Digiscope
voor 6502-systemen

M. Dohmen

Degenen die de beschikking hebben over een Grafisch Display en een computersysteem, gebaseerd op de 6502, zullen in dit artikel ontdekken dat de mogelijkheden van deze combinatie haast onbeperkt zijn. In het januarinummer van 1980 werd een frequentiemeter beschreven, die eens te meer aantoonde dat een computer zijn diensten kan bewijzen in het elektronicalab van de hobbyist. Het hier besproken programma zal, als toevoeging aan de frequentiemeter, voor velen een antwoord zijn op de vraag: „Schaf ik mij een computer aan?"

Het betreft hier een zogenoemde „digiscope". Dit is een soort oscilloscoop, die digitale signalen zichtbaar maakt op het grafisch display.

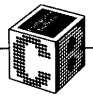
Deze digiscope is vrij snel. Zonder hardware-aanpassingen ligt de maximaal meetbare frequentie op 71,5 kHz. Hij kent acht kanalen en er kunnen dus acht frequenties tegelijkertijd worden gemeten. Met enige hardware-uitbreidingen kan een mini-logic-analyzer worden gerealiseerd. Wilt u alleen frequenties meten en bijvoorbeeld geen puls-pauzeverhoudingen, dan raad ik u aan gewone 10-delers te gebruiken om het frequentiegebied te vergroten. Een nadeel van de digiscope is dat hij slechts twee toestanden aan de ingangen kent, namelijk „hoog" en „laag", zodat deze alleen bruikbaar is bij experimenten met digitale bouwstenen. Voordeel is dat, dankzij de acht kanalen, de ligging van de signalen ten opzichte van andere zeer goed is te zien. Ook de vorm van het signaal is goed herkenbaar bij dit programma (mits de frequentie niet te hoog wordt). Het programma moest aan de volgende eisen voldoen:

1. Hoge snelheid en hoge resolutie.
2. Acht kanalen.
3. Periodetijd moet zichtbaar kunnen worden gemaakt.

Lijst 1 ▶

```
                0320 ;   INGANGSROUTINE VOOR EEN MAXIMALE
                0330 ;   FREQUENTIE VAN 38,5 KHZ
                0340 ;
                0350 ;
0200- A2 00     0360 INPUT      LDX #$00      ;HAAL 256 MAAL DE
0202- AD 40 17  0370 LUS        LDA PAD2      ;WAARDE BINNEN
0205- 95 00     0380            STA #BUFFER,X ;VAN DE PIA-POORT
0207- E8        0390            INX  ;EN ZET DEZE IN DE
0208- D0 F8     0400            BNE LUS       ;BUFFER
020A- 4C 00 07  0410            JMP DSPL      ;SPRING NAAR GD-ROUTINE
```

***Lijst 1** Korte invoerroutine.*



Lijst 2 Lange, doch snelle invoerroutine.
Lijst 3 Programma dat de snelle invoerroutine in het geheugen schrijft.

Een dergelijk programma kan op drie manieren worden verwezenlijkt. Allereerst kan gebruik worden gemaakt van het pennetje SO (soms ook RO) van de processor. Dit is de snelste methode. Pen SO staat rechtstreeks in verbinding met het Overflow-bit in het statusregister. Door het niveau op SO van hoog naar laag te laten gaan wordt het Overflow-bit gereset. Door dit pennetje als ingang te gebruiken is een zeer snelle invoer mogelijk. Voor ons doel is dit echter niet geschikt, omdat het om één ingang gaat in plaats van acht. We zullen hier gebruik moeten maken van een in/uit-poort. Een tweede mogelijkheid is het met vaste intervaltijden aftasten van de ingangspoort en vervolgens de desbetreffende bitjes omzetten naar acht puntjes voor het grafisch display. Een bezwarende factor is echter de beperkte snelheid. Het zetten van een punt op het grafisch display kost veel tijd (64 µs), zodat het frequentiegebied veel te klein wordt. Verreweg de effectiefste methode is wel achter elkaar 256 bytes in te lezen, deze op te slaan en vervolgens om te zetten naar acht grafieken voor het grafisch display. De uitvoering kan op twee manieren geschieden. Ze zullen beide worden besproken.

