hcc Commodore interessegroep

45 Info Bulletin 45 years anniversary



Jaargang 31
December 2026
Nummer 6

IN DEZE UITGAVE:

INLEIDING

LAATSTE NIEUWTJES

2-MAANDELIJKSE INTERVIEW

MAARTEN'S COMMODORE CORNER:

REU IN BASIC

ATARI NIEUWS

COLOFON EN AGENDA

Inleiding

Op het moment dat ik dit schrijf staan de hemelsluizen compleet open en vliegen de bladeren door de straat. De herfst is nu volledig van start gegaan en staan bij iedereen de Commodore, Amiga, MSX en Atari computers te snorren. Maar ook in allerlei verschillende vormen en maten, zoals bijvoorbeeld TheC64 mini en de Ultimate. Mijn fietstochten staan nu op een laag pitje, dat was tot een week geleden nog geen probleem. We hebben een prachtig najaar gehad met heel veel zon en we zijn er vaak op uit getrokken. Ook veel wandelingen gemaakt en we zaten begin november zelfs nog buiten op een terrasje te lunchen. Nu gaan we heel hard op weg naar de kerstdagen en een vuurwerkloos oud en nieuw. We zullen zien of dat inderdaad het geval is, ik denk het niet eigenlijk. Bij ons in de straat zijn een paar diehard vuurwerkfans die elk jaar met de grootste dozen aan komen sjouwen. De één nog groter dan de ander. Eerlijk gezegd hoop ik dat ze deze ook dit jaar weer hebben want het is elk jaar weer een prachtige show en het hoort er toch wel een beetje bij. Toch elke keer weer jammer dat een kleine minderheid het voor de rest weet te verpesten.



Er zijn de laatste jaren nogal veel ontwikkelingen geweest. Dat is ook wel te merken aan het bezoek op onze clubdagen, het blijft onverminderd druk en er wordt van alles en nog wat geprogrammeerd en ontworpen. AI heeft daar natuurlijk ook een rol in gekregen, zo werd er op de laatste bijeenkomst in enkele uren een nieuwe game gemaakt voor onder andere de Commodore 64. Het zijn bijzondere tijden, ook voor ons is het afwachten wat er allemaal nog gaat gebeuren op retro computer en game gebied in combinatie met de Artificial Intelligence. Mijn zoon heeft pas geleden één van de allernieuwste 3D printers gekocht en ook daar sta je met open mond naar te kijken. Nog even en dan vraag je aan AI om een computer met operating systeem te ontwerpen en print je deze zelf uit ☺. Maar goed alle gekheid op een stokje, laten we het maar niet teveel aan deze tools overlaten. Het is nog veel te leuk om zelf te knutselen en programmeren.



In deze uitgave staan weer een paar leuke artikelen, al zeg ik het zelf. Zo heeft Maarten weer een mooi artikel gemaakt over REU in Basic en hebben we een interview met onze reparateurtopper Jeroen Vlasveld. We gaan er weer een mooie december kerstclubdag van maken vandaag. We hebben weer een heleboel leuke prijzen bij elkaar gesprokkeld en de hoofdprijzen zijn weer super mooi! Ik wens jullie allemaal een hele fijne kerst en een super goed 2027 toe. We zien elkaar ook weer op zaterdag 20 februari met de ALV en een soldeer workshop.

Uw voorzitter én redacteur, Ron van Schaik

Laatste nieuwtjes



Rob O'Hara heeft een nieuwe podcast gemaakt. Het onderwerp van deze aflevering is het spel Bigfoot (1990) van CodeMasters voor de Commodore 64 computer:
https://podcast.robohara.com/sprite-castle/



Omdat CSDb een beetje moeilijk doet met betrekking tot Ultimate gerelateerde releases en dingen die met behulp van AI gemaakt heeft Richard-William Loerakker een website opgezet waar mensen hun "Ultimate" creaties kunnen showcasen. Gebruikers hebben het beheer over hun eigen data middels een persoonlijke JSON export. Verder kunnen mensen ook wanneer ze geregistreerd zijn hun eigen voorkeuren instellen, wil je geen AI dingen zien? Filter ze er dan gewoon uit. Zie verder op: https://beyond64.org/



Sidelsa C is een nieuw actiespel voor de Commodore VIC20 ontwikkeld door Javier Gonzalez. Verwijder alle blokken om elk level te voltooien. Botsingen met blokken of buitenaardse wezens kosten levenskracht. Het spel vereist een RAM-uitbreiding van 35 kB; het spel bevat 5 verschillende circuits en 3 moeilijkheidsgraden:
https://sleepingelephant.com/ipw-web/bulletin/bb/viewtopic.php?t=11760



Een nieuwe editie van Compute's Gazette is nu verkrijgbaar. In deze editie: The 2026 Holiday buyer's guide, Taking syrinx apart, Runnin' with the devils, Heartbeat Soundtracker, Star Wars, Alien Breed, Disk detective, The Underground BBS en Type-Ins: https://www.computesgazette.com/



North & South is een nieuw spel voor de Commodore Plus/4 computer, ontwikkeld door Proky en siz. Plan je campagne op de strategische kaart, verplaats en combineer legers, verover gebied en bescherm je aanvoerroutes. Wanneer legers elkaar ontmoeten, neem je de controle over van infanterie, cavalerie en artillerie:
https://plus4world.powweb.com/software/North_South

2-maandelijkse Interview

We gaan om de maand een interview doen met één of meerdere van onze clubleden, we starten de serie met ons bestuur en kernleden. Daarna kom jij misschien wel aan de beurt… Dit keer is het een interview met onze geweldenaar met de soldeerbout, kernlid Jeroen Vlasveld.

