

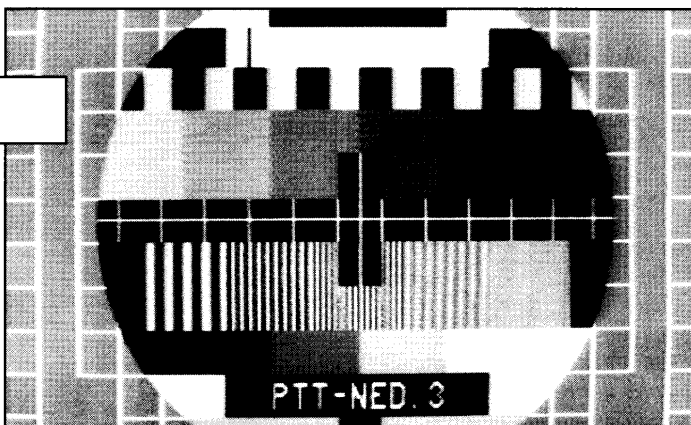
Digitaliseren van videobeelden via UNIFACE

Het kon eigenlijk ook niet anders na de fotoscanner: ook een video-digitiser moest er komen en dan liefst met dezelfde beeldkwaliteit. Daarom heb ik de soldeerbout maar weer eens warm gestookt. Het resultaat is een klein printje met elektronica en een programma. Daarmee kunnen videobeelden van televisie, videorecorder en videocamera gedigitaliseerd worden om ze vervolgens op het beeldscherm als zwart-wit foto te tonen. De beelden zijn op disk te bewaren en op een grafische printer af te drukken. Uiteraard werkt ook de video-digitizer via UNIFACE, want dat is de eenvoudigste manier en bovendien hebt u UNIFACE toch al in huis.

Het printje maakt het binnenkomende videosignaal geschikt om gedigitaliseerd te worden met de ADC-8 kaart. Zelfbouwers met een MSX-2 kunnen direct aan de slag, want zowel schema als programma worden hierbij afgedrukt. Degenen die wat minder handig zijn met de soldeerbout kunnen binnenkort een kant-en-klare UNIFACE-kaart kopen, waarop digitiser en ADC zijn samengevoegd. Programma's aangepast voor gebruik op andere Philips' thuiscomputers komen zo spoedig mogelijk beschikbaar.

Prestaties

In deze videodigitiser werd bewust gekozen voor een zwart-wit plaatje. Niet omdat kleur te moeilijk zou zijn, maar gewoon omdat met grijstinten mooiere plaatjes te maken zijn dan met het beperkte kleurengamma van de nu bestaande thuiscomputers. Dat zal ik op een andere plaats nog wel eens nader toelichten. Bij dit artikel treft u enkele voorbeelden aan van de vele plaatjes die ik met mijn digitiser opgenomen heb. Daarbij werd gebruik gemaakt van een camera of van een videorecorder met stilstaand beeld. Het direct van de televisie overgenomen testbeeld geeft een goede indruk van de kwaliteit van de digitiser. Omdat een MSX slechts 212 lijnen kan afbeelden ontbreekt aan de boven- en onderkant van



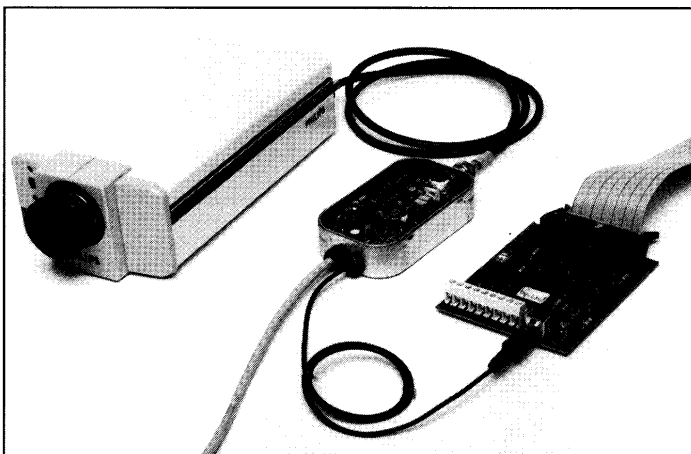
Schermafbeelding van een testbeeld

het beeld een stukje. Het achtergrond-raster laat zien dat de lineariteit helemaal in orde is. Van de lijnrasters zijn die van 1, 2 en 3 MHz goed zichtbaar. Let eens op de verticale lijntjes in het beeld: die zijn netjes recht zonder bibbers of hakkels. Hieruit kan afgelezen worden dat de nauwkeurigheid van het sampletijdstip zeer goed is. Vergelijk zelf maar eens met plaatjes van andere digitisers.

Wat er al was

Toen Philips in 1987 een MSX-computer aankondigde die externe videobeelden kon digitaliseren, was de belangstelling voor deze zo genoemde "video-computer" zeer groot. Dat ver-

anderde wel wat toen we de eerste resultaten van deze NMS 8280 te zien kregen. De kwaliteit van de gedigitaliseerde bleek een echte afknapper. Slechts drie bits voor rood en groen en niet meer dan twee bits blauw bleek deze aan te kunnen. Het bleek helaas niet mogelijk met méér bits in zwart-wit te digitaliseren. De belangstelling voor de NMS 8280 nam flink af, maar de interesse in een betere digitalisatie bleef bestaan. Dat bleek wel toen er voor enkele andere computers (zoals Apple en Archimedes) zelfbouw digitisers werden gepubliceerd. Deze werken in grijstinten en haalden daarmee goede resultaten. Wel viel op dat er grote printplaten vol met tientallen IC's no-