Uitvoering

Het binnenhalen en opslaan van de 256 bytes kan in een programmalus gebeuren, zie hiervoor lijst 1. Voordeel van dit programma is dat het heel kort is. Een nadeel, dat zwaar weegt, is echter de snelheid. Het binnenhalen van een byte duurt maar liefst 13 µs. Het frequentiegebied is dan slechts 38,5 kHz. De mensen, die hiermee tevreden zijn, kunnen het programma uit lijst 1 intypen. Dit gebruikt alle locaties op pagina nul, maar aangezien er geen gebruik wordt gemaakt van de monitor, is dit niet zo bezwaarlijk.
X is de teller die aanwijst, waar het opgehaalde byte in pagina nul moet komen te staan. De werking is nu vrij simpel te doorzien. Er wordt 256 maal naar de ingangspoort gekeken en de waarden worden opgeslagen. Wanneer dit is gebeurd, wordt naar de displayroutine gesprongen.
De nu besproken mogelijkheid is weliswaar veel sneller, maar gebruikt maar liefst 1¼K geheugenruimte. Dit is voor mensen met een klein geheugen dus niet zo aantrekkelijk. Wel wordt nu echter het beloofde frequentiegebied van 71,5 kHz gehaald. Bij de eerste methode werd veel tijd verspild met het ophogen van X, het vergelijken met nul en de relatieve sprong (alles bij elkaar is ca. 5 µs). Om deze tijd te winnen is het nodig de lus volledig uit te schrijven en geen geïndiceerde adressering toe te passen. Er wordt aldus 6 µs gewonnen, zodat de tijd om een byte in te lezen en weg te schrijven wordt gereduceerd tot 7 µs, wat neerkomt op een frequentiegebied van 71,5 kHz. Het programma ziet er dan uit, zoals is weergegeven in lijst 2. Het zou te ver voeren dit allemaal in te typen. Daarvoor hebben wij een computer, die dat sneller en nauwkeuriger kan. Een programma dat het invoerprogramma van 1¼K voor u intypt, is te vinden in lijst 3.

Werking

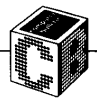
Het programma voor de snelle invoer, dat achter elkaar in het geheugen moet komen, staat in tabel 3 in lijst 3. Register X telt van 0 tot 255. Twee locaties op pagina 0 houden bij waar de tabel moet worden „afgedrukt". In dit geval van $0200 tot $0700. Op $0700 begint de displayroutine die de opgehaalde waarden omzet in acht grafieken. De tabel is vier bytes lang en bevat: AD 40 17 85.
Het vijfde byte is de locatie op pagina 0 waar het van poort A gehaalde byte moet worden neergezet. Dit vijfde byte moet echter oplopen van $00 tot $FF. Hiervoor zorgt nu register X.
Y wordt geladen met $04. X wordt in de accu gezet en het vijfde byte wordt in de desbetreffende geheugenlocatie geplaatst. Y wordt met

Lijst 2

```
                0450 ;   INGANGSROUTINE VOOR EEN MAXIMALE
                0460 ;   FREQUENTIE VAN 71.5 KHZ
                0470 ;
                0480 ;
                0490            .BA $0200
                0500 ;
                0510 ;
0200- AD 40 17  0520 INPT       LDA PAD2     ;HAAL WAARDE
0203- 85 00     0530            STA *BUFFER  ;EN ZET IN BUFFER
0205- AD 40 17  0540            LDA PAD2 ;
0208- 85 01     0550            STA *BUFFER+$01 ;
020A- AD 40 17  0560            LDA PAD2 ;
020D- 85 02     0570            STA *BUFFER+$02 ;
                0580 ; ENZ
                0590 ;
                0600 ;
                0610            .BA $6FB
                0620 ;
                0630 ;
06FB- AD 40 17  0640            LDA PAD2 ;
06FE- 85 FF     0650            STA *BUFFER+$FF ;
                0660 ;DSPL      LDA #$1F
                0670 ; ENZ
```

