie, niet de Lightgun-versie) ook alleen maar van links naar rechts ging, begon ik aan mijn muis te twijfelen.

Had de heer Maris niet ooit iets gezegt over VIER waardes voor iedere richting waarin de muis beweegt. (ook in mag.17.) Eens even kijken, even de joystick poort uitlezen met:

```
10 ? STICK(0):GOTO 10
RUN
```

Goh, da's lastig zeg, een muis maar in EEN richting te bewegen. Maar langs een lineaal is het net te doen. Bij alleen verticaal beweegt (lees: De cursor op je beeldscherm beweegt verticaal.) krijg ik maar twee waardes. Bijvoorbeeld 4 en 12.

De volgende morgen, in de badkuip (om de een of andere reden krijg ik de beste ideeën altijd op de badkamer of de w.c.) viel mij het volgende in: Iedere joystickpoort gebruikt vier pinnen voor joystick en muis. Oftewel twee pinnen voor horizontaal en twee voor verticaal. De uitgelezen waarde hangt af van het feit of er spanning (1) of geen spanning (0) op de pinnen staat.

In de badkuip bedacht ik dus, dat 12 min 4 is 8 (Nee, daar heb ik nog geen Nobelprijs voor gekregen!). En wie wel eens met nullen en eenen rekent, weet dat 8 en 4, net zoals 2 en 1 trouwens, waardes zijn die je kunt krijgen

door EEN bit (of signaal) te veranderen. Nu moet je weten (nauwja, MOET??), dat er in iedere muis twee asjes zitten, die de rotatie van de muisbal oppikken. Op ieder van die asjes, zit een wieltje, dat dus met de bal mee draait. In die wieltjes zitten weer gaatjes. Het wieltje draait tussen twee LED/LDR combinaties door. Een LED is een soort lampje. Een LDR is een dingetje dat stroom doorlaat als er licht op valt. (Ik ben bang dat als ik dat zo op een proefwerk in zou vullen, de leraar daar niet blij mee is. Maar op deze manier blijft iedereen bij de les :- ) Als je dus met de muis beweegt, zorgen de wieltjes met hun gaatjes ervoor, dat het getal dat STICK(0) leest, steeds verandert. Er zijn twee LED/LDR combinaties, dus vier verschillende waardes per wieltje/richting waarin je de muis beweegt. John had dus gelijk. En ik kreeg het vermoeden dat EEN van de vier LED/LDR combinaties kapot was.

Omdat iedere LED/LDR combinatie een eigen pin op de joystickstekker heeft, heb ik even een programmaatje geschreven dat de door STICK(0) gelezen waardes omzet naar de individuele signalen van de joystickpinnen. Als de muis niet aangesloten is, zijn alle waarden 1. Door de muis aan te sluiten en hem dan te verdraaien zul je zien dat de signalen veranderen. Door een voor een de asjes te verwijderen krijg je, dat de muis "denkt" dat ie maar in een richting wordt bewogen. Je kunt ze ook alletwee verwijderen. Door vervolgens een stukje papier tussen een LED en een LDR te houden, kun je iedere combinatie apart testen. Bij mij veranderde een van de vier signalen nooit. En ik vermoedde, zoals je al hebt kunnen lezen, dat een van de LED/LDR combinatie kapot was.

Op zo'n moment komt een digitale multimeter van pas. He, zullen sommige van jullie nu denken, die heeft verstand van electronica. Nee hoor, net als de muis was de multimeter een van die hebbedingetje die ik gewoon moest hebben. En als je zo'n ding hebt, wil je hem ook gebruiken. Dus eerst de spanning over de LED's gemeten. Die waren OK.

Toen de spanning over de LDR's. Viel er licht op, dan was er geen spanning, viel er geen licht op de LDR, dan stond er tussen vier en vijf Volt op. Ook daar klopte dus alles.

Vervolgens heb ik geprobeerd, om de signalen aan de connector (daar waar kabel en printplaatje met elkaar verbonden worden) te meten. Dat viel niet mee! Op de eerste plaats is het lastig op de banen op het printplaatje te volgen en dus het juiste pinnetje op de connector te vinden. Je moet testen terwijl de muis op de computer is aangesloten. Is dat namelijk niet het geval, dan gebeurt er in de muis helemaal niets. Is het zaakje aangesloten, dan houd je weinig ruimte over voor de testpennen van de multimeter. En dan moet je er ook nog voor zorgen dat de signalen die de muis afgeeft veranderen. Eigenlijk heb je dus minstens drie handen nodig. Zou het niet af en toe handig zijn als de mens een staart had?

Het zou natuurlijk handig zijn als we al eens wisten, welke draadje bij welke joystickpin hoort. Dus zetten we de multimeter op "doorpiepen" en gaan we een voor een alle draadjes na. He! Terwijl er aan de stekker (met plaats voor negen draadjes) acht draadjes zitten, vind ik maar op zeven gaatjes een bijpassend draadje. Pin 3 van de stekker en het groene draadje (de enig combinatie die dan nog mogelijk is/overblijft) maken geen contact.

Conclusie: Kabel breuk.  
Ergens is de kabel gebroken. Bij een electronica zaak bij ons in de beurt verkopen ze ATARI PC muizen voor maar 5 piek. "Ja, alle negen draden aangesloten en Nee, we draaien 'em niet open om te kijken of de connector ook hetzelfde is." Zucht. Dat ook de PC muizen van ATARI seriele muizen zijn (bij PC's wordt de muis op de seriele poort aangesloten), dat had ie al eens eerder verteld. De ST muis is gewoon een "parallele" muis en werkt dus heel anders.

muis, de teflon voetjes, het balletje en de schakelaars hetzelfde waren. De rest was totaal anders. Behalve misschien de kabel... Waarvan trouwens maar vijf draadjes waren aangesloten, gesoldeerd, niet via een connector en op z'n plaats gehouden door een soort lijm of kit. Tja, als er een firma goed was in sparen, dan was dat Atari wel, maar goed... "Zit daar niet draadje nummer 6?" "Dan hebben ze er misschien nog een paar afgeknipt!" Dus knip ik even de hele kabel af en leg vervolgens acht draadjes bloot. Weer de meter erbij en het zaakje door piepen. En dan...

Dan kijk ik nog eens goed naar de stekker van de PC muis en zie dat er een draadje zit op positie vijf. Die is bij de ST muis niet gebruikt. Oeps, dan zit er ergens anders geen draadje. En dat blijkt positie nummer negen te zijn. Uhhhh? Zat daar niet mijn rechter muisknop op...? Hoe lossen we dit netjes op? Dus met een originele muiskabel! Ernest Schreurs heeft al aangeboden om eens tussen zijn collectie muisonderdelen te kijken of ie de juiste kabel heeft. In de catalogus van Best Electronics in San Jose, California staan vier verschillende muiskabels. En die variëren in prijs van 8,5 tot 9 dollar (\$1,00 = fl.1.72). Hele muizen kosten bij Best \$50.

Ondertussen ben ik er ook achter waar het groene draadje gebroken is. Namelijk net achter de stekker. Hoe kom je daar nu achter? Och, de computer reageerde helemaal niet op de muis. En dan begin je met de stekker te wiebelen. Dat helpt niet. Dan maar eens stevig met de kabel "gerooid". En zie daar, de computer ziet de muis weer. En als je dan ook nog de draad wat tegen de stekker drukt, dan doet de hele muis het...