Videocamera + Digitiser + ADC-8 kaart

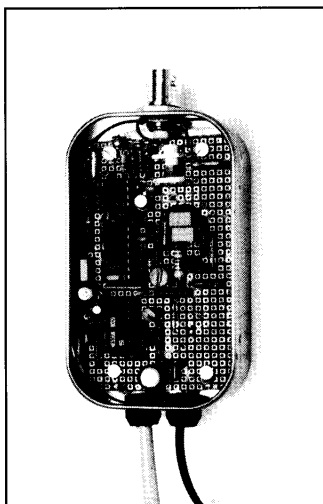
dig bleken te zijn. Toen ik op de PTC Open Dag van april dit jaar mijn foto-scanner liet zien, was één van de meest-gehoorde opmerkingen, dat er eigenlijk ook een digitiser zou moeten zijn in dezelfde beeldkwaliteit. Die wens heb ik goed in mijn oren geknoopt en kort daarna ben ik aan de slag gegaan. Uitgangspunt was het gebruik van UNIFA-CE en beslist niet méér elektronica dan nodig is, zoals u dat van mij gewend bent. Dat is dan gelukt.

Het videosignaal

Waarom kunnen we het videosignaal eigenlijk niet direct op de ingang van een ADC-kaart zetten? Omdat de computer en de ADC-kaart het tempo, waarmee het videosignaal er beeldpunten uitgooit (zo'n vijf miljoen per seconde) op geen stukken na bij kunnen houden. We hebben daarom óf razendsnelle en dus heel dure elektronica nodig die de videosnelheid wél bij kan houden, óf we moeten een truc gebruiken waardoor de computer en ADC-kaart de tijd krijgen om in hun eigen tempo te kunnen werken. Natuurlijk doen we het laatste. De truc is overigens al heel oud, want die stamt nog uit de tijd dat er nog helemaal geen voldoende snelle elektronica bestond, maar wordt ook nu nog in veel andere video-digitisers toegepast. Om te begrijpen wat er gebeurt bekijken we eerst even hoe een videosignaal er uit ziet. Een veel uitgebreidere beschrijving staat overigens in het artikel over monitoren in PRINT nr. 20. Het Europese televisiebeeld wordt 50 keer per seconde uit 312,5 horizontale lijnen opgebouwd. Dat zijn 15625 lijnen per seconde, dus elke lijn duurt 64 microsec (een microseconde is het miljoenste deel van een seconde). Daarvan is nog eens 12 microsec nodig voor de lijnterugschlag en de horizontale synchronisatie-impuls (HSYNC). Blijft over 52 microsec voor de over te dragen beeldpunten op de lijn. Wanneer dat er 256 zijn, dan worden elke microsec vijf beeldpunten aangeboden. Dat kan de microprocessor echt niet volgen, want die heeft voor één enkele instructie al meer dan een microseconde nodig.

De truc

De truc maakt gebruik van het feit dat het beeld telkens weer opnieuw geschreven wordt. Van het eerste beeld nemen we het eerste beeldpunt van de eerste videolijn, het eerste beeldpunt van de tweede videolijn en zo verder tot we alle lijnen gehad hebben. Deze eerste beeldpunten vormen samen een verticale lijn aan de linkerrand van het



Close-up van het digitiserprintje

beeld. Van het volgende beeld voeren we het tweede beeldpunt van elke lijn in. We krijgen zo weer een verticale lijn van beeldpunten, direct naast de vorige. Na 256 beelden hebben we via 256 verticale lijnen alle beeldpunten ingevoerd. Op deze manier voeren we maar één beeldpunt per videolijn in, zodat we 64 microsec de tijd hebben voordat het volgende punt zich aandient. Dat is ruimschoots voldoende voor de analog-naar-digitaal conversie, voor het

opzoeken van een kleur die de juiste grijsstint geeft en voor het in het geheugen opslaan van het resultaat. Er is niet voldoende tijd om het beeldpunt meteen op zijn juiste plaats in het videogeheugen van de computer te zetten, daarom parkeren we het tijdelijk in het normale geheugen. Gedurende de AD-conversie mag het videosignaal niet veranderen, anders deugt het resultaat niet. Dat lossen we op door het videosignaal tijdelijk vast te houden, in te vriezen als het ware, vanaf het moment dat het gewenste beeldpunt bereikt is. De enige beperking van deze truc is natuurlijk dat het resultaat uit 256 beelden wordt bepaald. De beeldinhoud moet gedurende die tijd zo goed mogelijk gelijk blijven, er mag niet teveel beweging in zitten! Goed stilzitten voor de videocamera of, nog beter, opnemen op de videorecorder en digitaliseren van stilstaand beeld.