Lijst 3

```
                1270 ;       PROGRAMMA DAT DE SNELLE INVOERROUTINE
                1280 ;       IN HET GEHEUGEN PLAATST
                1290 ;
                1300 ;
076A- A2 00     1310 INIT       LDX #$00
076C- A9 00     1320            LDA #$00     ;INITIALISEER
076E- 85 00     1330            STA *POINTL  ;POINTERS
0770- A9 02     1340            LDA #$02 ;
0772- 85 01     1350            STA *POINTH  ; STA POINTH ;
0774- A0 04     1360 BEGIN      LDY #$04     ;ZET WAARDE
0776- 8A        1370            TXA  ;VAN X IN ELK
0777- 91 00     1380            STA (POINTL),Y       ;UITGDE BYTE
0779- 88        1390            DEY  ;ZET DE REST
077A- B9 9F 07  1400 LUS1       LDA TABEL3,Y         ;VAN DE BYTES
077D- 91 00     1410            STA (POINTL),Y       ;OP ZIJN PLAATS
077F- 88        1420            DEY ;
0780- 10 F8     1430            BPL LUS1 ;
0782- 18        1440            CLC  ;TEL BIJ DE
0783- A9 05     1450            LDA #$05     ;POINTER
0785- 65 00     1460            ADC *POINTL  ;5 OP
0787- 85 00     1470            STA *POINTL ;
0789- A9 00     1480            LDA #$00 ;
078B- 65 01     1490            ADC *POINTH ;
078D- 85 01     1500            STA *POINTH ;
078F- E8        1510            INX  ;ALLES GEHAD?
0790- D0 E2     1520            BNE BEGIN    ;NEE, TERUG
0792- A0 02     1530            LDY #$02     ;PLAATS DE
0794- B9 A3 07  1540 LUS2       LDA TABEL4,Y         ;JMP-INSTRUCTIES
0797- 91 00     1550            STA (POINTL),Y ;
0799- 88        1560            DEY ;
079A- 10 F8     1570            BPL LUS2 ;
079C- 4C 00 02  1580            JMP INPUT    ;SPRING NAAR DE INVOERROUTINE
079F- AD 40 17  1590 TABEL3     .BY $AD $40 $17 $85
07A2- 85
07A3- 4C 00 07  1600 TABEL4     .BY $4C $00 $07
```



één verminderd en vervolgens worden de vier waarden uit de tabel achter elkaar in het geheugengebied geplaatst. Nu wordt X met één en de pointer met vijf opgehoogd ($0783). Is dit 256 maal gebeurd dan is het geheugengebied gevuld en wordt naar $0200 gesprongen, waar het zojuist geschreven programma start. Wanneer de displayroutine bij u niet direct achter de invoerroutine kan komen, moet het omlijnde gedeelte van lijst 3 worden tussengevoegd. Dit zorgt ervoor dat na de invoerroutine nog een sprong naar de displayroutine wordt geplaatst. Het haalt drie bytes uit tabel 4 van deze lijst en zet ze in het geheugen vanaf $0700. Uiteraard moet dan in tabel 4 het juiste startadres worden neergezet. Begint uw displayroutine bijvoorbeeld op $1000 dan moet tabel 4 worden: 4C 00 10. Let op de volgorde, lage byte voorop.

Displayroutine

Voor de displayroutine kunnen geen locaties op pagina 0 worden gebruikt, waardoor deze iets langer wordt. Het programma zet per X-coördinaat acht puntjes aan. De Y-coördinaat van die puntjes wordt enerzijds bepaald door het kanaal, anderzijds door het logische niveau van dat kanaal. De Y-coördinaten voor de desbetreffende kanalen zijn daartoe in een tabel gezet en wel een aparte tabel voor de „1"-niveaus van de acht kanalen (tabel 1 in lijst 4) en een voor de „0"-niveaus (tabel 2). Op deze wijze kan elk kanaal op een willekeurige plaats op het scherm worden gezet. Dit is handig wanneer er ook nog tekst op het scherm moet komen.