Wat is je naam, leeftijd, opleiding en achtergrond? Ben je getrouwd of samenwonend, heb je kinderen? Waar woon en (eventueel) werk je?
Ik ben Jeroen Vlasveld, 51 jaar en heb MBO elektronica gedaan. Ben getrouwd en heb 1 zoon en woon en werk in Dordrecht.



Wanneer ben je voor het eerst in aanraking gekomen met een Commodore, Amiga of soortgelijke computer en wat was het er voor één? Wat was de samenstelling van de complete setup? Heb je een eigen plek in huis?
Ik denk dat ik een jaar of 8 a 9 was toen er bij een vriend thuis een Commodore 64 kwam. Deze zat aan een kleuren tv'tje via de antenne kabel en had naast een datasette ook een 1541 drive welke voorzien was van een grote fan op de drive. De C64 en drive hadden ook een speeddos uitbreiding. Helemaal gek van dit systeem ben ik gaan sparen voor een C64, echter ging dit nog niet zo snel. Toen ging mijn vader een computer kopen, maar ik was teleurgesteld toen hij thuis kwam met een Schneider Joyce, dit was een Amstrad PCW 8512 met Schneider merk erop geplakt. Hij was hier heel zuinig op, en ik mocht er niet aankomen (maar dat deed ik natuurlijk wel ☺).

Eindelijk was ik een stuk op weg met sparen en kwam er een 2de hands ZX spectrum op mijn pad. Omdat ik heel graag een computer wilde hebben heb ik deze toen gekocht. Het setje was een ZX spectrum 48Kb met een grote TV en Philips cassette recorder. Daarnaast zat er een losse speaker bij en een "RAM Turbo" joystick interface. Dit alles voor maar 400 Gulden. Na een jaar wist ik met het in en eruit halen van de joystick interface de ZX spectrum te beroven van zijn leven. Deze is toen gerepareerd voor 125 gulden bij ELRA op de Zwartjan straat in Rotterdam dit was daar een "fixed" bedrag voor reparaties van de ZX spectrum. Toen begon de tijd van kranten wijken, ik was rond 12 jaar toen ik begon met het lopen van de krant (de "woensdag") op de woensdag middag, een huis aan huis krantje.

Heel snel hierna kwam er een Commodore 64 met "General electronics" datasette, gekocht van mijn kranten geld. Dit was een 2de hands die ik kocht bij Second Byte in Dordrecht. Echter waren spelletjes die ik deed best moeilijk, de videochip had een foutje waarbij bepaalde kleuren zwart weergegeven werden, dat waren precies de bruine balletjes die je in Jumpman Junior moest pakken. Het duurde even voordat ik dit in de gaten had, omdat de meeste spelletjes er redelijk goed eruit zagen. Ik ben toen met de C64 naar ESCON (niet ESCOM) gegaan, deze deden onder andere de reparaties van Commodore voor Dixons en V&D. Omdat zij de fout ook niet gelijk zagen en wel slechte soldeer verbindingen op de audio/video poort, kreeg ik de C64 terug met de defecte video chip. Toen we hiermee terug gingen mocht ik het probleem laten zien op de reparatie afdeling. Ik had nog nooit zoveel computers bij elkaar gezien, dit heeft toen best indruk gemaakt.



De krantenwijk maakte plaats voor een zaterdag baan als post bezorger bij PTT post. In december 1991 kocht ik mijn eerste Amiga 500 Plus met een 1048S monitor, dit model was toen net uit. Later heb ik nog een Amiga 4000 met 1960 monitor bij elkaar gespaard met een bijbaantje bij KCS. Ik deed hier voornamelijk het afregelen van de Power PC boarden. Rond 1995 ben ik begonnen als reparateur van computers (waaronder Commodore en Amstrad) bij Escon, waar eerder mijn Commodore 64 gerepareerd was. In 1999 ben ik uit de reparatie wereld gestapt en begonnen als systeem beheerder. Hierna ben ik overgestapt naar de PC wereld om pas rond 2010 mijn Commodore hobby weer op te pakken. Ik heb in huis mijn eigen "Hok" (7m2) welke (over)vol staat met diverse voornamelijk Commodore computers.

Wat doe of deed je er mee en waarom? Doe je alleen spellen? Of programmeer je? En op welk systeem en waarmee? Doe je ook dingen met hardware? Uitbouwen? Innovatie? Repareren?

Als kind zijnde waren het voornamelijk spelletjes. Mijn favoriete waren Jumpman Junior, Mission Impossible, Pac-Mania, Zak McKracken en Maniac Mansion. Naast het spelen vond ik het ook leuk om met de Final Cartridge III rond te neuzen in spelletjes en om kleine programmaatjes te maken. Tegenwoordig vind ik de hardware kant het leukst, interfaces/uitbreidingen ontwerpen (en bestaande uitproberen) en het repareren van voornamelijk de 8 bit Commodore. Ik heb twee

websites (myoldcomputer.nl en schematicreference.com), maar het onderhouden hiervan is best tijdrovend.

Kun je een klein beetje (en als het ingewikkeld is, in begrijpelijke taal) inzicht geven in wat je doet of in het verleden hebt gedaan?
Zoals ik al eerder aangaf repareer ik graag, ik zie dit als een soort puzzel waarbij je op zoek gaat naar het probleem door te zoeken wat er mis gaat. Dit is vaak met een diagnose cartridge al te vinden, maar soms moet je met een oscilloscoop zoeken (een oscilloscoop maakt een elektrisch signaal zichtbaar en lijkt een beetje op zo'n scherm waar je in het ziekenhuis de hartslag kan volgen of gewoon een platte lijn ziet als het niet goed gaat met de patiënt).