Interlacing

Bij standaard videosignalen hebben we nog te maken met een klein probleem. Twee opeenvolgende video-beelden worden namelijk niet precies over elkaar heen geschreven, maar een halve lijnafstand verschoven. Op die manier krijgt het menselijk oog de indruk dat er elke seconde 25 beelden van 625 lijnen geschreven worden. Dit wordt "interlacing" genoemd. Voor computerschermen wordt het niet toegepast, omdat het een trilling van het beeld veroorzaakt, die hinderlijk is als het beeldscherm van korte afstand wordt bekeken. (Op een MSX kan interlacing ingeschakeld worden met SCREEN,,,1. Dit wordt eigenlijk alleen



Pia digitaal over de koppeling

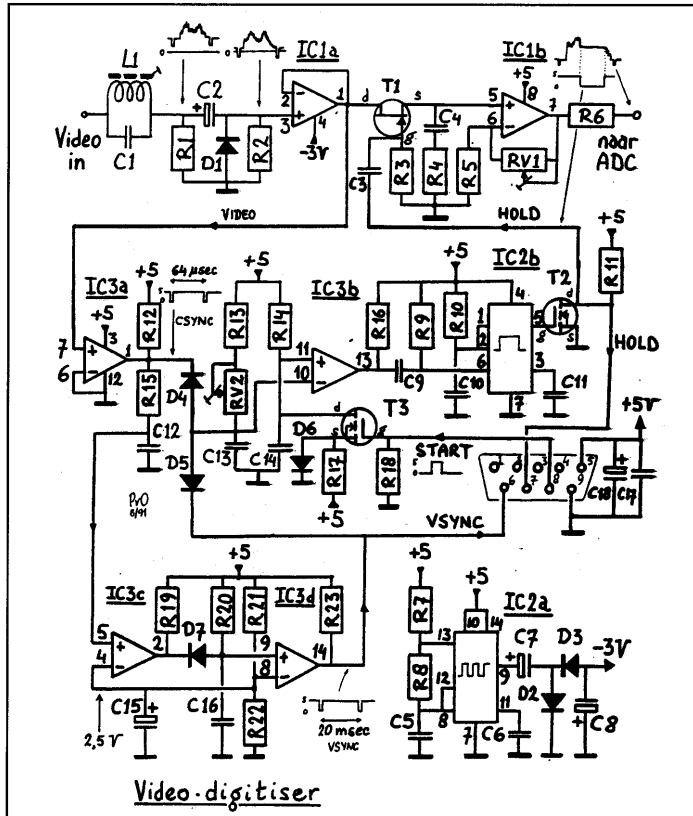
gebruikt om betere foto's van het beeldscherm te kunnen nemen.) De interlacing van een videobeeld kan tot gevolg hebben dat horizontale lijnen in het gedigitaliseerde beeld verspringen of onderbroken worden. De oplossing is eenvoudig: we digitaliseren alleen op de even of de oneven beelden, want die worden wel precies over elkaar heen geschreven. Na elk beeld wachten we gewoon één beeld af zonder te digitaliseren. Deze tijd kunnen we toch nog nuttig besteden, namelijk door de zojuist gedigitaliseerde verticale rij beeldpunten op de juiste plaats in het videogeheugen te zetten. Op deze manier hebben we voor het digitaliseren van 256 punten per videolijn in totaal 512 beelden nodig hebben. De tijd die nodig is voor het invoeren van een compleet videobeeld is nu eenvoudig te berekenen: 10,24 seconden.

Wat moet de digitiser doen?

Nu we zover zijn gekomen begint het te dagen wat de digitiser zoal moet doen. Ten eerste moet deze uit het videosignaal afleiden wanneer een nieuw beeld begint en dit aan de computer doorgeven. Dit signaal heet VSYNC, van verticale synchronisatie. Dat is al heel eenvoudig te doen. Vervolgens moet het juiste moment bepaald worden waarop een beeldpunt gedigitaliseerd moet worden: het eerste beeldpunt op elke videolijn uit het eerste beeld, het tweede beeldpunt op elke videolijn uit het derde beeld, en zo voort. (De even beelden sloegen we over, weet u nog wel?) Op dat juiste moment maken we een HOLD-sig-naal, dat het videosignaal lang genoeg vasthoudt om de AD-conversie ongestoord uit te kunnen voeren. Voor het vasthouden hebben we een zogenaamde "Sample and Hold"-schakeling nodig. Het HOLD-sig-naal sturen we ook naar de computer, want die moet nu de ADC-kaart opdracht geven om de conversie te starten en - na een korte wachttijd - het resultaat uitlezen ter verdere verwerking. Maar eerst moeten we dat HOLD-sig-naal nog zien te maken!

Het HOLD-sig-naal

We digitaliseren slechts één punt per videolijn, dus het HOLD-sig-naal moet voor elke videolijn éénmaal worden, zo ongeveer elke 64 microsec. In het begin van de digitalisatie moet HOLD vlak na het begin van elke videolijn komen. Dit begin wordt aangegeven met het signaal HSYNC, dat ook op een eenvoudige manier uit het videosignaal is af te leiden. Het HOLD-sig-naal moet zich nu langzaam van dat begin verwij-



Figuur 1

deren en wel met een snelheid van 0,1 microsec per beeld. Op die manier is het na 512 beelden aangekomen op 52 microsec van het begin van de videolijn, precies wat we nodig hebben. De nauwkeurigheid van het tijdstip waarop het HOLD-sig-naal komt bepaalt uiteindelijk de kwaliteit van het gedigitaliseerde plaatje. Een nauwkeurigheid van 100 nanosec of beter is gewenst. Hoe bepaal je dat tijdstip? Dat kun je op een ingewikkelde manier doen met snelle tellers, schuifregisters, 8-bits comparators en nog zo het een en het ander. Die treffen we dan ook aan in de andere video-digitisers. Maar als je eerst eens even goed nadenkt blijkt het ook te kunnen met maar één lineaire comparator. Die heb ik dus gebruikt. Want waarom zouden we iets op een ingewikkelde manier oplossen als het ook eenvoudig kan! Voor degenen die het naadje van de kous willen weten zal ik de werking nog wat uitgebreider uitleggen.