De invoerroutine heeft er voor gezorgd dat de binnengehaalde waarden op opeenvolgende plaatsen op pagina 0 zijn gezet. Register X wordt nu zowel voor de pointer die naar de adressen op pagina 0 verwijst, als voor de X-coördinaat van de punt op het grafisch display, gebruikt. Met deze kennis wordt het programma in lijst 4 wat eenvoudiger te begrijpen. Eerst worden de in/uit-registers goed gezet en wordt het scherm gewist. Vervolgens wordt X geladen met $00 (X-coördinaat is 0, pointer wijst naar locatie $00) en Y met $07, immers per byte moeten acht puntjes worden gezet. Door nu telkens een bitje van het byte in de carry te schuiven wordt elk bitje omgezet in een punt voor het grafisch display. Daartoe wordt Y bewaard op locatie $0769. Het eerste byte wordt opgehaald, een plaats naar rechts geschoven en weer teruggezet. Is de carry „1" dan moet een hoog niveau worden geschreven, is deze „0" dan moet een laag niveau worden geschreven. Daartoe dient plaats Y uit tabel 1 of plaats Y uit tabel 2 in Y zelf te worden gezet. Dit is als volgt gerealiseerd. Het programma gaat

Lijst 4 Routine voor het weergeven van logische signalen op het grafisch display.
Lijst 5 Testprogramma.
Lijst 6 Programma om periodetijden te meten.

Lijst 4

```
                0710 ;      ROUTINE VOOR HET WEERGEVEN VAN LOGISCHE
                0720 ;      SIGNALEN OP HET GRAFISCH DISPLAY
                0730 ;
                0740 ;
                0750            .BA $0700
                0760 ;
                0770 ;
0700- A9 1F     0780 DSPL       LDA #$1F        ;INITIALISEER
0702- 8D 03 17  0790            STA PBDD1       ;DE IN- EN
0705- A9 FF     0800            LDA #$FF        ;UITGANGEN
0707- 8D 01 17  0810            STA PADD1 ;
070A- A9 1C     0820 CLSC       LDA #$1C        ;WIS-CODE
070C- 8D 02 17  0830            STA PBD1        ;WIS SCHERM
070F- A9 20     0840            LDA #$20        ;WACHT
0711- 8D F7 17  0850            STA TIMER       ;32 MS
0714- 2C F7 17  0860 TT         BIT TIMER ;
0717- 10 FB     0870            BPL TT ;
0719- A2 00     0880 START      LDX #$00        ;BITTELLER
071B- A0 07     0890 NEXTBY     LDY #$07        ;KANAALENTELLER
071D- BC 69 07  0900            STY VSAV        ;ACHT KANALEN
0720- B5 00     0910 NEXBI      LDA *BUFFER,X           ;SELECTEER BIT
0722- 4A        0920            LSR A ;
0723- 95 00     0930            STA *BUFFER,X           ;BEWAAR EVEN
0725- AC 69 07  0940            LDY VSAV        ;HAAL '1'-NIVEAU
0728- B9 59 07  0950            LDA TABEL1,Y ;
072B- B0 0C     0960            BCS SKIP        ;SPRING ALS BIT = 1
072D- B9 61 07  0970            LDA TABEL2,Y            ;HAAL '0'-NIVEAU
0730- A8        0980 SKIP       TAY ;
0731- 20 3F 07  0990            JSR PUNT        ;ZET PUNT AAN
0734- CE 69 07  1000            DEC VSAV        ;VOLGENDE BIT
0737- 10 E7     1010            BPL NEXBI ;
0739- E8        1020            INX  ;VOLGENDE BYTE
073A- D0 DF     1030            BNE NEXTBY ;
073C- 4C 1D 1C  1040            JMP MONIT       ;TERUG NAAR MONITOR
                1050 ;   ROUTINE VOOR HET WEERGEVEN VAN PUNTEN
                1060 ;   OP HET GRAFISCH DISPLAY
                1100 ;
                1110 ;
073F- AD 02 17  1120 PUNT       LDA PBD1        ;SYNCHRONISEER
0742- 10 FB     1130            BPL PUNT        ;MET DISPLAY
0744- 8E 00 17  1140            STX PAD1        ;ZET X-COORDINAAT
0747- EE 02 17  1150            INC PBD1 ;
074A- EE 02 17  1160            INC PBD1 ;
074D- 8C 00 17  1170            STY PAD1        ;ZET Y-COORDINAAT
0750- EE 02 17  1180            INC PBD1 ;
0753- A9 10     1190            LDA #$10        ;ZET KLEUR
0755- 8D 02 17  1200            STA PBD1 ;
0758- 60        1210            RTS ;
0759- E8 C8 A8  1220 TABEL1     .BY $E8 $C8 $A8 $88 $68 $48 $28 $08 ; 1'-NIVEAUS

075C- 88 68 48
075F- 28 08
0761- F8 D8 B8  1230 TABEL2     .BY $F8 $D8 $B8 $98 $78 $58 $38 $18 ; 0'-NIVEAUS

0764- 98 78 58
0767- 38 18
0769- 00        1240 VSAV       .BY $00
```