Heb je nu nog veel soorten Commodore/Amiga, retro computers, randapparatuur? En welke? Kun je er wat over vertellen?
Ik heb voornamelijk 8-bit Commodore computers zoals de C64, VIC20, C16, C116, Plus4, C128 en C128D(cr). Maar ik heb ook nog een paar Amiga's en Spectrums liggen, hiernaast een kleine doos met wat Nintendo spul (NES en diverse Gameboy versies). Mijn meest bijzondere items in de collectie zijn een C64 (silverlabel) met serienummer 65 en een Micropower 2000. Laatste is een dualfloppy drive met seriële IEC poort en een IEEE parallel port (gebruikt door onder andere de PET). De drive lijkt mij een clone van de MSD-2 drive.



Heb je een verklaring waarom Commodore/Amiga en andere retro computers zo populair zijn?
Denk dat en toch een stukje nostalgie is van wat men had toen men jonger was, misschien ook mede doordat de systemen minder ingewikkeld waren en toch wel een eigen karakter hadden. Tegenwoordig is een PC een PC één uit duizend. Hoe vind je de ontwikkelingen binnen onze club (HCC!commodore) op dit moment? Zijn er dingen die je anders zou willen zien of hebben je ideeën? Je bent nu kernlid, ambieer je ooit nog een bestuursfunctie? Zo ja, penningmeester, secretaris of voorzitter?

Bestuursfunctie… Nee, ik ben heel dankbaar voor de personen die deze taak (die ze veel beter dan ik kunnen) op zich nemen. Andere Amibies, niet echt, voel me erg op mijn plek in de reparatie hoek. Ontwikkelingen binnen de club, ik vind de locatie perfect en heb het idee dat de WhatsApp groepen ook mensen meer naar elkaar brengt.

Ben je ook een verzamelaar? En zo ja, waar richt je je op?
Ik vind het leuk om Commodore items te verzamelen, maar doe vaak een stapje achteruit om de andere ook iets te gunnen. Ik richt me voornamelijk in de 8bit spullen van tussen 1980-1990. Ik zou nog graag een PET 8296-D willen, maar als er 1 aangeboden wordt, zal ik hem waarschijnlijk toch niet nemen, de enige ruimte die ik er nog voor heb is namelijk mijn bureaustoel…



Hoe kijk je tegen de laatste ontwikkelingen aan, C 64 Ultimate, TheC64, THEA1200, Apollo A6000, ApolloOS, The Commodore Callback (als je hier wat van af weet).
Ik vind het een leuke ontwikkeling, je ziet hieraan voornamelijk dat de Commodore en Amiga toch een goede fan base heeft, en uiteraard zijn de "moderne" machines voor de één een uitkomst (software ontwikkeling/makkelijk aan te sluiten op moderne beeldschermen/lekker snel met veel geheugen) terwijl het voor de andere niet het "echte" werk is. En wat betreft de callback… tja,.. persoonlijk dacht ik dat het een grap was. Maar ik moet ook wel bedenken dat het voor Commodore nu best moeilijk is om iets uit te brengen dat zowel "nieuwe" klanten en door de retro community interessant gevonden wordt.

Ben je weleens op andere Commodore (of retro computer) dagen geweest (in binnen- of buitenland)? Misschien heb je nog wat leuke anekdotes of dingen die je hebt meegemaakt in het buitenland of op reis naar een buitenlandse party?

Niet echt retro, maar toch wel een herinnering, ik ging vroeger wel eens naar een computer club in Maassluis, daar woonde een klasgenoot. Deze club organiseerde elk jaar een busrit naar computer beurs in Hannover. Echter was de beurs op een bepaald moment veranderd en kon men geen kaarten meer vooraf kopen. Dit was wel een issue omdat we niet wisten of iedereen dan binnen kwam (lange rijen voor kassa in combinatie met de tijd die we hadden voor we weer met de bus terug moesten). Dus hadden we een trip naar een computer beurs in Londen geregeld…

inclusief één overnachting. Wij naar Engeland, waar we overnacht hebben en genoten van een Engels ontbijt. Daarna met zijn allen op naar de beurs… bleken er maar 5 kraampjes te staan…. Maar we hebben toch een leuke dag gehad, maar vooral in de pub en een tour door Londen. Ik ben één keer op een Amiga party in Nederland geweest die georganiseerd was door Orgin (Amiga groep). Hier zat toen een kennis (Nico) in, die ik kende van een Commodore club in Dordrecht en waar ik regelmatig mee in de trein richting school zat. Ik ben hier geweest samen met Remi (van REKO Productions), dit was wel een bijzondere ervaring. Buiten dit ben ik al een paar keer bij de BAC (Belgian Amiga Club) geweest. Zij organiseren eens per jaar een clubdag in Antwerpen.



Zijn er nog dingen die je wil doen in de toekomst? Heb je nog ideeën die wij als club kunnen doen?
Dingen die ik nog zou willen doen in de toekomst… Misschien het bezoeken van de X party. Ideeën voor de club… verder dan een soort "ruil hoek" waar mensen dingen kunnen ruilen, of een gezamenlijk tripje naar een computer party kom ik niet.

Je bent altijd druk in de reparatiehoek en met de soldeer workshop, heb je daar nog één of meerdere leuke ervaringen mee gehad die je met ons wil delen?
Komt niet direct een specifiek iets in gedachten, wat ik in de tijd dat ik in de reparatie hoek zit wel merk, is dat veel mensen de computer hobby op hun eigen manier beleven. De één verzamelt, de ander programmeert en de ander hobby't met de hardware. De één wil een functioneel systeem die uitgebreid is, terwijl anderen weer trouw blijven aan de traditionele configuraties en/of toebehoren. Wat ik ook zie is dat mensen soms ook meerdere systemen hebben zoals een Commodore en een Atari, dit zag je toen de systemen uitkwamen niet echt. Het was toen vaak dat men de "vijand" probeerde te overtuigen waarom een bepaald systeem beter was dan die van de ander.