Heel even BOMBDOWN uitgeprobeerd. Werkt prima, maar het is wel lastig om met de linkerhand de muiskabel goed te houden en met de rechter de muis te bedienen. En andersom wordt nog leuker...

Thuisgekomen bleek dat de behuizing van de CU Mathy Van Nisselroy

## BBS-sen in Nederland:

**STack BBS**

*Door Fred Meijer*

Dit is het tweede artikel in de reeks 'BBS-sen in Nederland'. Dit keer zullen we aandacht besteden aan een van de oudste Atari BBS-sen in Nederland, namelijk het STack (SAG) BBS.

Dit BBS wordt draaiende gehouden door twee van de bekendste Atariërs in Nederland, namelijk Bo en Ernest Schreurs. Tijdens de autoreis van het Duitse Hanau naar Den Bosch stelde ik ze een aantal vragen. Uiteindelijk is er ook nog uitvoerig gesproken over andere activiteiten dan het BBS, maar die willen wij u niet onthouden...

***Hoe en wanneer kwam je voor het eerst in aanraking met een Atari computer?***

(Na erg lang nadenken) Ehh, da's wel erg lang geleden. Via Engelse en Amerikaanse bladen kwamen wij op de hoogte van het bestaan van Atari homecomputers. De eerste computer die wij hadden, was echter geen Atari, maar een 8080 S100 bus computer. Dit was een zelfbouwsysteem, dat door Bo in elkaar gezet was. Deze computer werd hoofdzakelijk gebruikt om te programmeren. Op de beurs 'Firato' in Amsterdam zagen we de eerste Atari's, de 400 en de 800. Deze computers werden daar uitgebreid gedemonstreerd. Na die demonstratie wilden we er gelijk een kopen. Dat was echter moeilijker dan we dachten. In Nederland waren deze computers namelijk niet verkrijgbaar. We zijn toen naar Duitsland gegaan om er een te bemachtigen. In Duitsland hebben we een 800, een 410 tape-recorder en een 810 disk-drive gekocht. De 810 hebben we een week daarna zelfs nog een keer moeten ruilen,

omdat hij stuk was. In totaal hebben we toen zo'n 4000 Mark uitgegeven.

***Wat deed je vroeger met je Atari 800?***

De computer werd hoofdzakelijk gebruikt om te programmeren. We hadden wel enkele spellen (o.a. Star Raiders en Galaxians). We moesten echter wel elke keer naar Duitsland toe om software te kopen. Gelukkig werd al snel de XL ook in Nederland verkocht en konden we dus ook software in plaatsen als Rotterdam en Amsterdam kopen.

***Wat waren de verder activiteiten?***

We hebben diverse listings uit bladen ingetypt, daarnaast probeerden we diverse andere zaken voor onze Atari te krijgen, waaronder diverse interfaces en bijvoorbeeld een tweedehands voice-box. Helaas was op software gebied nog steeds weinig te krijgen en namen ook wij onze toevlucht tot de illegale handel.

***Hoe zijn jullie in aanraking gekomen met de Stichting Atari Gebruikers (SAG)?***

In 1985 hadden we onze eerste ST gekocht en een Atari 130XE. We hebben toen ook een SAG Magazine gekocht. We zijn toen lid geworden en zijn op de SAG-beurs in contact gekomen met Eli Maas.

In 1986 zijn we voor twee jaar vertrokken naar Amerika. We zijn daar lid geworden van de Atari club SPACE. Daarnaast zijn we in Amerika voor het eerst in aanraking gekomen met het fenomeen BBS. Een groot aantal Amerikaanse BBS-sen boden de mogelijkheid om gratis software te downloaden en te uploaden. Dit sprak ons erg aan.

Tijdens ons verblijf hebben we altijd contact gehouden met Eli Maas. We hebben diverse produkten meegenomen en aan hem gegeven om te bespreken in het SAG-Magazine. Omdat ook bij het SAG-Magazine er altijd een groot tekort was aan artikelen hebben we nog een aantal Amerikaanse vertaald voor het SAG-Magazine.

### ***Wanneer zijn jullie begonnen met het BBS?***

Toen wij terug waren uit Amerika wilden we direkt beginnen met een BBS. In die tijd was het aantal BBS-sen erg klein en was het erg nieuw. We hebben toen een erg duur modem aangeschaft, met de toen enorme snelheid van 300 baud. We wilden met ons BBS een centraal punt vormen voor Atari gebruikers. We hadden onze plannen aan Eli Maas verteld. Hij vroeg ons toen of wij niet het SAG-BBS wilden overnemen. We hebben toen onze eigen plannen in de ijskast gezet en hebben het SAG-BBS overgenomen. Het SAG-BBS draaide met het programma 'Michitron BBS' op een PC. Het BBS kreeg twee lijnen, die draaiden op 2400 baud. Het nieuwe SStack/SAG BBS kreeg veel aandacht in het SAG-Magazine.

We zijn niet kort daarna overgestapt op een ST, de PC en een modem waren ons in bruikleen gegeven. Op de ST was het Michitron BBS-programma gelukkig ook verkrijgbaar. We hebben toen een peperdure knots van een harde schijf op het systeem aangesloten. De harddisk van 340MB heeft ons nog f 4000,- gekost!

### ***Hoe zijn jullie aan de naam SStack gekomen?***

Allereerst hadden we besloten een stichting voor het BBS op te richten. Zo werd er een rechtspersoon gecreëerd en waren we niet meer persoonlijk aansprakelijk voor bijvoorbeeld door anderen ge-uploade software op het BBS.

Het woord SStack staat voor ST acknowledge, wij wilden hiermee een leuke naam verzinnen. Bo dacht aan STichting Atari ComputerKenners. Echter de notaris dacht daar

anders over, want Atari was tenslotte een beschermdde naam. Met de vooruitzichten die Atari nu biedt, is die keuze achteraf gezien, nog geneeens zo slecht geweest.

### ***Hoe liep het BBS?***

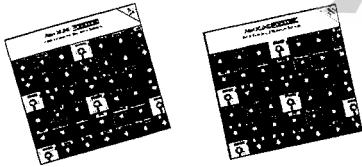
Het ging redelijk goed. De ST en de XL/XE waren nog zeer populair. De grootste groep bellers waren wel in het bezit van een 8-bitter. We hadden zelf wel een hekel aan de vaak bezette lijnen, zodat we besloten een derde lijn te nemen. We hadden tenslotte een cratridge die drie lijnen aankon. Dit was toendertijd echt uniek, omdat de PC nog geen drie lijnen aankon en de ST wel. Daarnaast lazen we in bladen over CD-ROM spelers. We hebben er toen een gekocht en aan het BBS gehangen. We moesten echter wel zelf de Atari-driver schrijven. Uiteindelijk hadden we 4 CD-spelers aangesloten en nog was het niet genoeg. We zijn toen CD-wisselaars gaan kopen. Ook daar moesten we zelf de drivers voor schrijven. Nu hangen er 7 wisselaars en 1 losse CD-ROM speler aan het BBS, zodat er 43 CD-ROMS online zijn. Zo'n twee jaar geleden besloot Atari Benelux hun BBS 'Atari-net' op te heffen. Men heeft toen ons gevraagd het BBS op het SStack BBS een plaatsje te geven. We hebben toen maar een nieuwe harddisk gekocht om de software op ons BBS te zetten, want de software van Atari Benelux mocht natuurlijk niet verloren gaan.