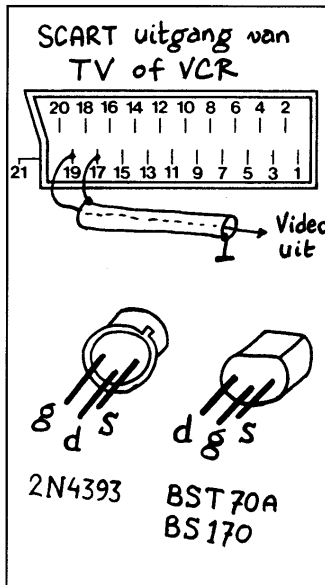
Schemabeschrijving

We bekijken de werking van de schakeling aan de hand van het schema (fig.1). Dit is opgebouwd rond drie goedkope IC's: een dubbele Opamp LF412, een dubbele Timer NE556 en een viervoudige Comparator LM339. Direct na binnenkomst van het videosignaal wordt de 4,4 MHz kleurdraaggolf daaruit verwijderd met het spierfilter L1 en C1. De combinatie C2 en D1 zorgt ervoor dat het nulniveau midden door de synchronisatiepulsen loopt. Deze zijn maar ca. 0,5 Volt hoog, daarom moet D1 een Schottky- of Germaniumtype zijn met een doorlaatspanning van 0,45 Volt of minder! Hetzelfde is overigens gewenst voor de dioden D2, D3, D4 en D5. De dubbele Opamp met JFET ingangen LF412 vormt samen met de FET schakelaar 2N4393 een Sample & Hold schakeling, die het videosignaal tijdens het digitaliseren even vasthoudt. Het uitgangssig-naal gaat direct naar de ADC-8 kaart. Er bestaan ook

kant-en klare Sample & Hold-IC's, maar a) die had ik niet in huis, b) die zijn nogal duur en c) nu kan de signaalsterkte met RV1 op de gevoeligheid van de ADC-kaart aangepast worden. De LF412 heeft een negatieve spanning van ongeveer -3 Volt nodig, die echter niet aanwezig is in UNIFACE. Daarom maken we die maar zelf met IC2a, een helft van de dubbele timer NE556. Het videosignaal gaat ook naar IC3, de eenvoudige comparator LM339. IC3a splitst hieruit het gecombineerde synchronisatiesignaal CSYNC af. IC3c haalt daaruit weer de beeldsynchronisatie VSYNC. IC3d verlengt VSYNC tot ongeveer 1,5 msec, lang genoeg om de egalisatiepuls te overbruggen. Dit signaal gaat naar de computer.

Het juiste moment

Nu moeten we nog op het juiste moment het HOLD-signaal maken. Dit doen we met de comparator IC3b. C14 wordt via T3 ontladen door het START-signaal uit de computer ontladen en vervolgens weer langzaam opgeladen via R14. Na 11 sec is de spanning tot zo'n 3 Volt gestegen. C13 wordt aan het begin van elke videolijn via D4 ontladen door CSYNC en vervolgens weer snel opgeladen via R13 en RV2. Het opladen tot 3 Volt duurt hier ongeveer 55 microsec. De comparator IC3b vergelijkt de spanning over C14 met die over C13 en geeft een triggersignaal aan IC2b wanneer de laatste groter wordt. Kort na de START is de spanning over C14 nog erg laag en zal dit dus aan het begin van de videolijn gebeuren. Maar naarmate C14 verder opgeladen is, duurt het steeds langer. Zo'n elf seconden na START gebeurt het pas op 55 microsec na het begin van de videolijn. IC2b, de tweede timer van NE556, is geschakeld als een One Shot van ca 40 microseconde. Het uitgangssignaal wordt geïnverteerd door T2 en vormt het HOLD-signaal voor de Sample & Hold schakeling. Dit HOLD-signaal gaat ook naar de computer. Het opladen van een condensator via een weerstand uit een constante spanning heeft een allesbehalve lineair verloop. Maar de beide takken R13/C13 en R14/C14 vertonen wel precies dezelfde afwijking. Bij het vergelijken van de spanningen heft de comparator deze afwijking exact op, zodat het resultaat toch weer zeer goed lineair is. Het is wel belangrijk dat R13 en R14 uit dezelfde spanning worden gevoed en dat de condensatoren C13 en C14 een verwaarloosbare lekstroom hebben. Geen elco gebruiken voor C14!



Programmabeschrijving

Het BASIC-programma installeert een stukje machinetaal op adres &HD000 en wacht op een toets om het digitaliseren te starten. Nadat het plaatje op het scherm staat wacht de computer weer op een toets. Het plaatje kan dan op disk worden bewaard door de toets "S" in te drukken. Het stukje machinetaal voert de volgende functies uit (de stappen 2 en 4 worden verderop nader toegelicht):

1. Zet de interrupt uit en geef een start-signaal.
2. Wacht 34 VSYNC-signalen.
3. Wacht op het eerstvolgende VSYNC-signaal.
4. Wacht 50 HOLD-signalen.
5. Wacht op het eerstvolgende HOLD-signaal.
6. Start de AD-conversie en wacht even tot die klaar.
7. Zet het resultaat in de Accu.
8. Zoek in de Look Up Table de bijbehorende kleur (grijstint) op.
9. Zet deze tijdelijk in RAM.
10. Zijn 212 beeldpunten ingevoerd? Zo nee, ga terug naar 5.
11. Verplaats de 212 punten van RAM naar het videogeheugen. Dit duurt zo lang dat het volgende beeld vanzelf overgeslagen wordt!
12. Zijn 256 verticale lijnen geplaatst? Zo nee, ga terug naar 3.
13. Zet de interrupt weer aan en keer terug naar BASIC.