Ik wil je bedanken voor je tijd en medewerking aan dit leuke interview!
Ron van Schaik

Maartens Commodore Corner

Deze maand: REU in Basic



Dit artikel is een verkorte Nederlandse versie van een Engels artikel. Dat bevat meer details en bijvoorbeeld ook source files. Zie:

https://github.com/maarten-pennings/C64howto/blob/main/reu/readme.md

Introductie

In 1985 heeft Commodore de eerste REU (RAM Expansion Unit) op de markt gebracht. Een REU breidt het geheugen van de Commodore 64 behoorlijk uit, met 128 kB (REU 1700), 256 kB (REU 1764), of zelfs 512 kB (REU 1750). Ook moderne systemen zoals VICE, de Commodore Ultimate, The C64, BMC64, en Kung Fu Flash 2 hebben ondersteuning voor een REU. In dit artikel bekijken we hoe een REU werkt, en we schrijven een BASIC programma ("10 PRINT") dat de REU gebruikt.

Een REU gebruikt geen banking mechanisme. We spreken van banking wanneer er meerdere RAM chips op hetzelfde adres gebied zitten, en er een mechanisme is om te kiezen welke chip (welke bank) op enig moment actief is. Bij een banking oplossing kiest de CPU, in ons geval de 6510, welke bank actief is, en daarna kan de CPU de bank lezen en schrijven.

De REU werkt anders. Een REU is een memory mapped I/O device. Het wordt bediend met registers. Via deze registers geeft de 6510 commando's (met POKEs) om data uit het C64 geheugen ergens in het REU geheugen te stashen, of om data uit het REU geheugen te fetchen en ergens in het C64 geheugen te plaatsen. Er zijn nog twee andere commando's: *swap en compare* maar die zullen we in dit artikel niet bekijken. We gebruiken de term kopiëren als overkoepelend begrip voor alle 4 de commando's.

De 6510 kan dus niet het geheugen van de REU lezen of schrijven. Het kan alleen een kopieer opdracht aan de REU geven. De REU is een soort DMA device (*direct memory access*) dat data kopieert tussen de REU en de C64. Dat klinkt langzaam, en ten opzichte van *bank switching* is het dat ook. *Bank switching* kost een of twee 6510 instructies. Maar in de praktijk is de REU oplossing snel. De REU kopieert op

C64 klok snelheid: elke tik van de ongeveer 1MHz klok wordt 1 byte gekopieerd. Dat betekent dat 1 miljoen bytes in 1 seconde worden gekopieerd, ofwel 1000 bytes in 1 ms. Het hele C64 geheugen kopiëren kost dus 64 ms.

Registers
De REU registers zijn *memory mapped* in de "I/O 2" regio; die begint op $DF00. De REU heeft 7 registers variërend in grootte van 1 tot 3 bytes.

Register	Size	Offset	Hex	Dec
`status`	1	0	DF00	57088
`command`	1	1	DF01	57089
`c64base`	2	2,3	DF02-DF03	57090-57091
`reubase`	3	4,5,6	DF04-DF06	57092-57094
`translen`	2	7,8	DF07-DF08	57095-57096
`irqmask`	1	9	DF09	57097
`addrctrl`	1	10	DF0A	57098

Het status register is eigenlijk niet belangrijk. Het geeft aan of de REU al klaar is met het uitvoeren van het kopieer commando (via bit 6). In de praktijk blokkeert (*stalls*) de REU de 6510 tijdens het kopiëren. Als de 6510 dus weer begint te lopen is het niet nodig het status register te bekijken; de REU is klaar. We slaan command even over en springen naar registers c64base, reubase en translen. Het register c64base is het start adres van de kopieer actie aan de C64 kant, het register reubase is het start adres van de kopieer actie aan de REU kant en translen is het aantal te kopiëren bytes. Merk op dat c64base en translen beide 2 bytes zijn en dus de hele 64 kB van de 6510 omvat, terwijl reubase zelfs 3 bytes is, zodat het REU geheugen maximaal 256 × 64 kB kan zijn (16777216 bytes of 16 Mbytes).

Terug naar command.

Bits	Function	Details `command`
7	EXECUTE	1 start een kopieer actie (van type TRANSFERTYPE)
6	reserved	
5	LOAD	1 = `c64base`, `reubase`, `translen` zet start waardes terug
4	NOFF00	1 = start meteen, 0 = start na schrijven naar $FF00
3:2	reserved	
1:0	TRANSFERTYPE	00=*stash* (C64→REU), 01=*fetch* (REU→C64), 10=*swap*, 11=*compare*

Het command register bestuurt de kopieer actie. De onderste twee bits TRANSFERTYPE bepalen welk van de 4 kopieer acties wordt uitgevoerd. Het EXECUTE bit (bit 7: 128) start de actie. Tijdens de kopieer actie zal de REU de registers c64base, reubase stap voor stap ophogen en translen aflagen. Als LOAD (bit 5: 32) hoog is zal de REU de startwaardes herstellen aan het einde van de kopie. De vlag NOFF00 stelt je in staat de start nog even uit te stellen; die begint pas na een schrijfactie naar $FF00. Dit is nodig in één speciaal geval: als de REU het C64 geheugen moet lezen of schrijven, dat *onder* het I/O 2 gebied ligt, dan is er na de *execute* een geheugen bank wissel nodig, gevolgd door een schrijfactie naar $FF00.