### ***Hoe staan de zaken nu met het BBS?***

De laatste tijd is het erg stil. We krijgen nog maar zo'n 5 a 10 bellers per dag. Zelf geven we door een aantal andere activiteiten wat minder aandacht aan het BBS. Het leeft ook onder de bellers niet meer zo. Er wordt niet meer ge-upload door hen. Er staat echter al zo'n 40 Gigabyte (!! ) aan programmatuur op het BBS.

### ***Hebben jullie er wel eens aan gedacht om te stoppen?***

NOOIT! We hebben al diverse tegenslagen gehad, waaronder blikseminslag. Daarnaast hebben we al regelmatig onze harddisk moeten vervangen. Andere BBS-sen zouden



met die tegenslagen waarschijnlijk al gestopt zijn, wij zijn echter van mening dat als je een commitment bent aangegaan, je je daaraan moet houden!

### ***Wat doen jullie naast het Stack BBS voor de Atari 8-bitter?***

Wij helpen andere Atari BBS-sen. Zo proberen wij hun problemen op te lossen en sponsoren wij het een en ander. Komt een collega-SySop er niet meer uit, dan willen wij nog wel eens persoonlijk het probleem oplossen. Daarnaast importeert Bo uit het buitenland hard- en software.

### ***En natuurlijk de bekende Atari Pooldisk CD-ROM.***

Omdat we zelf wel eens kapotte diskettes hadden besloten we door middel van SIO2PC een back-up van onze software op CD te maken. Zo'n back-up zou ons f 125,- gaan kosten. We wilden dan wel dat die CD dan helemaal vol kwam te staan. We zijn toen bakken met PD-Software gaan aanschaffen om op de CD te zetten. Voor de overzichtelijkheid wilden we van elk programma een beschrijving maken. Dat kost tijd zeg! Al snel hoorden andere mensen van ons plan. Toen bleek dat veel mensen interesse hadden in een kopie van de CD. We hadden zo'n 2 jaar gereserveerd voor dit project.

Uiteindelijk duurde het nog langer. De programmatuur werd eerst naar de harddisk geschreven. Dit deden we in twee formaten: het ATR en het XFD formaat. Hierdoor kon de CD op de PC en de ST gebruikt worden. We besloten de CD in het ISO formaat te laten beschrijven, zodat ook zelfs in de toemst de Atari XL de CD zou kunnen lezen. Om te peilen hoeveel interesse de mensen zouden hebben, hebben we op de JHV in Herten van 1995 gevraagd hoeveel mensen geïnteresseerd waren. Daar hebben we ook beloofd het

volgende jaar met de CD te komen. Een zware belofte.

Op het BBS hebben we een proef-versie gezet, om te kijken wat de bellers er van vonden. Helaas kregen we daar zeer weinig reacties op. Men was erg passief. Dit stimuleerde ons niet erg, maar we deden het tenslotte voor ons zelf. Gelukkig bleven de prijzen van de CD-ROM schrijvers zakken. We hebben toen zelf maar een schrijver aangeschaft. We konden een bedrijf het geld geven of ze zelf maken. We hielden er dan ook een schrijver aan over! De kosten van de schrijver (niet de software en de tijd!) zouden dan gedekt worden als we 100 CD's a f 50,- verkochten.

### ***Hoe loopt de CD?***

Boven verwachting. We hebben er in korte tijd zo'n 50 verkocht en we hebben er zo'n 10 geschonken. Tot nu toe zijn er zo'n 125 geproduceerd en zo'n 100 stuks verkocht.

### ***Wat zijn jullie toekomstplannen?***

Ach, Atari is niet meer. Bo: "Maar ik zal Atari nooit opgeven." We hebben al veel aangeschaft, zodat als de Atari's niet meer te krijgen zijn, wij niet zonder zitten. Er wordt nu veel tegen spotlage prijzen aangeboden. Ernest: "Anders gaan ze toch de vuilnisbak in."

Als we veel nieuwe software krijgen dan gaan we een tweede CD maken. We hebben met de ABBUC afgesproken dat we de gehele SAG-bibliotheek krijgen. We hebben daarnaast nog veel software van de gebruikers nodig. Echter we krijgen nauwelijks reacties, dus of die CD er van komt...

Voor het BBS willen we graag 28k8 modems aanschaffen. Daarnaast willen we het ZModem protocol toevoegen. Hiervoor moeten we zelf de software van het BBS aanpassen. Dit is een immens karwei. We hopen hiermee aan het eind van 1996 klaar te zijn. Door de 28k8 kan de snelheid van 300 baud afvallen. We hebben onze bellers om reactie gevraagd, ook krijgen we helaas hierop nauwelijks antwoord.


# EEN KIDS-CONTROLLER EN DAN?

Door Ernest Schreurs

Zoals zo langzamerhand wel bekend is verzamel ik alles waar het Fuji logo op voorkomt. Dat geldt niet alleen voor apparatuur. Ook boeken en allerlei literatuur die iets met Atari te maken hebben worden verwoed verzameld. Omdat ik me eigenlijk bijna 24 uur per dag met programma's schrijven bezig houd, is het voor mij ook noodzaak om alle informatie die daarvoor nodig is onder handbereik te hebben. Het is daarom ook bijna vanzelfsprekend dat ik flinke stapels literatuur heb over dit onderwerp. Niet alleen voor de classic Atari, maar dus ook voor de ST.

Daarnaast verzamel ik ook nog Wang computers, ook met de bijbehorende niet onaanzienlijke stapels handboeken. Daar dit de meeste lezers verder niet veel zal kunnen schelen zal ik me beperken tot het classic

Atari gedeelte. Iedereen die wel eens een boek over programmeren heeft ingekeken zal weten dat er een behoorlijk scala aan onderwerpen bestaat. Sommige boeken gaan diep in op het programmeren van de hardware. Andere boeken zijn meer bestemd voor beginners. Er zijn boeken die zich toespitsen op een bepaalde programmeertaal. Er is zo ongeloflijk veel geschreven over en voor de Atari, dat het zelfs bij ons nog wel eens gebeurt dat je een boek tegenkomt dat je nog nooit gezien hebt. De informatie in een enkel boek kan tegenwoordig niet meer zo gek veel kennis toevoegen aan de informatie die reeds in ons bezit is, en dat soort boeken wordt dan eigenlijk alleen nog aangeschaft om de verzameling completer te maken.



**ANG**  
COMPUTERS

Groenelaan 50a  
3114 CJ Schiedam  
010-2467313

Geopend  
di-vr 12.00-18.00 uur  
DO KOOPAVOND  
za 10.00-17.00 uur

**MULTIMEDIA**

Minimicrofoon	7,50
Microfoon op stand.	15,-
Koptelefoon	15,-
Headset met micr.	20,-
8 speed CD rom	210,-
12 speed CD rom	260,-
16 b. audiokaart 3d	60,-
Soundblaster 32 PnP	220,-

**PENTIUM 120**

**MULTIMEDIA**

Minitower, VX moederbord  
256 kb burst cache, Intel P120,  
16 mb EDO RAM, 1600 mb  
harddisk, 1.44 mb diskdrive, 1  
mb videokaart, 8 speed CD  
speler, 16 bits audiokaart, 50  
Watt speakerset, Windows 95  
toetsenbord, muis met mat, 14"  
Cheer kleurenmonitor .28,  
Windows 95 op CD en  
geïnstalleerd.  
Van 2395,- voor

**2295,-**

Vroeger was dat heel anders. Om te beginnen waren er niet zoveel boeken verkrijgbaar. En de boeken die er dan waren, die waren zeker niet goedkoop. Tegenwoordig kun je voor weinig geld een gebruikt boek aanschaffen. Toen die boeken nieuw waren moest je er soms wel tegen de honderd gulden voor betalen. Het was dus zaak een keuze te maken, en dat boek te kopen, waar je op dat moment mee verder kon. Soms kocht je zo'n boek omdat er een specifiek onderwerp in werd behandeld, zoals bijvoorbeeld player/missile graphics. De andere onderwerpen in zo'n boek waren dan soms zo ingewikkeld, dat je er niets van begreep. Die hoofdstukken sloeg je dan maar over.

