

Testprogramma voor EPROM-programmer

MSX CLUB MAGAZINE 30

Theo van Dooren

Scanned, ocr'ed and converted to PDF by HansO, 2001

Het programma EPROMTST.BAS is een testprogramma voor de MSX eprom-programmer, zoals die gepubliceerd is in *Elektuur*, in de uitgaven van maart en april 1987.

Deze eprom-programmer is door mij gebouwd en voldoet uitstekend. Het is hiermee mogelijk om programma's, zowel in basic als in machinetaal, in een eprom op te slaan, zodat deze als een insteek-cartridge te gebruiken is. Ook is het hiermee mogelijk om programma's in een eprom te kopiëren naar je eigen eprom. Een toepassing daarvan is bijvoorbeeld de disk-rom in de Philips VG 8235, die alleen geschikt is voor de besturing van een enkelzijdig diskette-station, te vervangen door een eprom waarin de disk-basic van de NMS 8250 is geprogrammeerd. Deze methode kan ook toegepast worden om de oude versie van de Philips Floppy Discdrive Interface geschikt te maken voor de besturing van een dubbelzijdig diskette-station. Bovendien is de besturing van een enkelzijdige drive met deze interface dan ook heel wat sneller. BASIC wijzigen of uitbreiden...

Een andere interessante toepassing is om de MSX-Basic-Rom in je computer aan je eigen wensen aan te passen. Zo heb ik de data van deze rom in een gewijzigde vorm in een eprom gezet, zodanig dat mijn NMS 8250 nu opstart met een eigen tekst op het scherm, en een aan mijn eigen wensen aangepaste funktietoetsenindeling. Door in de computer nog een eprom toe te passen, waarin een monitor-, een debugger- en een disassembler-programma zijn geprogrammeerd, kan ik met een eenvoudige call vanuit basic een van deze toepassingen aanroepen. Tevens kan ik ook enkele veel door mij geraadpleegde tabellen, bijvoorbeeld de ASCII-tabel of de BIOS-adressenlijst, die eveneens in deze eprom zijn opgeslagen, met een call op het scherm toveren. Men kan natuurlijk ook eprom's programmeren voor een andere toepassing dan in een MSX, vooral als men de electronica-hobby beoefent.

Terug echter naar het test-programma. De MSX-eprom-programmer wordt op de computer aangesloten via de 32-bits I/O-cartridge, die in de uitgave van *Elektuur*, januari 1987, beschreven is. Met deze interface worden de mogelijkheden van de MSX, wat betreft de verbindingen met de buitenwereld, enorm uitgebreid. De interface bevat namelijk twee Z80-PIO's en een 280-CTC plus uiteraard de nodige aan-stuur-logica. Nu beschikt een PIO (Parallel Input/Output chip) over twee 8-bit poorten, die elk zowel voor input als voor output (zelfs per bit) geprogrammeerd kunnen worden, iedere poort heeft ook nog twee control-lijnen (Ready en Strobe) voor handshaking tussen het aangesloten

apparaat en de poort. Ook kan de PIO op basis van interrupt met de Z80 in de MSX samenwerken.

Met twee PIO's beschikt deze interface dus over vier 8-bits poorten, ofwel over 32-bits input/output faciliteiten. De Z80-CTC (Counter-Timer-Circuit) is een programmeerbare chip met vier kanalen die ieder voor tijd- of telfuncties kunnen worden ingezet. Alle kanalen hebben een trigger-in-put, om een bepaalde timer- of telfunctie te starten, dit kan echter ook software-matig gebeuren. Drie van de vier kanalen hebben een uitgang waarop een puls wordt gegeven als de timer of teller de nulstand heeft bereikt. Hiermee kan een randapparaat aangestuurd worden, of kan via een interrupt naar een bepaalde routine in een programma gesprongen worden. Alle vier kanalen werken geheel zelfstandig, zonder dat de MSX hier verder naar om hoeft te kijken.