Het irqmask register geeft de mogelijkheid een interrupt te genereren als de kopieer actie klaar is. Zoals bij status vermeld *stallt* de REU de 6510 tot het kopiëren klaar is dus interrupts zijn niet nodig.

Het laatste register addrctrl bepaalt of de registers c64base en reubase tijdens de kopieer actie opgehoogd worden of niet. Bit 7 (128) is voor c64base en bit 6 (64) voor reubase. Niet ophogen is zinvol als er een *memory fill* nodig is: steeds hetzelfde byte schrijven naar alle adressen in het doel geheugen. Zie het Engelse artikel op GitHub voor meer details en BASIC voorbeelden voor alle registers.

10 PRINT met REU - ontwerp

Nu we weten hoe de REU werkt gaan we hem gebruiken in een BASIC programma. We gaan het klassieke 10 PRINT programma nabouwen. 10 PRINT omdat iedereen het kent en omdat het goed past bij de mogelijkheden van de REU.

10 PRINT

Het klassieke 10 PRINT programma bestaat uit één regel:

```
10 PRINT CHR$(205.5+RND(1));:GOTO 10
```

Het gebruikt de RND(1) functie om een willekeurig getal tussen 0 en 1 te genereren. Hierdoor ligt 205.5+RND(1) tussen 205.5 en 206.5. De CHR$() functie kapt het argument af naar 205 of 206, waardoor een \ of / streep wordt afgedrukt. De GOTO 10 vormt een lus, wat leidt tot het afdrukken van oneindig veel schuine strepen, zodat er een doolhof patroon ontstaat.

REU en 10 PRINT

De REU kan snel 1000 tekens kopiëren van zijn geheugen naar het C64-geheugen. We zullen de REU opdracht geven te kopiëren naar $0400, het scherm geheugen van de C64. Zoals we eerder zagen, de REU kopieert 1000 tekens, één scherm, in 1 ms.

Het idee is dat we veel schuine strepen van het 10 PRINT-programma opslaan in het REU-geheugen. We slaan ze op vanaf adres 0 ($000000) in de REU. Voor de animatie halen we 1000 bytes op vanaf REU adres 0 en plaatsen deze op C64 adres $0400 (het scherm). Daarna halen we 1000 bytes op vanaf REU adres 40 en plaatsen die opnieuw op $0400. Dit scrolt het scherm één rij omhoog. Vervolgens halen we 1000 bytes op vanaf REU adres 80 en plaatsen die op $0400; weer *scrollt* het scherm een rij omhoog. En zo verder.

Dit gaat snel. Het is niet meer nodig om strepen te berekenen en ook de *kernel-scroll* routine hoeft niet meer aangeroepen te worden. Elke kopie duurt slechts 1 ms (we hebben nog wel een paar BASIC POKE opdrachten nodig om de kopie te starten).

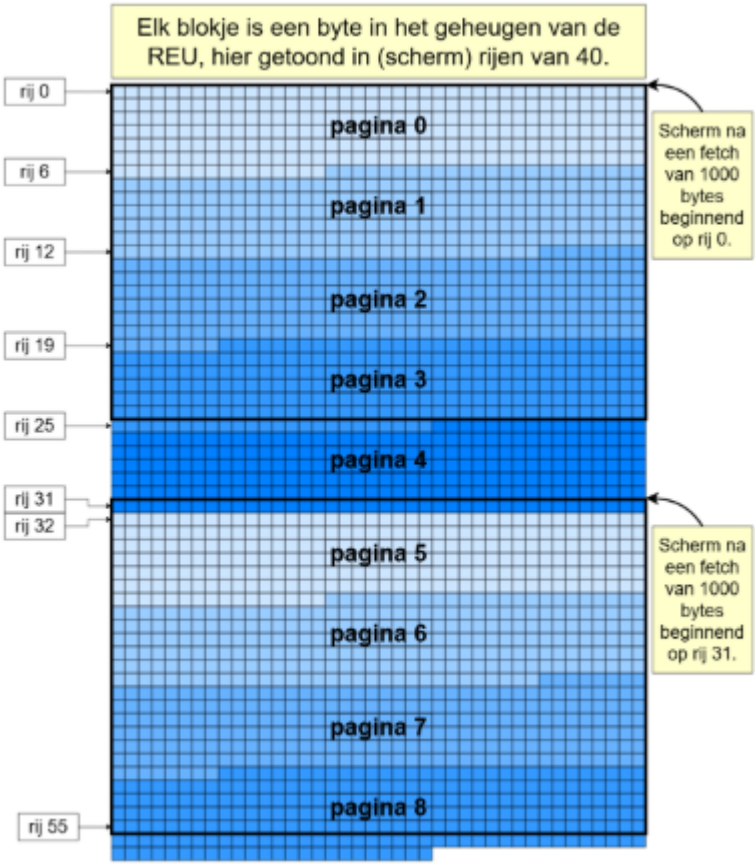
REU geheugen indeling

Er zijn twee problemen met de hierboven geschetste REU versie van 10 PRINT. Het eerste probleem is dat elke schuine streep, in de REU moet staan; we moeten ze dus toch allemaal berekenen. Om dit op te lossen, spelen we een beetje vals. We genereren slechts 256 schuine strepen en slaan meerder kopieën van dit setje achtereenvolgens op in het REU geheugen. Als je goed kijkt kun je dat zien, maar 256 schuine strepen is 6 scherm rijen (van 40) plus nog 16 tekens. Hierdoor is de tweede reeks van 256 schuine strepen met 16 tekens verschoven. Dit maakt het lastiger om het herhalende patroon te zien. Goed genoeg voor onze eenvoudige demo. Vanaf nu noemen we een reeks van 256 schuine strepen één pagina.