Soms ook schafte ik een handboek aan gewoon om de gegevens bij de hand te hebben voor het geval ik daar behoefte aan zou hebben. Tenslotte zijn er niet zoveel boekwinkels die om drie uur 's-nachts open zijn, het tijdstip waarop je meestal vast komt te zitten, omdat je een klein stukje informatie nodig hebt over het een of ander.

Er zijn een aantal boeken die een schat aan informatie bevatten. Dit soort boeken heeft dan ook een grote reputatie en ze zijn bij iedereen bekend. Nu nog kom je regelmatig iemand tegen die nog zoekt naar deze klassiekers. Zo zijn er onder andere 'De Re-Atari' en 'Mapping The Atari'. Een van de klassieke boeken is ook het door Atari zelf uitgegeven, namelijk de 'Technical Reference Notes'. Naast een officiële beschrijving van hoe je sommige zaken volgens Atari moet programmeren, bevat dit boek ook een afdruck van de source listing van het Operating System van de Atari 400/800. Omdat ik zelf veel programmeer, vind ik het altijd prettig als je iets kunt opzoeken in een source listing. Een dergelijk boek ga je eigenlijk nooit van kaft tot kaft uitlezen. Je gebruikt het puur als naslagwerk. Op het moment dat je zo'n boek aanschaf weet je ook nog vrijwel niets van de computer af, dus zelfs al zou je het lezen, je zou er vrijwel niets van begrijpen. Als je met een bepaald onderwerp aan het programmeren slaat, vind je echter altijd wel de infor-

matie die je op dat moment nodig hebt. Jammer genoeg staat een en ander dan niet altijd helemaal duidelijk beschreven, dus is het toch vaak een kwestie van met de gevonden informatie experimenteren. Maar dat is ook de leuke kant van de hobby.

Als je zo aan het verzamelen bent, kom je ook wel eens wat vreemde dingen tegen. Zo heb ik op een gegeven moment een Kids-controller gekocht. Het stond in de prijslijst van Best Electronics, dus ik belde Brad Koda op en ik zei dat ik er een wilde hebben. Er stond tenslotte een Fuji op. Het bleek een vrij groot en kleurig paneel te zijn, met toetsen die lijken op die van een telefoontoestel. Het formaat bleek dus vooral voor kinderen niet over het hoofd te zien. Later heb ik nog een paar van hetzelfde type controllers gekocht in een formaat dat meer gebruikelijk is. Ook hier de fraaie druktoetsen. Echter geen enkele beschrijving erbij van hoe je nou het zaakje bestuurt. Kennelijk waren die dingen bedoeld om bij bestaande spellen of programma's te gebruiken. Erg jammer. Maar een beetje experimenteren is ook wel leuk. Na een avondje of drie experimenteren begon de lol toch wat af te nemen. Ik snapte er echt niets van. Nou ja, toch een leuk verzamelobject.

Vorige week kreeg ik van een Amerikaan een doos met een aantal boeken opgestuurd. Hij had deze via Internet te koop aangeboden, en ik was natuurlijk simpel te verleiden om een bod op deze boeken uit te brengen. De koop was gesloten, alleen moest het spul nog verzonden worden. Dat bleek ongeveer evenveel te kosten als het bedrag dat ik voor de boeken had geboden. Nou ja, er zat een en ander bij wat ik nog niet had. Hoewel er bij die set boeken ook een aantal boeken zat dat ik al had, heeft hij toch alles opgestuurd. Bij aankomst bleek daar ook het eerder genoemde Technical Reference Notes bij te zitten. Dat had ik dus al. Hoewel, het leek wel of dit boek anders was. Toch maar eens wat beter kijken. Tenslotte moest ik mijn eigen boek erbij pakken om te zien wat het verschil was. Er bestaan twee revisies van het Operating System van de 400/800. Een "A" en een "B"

revisie. Het bleek dat dit boek een "A" revisie listing bevatte. Toch nog de collectie uitgebreid. Toen ik die boeken toch op tafel had liggen, kon ik het niet laten om er weer eens wat in te bladeren. Nu heb ik inmiddels heel wat jaartjes ervaring op de classic Atari, dus dingen die ik doorbladerde kwamen mij nu allemaal best wel bekend voor. Je leert tenslotte toch ooit wat al die abacadabra nu eigenlijk allemaal voorstelt. Toen viel mijn blik op een stukje over keyboard controllers. Er stond nota bene een stukje basic bij, waarmee je deze beruchte controllers kunt uitlezen. Tja, toen ik dat boek kocht had ik die dingen nog niet, en begreep ik ook niet veel van wat er nou allemaal ge-peek't en ge-poke't werd. Dat stukje zal ik destijds dus wel snel overgeslagen hebben. Maar nu was alles glashelder.

Maar ja, als er iets is waar ik een hekel aan heb, dan is dat wel listings intikken. Ik heb nog wel wat diskettes met half ingetikte listings uit 'Compute!' en dergelijke bladen. Als je het al compleet intikte, dan zaten er altijd fouten in. En als er geen fouten in zaten, dan bleek altijd wel dat de programmeur nog wat foutjes had laten zitten om zelf op te lossen. Maar ja, als de listing als voorbeeld dient in een handboek van Atari zelf, dan mag je verwachten dat het toch wel goed werkt. Niettemin heb ik nog steeds die hekel aan het

intikken van listings. Toch maar even de flatbed scanner van de PC opstoken. Blaadje onder de scanner en OCR software erop los. Gaat wel niet helemaal vanzelf, maar is toch leuker dan domweg overtikken. Dit had ik jaren geleden moeten hebben. Intussen staat de listing als tekstbestand op de PC. Nu nog even dit omzetten naar ATASCII. Daar zijn utilities voor. Vervolgens het zaakje inladen in de PC-Xformer omdat ik te lui ben het bestandje om te zetten naar de echte XE. Hmm, het werkt dus niet. Even nadenken. Tja, hij laat vier controllers op het scherm zien. De originele 400 en 800 hebben inderdaad vier joystick poorten. Maar even de Xformer in de 800 stand zetten, en ja hoor, het voorbeeld programmaatje lijkt te werken. Nu maar eens op een echte XE proberen, echter eerst wat slapen.