Ingebruikname

Wanneer de schakeling gebouwd en goed nagekeken is, controleren we (met een videosignaal op de ingang), of de juiste signalen op joystickingang 2 binnenkomen. PRINT INP(&HA2) geeft normaal 191, maar dit wordt 175 voor een VSYNC en 159 voor een HOLD-sig-naal. Met het volgende testprogramma geven we de digitiser een START-sig-naal en bekijken wat er daarna gebeurt:

```
10 OUT &HA0, 15 :
OUT &HA1, &H6F
20 PRINT INP (&HA2) ;
30 GOTO 20
```



Uit de oude doos...

ar boven
r het ge-
te wijzi-
&HDOFF
et de ge-
van het
e op een
ste grijs-
lop mijn
elijk dat
ig betere
de wijzi-
, vergeet
IC in re-
zien met

De eerste 10 seconden moeten er flink wat HOLD-signalen komen met af en toe een VSYNC, daarna zien we alleen de VSYNC nog zo nu en dan. Als deze signalen er zijn, kunnen we gaan digitaliseren. Het afregelen gaat het beste met een testbeeld van de televisie. De regelbare weerstand RV1 stellen we zo in, dat de hele grijstrap onder in het testbeeld goed te zien is en het wit ook in het resultaat echt wit is. Met RV2, de andere regelbare weerstand, wordt de beeldbreedte ingesteld, zodat het testbeeld netjes symmetrisch op het scherm komt. Regel de kern van het



Postzegels, maar ook andere verzamelingen kunnen zo in de computer worden opgeslagen.

drukking van de kleurdraaggolf op 4,4 MHz. Dit signaal kan namelijk interferentiestrepen veroorzaken, zichtbaar als strepen in de kleurbalken van het testbeeld.

Positioneren van het beeld

Na het synchronisatie-signaal HSYNC aan het begin van elke nieuwe videolijn duurt het nog ongeveer 7 microsec vóórdat het eerste beeldpunt komt. Deze tijd moeten we overbruggen, omdat anders de linkerrand van het gedigitaliseerde beeld niet overeenkomt met die van het videobeeld. We moeten even wachten totdat C14 voldoende ver is opgeladen. Het programma doet dat door 34 (&H22) keer het signaal VSYNC

af te wachten. Het juiste aantal is wel een beetje afhankelijk van de gebruikte onderdelen rond C14. Experimenteer hier maar eens mee door een ander getal op adres &HD01F te POKEn. Van de 312,5 videolijnen per videobeeld zijn er op een televisie 285 zichtbaar en kan een MSX-computer er maar 212 tonen. We moeten dus 100 lijnen kwijttraken: 50 aan de bovenkant en 50 aan de onderkant van het beeld, zodat we het centrale deel van het videobeeld op ons scherm krijgen. Voor elke videolijn krijgt de computer een HOLD-signaal, dus na VSYNC wacht het programma eerst 50 HOLD-signalen en begint dan pas beeldpunten te digitaliseren. Wan-

neer u het beeld liever wat na of beneden plaats kan dat door tal 50 (&H32) op adres &HD02F gen. Van adres &HD0C0 tot staat de Look Up Table. Deze is digitaliseerde helderheid v beeldpunt om in een kleur, di monochroom monitor de jui tint geeft. De tabel is afgestemc NMS 8250 en het is best mog voor uw MSX-computer een nc tabel is te maken. Wanneer u gingen in de DATA-regels zet, dan niet ook het controlegeta gel 80 aan te passen. Veel ple

Peter van Overbeek

Foto's Henk Keijzer

MSX-Programma

```

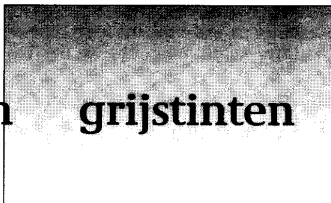
20 '(C) 1991, Peter van Overbeek
30 'ADC-kaart op adres &HDF (223)
40 'HOLD op pen 7 van joystick 2
50 'VSYNC op pen 6 van joystick 2
60 'START op pen 8 van joystick 2
70 IF PEEK (&HD0C0)-1 THEN GOTO 110
80 A-&HD000:C-32916!:FOR I=0 TO &HFF:READ D$
90 D=VAL("&H"+D$):POKE A+I,D:C=C-D:NEXT
100 IF C THEN PRINT "Fout in data!":END
110 PRINT"Druk een toets om te digitaliseren"
120 I$=INPUT$(1):SCREEN 8
130 DEFUSR=&HD000:X=USR(0):I$=INPUT$(1)
140 IF I$="g" OR I$="s" THEN BSAVE"digit.pic",0,&HD3FF,S
150 'data voor machinetaalprogramma
160 DATA 3E,FF,D3,30,3E,DF,D3,31,F3,3E,0F,D3,A0,3E,6F,D3
170 DATA A1,06,FF,10,FE,3E,4F,D3,A1,3E,0E,D3,A0,01,10,22
180 DATA CD,86,D0,C5,06,01,CD,86,D0,01,20,32,CD,86,D0,06
190 DATA D4,21,00,D1,DB,A2,A1,20,FB,3E,FC,D3,30,E3,E3,DB
200 DATA 30,E5,26,D0,0F,0F,F6,C0,6F,7E,E1,77,23,3E,FF,D3
210 DATA 30,10,E1,26,00,C1,C5,68,11,00,D1,06,D4,7C,E6,3F
220 DATA F6,40,F5,7C,E6,C0,F6,32,07,07,D3,99,3E,8E,D3,99
230 DATA 7D,D3,99,F1,D3,99,1A,E3,E3,D3,98,13,24,10,DE,C1
240 DATA 04,C2,23,D0,FB,C9,DB,A2,A1,20,FB,DB,A2,A1,28,FB
250 DATA 10,F4,C9,56,49,44,45,4F,44,49,47,49,54,49,53,45
260 DATA 52,20,28,43,29,20,31,39,39,31,20,62,79,20,50,65
270 DATA 74,65,72,20,76,61,6E,20,4F,76,65,72,62,65,65,6B
280 DATA 01,04,02,05,08,06,21,24,22,25,28,26,41,2C,42,45
290 DATA 30,46,61,64,62,65,68,66,81,84,82,85,A0,86,A1,A4
300 DATA A2,A5,A8,A6,A9,C2,C5,C8,C6,C9,E2,E5,E8,E6,E9,EC
310 DATA EA,ED,F0,EE,F1,F4,F2,F5,F8,F6,F9,FC,FA,FD,FE,FF