Het tweede probleem is dat we willen dat de animatie oneindig door loopt, net als in het originele 10 PRINT programma. Het is duidelijk dat we geen oneindige hoeveelheid (identieke) pagina's van 256 schuine strepen in het REU geheugen kunnen hebben. Hier schiet de wiskunde ons te hulp. Het kleinste gemene veelvoud van de rijgrootte 40 en de paginagrootte 256 is 1280. We kunnen precies 5 pagina's (van 256 tekens) kwijt in 1280 bytes, én we kunnen precies 32 rijen (van

40 tekens) kwijt in 1280 bytes. Als de REU gevuld is met voldoende pagina's en we beginnen met het ophalen van (1000 bytes vanaf) rij 0 (adres 0), vervolgens 1000 vanaf rij 1 (adres 40), ..., en op een gegeven moment 1000 bytes vanaf rij 32 (adres 1280), dan bereiken we een cyclus: de bytes (vanaf) rij 32 zijn gelijk aan die op rij 0. Dus na het ophalen van het scherm vanaf rij 31, halen we het volgende scherm niet op vanaf rij 32, maar springen we terug en halen we het scherm weer op vanaf rij 0. Dat is immers identiek aan dat op rij 32.

Het laatste scherm dat we ophalen begint op rij 31 en is 1000 bytes groot. Zoals het geheugen plaatje hieronder laat zien, hebben we dus 9 pagina's in de REU nodig om genoeg schuine strepen te hebben om helemaal naar rij 31 te scrollen. Wij denken dat dit genoeg variatie biedt voor onze demo.



10 PRINT met REU - BASIC programma

We bespreken nu het BASIC programma 10PRINT-WITH-REU. We gebruiken de petcat conventie om speciale symbolen als "wissel naar licht blauw" af te drukken als {lblu}. Ook hakken we bij de bespreking het programma op in blokken. Alle blokken achter elkaar vormen het uitvoerbare programma.

Kleuren

```
100 r=57088:rem 10print-with-reu
110 poke 53280,14:poke 53281,6
120 print "{clr}{lblu}";tab(54);"please wait"
```

Het eerste blok regelt de kleuren. Regel 100 is de uitzondering; deze definieert variabele R als het basisadres van de REU en het documenteert de naam van het programma. Zoals uitgelegd gebruiken we het *fetch* commando van de REU om 1000 (schuine) strepen te kopiëren uit het REU-geheugen naar het schermgeheugen van de C64. Om de strepen zichtbaar te maken, moet ook het *kleurengeheugen* worden gevuld. Welke kleur zullen we gebruiken? De ontwerpkeuze is dat we de C64 standaard kleuren gebruiken. Regel 110 zet de rand (53280) op lichtblauw (kleur 14) en de achtergrond van het scherm (53281) op donkerblauw (kleur 6). Tot slot zet regel 120 de cursor of voorgrond kleur op lichtblauw ({lblu} kleur 14) en drukt PLEASE WAIT af. Hiermee zijn alle standaardkleuren ingesteld.

Schuine strepen

```
200 print "{blu}";
210 print chr$(206);
220 for i=0 to 254
230 :print chr$(205.5+rnd(1));
240 next
250 print "{lblu}"
```

Het tweede blok drukt de 256 (schuine) strepen af op de bekende 10 PRINT manier. We willen echter niet dat de gebruiker ze (nu al) ziet. Daarom schakelt regel 200 de cursor kleur om naar donkerblauw (6, {blu}) dezelfde als de achtergrond. Regel 250 schakelt weer terug naar lichtblauw (14, {lblu}). Eén aspect is bijzonder: het allereerste teken van de 256 wordt door regel 210 geforceerd ingesteld op 206 (/). Toevallig is 206 AND 15 gelijk aan

14. We hergebruiken later het eerste teken van de pagina om de REU het kleurengeheugen te laten vullen (via *fill*) met 1000 bytes met de waarde 206 (zie regels 500-550 verderop). Omdat het kleurengeheugen alleen de onderste 4 bits van 206 opslaat, krijgt het kleurengeheugen 1000 keer de waarde 14, wat lichtblauw is, onze gekozen cursor kleur.

Pagina's *stashen*

```
300 poke r+2,80:poke r+3,4
310 poke r+4, 0:poke r+5,0:poke r+6,0
320 poke r+7, 0:poke r+8,1
330 poke r+9,0
340 poke r+10,0
350 for i=0 to 8
360 :poke r+5,i
370 :poke r+1,128+32+16+0
380 next
```

Dit blok slaat de pagina van 256 bytes aan schuine strepen negen keer op in het REU-geheugen. We configureren de REU via de registers.

- Regel 300 stelt de MSB van c64base in op 4 (het schermgeheugen begint op $0400). De LSB wordt ingesteld op 80 omdat vanwege de PRINT TAB(54);"PLEASE WAIT", de (schuine) strepen beginnen op de derde rij, oftewel met een *offset* van 80.
- Regel 310 stelt reubase in op $000000. Maar in de FOR-NEXT lus (regel 360) wordt de middelste byte overschreven met het paginanummer in de REU.
- Regel 320 stelt de translen in op $0100, wat precies 1 pagina (256 bytes) is.
- We willen geen interrupts (regel 330) en zetten dus irqmask op 0.
- We willen wél dat de adressen automatisch ophogen (regel 340) en zetten addrctrl dus op 0.
- De lus heeft 9 iteraties (regel 350), waarbij steeds een pagina wordt gekopieerd. Regel 360 selecteert de bestemmingspagina in de REU, en regel 370 is de daadwerkelijke *stash* opdracht; we zetten deze bits 128 (EXECUTE) + 32 (LOAD) + 16 (NOFF00) + 0 (STASH). De

LOAD is nodig omdat elke *stash* met dezelfde *c64base* moet beginnen.