De PC staat ondertussen andere dingen te doen. Zou je eigenlijk SIO2PC kunnen gebruiken terwijl je op Internet aan het surfen bent? Niet met deze PC, want hij heeft maar twee COM-poorten, en met een modem en een muis is het dan wel bekeken. Moet ik ook nog eens wat aan doen. Tja, dus die ingescande listing kunnen we even niet overhalen naar de XE. Nou ja, ik wilde eigenlijk ook liever zelf een beetje hobbyen met de informatie die ik nu gevonden had. Hoe bestuur je nu eigen-

**ANG**  
COMPUTERS

Groenelaan 50a  
3114 CJ Schiedam  
010-2467313

Geopend  
di-vr 12.00-18.00 uur  
DO KOOPAVOND  
za 10.00-17.00 uur

**Nieuwe Supplies**

Div. stofhoezen	10,-
Tigermouse	20,-
Ergo trackball	69,-
Diskettebox	10,-
8Muismat	5,-
CD/Disketterek	30,-
Printerpapier	8,-
120 Watt speakerset	55,-

**SUPERSTUNT!**

OP - OP  
Hewlett Packard  
**INKT-CARTRIDGES**  
voor

**45,-**

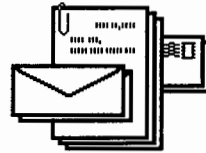
per stuk.  
*(maximaal 5 per klant)*

lijk zo'n keyboard controller. Even zo'n ding aansluiten maar. Dat blijkt niet mee te vallen, omdat ze weer eens onvindbaar zijn. Er ligt wel een video controller in een doos in de gang. Er zit een leuker kastje omheen, maar voor de rest is ook dit een keyboard controller. Het wordt toch tijd dat daar eens wat software voor gemaakt wordt. Uit de listing wordt duidelijk dat deze keyboard controllers gebruik maken van de PIA. De aansluitingen die normaal gesproken voor de joystick en de paddles worden gebruikt, worden in deze controllers gebruikt om aan te geven welke toets er is ingedrukt.

De keyboard controllers bestaan uit 12 druktoetsen, opgesteld in vier rijtjes van drie toetsen. Intern zijn ze blijkbaar ook zo opgebouwd. De toetsen die op dezelfde rij zitten, zijn met elkaar verbonden, en het draadje dat ze verbindt, is aangesloten op een van de pennen van de joystick poort. Samen dus vier pennetjes. De andere contactpunten van de druktoetsjes zijn per kolom aan elkaar verbonden, en deze draadjes zijn aangesloten op de twee paddle aansluitingen van de joystick poort en op de trigger van de joystickpoort. Met deze kennis en de listing wordt duidelijk hoe het zaakje eigenlijk werkt en waarom de pogingen om wat rond te peek'en in het verleden zo jammerlijk mislukten. Het is de bedoeling dat je het matrixje van de keyboard controller zelf gaat zitten afscannen met behulp van een programmatje.

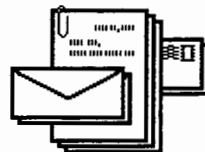
Hoe kun je nu met deze kennis bepalen welke toets er is ingedrukt? Je wilt weten welke rij met welke kolom wordt verbonden door een van deze druktoetsen. De wijze waarop dit gedaan wordt is door zelf een spanning te zetten op de contacten van de druktoetsen. Als een van de druktoetsen verbinding maakt, dan kun je die spanning meten op het corresponderende andere contact van de druktoets. De truc hierbij is dat je zelf bestuurt welke van de rijtjes van de controller je van een spanning voorziet middels hun aansluiting aan de PIA. Om te beginnen moeten we dus ervoor zorgen dat de joystick pennen van de PIA voor de verandering eens als output

werken. Met behulp van een poke 54018,48 kun je in PACTL aangeven dat je de instellingen voor invoer en uitvoer van deze pennen wilt aanpassen. Door daarna een poke 54016,255 uit te voeren op PORTA, zet je alle bitjes van die poort op output, waardoor de joystick pennen als output gaan fungeren. Als je de waarde 255 opschrijft in 8 bitjes, krijg je namelijk 11111111, dus alle 8 de pennen zijn dan output. Door nu weer een poke 54018,52 uit te voeren geef je aan dat je klaar bent met het programmeren van de invoer en uitvoer instellingen. Vanaf nu worden de waarden die we in PORTA schrijven weergegeven op de pennen van de joystick poort.



**VRAGEN OVER DIT ARTIKEL  
KUNT U STELLEN AAN DE  
SCHRIJVER VIA HET STACK  
BBS.**

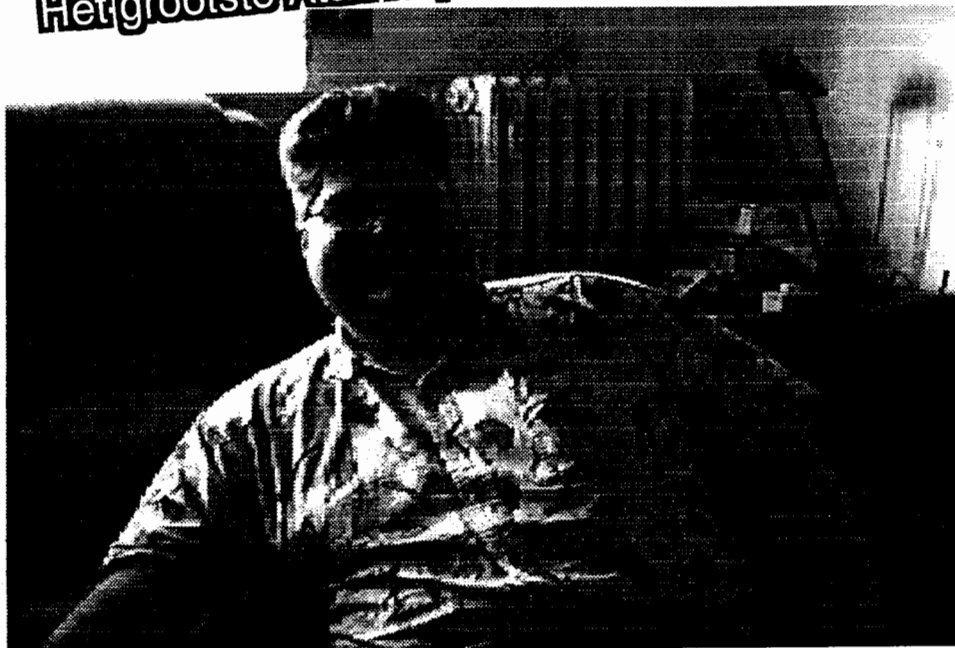
**HET NUMMER VINDT U  
ELDERS IN DIT MAGAZINE**



ATARI  
MUSEUM.NL

# BEcotel

Het grootste Atari XL/XE BBS in Nederland



BecoTel biedt als enige: Nederlandse Atari post, Duitse Atari post en Amerikaanse Atari post via diverse netwerken! Daarnaast hebben wij MEGABYTES aan Atari 8-bit programmatuur die u GRATIS kunt downloaden, online games, verhaaltjes voor het slapen gaan en nog veeeeeel meer. Pak uw computer (merk en type maakt niet uit) en bel naar BecoTel, het BBS voor Atari XL/XE fanaten!

BecoTel is elke dag online van 18.00 tot 06.00 uur en is te bereiken op 300, 1200, 2400, 9600 en 14400 baud.

# 020-6326859

# GTIA GEDOE

*Door Marius Diepenhorst*

Al geruime tijd vond ik het behoorlijk vervelend, dat er in mijn verzameling 800/65XE's (o.a. bij Intertoys gekocht) een probleem leek te bestaan met de GTIA chip. Met name grafische dingen die gebruik maken van Graphics mode 9 bleken niet mooi op het scherm te verschijnen. Er zaten lelijke blokjes op de plaats waar bij een "gezonde" XL/XE prachtige tinten moeten verschijnen.