```

Onderdelenlijst

IC1	LF412
IC2	NE556
IC3	LM339
T1	2N4393
T2,T3	BST70A of BS170
D1,D2,D3	BYV10 of BAT 85
D4,D5	BAT 82 of BAT85
D6,D7	BAW62
L1	1µH
RV1,RV2	4k7
R1	82
R2,R18	100k
R3	1M
R4	510
R5,R9,R12	1k
R6	620
R7,R17	5k1
R8	30k
R10	15k
R11	10k
R13	5k6
R14	10M
R15	7k5
R16,R19,R23	1k
R20	4k7
R21,R22	2k2
C1	1n2
C2,C15	4,7µ
C3,C5,C10,C12	2n2
C4	1n5
C6,C11	10n
C7	10µ
C8	100µ
C9	1n0
C13	4n7
C14	1200n
C16,C17	100n
C18	220µ

Kleurtonen en grijstinten



Kunnen de fotoscanner en de videodigitiser ook in kleur werken?

De allereerste thuiscomputers kenden alleen maar zwart en wit. Al spoedig deden kleurenbeelden hun intrede: de P2000 liep ook hierin voorop. Dat kwam vooral omdat in die tijd meestal een (kleuren-)TV als beeldscherm werd gebruikt. Vorig jaar had meer dan de helft van alle PRINT-lezers een kleurenmonitor. Voor de MSX-ers was dat zelfs al tweederde, zo bleek uit het lezersonderzoek.

Naar aanleiding van de fotoscanner en nu weer voor de video-digitiser vraagt men mij vaak, of deze ook niet in kleur kunnen werken? Het antwoord is: ja en nee. Het is niet moeilijk zowel de fotoscanner als de video-digitiser zo uit te breiden, dat ook kleurenplaatjes ingevoerd kunnen worden. Het probleem ligt echter bij het tonen van de aldus verkregen plaatjes op het beeldscherm. Voor een natuurgetrouwe afbeelding gelden namelijk veel zwaardere eisen dan voor een tekening of een ander kunstmatig plaatje. Dit zal ik hier wat nader toelichten.

Beeldinvoer in kleur

Om de fotoscanner (zie PTC-Print nr. 46, april 1991) uit te breiden naar kleur hoeft er niet eens zoveel te gebeuren. In plaats van één opnemer hebben we er drie nodig. Deze moeten worden voorzien van kleurfilters, zodat ze elk voor één van de drie basiskleuren rood, groen en blauw gevoelig zijn. De opnemers moeten zo goed mogelijk op hetzelfde punt van de foto gericht worden en de drie signalen moeten tegelijk of in elk geval zeer snel na elkaar via de ADC-kaart uitgelezen worden. Ook de video-digitiser (zie elders in dit nummer) is betrekkelijk eenvoudig aan te passen. In plaats van het composite video-signaal (CVBS) moeten we de drie RGB-signalen gebruiken, die aanwezig zijn op de SCART-plug van de televisie of de videorecorder. Er is wat extra elektronica voor nodig, maar ook hier zijn er geen onoverkomelijke problemen. Die komen pas als we het resultaat willen tonen.

Wat het oog ziet

Een geoefende waarnemer kan, zittend in een verduisterde kamer, kijkend naar een goede kwaliteit monochroom monitor die perfect afgere-

geld is, ongeveer honderd verschillende grijstinten onderscheiden. Onder wat minder gunstige omstandigheden, zoals met omgevingslicht, zakt dit al snel tot zo'n 50 à 60 grijstinten. Voor toepassing thuis hebben we voldoende aan 64 grijstinten. Wat wil dat zeggen? Stel dat we de helderheid van het beeldscherm tussen diepzwart en helderwit opdelen in 64 even grote stapjes. Onder normale omstandigheden zal nu de overgang van de ene grijstint naar de volgende (of vorige) juist op de grens van zichtbaarheid liggen. Zouden we echter maar 32 of zelfs nog minder grijstinten gebruiken, dan worden deze overgangen wel duidelijk zichtbaar. Dat kan heel hinderlijk zijn in een plaatje met geleidelijk verlopende helderheid. Omdat de overgang van de ene naar de andere grijstint zo duidelijk zichtbaar wordt, lijkt die op een rand of contour. In dit geval is er geen sprake is van een echte rand of contour, daarom noemt men dit een valse contour. Bij het bekijken van zo'n plaatje hebben de menselijke hersenen moeite met het onderscheid tussen de echte en de valse contouren. Het geheel geeft daardoor een onnatuurlijke indruk. Maar met 64 goed verdeelde grijstinten (6 bits grijs) hebt u daar geen last van. Professionele systemen gebruiken zelfs 8 bits: 256 grijstinten dus, dan is er ook nog wat reserve om beeldbewerkingen uit te voeren.