Besturingen

Deze interface opent vele perspectieven voor diverse besturingen met een MSX. Om een voorbeeld te stellen : een klimaatregeling in een kweekkas. Men kan via een PIO-poort en een analoog/digitaal-omzetter de temperatuur en de vochtigheidsgraad in de kas met vaste tijds-intervallen opnemen. Aan de hand hiervan kunnen via een andere PIO-poort een verwarmings-element, een ventilator, de sproei-installatie en de beluchtungs-ramen onafhankelijk van elkaar worden bestuurd. Met behulp van de CTC kan een nauwkeurige klok gemaakt worden, die de tijds-intervallen en de omschakeling van een dag- en nacht-programma verzorgt. Via de twee andere PIO-poorten kan dan een display worden aangesloten waarop de tijd, de temperatuur en de vochtigheidsgraad achtereenvolgens doorlopend worden weergegeven. Op het scherm kan het verloop van het proces in een diagram worden opgetekend, terwijl een aangesloten printer hiervan iedere dag een uitdraai maakt. De gewenste temperatuur en vochtigheidsgraad en de nauwkeurigheid waarbinnen deze geregeld moeten worden kan via het toetsenbord aan het programma worden doorgegeven, en op ieder moment gewijzigd worden. Andere toepassingen van deze interface zijn onder andere te verwezenlijken in de model(spoor)bouw. Maar ook kan men een printer op een PIO-poort aansluiten, waarbij de mogelijkheid bestaat, om via de interrupt, alleen dan de printerbuffer met data te vullen wanneer de printer met het BUZY-sigitaal hierom vraagt. Zodat de computer met het lopende programma verder kan gaan en de relatief trage printer tussendoor op zijn wenken kan bedienen. Normaal is tijdens het printen de rest van de computer buiten gebruik, omdat de micro-processor voortdurend de BUZY-lijn van de printer blijft controleren. Uiteraard moet bij al deze toepassingen de nodige programmatuur geschreven worden. Om terug te komen op de toepassing van de interface met de eprom-programmer, eerst worden de PIO's geïnitieerd. Dit betekent dat vooraan in het programma een routine wordt opgenomen die de PIO-poorten voor input of voor output geschikt maken, en of dat er van de interrupt en handshake gebruik wordt gemaakt. Voor de CTC wordt hiermee ingesteld welke kanalen als timer en welke als teller moeten fungeren, en worden bovendien de instelwaarden doorgegeven. Deze instellingen worden verricht door bepaalde binaire code's naar de control-adressen te sturen door middel van de OUT-instructie. Het adresgebied wordt op de interface vooraf d.m.v. jumpers ingesteld, en ligt zodoende hardware-matig vast. De instellingen kunnen tijdens het uitvoeren van een programma bovendien gewijzigd worden. Dit om bijvoorbeeld een timer opnieuw in te stellen, of om een PIO-poort om te

schakelen van input naar output, als deze de data op de databus moet in- of uitvoeren. De code's voor alle mogelijke instellingen kan men vinden in de diverse boeken die hierop betrekking hebben, bijvoorbeeld " Sturen met de Z-80 microprocessor" van R. Lingier, of " Z80 Applications " van James W Coffron. Ook in de betreffende uitgaven van Elektuur vindt men een beknopte omschrijving hiervan. De instelcode's voor de CTC heb ik in een bijgaande tabel weergegeven. Voor de eprom-programmer wordt de volgende instelling verricht: Twee 8-bits PIO-poorten A en B worden in de output-mode gezet, omdat deze met de adreslijnen van de eprom zijn doorverbonden, en deze adresbus kan maximaal 16 bits breed zijn, afhankelijk van de grootte van de te programmeren eprom. Van de 8-bits C-poort, worden 7 bits in de output-mode gezet, voor de software-matige bediening van de diverse instellingen op de programmer, zoals de programmeerspanning, het in- en uitschakelen van de voedingsspanning e.d. Dit zijn de control-lijnen. Het zevende bit van deze C-poort wordt in de input-mode gezet, hierdoor moet namelijk het reset-sigitaal vanaf de programmer worden doorgegeven naar de computer. De D-poort van de interface is aan de ene kant verbonden met 8-bits databus in de computerr en aan de andere kant met de datalijnen van de eprom. Deze poort wordt tijdens het uitvoeren van het programma ingesteld in de in- of output-mode. Dit is afhankelijk van de instelling in het programma, of men een eprom wil uitlezen, of dat een eprom geprogrammeerd moet worden. Eveneens worden tijdens het uitvoeren van het programma, drie kanalen van de CTC ingesteld.

