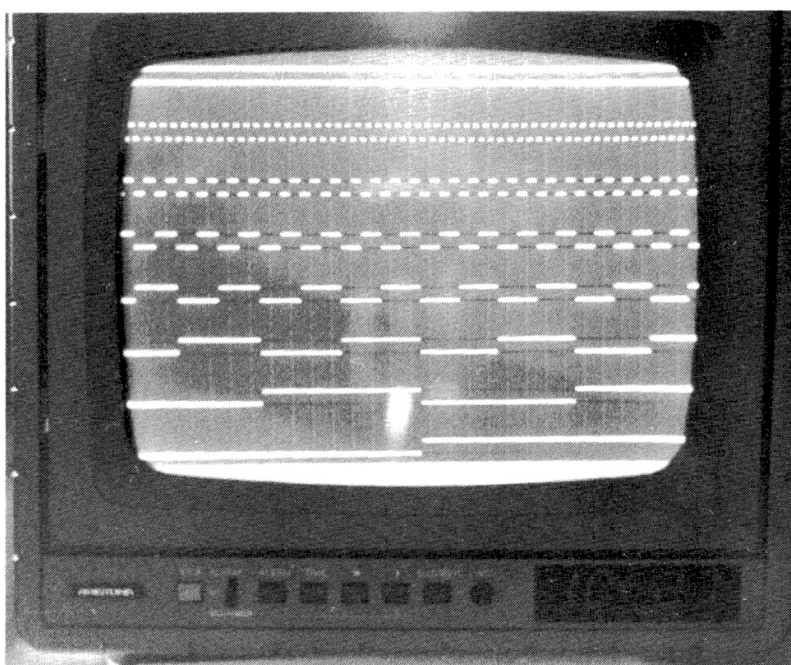


Digiscope

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 januarinumnummer 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.

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 μ s), 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.

Lijst 1

	0320 ;	INGANGSRoutine voor een maximale	
	0330 ;	Frequentie van 38,5 kHz	
	0340 ;		
	0350 ;		
0200- A2 00	0360 INPUT	LDX #256	#HAAL 256 MAAL DE
0202- AD 40 17	0370 LUS	LDA P002	#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	RNE LUS	#BUFFER
020A- 4C 00 07	0410	JMP DSPL	#SPRING NAAR GO-ROUTINE

Lijst 1 Korte invoerroutine.

Uitvoering

Het binnenhalen en opslaan van de 256 bytes kan in een programma dus 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 μ s. 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 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.

Y wordt geladen met \$04. X wordt in de accu gezet en het vijfde byte wordt in de desbetreffende geheugenlocatie geplaatst. Y wordt met 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 omliggende 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.

Lijst 4

0710 ;	ROUTINE VOOR HET VERPOESEN VAN LOGISCHE
0720 ;	SIGNALEN OP HET GRAFISCH DISPLAY
0730 ;	
0740 ;	
0750 ;	.BR \$0700
0760 ;	
0770 ;	
0780- A9 1F	0780 DSPL LDA #1F ;INITIALISEER
0782- 80 03 17	0782 STA PRCOJ ;ADE IN- EN
0785- A9 FF	0785 LDA #FF ;AUTORAGEN
0787- 80 01 17	0787 STA PRCOJ ;
0789- A9 1C	0789 LDA #1C ;AUS-CODE
078C- 80 02 17	078C STA PRCOJ ;AUS-SCHERM
078F- A9 20	078F LDA #20 ;AUSCH
0711- 80 F7 17	0711 STA THER ;32 MS
0714- 20 F7 17	0714 BIT THER ;
0717- 10 F8	0717 BPL TT ;
0719- A2 00	0719 LDW #000 ;BITTELLER
071B- 80 07	071B LDW #007 ;AUSCHTELLER
071D- 8C 69 07	071D STV VSRU ;RECHT KWAKLEN
0720- 85 00	0720 LDA #BUFFER,X ;SELECTEER BIT
0722- 4A	0722 LSR A ;
0723- 95 00	0723 STA #BUFFER,X ;BEWAAR EVEN
0725- 8C 69 07	0725 LDW VSRU ;AUSCH 1"-NIVEAU
0728- 89 59 07	0728 LDA TABEL1,V ;
072B- 80 05	072B BCS SKIP ;SPRING ALS BIT = 1
072D- 89 61 07	072D LDA TABEL2,V ;AUSCH 0"-NIVEAU
0730- 88	0730 TRV ;
0731- 28 3F 07	0731 VCR PUNT ;ZET PUNT AAN
0734- CE 69 07	0734 DEC VSRU ;VOLGENDE BIT
0737- 10 E7	0737 BPL NEXT1 ;
0739- E8	0739 INC ;VOLGENDE BYTE
073A- E4 0F	073A BNE NEXT1 ;
073C- 4C 10 1C	073C JMP PUNT ;TERUG NAAR MONITOR
1000 ;	ROUTINE VOOR HET MEERGEVEN VAN PUNTEN
1000 ;	OP HET GRAFISCH DISPLAY
1100 ;	
1110 ;	
073F- A0 02 17	073F LDA PRCOJ ;SYNCHRONISEER
0742- 10 F8	0742 BPL PUNT ;
0744- 8E 00 17	0744 STX PRCOJ ;
0747- EE 02 17	0747 INC PRCOJ ;
0749- EE 02 17	0749 INC PRCOJ ;
074D- 8C 00 17	074D STV PRCOJ ;
074E- EE 02 17	074E INC PRCOJ ;
0753- A9 10	0753 LDA #10 ;
0755- 80 02 17	0755 STA PRCOJ ;
0758- 68	0758 RTS ;
0759- E8 C8 08	0759 .BY #E8 #C8 #08 #08 #08 #08 #08 #08 ;1"-NIVEAUS
075C- 88 68 48	
075F- 28 08	
0761- F8 08 08	0761 .BY #F8 #08 #08 #08 #08 #08 #08 #08 ;0"-NIVEAUS
0764- 98 78 58	
0767- 38 18	
0769- 00	0769 .BY #00