Kleurenweergave

Dezelfde eisen gelden voor kleur, maar dan wel toegepast op elk van de drie basiskleuren rood, groen en blauw. Voor een natuurgetrouwe weergave van de kleuren moet elke basiskleur in minimaal 64 tinten gemaakt kunnen worden. In totaal zijn zo dus $64 \cdot 64 \cdot 64 = 262144$ verschillende kleuren nodig (18 bits kleur). Professionele grafische

computersystemen, werkstations genaamd, gebruiken ook hier weer 8 bits per basiskleur en komen zo aan meer dan 16 miljoen kleuren! Ook de foto-CD die Philips en Kodak binnenkort op de markt brengen, werkt met 24 bits kleur per beeldpunt. De bestaande kleurentelevisies en kleurenmonitoren met lineaire RGB-ingang hebben geen enkel probleem met het tonen van zoveel kleuren. Maar thuiscomputers die ze kunnen maken zijn er nog niet, al zal dat niet lang meer duren, verwacht ik. Ook kleurenprinters en kleurenkopieerautomaten zijn nog niet zo ver. Daar gebruikt men vaak trucjes om de aanwezigheid van valse contouren te verhullen. Zoals het gebruik van voorbeelden waarin kleurfouten niet zo snel opvallen en waarin zich vooral één geleidelijk verlopende kleurtinten bevinden. Nu begrijpt u ook waarom in brochures van deze apparaten de plaatjes van harige apen zo populair zijn!

Kleur in de thuiscomputer

De meeste thuiscomputers komen op dit moment niet verder dan 256 kleuren die tegelijk in beeld kunnen worden getoond. Daarmee zijn hele fraaie plaatjes in kermiskleuren te maken, maar een natuurgetrouwe weergave van de werkelijkheid is beslist niet mogelijk. Een PC heeft voor 256 kleuren een VGA-kaart nodig. Met een EGA-kaart komt men niet verder dan 64 kleuren. Een MSX-2 kan in SCREEN 8 256 kleuren tonen. De NMS 8280 computer heeft een ingebouwde digitiser met drie bits voor rood, drie bits voor groen en twee bits voor blauw. Er kunnen acht tinten rood, acht tinten groen en vier tinten blauw worden gemaakt: in totaal 256 kleuren. De kleurweergave lijkt helaas nergens op en valse contouren zijn uiterst storend zichtbaar. Met de ingebouwde videoprocessor is zwart-wit digitalisatie in zes bits of meer niet mogelijk. Jammer en voor mij de belangrijkste reden om niet in een NMS 8280 geïnteresseerd te zijn. Een MSX-2+ komt met 19268 kleuren al een stuk verder, echt tevreden over de kleurweergave kan ik nog lang niet zijn. Niet alleen is dit nog geen 8% van het minimaal aantal kleuren dat nodig is, ook zijn er nogal wat extra beperkingen op het kleurgebruik. Zo mag per rijtje van vier pixels maar één kleur uit 4096 gekozen worden, terwijl voor elk van de beeldpunten de kleurverzadiging in slechts 32 stapjes (5 bits) kan worden vastgelegd. De kleuren blijven onnatuurlijk en er zijn ook nog steeds valse contouren zichtbaar.

Toekomst

Is kleurweergave met 18 of liever nog met 24 bits per beeldpunt een overtrokken eis voor een thuiscomputer? Ik ben er zeker van dat het over enkele jaren heel normaal zal zijn. Technisch is er ook vandaag al geen probleem meer: voor professionele toepassingen zijn 24 bits videoprocessors allang gemeengoed. Wel zijn ze nu nog tamelijk duur. Maar de ontwikkelingen en prijsverlagingen gaan razendsnel. Kijk maar naar de geheugenchips. Nog maar weinig jaren geleden was geheugen duur en werd méér dan 64 kBytes volstrekt overbodig geacht voor thuiscomputers. Er waren trouwens ook geen programma's groter dan enkele tientallen kBytes. De lagere kosten voor geheugenchips hebben niet alleen geleid tot veel meer geheugen in thuiscomputers, het is ook van belang voor het toepassen van kleur. Immers, voor een plaatje met 24 bits per beeldpunt is drie maal zoveel geheugenruimte nodig als voor de huidige plaatjes met maximaal 8 bits per beeldpunt. Voor de Foto-CD zullen 24 bits videoprocessors in zeer grote aantallen gemaakt worden. Daardoor zullen de prijzen zakken en het ligt eigenlijk voor de hand dat deze videoprocessors ook in toekomstige thuiscomputers gebruikt zullen worden.