Dit is nodig tijdens het programmeren van een eprom. Een timer zorgt dan voor een vertragingstijd van 4 microseconden, zodat de data- en adresbus zeker stabiel zijn alvorens ze in de te programmeren eprom worden toegelaten. Een tweede timer genereert pulsen met een tijdsduur van 0,5 msec., die door het derde kanaal, dat als teller is ingesteld, worden geteld. Door de waarde in deze teller wordt aldus de lengte van de programmeer-puls bepaald. Deze waarde is weer afhankelijk van de instellingen in het programma, die worden bepaald door het type eprom dat men wil programmeren.

EPROMTST.BAS

Met het programma "EPROMTST.BAS" kan men nu met de hand, via de functie-toetsen, allerlei instellingen van de eprom-programmer verrichten. Dit kan erg makkelijk zijn als men bijvoorbeeld de programmeer-spanningen moet afregelen, of ingeval van een onjuiste werking van de programmer. Tevens kan men op het scherm zien wat de instelling op dat moment is.

De functie-toetsen zijn als volgt ingedeeld:

- • F1 rmaakt de adreslijnen achtereenvolgens "hoog".
- • F2 :maakt de datalijnen achtereenvolgens "hoog".
- • F3 :geeft een 50 msec programmeer-puls (als hulp-jumper geplaatst is).
- • F4 :zet de voedingsspanning aan de eprom op 5 Volt of op 6 Volt.
- • F5 :zet de programmeerspanning achtereenvolgens op 5, 12.5. 21 of 25 Volt.
- • F6 :reset (alle instellingen terug op beginsituatie).
- • F7 :schakelt de eprom in of uit dmv CE-lijn (Chip Enable).
- • F8 :schakelt het datatransport om dmv OE-lijn (Output Enable).
- • F9 :schakelt de voedingsspanning aan de eprom in of uit.

- F10: schakelt de programmeerspanning in of uit.

Dit test programma is dus alleen te gebruiken in combinatie met de eprom-programmer. Indien men de interface voor een andere toepassing gebruikt, is het wellicht mogelijk, het programma hierop aan te passen. De bedoeling van dit artikel is niet om een volledige beschrijving te geven van de 32-bits I/O-interface met de daaraan gekoppelde eprom-programmer, het is meer bedoeld als een kennismaking met deze interessante toepassing van de MSX-computer. Voor de meer geïnteresseerden onder de lezers, verwijs ik dezen naar de desbetreffende uitgaven van Elektuur,

Programmeren van de Z80-CTC

ADRESSERING :

ADRES + 0 = kanaal 0, en interrupt-vector register

ADRES + 1 = kanaal 1

ADRES + 2 = kanaal 2

ADRES + 3 = kanaal 3

CONTROLWORD :

b7	b6	b5	b4	b3	b2	b1	b0
interrupt	mode	range	slope	trieeer	time-constant	reset	herkenbit
1 = enable interrupt 0 = disable interrupt	1 = counter-mode 0 = timermode	1 = 256 0 = 16 prescale n'system-clock, deer. downcounter (alleen timer)	1 = dalende 0 = stijgende flank, start timer /counter	1 = ext. trigger start 0 = start na load timeconstant register (alleen timer)	1 = volgende woord is voor timeconstant register 0 = geen volgend woord	1 = stopt timer/counter tot load volgende timeconstant	1 = controlwoord 0 = interrupt-vector