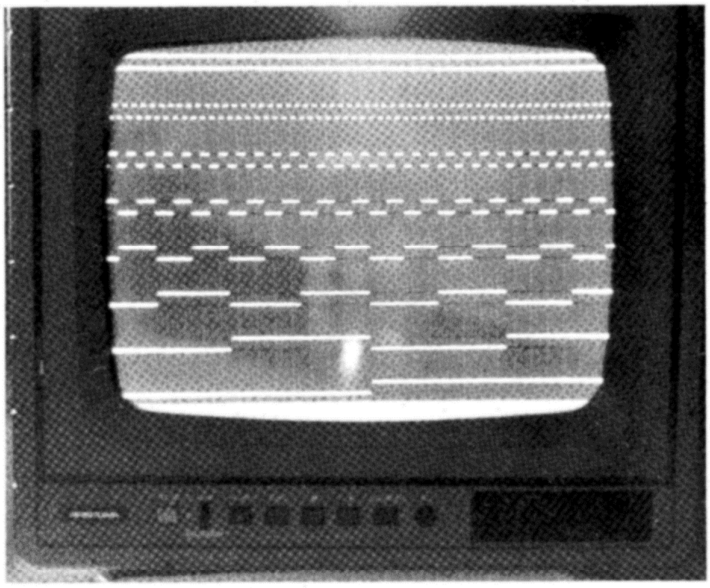
Lijst 5

```
                1640 ;      TESTROUTINE
                1650 ;
                1660 ;
                1670            .BA $0800
                1680 ;
                1690 ;
0800- A2 00     1700 TEST       LDX #$00        ;SCHRIJF BUFFER
0802- 8A        1710 LUS3       TXA  ;VOL MET
0803- 95 00     1720            STA *BUFFER,X           ;NULLEN
0805- E8        1730            INX ;
0806- D0 FA     1740            BNE LUS3 ;
0808- 4C 00 07  1750            JMP DSPL        ;SPRING NAAR DISPLAYROUTINE
```

Lijst 6

```
                1790            .BA $0800
                1800 ;
                1810 ;
                1820 ;      ROUTINE VOOR HET METEN
                1830 ;      VAN PERIODETIJDEN
                1840 ;
                1850 ;
0800- A2 00     1860 MEET       LDX #$00        ;WACHT OP
0802- B5 00     1870 LUSA       LDA *BUFFER,X           ;EEN
0804- E8        1880            INX  ;NEERGAANDE
0805- 2D FF 07  1890            AND MEM         ;FLANK
0808- D0 F8     1900            BNE LUSA ;
080A- B5 00     1910 LUSB       LDA *BUFFER,X           ;WACHT OP
080C- E8        1920            INX  ;DE EERSTE
080D- 2D FF 07  1930            AND MEM         ;OPGAANDE
0810- F0 F8     1940            BEQ LUSB        ;FLANK
0812- A0 01     1950            LDY #$01        ;TELLER = 1
0814- B5 00     1960 LUSC       LDA *BUFFER,X           ;WACHT OP
0816- E8        1970            INX  ;VOLGENDE
0817- C8        1980            INY  ;NEERGAANDE
0818- 2D FF 07  1990            AND MEM         ;FLANK
081B- D0 F7     2000            BNE LUSC ;
081D- B5 00     2010 LUSD       LDA *BUFFER,X           ;WACHT OP
081F- E8        2020            INX  ;OPGAANDE
0820- C8        2030            INY  ;FLANK
0821- 2D FF 07  2040            AND MEM         ;EN TEL
0824- F0 F7     2050            BEQ LUSD        ;HET AANTAL
0826- 98        2060            TYA  ;PERIODEN
0827- 8D 69 07  2070            STA VSAV        ;BEWAAR DIT
082A- 0A        2080            ASL A           ;VERMENIGVULDIG
082B- 0A        2090            ASL A           ;MET 7
082C- 0A        2100            ASL A           ;EN TREK ER
082D- 38        2110            SEC  ;AF VANAF
082E- ED 69 07  2120            SBC VSAV ;
0831- 4C 1D 1C  2130            JMP MONIT
```