REU aanwezig?

```
400 if (peek(r) and 64) = 64 then 420
410 print "error:reu expected":end
420 if (peek(r) and 64) <> 0 then 410
```

Het doel van dit blok code is om te controleren of de *stashes* gelukt zijn; is er een REU actief in de C64. Na de *stashes* van het vorige blok zou de vlag STATUS.ENDOFBLOCK in het status register hoog moeten zijn. Dit wordt gecontroleerd in regel 400. Als dat niet zo was, drukt regel 410 een foutmelding af en stopt het programma. Er is dan hoogstwaarschijnlijk geen REU. Als de vlag wél hoog was, moet deze nu laag zijn, aangezien STATUS.ENDOFBLOCK gewist wordt bij lezen (*clear on read*). Is dat niet het geval, dan volgt er een sprong terug naar regel 410 om een foutmelding af te drukken en te stoppen.

Kleurgeheugen *fill*

```
500 poke r+2,0:poke r+3,216
510 poke r+4,0:poke r+5,0:poke r+6,0
520 poke r+7,0:poke r+8,4
530 poke r+9,0
540 poke r+10,64
550 poke r+1,128+32+16+1
```

De laatste voorbereidingsstap voordat de animatie begint, is het vullen van het kleurengeheugen van de C64. Dit is een *fetch* (geen *stash*), maar dan zonder het ophogen van reubase, waardoor de *fetch* als een *fill* werkt.
We bespreken weer hoe alle REU-registers (opnieuw) beschreven worden.

- Regel 500 stelt c64base in op 216 (MSB) en 0 (LSB), oftewel $D800 of 55296, het kleurengeheugen.
- De reubase wordt $000000 (regel 510) , de locatie waar regel 200 de waarde 206 ($CE) heeft neergezet. De onderste *nibble* is dus $E (14, oftewel lichtblauw).

- In regel 520 zijn we lui en stellen translen in op 4 pagina's (1024 bytes), waar 1000 voldoende zou zijn.
- Wederom willen we geen interrupts (regel 530).
- De addrctrl is bijzonder: we willen wel het ophogen voor c64base, maar we willen een vast adres voor reubase (regel 540). Dit is een *fill* configuratie.
- Regel 550 voert de daadwerkelijke *fetch* uit met deze bits: 128 (EXECUTE) + 32 (LOAD) + 16 (NOFF00) + 1 (FETCH). De LOAD vlag is niet nodig, maar kan ook geen kwaad.

De animatie

```
600 poke r+2,0:poke r+3,4
610 poke r+4,0:poke r+5,0:poke r+6,0
620 poke r+7,0:poke r+8,4
630 poke r+9,0
640 poke r+10,0
650 for i=0 to 31
660 :a=i*40:ah=int(a/256):al=a and 255
670 :poke r+4,al:poke r+5,ah
680 :poke r+1,128+32+16+1
690 next:goto 650
```

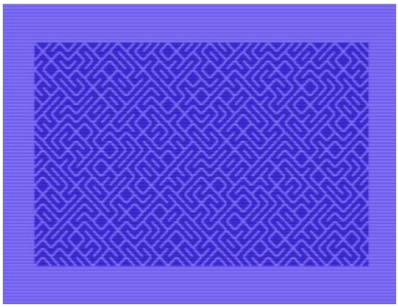
Dit laatste blok zorgt voor de animatie. Alle REU-registers worden (nogmaals) beschreven.

- Regel 600 stelt c64base in op 4 (MSB) en 0 (LSB), oftewel $0400 of 1024, het schermgeheugen.
- Regel 610 stelt reubase in op $000000, maar de lus overschrijft de twee onderste bytes (regel 670) met het rij adres in de REU.
- In regel 620 zijn we opnieuw lui en stellen we translen in op 4 pagina's.
- Wederom willen we geen interrupts (regel 630) en willen we wel weer dat de adressen automatisch ophogen (regel 640).
- Regel 650 laat I over de 32 rijen itereren die zich in het REU-geheugen bevinden.
- Regel 660 berekent eerst het adres A van rij I in de REU, en splits A vervolgens op in een hoge byte (AH) en een lage byte (AL). In regel 670 worden deze gebruikt om de reubase in te stellen.

- Regel 680 voert de daadwerkelijke *fetch* uit met deze bits: 128 (EXECUTE) + 32 (LOAD) + 16 (NOFF00) + 1 (FETCH). De LOAD is nodig omdat elke *fetch* hetzelfde *c64base* adres als bestemming heeft.
- Regel 690 zorgt ervoor dat we, na het kopiëren van het "scherm beginnend bij rij 31", weer teruggaan naar het "scherm beginnend bij rij 0". Dit wordt bereikt door de GOTO 650 na de NEXT.

Opmerkingen over het BASIC programma

De animatie haalt één compleet scherm op, rij voor rij. Dit geeft de indruk van scrollen, zonder gebruik te maken van de scroll functie van de kernel. Het resultaat is dat er, in tegenstelling tot de originele 10 PRINT, nooit twee lege rijen onderaan het scherm staan, ook nooit één lege rij, en dat er zelfs geen lege schermpositie is in kolom 40 op rij 25.