INTERRUPT-VECTOR

b7	b6	b5	b4	b3	b2	b1	b0
X	x	X	X	X	n	n	0

xxxxx = door programmeur zelf op te geven deel van vector-byte

nn0 = kanaalnummer wordt door CTC in het vector geplaatst (nn = 0...3)

Voorbeeld :

Interruptvector in IR van Z80 is 19H

Interruptvector in IR van CTC is F0H

De adressen van de interrupt-routines staan dan als volgt:

voor kanaal 0 op de adressen 19F0H en 19F1H

voor kanaal 1 op de adressen 19F2H en 19F3H

voor kanaal 2 op de adressen 19F4H en 19F5H

voor kanaal 3 op de adressen 19F6H en 19F7H

```

10 ' *****
11 ' **
12 ' **          TEST  PROGRAMMA      MSX-EPROM-PROGRAMMER          **
13 ' **
14 ' **          (uit Elektuur maart 1987, blz. 72)                **
15 ' **          aangepast door Theo van Dooren (voor MSX-2)      **
16 ' **
17 ' *****
18 '
19 ' Programmanaam: "EPROMTST.BAS"
20 SCREEN 0:COLOR 15,1,1:CLS:WIDTH 80:KEY OFF
30 PRINT
40 PRINT"          ===== TEST  PROGRAMMA  EPROMMER ===== "
50 PRINT
60 PRINT
70 PRINT
80 PRINT"
90 PRINT"          F1          F2          F3          F4          F5          "
100 PRINT
110 PRINT"  Adreslijn      Datalijn      PGM-puls      Vcc =    V    Vpp =    V"
120 PRINT
130 PRINT"          hoog          hoog          (jumper J1)          "
140 PRINT
150 PRINT
160 PRINT
170 PRINT"          F6          F7          F8          F9          F10 "
180 PRINT
190 PRINT"  Reset      Chip Enable      Richting data      Vcc          Vpp "
200 PRINT
210 PRINT"          off          in          off          off"
220 PRINT
230 PRINT
240 PRINT"          Verlaat het programma met CTRL/STOP"
250 PRINT
255 LOCATE , ,0
260 '===== address-area
270 A=0*16 : ' instelling adresgebied van de interface
280 F1=0: F2=0: F4=5 :F5$=" 5 " :F7$="off": F8$="in " : F9$="off": F10$="off"
290 DA= 4+A : DB= 5+A : DC= 8+A : DD= 9+A
300 CA= 6+A : CB= 7+A : CC=10+A : CD=11+A
310 T0=12+A : T1=13+A : T2=14+A : T3=15+A
320 '===== PIO-port a,b and d as output (mode 3)
330 OUT CA,255 : OUT CA,0 : OUT CA,7 : OUT CA,3
340 OUT CB,255 : OUT CB,0 : OUT CB,7 : OUT CB,3
350 OUT CD,255 : OUT CD,0 : OUT CD,7 : OUT CD,3
360 '===== PIO-port c with 7outputs and 1 input (mode 3)
370 OUT CC,255 : OUT CC,128 : OUT CC,7 : OUT CC,3
380 '===== reset configuration
390 OUT DA,0 : OUT DB,0 : OUT DC,255 : OUT DD,0
400 OUT T2,5 : OUT T2,1
410 OUT T0,3 : OUT T1,3 : OUT T2,3 : OUT T3,3
420 '===== initialisation
430 ON KEY GOSUB 530,590,630,660,710,790,870,920,970,1020
440 FOR I=1 TO 10 : KEY(I) ON : NEXT I
450 ON STOP GOSUB 520 : STOP ON
460 A=1 : B=0 : C=255 : D=1:F1=1:F2=1
470 '===== execution loop
480 OUT DA,A : OUT DB,B : OUT DC,C : OUT DD,D
490 IF INP(DC) < 128 THEN 520 : '===== reset pressed ?
500 GOTO 480
510 '===== on stop routine
520 STOP OFF : GOSUB 790 : OUT DC,C :CLS: END
530 'Rotate address line high ===== key 1 routine
540 KEY(1) OFF
550 IF A=128 THEN A=0 : B=1 :GOTO 570 ELSE A=A*2
560 IF B=128 THEN B=0 : A=1  ELSE B=B*2
570 IF F1=16 THEN F1=0
580 LOCATE 12,8,0:PRINTF1:F1=F1+1:KEY(1) ON:RETURN