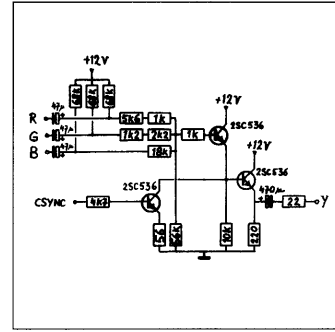
Plaatjes in grijs tinten

Liever een goed plaatje in grijs tinten dan een onnatuurlijk kleurenplaatje. Uiteraard kun je een plaatje zonder kleur ook niet echt natuurgetrouw noemen, maar het ontbreken van kleur blijkt veel minder storend te zijn dan de aanwezigheid van verkeerde kleuren, zoals bijvoorbeeld groengetinte gezichten. Ook heel belangrijk is het ontbreken van valse contouren in een plaatje. Met 64 grijs tinten zijn die niet meer te zien. In een kleurenplaatje zijn daarvoor al zo'n kwart miljoen kleuren nodig. Het mag dan mogelijk de fotoscanner en de video-digitizer geschikt te maken voor het invoeren van kleurenplaatjes, over de weergave op de huidige thuiscomputers zou u beslist ontvreden zijn. Voorlopig houden we het voor deze toepassingen daarom maar gewoon bij grijs tinten. Maar, zo hoor ik u al zeggen, volgens de gebruiksaanwijzing van mijn computer kan ik wel 256 kleuren maar slechts acht of 16 grijs tinten maken! In het artikel over de fotoscanner heb ik beschreven hoe dit is op te lossen. Met een monochroom monitor is het al heel eenvoudig. Die geeft immers de kleuren weer als verschillende grijs tinten.

Groen is wat lichter grijs dan rood en blauw komt vrij donker over. Bij een kleurenmonitor gebruiken we het helderheids-signaal Y (luminance) van de computer in plaats van het RGB-signaal. Op veel kleurenmonitoren zit een knop om de kleurverzadiging te regelen. Hiermee kan ook alle kleur wegge-regeld worden. Voor het plaatje komt het er nu op neer om een zodanige volgorde van kleuren te kiezen, dat de daarbij horende grijs tinten een geleidelijk oplopende helderheid te zien geven. Hoewel sommige heel verschillende kleuren vrijwel dezelfde helderheid zullen opleveren, zijn er uit 256 kleuren zeker 64 verschillende helderheden te definiëren. Hoe vinden we die? Dat kan door uitproberen, maar het kan ook nuttig zijn om er een beetje theorie bij te halen.

Een beetje theorie

Om uit de drie basiskleuren rood, groen en blauw zuiver wit te maken, moeten deze in bepaalde verhoudingen gemengd worden: 30% rood, 59% groen en 11% blauw. Het helderheids-signaal Y wordt nu als volgt gedefinieerd: $Y = 0,30 \cdot R + 0,59 \cdot G + 0,11 \cdot B$. Waarom zo? Omdat we op die manier op een kleurenmonitor wit of grijs krijgen als we de drie basiskleuren in gelijke mate mengen, dus als $R=G=B$. Op een monochroom scherm zien we alleen maar grijs tinten. Uit de formule volgt wel dat een rood scherm ($G=0$ en $B=0$) 30% van de helderheid van een wit scherm heeft, een groen scherm 59% en een blauw scherm 11%, dit alles bij maximale kleurintensiteit. Geel ontstaat uit rood plus groen en geeft 89%. Maar de kleurtinten zijn zelf ook nog in te stellen. Meestal zijn drie bits per basiskleur beschikbaar, zodat de waarden voor R, G en B van 0 tot 7 kunnen lopen. $R=G=B=0$ levert $Y=0$ en dat is diepzwart. $R=G=B=7$ levert $Y=7$ en dat is helderwit. $R=B=G=2$ levert een grijs tint op met $Y=2$. $R=7$ en $G=B=0$ levert op een kleurenmonitor een helderrood scherm op. Op een monochroom scherm zien we een grijs tint met $Y = 7 \cdot 0,30 = 2,1$, dat is een heel klein beetje lichter dan de grijs tint $Y=2$. De weer iets lichtere grijs tint $Y=2,2$ krijgen we met $R=5$, $G=1$ en $B=1$. Maar ook $R=7$, $G=0$ en $B=1$ levert vrijwel dezelfde grijs tint op, reken maar na. Het lukt echter bijna altijd de helderheid Y in nog net zichtbare stapjes van 0,1 te verhogen door geschikte keuze te maken voor R, G en B. Daarmee kunnen in het helderheidsbereik van $Y=0$ tot $Y=7$ zo'n 70 grijs tinten worden gekozen: precies



Zó maakt en NMS 8250 het helderheids-signaal Y uit de basiskleuren R, G en B plus het synchronisatiesignaal CSYNC. (Sterk vergroot afgebeeld).

genoeg om een goed plaatje te kunnen maken!

En nu de praktijk

Alle MSX-computers hebben een Luminance-uitgang, die zó op de CVBS-ingang van de kleurenmonitor aangesloten kan worden om fraaie plaatjes in grijs tinten te vertonen. Wegens de tolerantie op de gebruikte weerstanden kan het zijn dat het helderheids-signaal iets afwijkt van de hierboven gegeven formule. In SCREEN 8 zijn maar twee bits beschikbaar voor blauw, waarmee uit vier waarden voor B kan worden gekozen. Deze zijn echter niet 0, 1, 2 en 3 maar 0, 2, 4 en 7! Het is echter goed mogelijk hiermee een grijs tintentabel te maken, kijk maar in het verhaal over de fotoscanner in PRINT nr.46. Voor een VG 8230/8235 zullen sommige waarden iets aangepast moeten worden, het Y-signaal wijkt hier weer iets af van dat van de NMS 8250. Voor de PC's ligt de zaak wat minder eenvoudig. Wat grafische toepassingen betreft lopen deze duidelijk achter op de mogelijkheden van een MSX. Men moet al minstens een VGA-kaart hebben om te kunnen beschikken over lineair RGB, zodat een voldoende groot aantal kleuren of grijs tinten gedefinieerd kan worden.

Peter van Overbeek