Ik heb T0=TI toegevoegd vóór de FOR - lus op regel 650, en T1=TI net ná NEXT (regel 690). Het afdrukken van T1-T0 laat zien dat het animeren van 32 rijen 86 jiffies kost, ofwel dat het 1,5 seconde duurt. Dat is 45 ms per fetch/scroll. In een tweede experiment zet ik ook nog R=49152 voor de lus. Dit zorgt ervoor dat alle BASIC code hetzelfde is, maar dat de REU geen commando's uitvoert. Nu kosten de 32 iteraties 84 jiffies. Het verschil is 2 jiffies ofwel 2/60*1000 = 33 ms voor 32 kopieer acties. Kortom, 1 REU commando voor een 1000 byte fetch kost inderdaad de 1 ms die we in de inleiding schatte. De animatie is veel te snel. Ik wilde hem vertragen door een WAIT 53265,128 in te voegen voor de raster positie van de VIC-II. Dat werkte niet. Ik gebruik nu een eenvoudige wachtlus 685 FOR T=0 TO 222:NEXT als ik een tragere scroll wil.

Conclusie

In eerste instantie leek de REU een ingewikkelde uitbreiding. Na enige bestudering bleek hij eenvoudig maar slim in elkaar te zitten. Hij blijkt zelfs vanuit BASIC goed te gebruiken. En hij is snel.

(end)

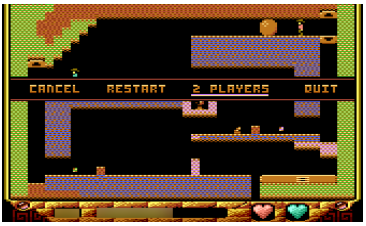
Atari nieuws (door Dennis Roos)

Laatste loodjes



Het jaar is alweer bijna voorbij. Gelukkig hebben we de laatste clubdag nog waar het in December altijd extra druk is, met natuurlijk de eindejaarsloterij waar weer mooie prijzen worden verloot. Als we terugblikken op het afgelopen jaar, was dit vooral het jaar waarin AI slop een veel gebezigde term was. Het resultaat van de AI opmars is dat de homebrew scene wordt overspoeld met nieuwe releases en het bijhouden ervan een monsterklus is geworden. Er zijn al wat hobbyisten gestopt met het volgen en publiceren van releases. Ook wij testen lang niet alles wat er uitgebracht wordt, maar een goed product is een goed product en daar houden wij jullie nu en in de toekomst van op de hoogte.

Evenementen



De Silly Venture Winter Editie heeft weer plaatsgevonden in Gdansk en de resultaten zijn bekend: https://tinyurl.com/sve2026we Voor bijna elk Atari platform vind je een release terug in de lijst. Dit jaar zagen we ook een aantal Atari 8-bit inzendingen specifiek voor de VBXE, een grafische kaart voor de Atari 8-bit computers. Op een standaard Atari 8-bit niet te spelen, maar onder emulatie gaat dat prima. Ook de ABBUC competities zitten er weer op: https://tinyurl.com/ABBUC2026SC De winnaar, Spectrum Gem is een puzzel/platform spel waar je menig verloren uurtje plezier van beleeft. Andere releases, zoals Marian (Giana Sisters), Great Axe (Golden Axe) en Endurance racer vielen bij ons in de smaak. Meer informatie en links naar deze en veel meer nieuwe titels vind je in de lijsten van Zero Page Homebrew op het Atari Age forum: https://tinyurl.com/2026ZPH2600

A8 Pico en Competitie



Dit jaar geen inzending van F#Ready en Fred_M voor de ABBUC Software competitie, maar een inzending voor de hardware competitie: De 'Rotary controller'. Het spel RotoR werkt al met deze 'spinner'. Het leverde ze een derde plek op, achter de 'Smart Reset+' en de 'Insane Pokey'. Een ander nieuwtje dat we jullie niet willen onthouden is de ontwikkeling van verscheidene firmware varianten voor de goedkope A8PicoCart: 'eightbit Enhanced Firmware' met screensavers, muziekspelers en meer. 'Mean Hamster Power Engine' met emulatoren, Doom, etc. en Ivo_P's RAM cart om er een paar te noemen. In het Atari 8-bit subforum van Atari Age vind je deze en nog een paar andere zelf terug.

Colofon

Bestuur:
Ron van Schaik, voorzitter
Marien Kaptein, penningmeester
Robert Sprokholt, secretaris

Kernleden:
Hans Kessels
Ruud Baltissen
Antony Mo
Erwin van Betten
Jeroen Vlasveld

Kidscorner:
Rinus van de Reep
Youri Wagenaar

Redactie:
Ron van Schaik

Websites, social media, e-mail:
https://commodore.hcc.nl/
http://twitter.com/commodoreig
http://www.youtube.com/commodoregg
http://www.facebook.com/commodoreclubnederland
https://www.instagram.com/commodoreclubnederland
Email: bestuur@commodore.hcc.nl

Dit magazine verschijnt twee maandelijks op de derde zaterdag van de even maand.

Agenda

20 februari 2027 10:00 – 16:00 uur
Clubdag met ALV en Soldeer Workshop in ‘t Schuurtje, Maarssen

26 t/m 28 februari 2027
BCC Party #21 Berlijn, Duitsland
https://csdb.dk/release/?id=264783

12 t/m 14 maart Fioniadata 2027
Demo Party/Meeting
Langeskov, Denemarken
https://www.fioniadata.dk/

19 t/m 21 maart Forever 2027
Demo Party/8 bit Party
Trnava, Slowakije
https://forever.zeroteam.sk/

17 april 2027 10:00 – 16:00 uur
Clubdag met Game Competitie in ‘t Schuurtje, Maarssen