Afb. 1 Resultaat van het testprogramma op het scherm.



ervan uit dat de carry gelijk is aan „0" en er wordt dus een waarde uit tabel 2 opgehaald ($0728), afhankelijk van de waarde van Y. De instructies STA, LDA respectievelijk LDY tasten het carry-bit niet aan, zodat de carry nog steeds bepaalt of er een hoog dan wel een laag niveau moet worden getekend. Op adres $072B staat nu dat de volgende instructie (haal de waarde uit tabel 2) moet worden overgeslagen, als het carry-bit gelijk is aan „1". Aangekomen op adres $0731 staat dus de X-coördinaat van het puntje in X en de Y-coördinaat in Y. Vervolgens wordt naar de routine gesprongen, die op deze coördinaten een wit puntje zet. Dit zelfde geschiedt voor de zeven andere kanalen, waarna X wordt opgehoogd en een nieuwe waarde van pagina 0 wordt gehaald. Na 2048 puntjes te hebben gezet wordt teruggesprongen naar de monitor.

Laatste loodjes

Wanneer alles is ingetypt, kan het programma worden getest. Controleer vooral het „invoerroutine-schrijvende" gedeelte ($076A en verder), omdat deze routine in RAM schrijft en het risico groot is dat u al uw programma's kwijt raakt, als hierin bij het intypen een foutje is geslopen. In lijst 5 ziet u een testprogramma dat het in afb. 1 getoonde resultaat geeft, als alles werkt. Is dat niet het geval dan is er iets mis.

Uitbreidingen

Door de opzet van het programma is het vrij eenvoudig om bijvoorbeeld periodetijden weer te geven op het grafisch display. Met behulp van het programma „Letters op het grafisch display" van D. M. de Boer (RB, mei en juli 1979) kunt u tekst en cijfers toevoegen. In lijst 6 ziet u het programma om de periodetijd van een signaal te meten. Het principe is als volgt. Haal een byte, selecteer een bit en tel af, hoelang het duurt, voordat het bit van „0" naar „1" en van „1" weer naar „0" is gegaan. Eerst moet worden opgezocht, wanneer het bitje „0" wordt. Van daaraf moet worden geteld. Register X is de pointer die het desbetreffende byte selecteert. Op adres 07FF moet komen te staan, welk kanaal wordt gekozen in de vorm 2^{n-1}, waarbij n het desbetreffende kanaal is. Als kanaal 3 moet worden geselecteerd moet dus $04 in $07FF worden gezet. Als het in lijst 6 afgebeelde programma wordt uitgevoerd moet er op adres 0700 komen te staan: 4C0008. Dit moet omdat de displayroutine de meetwaarden wist en er geen periodetijd meer kan worden gemeten.

Meting van de periodetijd

Allereerst wordt X gelijk gemaakt aan nul. Er wordt vervolgens een byte van pagina 0 gehaald en het desbetreffende bit wordt gemaskeerd. Is dit bit „1" dan moet het volgende bit worden gehaald. Immers, er wordt pas begonnen met tellen bij een overgang van „0" naar „1". X wordt telkens opgehoogd, totdat het bit „0" is geworden. Dan wordt X opgehoogd, tot dat het bit weer „1" wordt. We weten nu zeker dat we aan het begin van een nieuwe periode zitten. Y wordt geladen met „1" (we hebben al een cyclus gehad) en X en Y worden telkens opgehoogd, totdat het desbetreffende bit van „0" naar „1" gaat. In Y staat nu de periodetijd gedeeld door zeven. Elk puntje stelt immers 7 µs voor. De waarde in Y moet nog met zeven worden vermenigvuldigd. Dat gaat als volgt. Y wordt bewaard in $0801 en tevens in de accu gezet. De accu wordt driemaal naar links geschoven (dus met acht vermenigvuldigd), waarna er éénmaal Y vanaf wordt getrokken. Resultaat is:
Y * (8 – 1) in de accu. De waarde in de accu kan nu met het programma „Letters op het grafisch display" op het scherm worden gezet.