Lijst 5

1640 ;	TESTROUTINE
1650 ;	
1660 ;	
1670 ;	.BR \$0800
1680 ;	
1690 ;	
0800- A2 00	0800 LDW #000 ;SCHRIFJ. BUFFER
0802- 8A	0802 TWA ;AUSCH. NET
0803- 9C 00	0803 STA #BUFFER,X ;NULLEN
0805- E8	0805 TRV ;
0806- D0 FA	0806 BNE LUST ;
0808- 4C 00 07	0808 JMP DSPL ;SPRING NAAR DISPLAYROUTINE

Lijst 6

1730 ;	.BR \$0800
1800 ;	
1810 ;	
1820 ;	ROUTINE VOOR HET NIETEN
1830 ;	VAN PERIODETIDEN
1840 ;	
1850 ;	
0800- A2 00	0800 LDW #000 ;AUSCH. OF
0802- B5 00	0802 LDA #BUFFER,X ;ZEH
0804- E8	0804 TRV ;AUSCH. PERIODE
0805- 2D FF 07	0805 AND MEM ;FLANK
0808- D0 F8	0808 BNE LUSR ;
080A- 85 00	080A LDA #BUFFER,X ;AUSCH. OF
080C- E8	080C TRV ;
080D- 2D FF 07	080D AND MEM ;OPGANGDE
0810- F0 F8	0810 BEQ LUSD ;FLANK
0812- A0 01	0812 LDW #001 ;TELLER = 1
0814- B5 00	0814 LDA #BUFFER,X ;AUSCH. OF
0815- E8	0815 TRV ;
0817- C9	0817 INC ;VOLGENDE
0818- 2D FF 07	0818 AND MEM ;FLANK
081B- D0 F7	081B BNE LUSD ;
081D- B5 00	081D LDA #BUFFER,X ;AUSCH. OF
081F- E8	081F TRV ;
0820- C8	0820 INC ;OPGANGDE
0821- 2D FF 07	0821 AND MEM ;ZEH TEL
0824- F0 F7	0824 BEQ LUSD ;ZEH. NUTTEL
0826- 98	0826 TRV ;PERIODEN
0827- 8D 69 07	0827 STA VSRU ;BEWAAR DIT
082A- 8A	082A RSL A ;AUSCH. GAA. DIG
082B- 8A	082B RSL A ;ZET 7
082C- 8A	082C RSL A ;ZEH TREK. ER
082D- 38	082D SEC ;Y. VANAF
082E- ED 69 07	082E DEC VSRU ;
0831- 4C 1D 1C	0831 JMP MONIT

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 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 „invoeroutine-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 μ s 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.

Lijst 1 Korte invoerroutine.

Lijst 2 Lange, doch snelle invoerroutine.

Lijst 3 Programma dat de snelle invoerroutine in het geheugen schrijft.

Lijst 4 Routine voor het weergeven van logische signalen op het grafisch display.

Lijst 5 Testprogramma.

Lijst 6 Programma om periodetijden te meten.