```

```

590 'Rotate dataline high ===== key 2 routine
600 KEY(2) OFF : IF D=128 THEN D=1 ELSE D=D*2
610 IF F2=8 THEN F2=0
620 LOCATE 27,8,0:PRINT F2: F2=F2+1:KEY(2) ON:RETURN
630 'One program pulse of 50 ms ===== key 3 routine
640 KEY(3) OFF : OUT T2,&B01010101 : OUT T2,100 : OUT T1,&B00111101
650 OUT T1,7 : OUT T0,&B01010101 : OUT T0,0 : KEY(3) ON : RETURN
660 'Vcc change ===== key 4 routine
670 KEY(4) OFF : C=C AND 8 : C=C+8 : C=C AND 8
680 IF F4=6 THEN F4=5 ELSE F4=6
690 LOCATE 56,8,0:PRINT F4
700 C=(INP (DC) AND 247) OR C : KEY(4) ON : RETURN
710 'Vpp change ===== key 5 routine
720 KEY(5) OFF : C=C AND 3 : C=C-1 : C=C AND 3
730 IF F5$=" 5 " THEN F5$="12.5": GOTO 770
740 IF F5$="12.5" THEN F5$="21 " : GOTO 770
750 IF F5$="21 " THEN F5$="25 " : GOTO 770
760 IF F5$="25 " THEN F5$=" 5 " : GOTO 770
770 LOCATE 70,8,0:PRINT F5$
780 C=(INP (DC) AND 252) OR C : KEY(5) ON : RETURN
790 'Reset ===== key 6 routine
800 KEY(6) OFF : C=255
810 F1=0: F2=0: F4=5 :F5$=" 5 " :F7$="off": F8$="in " : F9$="off": F10$="off"
820 F$=" " :LOCATE 13,8,0: PRINT F$ :LOCATE 28,8,0: PRINTF$
830 LOCATE 56,8,0:PRINT F4 :LOCATE 70,8,0:PRINT F5$
840 LOCATE 22,18,0:PRINT F7$:LOCATE 37,18,0:PRINT F8$
850 LOCATE 52,18,0:PRINT F9$:LOCATE 68,18,0:PRINT F10$
860 KEY(6) ON : RETURN
870 'Chip enable ===== key 7 routine
880 KEY(7) OFF : C=C AND 64: C=C+64: C=C AND 64
890 IF F7$="off" THEN F7$="on "ELSE F7$="off"
900 LOCATE 22,18,0:PRINT F7$
910 C=(INP (DC) AND 191) OR C : KEY(7) ON : RETURN
920 'Output enable ===== key 8 routine
930 KEY(8) OFF : C=C AND 32: C=C+32: C=C AND 32
940 IF F8$="in " THEN F8$="out"ELSE F8$="in "
950 LOCATE 37,18,0:PRINT F8$
960 C=(INP (DC) AND 223) OR C : KEY(8) ON : RETURN
970 'Vcc on/off ===== key 9 routine
980 KEY(9) OFF : C=C AND 16: C=C+16: C=C AND 16
990 IF F9$="off" THEN F9$="on "ELSE F9$="off"
1000 LOCATE 52,18,0:PRINT F9$
1010 C=(INP (DC) AND 239) OR C : KEY(9) ON : RETURN
1020 'Vpp on/off ===== key 10 routine
1030 KEY(10) OFF : C=C AND 4 : C=C+4 : C=C AND 4
1040 IF F10$="off" THEN F10$="on "ELSE F10$="off"
1050 LOCATE 68,18,0:PRINT F10$
1060 C=(INP (DC) AND 251) OR C : KEY(10) ON : RETURN

```