Het leek me leuk om het probleem op te lossen, te meer omdat ik die grafische mogelijkheden van die XE graag aan andere mensen laat zien en dan moet het uiteraard wel in orde zijn; het is per slot van rekening wel de bedoeling dat je andere mensen enthousiast maakt voor de XE!!!

Het hele onderzoeksverhaal zal ik iedereen besparen, want het gaat toch uiteindelijk om het resultaat... en dat is, al zeg ik het zelf, best de moeite waard.

Hoe weet ik of er een goede GTIA in mijn computer zit?

Tja dat is nogal moeilijk te omschrijven op papier... Hieronder staat een programmaatje (basic) dat ingetoetst kan worden. Als het programma geRUNd wordt, is het de bedoeling dat je boven aan het scherm een soort van STAAF ziet. Deze staaf lijkt een prachtige cylinder te zijn, doordat er gebruik gemaakt is van verschillende grijs-TINTEN.

Als uw XE echter kampt met het GTIA probleem, en die kans is erg groot als het een 800/65XE van Intertoys is, dan wordt er wel een staaf getekend, en er zullen waarschijnlijk wel ENKELE grijstinten te zien zijn,

maar in deze grijstinten zal in meer of mindere mate een STRAMIEN van hokjes, blokjes en/of streepjes te zien zijn!! Dat is dus niet de bedoeling, het moet zagezegd dus een hele mooie "glimmende" en "gladde" buis zijn.

```

10 GRAPHICS 9
20 POKE 559,0
30 FOR T=0 TO 15
40 COLOR T
50 PLOT 0,T:DRAWTO 79,T
60 PLOT 0,31-T:DRAWTO 79,31-T
70 NEXT T
80 POKE 559,34
90 GOTO 90
  
```

De volgende modificatie is het resultaat van erg lang proberen en testen. Mijn gemodificeerde XE die hier nu al 4 dagen continu GTIA schermen staat te produceren doet niet raar, en is ook niet uitgebrand of iets dergelijks. Op zich is dat wel eigenaardig want de manier waarop de GTIA het nu goed gaat doen, is eigenlijk een rare manier. Ik laat het ANTIC signaal afzwakken door het aan de aarde te verbinden. Met digitaal gedoe als chips etc. is het echter regel bij dit pulldownen weerstanden van toch minimaal 1kOhm te gebruiken. Hiermee werkt de modificatie echter niet goed, dus moest er een 4700Ohm weerstand aan te pas komen. Des al niet te min functioneert de XE nu dus al een aantal dagen zonder oververhit te raken, dus ga ik er maar van uit dat het niet schadelijk is voor de XE... maar goed alles uiteraard op eigen risico.

### *Benodigheden*

- 2 weerstanden van 1 KiloOhm (bruin, zwart, rood)
- 1 weerstand van 470 Ohm (Geel Paars Bruin)

*Werkwijze:*

Gebruik voor het soldeerwerk een soldeerbout van maximaal 15 Watt!

1. Open de computer door de 4 schroeven te verwijderen.
2. Trek voorzichtig met twee handen het toetsenbord los.
3. Verwijder het afscherm-blik
4. Zoek de GTIA chip op (40 pins; (CO)14889)
5. Soldeer voorzichtig een 1Kohm weerstand met 1 zijde aan pin 18 van de GTIA
6. Soldeer voorzichtig de 470Ohm weerstand met 1 zijde aan pin 19 van de GTIA
7. Soldeer voorzichtig een 1Kohm weerstand met 1 zijde aan pin 20 van de GTIA.
8. Soldeer de drie losse kanten van de 3 weerstanden nu allemaal aan pootje 3 van de GTIA. (Kijk uit dat de weerstanden elkaar aan de kant van pin 18, 19 en 20 NIET raken!)
9. Buig de weerstanden voorzichtig zo plat mogelijk! Let er op dat ze geen andere contacten maken.

U kunt nu testen of de computer het nog doet....

Zoja: gefeliciteerd! U kunt de computer nu weer netjes in orde maken door het afscherm-blik terug te plaatsen (let op dat de weerstanden plat genoeg zijn gebogen, zodat ze niet het afscherm-blik raken), het toetsenbord voorzichtig weer aan te sluiten en de computer weer dicht te maken.

Ik heb de modificatie hier met 6 XE's getest en met al deze XE's werkt het perfect. Het kan natuurlijk zijn dat het bij u niet werkt, maar dan is er misschien nog iets anders mis met die computer.

Succes ermee. Vergeet niet NA de ombouw even dat programmatje nog een keer te testen of de cylinder nu niet veel mooier is....

# WAARGEBEURDE DE LAATSTE! VERHALEN

*Door Cosimicus*

Een persoon van de firma X gebruikte een 5 1/4 diskette om daar belangrijke informatie op veilig te stellen. Het was belangrijk om deze gevoelige informatie niet op de harde schijf te laten staan. Vandaar deze wijze van opslag. Tegenwoordig zouden we waarschijnlijk een portable syquest hard disk backup eenheid gebruiken, maar toen waren die er nog niet.. Nou zijn dergelijke grote diskettes nogal gevoelig voor stof en zo, en niet echt geschikt voor langdurige opslag, transportdoeleinden of backup. Het kan dan ook nogal eens gebeuren dat bij het lezen van gegevens van de diskette, leesfouten optreden. Deze persoon belde dan ook de Helpdesk op met de mededeling dat de desbetreffende diskette niet meer te lezen was op zijn computer.

De Helpdesk vroeg vervolgens de diskette maar naar hun op te sturen. Als het kon zag men er ook graag een briefje bij met een probleemomschrijving en de foutmelding(en) die eventueel werden weergegeven op zijn scherm. Dat was allemaal geen probleem, werd stellig aan de andere zijde verklaard. Er zijn nogal wat hulpmiddelen voor de PC (zoals de Norton Utilities), waarmee men data kan "redden" van diskettes, zelfs als deze beschadigd zijn, vandaar dat men vroeg of de diskette kon worden opgestuurd. De volgende dag lag er dan ook een enveloppe in de bus, met de desbetreffende diskette, en er zat ook een kort briefje bij waarop vermeld stond wat er allemaal niet goed ging met de diskette. De Helpdesk medewerker kon echter vanaf een afstand al zien dat deze diskette absoluut niet meer te redden was. De verzender had namelijk het briefje met de foutmeldingen AAN de diskette GENIET!

# MUISKABEL ERGERNISSEN

*Door Mathy van Nisselroy*

Tja, wat doe je, als een draadje in je muiskabel kapot is? Een nieuwe bestellen is duur. Tweedehands? Dan kom je dus al snel bij Bo en Ernest Schreurs uit. Die hebben van zo ongeveer alles wel wat (als er maar ATARI op staat). Maar het merendeel van hun collectie muizen had ook een of andere fout. Zouden muiskabels dan toch van die hoge (ahum!) kwaliteit zijn als ik al verwachtte?

Dus eerst maar eens de stekker van mijn eigen kabel opengemaakt. Doe dit niet, als je kabeltje nog werkt. Wie goed kijkt krijgt misschien de indruk, dat je joystick-, muis- en andere kabels uit elkaar kunt halen zonder ze echt kapot te maken, maar da's echt niet zo.

Het buitenste gedeelte (de mantel) van de stekker wil nog wel schuiven. Maar daar waar mantel en kabel bij elkaar komen, zitten ze op de een of andere manier aan elkaar geplakt. Wie dan aan de mantel gaat liggen trekken beschadigt de draadjes in de kabel.

Het binnenste gedeelte van de stekker bestaat uit twee (onafscheidelijke) delen. Een van deze delen kun je half zien. Dat is namelijk het deel dat je in de joystickpoort steekt. Aan de achterkant van dit deel (dus tegenover de "gaatjes") zitten busjes. Ieder busje is verbonden (ik meen geperst, maar ik kan het \$%&@# stekkerkje niet meer vinden...) met



een draadje. Zit er geen draadje, dan vind je ook geen busje. Aan de "voorkant" herken je dit, doordat het gaatje iets groter is dan de andere gaatjes die WEL met een draadje verbonden zijn. Zit er in een gaatje een busje, dan zie je dat als je goed kijkt ook aan de voorkant.

Het tweede deel bestaat uit een andere kunststof die doet denken aan een (drop/salmiak) toffee die om de draadjes en busjes is gegoten. Deze zwarte, harde plastic vult de mantel helemaal op en zit vast aan "deel 1". Helaas kun je deze plastic niet verwijderen, zonder de draadjes en/of het stekkerkje te beschadigen. Tenzij je een engelengeduld hebt en ZEEER precies bent. Ik heb het

uitgeprobeerd en kan jullie vertellen, dat het de moeite en tijd niet waard is (tenzij je zeer nieuwsgierig bent en je de kabel toch al wilt/moet vervangen). Zonder het zwarte goedje blijft het stekkertje trouwens toch niet in de mantel zitten. Die bij mij de operatie trouwens ook niet heeft overleefd.

Dus de volgende dag maar even een DB-9 stekker bij de electronica winkel gehaald. Een DB-9 stekker ziet er uit als een standaard joystickstekker, maar is van metaal en heeft twee "oortjes". En die oortjes zorgen er meestal voor dat je het stekkertje niet in de ATARI krijgt. En de mantel moet je er meestal los bij kopen. 's Avond heb ik de oude stekker van de muiskabel afgeknipt en de nieuwe eraan gesoldeerd. Even uitproberen. Joepie! Het groene draadje doet het weer.... m..m..m..maar wat is dat? Nu doet het blauwe draadje het niet meer. Ergens gebroken. Tja, dat krijg je als je probeert de trek-ontlaster of de stekker van de kabel te trekken. Natuurlijk kun je dan proberen om de kabel iedere keer met een paar centimeters in te korten, maar de kans dat je de kabelbreuk vindt voordat de kabel te kort wordt is nog al klein.

Er moest dus toch een andere muiskabel komen. M'n opties:

- Een muis(kabel) overnemen van Ernest. Een hele muis versturen met de post (je kunt niet verlangen dat ze voor een muis(kabel) even naar Heerlen komen) kost fl. 7.50. En dan heb je dus nogsteeds die "fantastische" muiskabel kwaliteit.
- Een nieuwe laten komen. Da's nog duurder.
- Een joystickverlengkabel doormidden knippen. Bij Bart Smit kosten die dingen fl. 19.95 per twee. En je moet een ronde kabel door een vierkant gat duwen, zonder trek-ontlasting.
- He, had ik niet nog ergens zo'n joystickkabel liggen net alle pinnetjes aangesloten? Ja hoor. En er zit nog een trek-ontlasting op

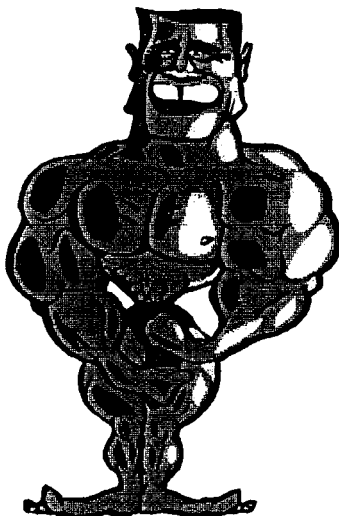
ook. Joepie!

Dus heb ik de oude muiskabel net achter de connector in de muis afgeknipt en die ouwe joystickkabel er voorlopig aan vastgesoldeert. M'n muis doet het. Hoera!

't is wel even wennen aan een muis. Zowel met Operation Blood and met BOMB-DOWN. Minder "Hoera" was, dat ik een dag later, toen ik een tekst zocht, niet alleen die tekst, maar ook de muis die erop lag van de stapel trok. Natuurlijk had ik dit iets te laat in de gaten en de muis maakte een enorme smak. Dat krijg je als ie boven op een rek van twee meter hoog rek ligt. Er braken dan ook een paar zeer kleine stukjes plastic af. En jawel hoor, precies die stukjes die ervoor zorgen dat de muisknoppen op hun plaats blijven zitten. Met een beetje lijm en wat geduld is echter ook dat probleem opgelost.

Mijn muis doet het nu prima. Gebruik ik hem veel? Nee! Maar het is ook meer een verzamelobject aan iets wat ik echt nodig hebt...

CU Mathy Van Nisselroy



# WIE ZIJN ER NOG??

*Door Fred Meijer*

Langzamerhand wordt het steeds stiller in Atari land. Niet alleen voor de XL/XE, maar ook voor de ST. Nederland heeft nog een groot ST-tijdschrift (ST-Nieuws), dat de laatste tijd zeer onregelmatig uitkomt. Ook de ST is nu echt in het hobby circuit terecht gekomen, wat betekent dat het alleen nog maar bergafwaarts zal gaan.... Helaas...

Maar dit is een Atari 8-bit magazine, dus we gaan eerst eens vertellen wie het afgelopen jaar gestopt zijn met het ondersteunen van de Atari 8-bit computer. Allereest wij dus. Door een te klein donateurs aantal en te weinig tijd van bestuursleden (drukke banen!) is het voor Stichting Pokey onmogelijk om door te gaan. Echter wij zijn niet de enige. Ook in Duitsland is het nu kommer en kwel. Ulf Petersen, importeur van onder andere de blackbox in Duitsland, is gestopt met de Atari 8-bit. Markus Roesner, ook wel bekend als Powersoft of metal-freak, heeft vorig jaar zijn bedrijfje opgeheven. Powersoft werd vooral bekend door het distribueren van hoofdzakelijk ANG Software produkten in Duitsland. Een andere zeer bekende naam Kemal Ezcan, oftewel KE-soft, is gestopt met het publiceren van zijn tijdschrift Zong. Ook voor hem werden de kosten te hoog en het aantal actieve leden te klein.

Maar nu het betere nieuws. Met de Abbuc gaat het nog steeds goed. Welliswaar hebben

ook zij te kampen met een dalend ledenaantal, maar dat schijnt zich nu te stabiliseren op zo'n 500 leden, wat toch zeker niet weinig te noemen valt. Zes keer per jaar geven zij een magazine uit op a5 formaat tezamen met een diskette met hoofdzakelijk artikelen die zeer ininteressant zijn voor min of meer gevorderde gebruikers. Onderaan dit artikel vindt u hoe u in contact kunt komen met Abbuc.

Ook in Nederland zijn er nieuwe initiatieven ontwikkeld. Nog nooit heeft Nederland zoveel Atari 8-bit BBS-sen gekend. Deze BBS-sen draaien allemaal met hetzelfde pakket, waardoor het mogelijk is post uit te wisselen. Hierdoor hoeft u (naar mijn inziens gelukkig) maar een van de BBS-sen te bellen om toch alle berichten te kunnen lezen. Ook op internet staat een schat aan informatie voor de Atari 8-bit computer. Denk maar aan 'The Atari 8-bit homepage' van Ivo van Poorten (The Gatekeeper), daarnaast zijn daar het afgelopen jaar de 'Becotel-homepage' en 'FM Atari 8-bit magazine' bijgekomen. Al deze sites zijn Nederlands en zijn met iedere computer geschikt voor internet te bereiken. Ook in het buitenland vindt u een groot aantal interessante pagina's voor Atari 8-bit computers. Onder dit artikel vindt u informatie over deze pagina's.

Ook in Engeland begint het stil te worden.



Echter Micro Discount en Page 6/New Atari User zijn nog steeds actief. Page 6 heeft pas bekend gemaakt nog een jaar door te gaan met het publiceren van hun tijdschrift. Dit tijdschrift op a5 formaat is voornamelijk gericht op de Engelse markt en bevat hoofdzakelijk artikelen die zeer geschikt zijn voor beginnende gebruikers. Er kan gekozen worden voor een abonnement met of zonder diskette.

Er is niet veel meer over, maar er zijn gelukkig nog steeds manieren om op de hoogte blijven op het gebied van de Atari 8-bit. Enkele van die manieren (BBS-sen en Internet) zijn zelfs bijna gratis (alleen de PTT verdient eraan) en zonder verplichtingen.

Hieronder vindt u alle belangrijke gegevens van de initiatieven en groepen die hierboven genoemd zijn.

---

## WIE ZIJN ER NOG OVERZICHT

### *Duitsland:*

ABBUC e.V.  
Wolfgang Burger  
Wieschenbeck 45  
D-45699 Herten

Tel/Fax. 00-49-2366-39623

Tijdschrift kost DM 60,- per jaar inclusief diskette. Een redelijk aantal Nederlanders zijn lid van deze organisatie.

Internet adres: <http://home.t-online.de/home/WBurger>

### *Engeland:*

Page 6/New Atari User  
P.O. box 54  
Stafford  
ST16 1DR

Tel. 00-41-1785-241153

Tijdschrift kost 17 Pond per jaar zonder diskette en 32 Pond per jaar met diskette.

### *Nederland:*

*Internet:* FM Atari 8-bit Magazine: <http://www.meijer.demon.nl>  
BecoTel Homepage <http://www.becotel.demon.nl>  
The 8-bit homepage  
<http://pmw.cs.vu.nl/home/ipoorten/Atari.8bit.Homepage>

*BBS-sen:* BecoTel: 020-6326859 van 18.00 tot 06.00 uur  
STack: 073-6122766 24 uur per dag

# Het hobby-isme is terug...

Een jaar of twaalf geleden, begon ik met een Atari 8-bit computertje te stoeien. Minuten werden uren en uren werden nachten... Mijn bed heb ik in die tijd niet erg veel gezien. Ik begon namelijk met programmeren en dat heeft mij jaren van mijn leven gekost. Sterker nog: er werd een heuse stichting opgericht om ook anderen nachten lang wakker te houden.

Het gevolg was dat ik overdag ongeveer 30 telefoontjes mocht afhandelen tijdens maar ook na de helplijn-uren. Gedurende twee jaar was het een echt gekkenhuis. M'n hobby was een onbetaalde job geworden. Pokey ging bijna failliet, dus ben ik met ANG Software verder gegaan. Beurzen afsjowen, software maken, magazines programmeren. Het hobby-isme was ver te zoeken....

Na twee jaar ANG Software zijn we toch weer verder gegaan met stichting Pokey. Het eerste jaar was een succes. Het tweede jaar hebben we door drukte niet meer af kunnen maken. Jammer maar helaas! Nu zijn we zover dat we echt afscheid moeten nemen....

Het afscheid geeft een dubbel gevoel. 1. Stoppen met Atari 8-bit en 2. geen verplichtingen meer. En juist dat tweede punt spreekt mij zo aan. Atari computeren zonder verplichtingen. Iets dat ik in de afgelopen 7 jaar niet heb kunnen doen. Ik draai nu alweer bijna 3 jaar een PC shopje, maar ik heb met de PC's nog niet zoveel plezier gehad als dat ik had met ons trouwe 8-bittertje... Ik ga daarom de komende jaren weer lekker hobby-matig knoeien met een Atari. Ik ga weer programmeren (voor mezelf oefenen) en ik ga er hardwarematig weer mee aan de slag.

Ik heb al mijn verzamelde 8-bit troep op een hoop gegooid en naar het grof vuil gebracht, behalve de voor mij bruikbare spullen. Een heerlijk gevoel gaf dat. Fred en ik hebben samen oude onderdelen kapot getrapt. Zalig... Een echt afscheid van verplicht onderdelen bewaren, verplicht teksten maken, verplicht software schrijven, verplicht....

Ik ga weer aan de hobby. Dit was mijn laatste artikel... Ik bedank iedereen die de moeite heeft genomen om mijn artikelen te lezen, er op te reageren en er zelfs aan mee te werken. Wellicht tot ziens op een meeting, op internet of waar dan ook... Succes met je hobby!

*John Maris*

The Missing Link  
Manus  
Jomaware



De ABBUC e.V. is de grootste Atari 8-bit gebruikersgroep van de wereld. Op dit moment heeft de ABBUC 543 leden, die nog steeds zeer geïnteresseerd zijn in de kleine Atari. De ABBUC e.V. is 11 jaar geleden in Herten opgericht en daarom komen de meeste leden uit Duitsland. Desondanks wonen er ook een flink aantal leden in het buitenland (ook in Nederland!) met wie de ABBUC haar contacten in het buitenland onderhoudt.

Daarnaast heeft de ABBUC ook meerder Regiogroepen. Deze Regiogroepen bestaan uit geïnteresseerde mensen uit een regio, die meerdere keren per jaar bijelkaar komen en zo de mogelijkheid bieden informatie en ervaringen uit te wisselen. Bovendien beschikt de ABBUC over meerder specialisten die u kunnen helpen bij diverse problemen, zoals printer, computer en disk-drive-problemen.

Vier keer per jaar verschijnt ons tijdschrift op a5 formaat met daarbij een diskette met diverse artikelen en programma's. Daarnaast zijn er ook nog 2 extra uitgaves per jaar. Deze uitgaves bestaan uit een groter programma, die door de ABBUC in licentie uitgegeven wordt.

Al deze activiteiten zijn alleen mogelijk door het grote aantal leden, die door hun financiële bijdrage de ruggegraat voor de komende jaren gevormd hebben. De ABBUC wordt hoofdzakelijk gefinancierd door haar leden, daarnaast ontvangen wij ook geld van bedrijven en prive-personen.

Het lidmaatschap bedraagt DM 5,- per maand, dus dat is DM 30,- per half jaar. U kunt het makkelijkste betalen door middel van een eurocheque. Heeft u die niet, neem dan contact met ons op.

Ons Adres:

Atari Bit Byter User Club e.V.  
c/o Wolfgang Burger  
Wieschenbeck 45  
D-45699 Herten  
Duitsland

Tel. Fax. 00 49 2366 39623

BBS: 00 49 5439 3265

Online : ma. t/m vrij. 19.00 uur - 06.45 uur  
za. en zon. 15.00 uur - 06.45 uur