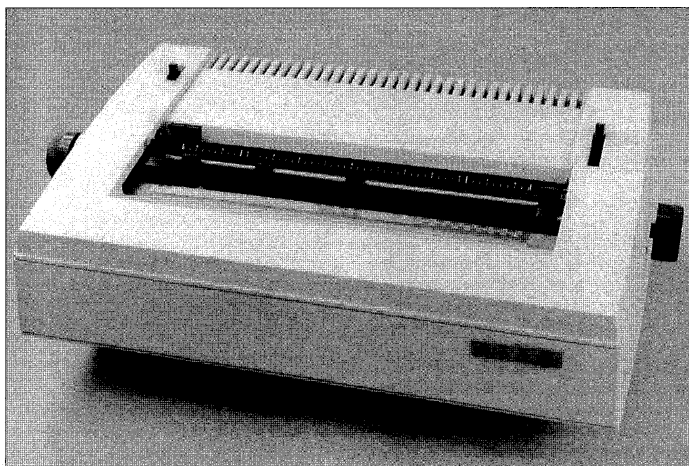


Aansturen van de QUME printer (2)



In de vorige PTC-print stond hoe we de QUME letterwiel printer met behulp van UNIFACE eenvoudig op onze computer konden aansluiten. Nu wordt beschreven hoe vanuit een programma de printer te besturen is. De beschrijving geldt voor de PC, maar door één getalletje te veranderen werkt het op de P2000 of op de MSX. Zo gaat dat met UNIFACE.

BASIC

Ieder heeft een eigen voorkeur voor een programmeertaal. Persoonlijk denk ik dat een besturing voor een printer in machinetaal zou moeten. Maar omdat we het nu alleen over de principes willen hebben gaan we de printer direct vanuit BASIC aansturen. Iedereen die kan programmeren kent ook wel een beetje BASIC. En wat in BASIC kan, kan in uw favoriete programmeertaal ongetwijfeld beter, sneller en compacter.

De printer

Eerst nog even wat over de printer zelf. Dit om, afgezien van de gebouwde interface, te zien of hij wel werkt. Het meest eenvoudig is om hem gewoon maar eens aan te zetten. Dan moet het eerste lampje aangaan (power) en de printkop zoekt zijn uitgangpositie op. Zet hem maar weer eens uit. De kop kan dan met de hand naar het midden van de rol worden bewogen. Als je hem dan weer aan zet beweegt hij piepend naar links. Dat hoort zo.

Het rechtse lampje RIBBON gaat aan als er een inktlint in zit.
Het middelste lampje READY gaat aan als:

- alle interne controles goed zijn verlopen,
- de kop in zijn beginpositie is gekomen,
- de deksel gesloten is,
- de interface is aangesloten (PRINTER SELECT)

Als het READY lampje niet brandt accepteert de printer geen commando's, behalve RESTORE. Branden alle drie de lampjes dan is de printer in orde en kunnen we verder gaan.

RESTORE

We beginnen nu met ons programma. Omdat we niet weten in elke toestand we de printer hebben achtergelaten doen we eerst een RESTORE. De printer controleert zichzelf daarbij en zet de printkop in de uitgangpositie. Daarvoor maken we het volgende programma in BASIC:

```
10 GOSUB 100
90 END
100 REM *** Restore ***
110 D=&H310:A=D+1
120 OUT A,0:OUT D,0
150 GOTO 900
900 REM *** Geef STROBE-puls ***
910 OUT A,1:OUT A,0:RETURN
```

Regel 10 tot 90 is het hoofdprogramma, niet zo lang, maar dat houdt het overzichtelijk.

De subroutine van regel 100 doet een RESTORE. We zetten dat er maar even boven tussen sterretjes, dan weten we straks nog wat alles voorstelt.

In regel 110 worden de UNIFACE-adressen vastgelegd. Voor de PC hex 310 voor data (D) en hex 311 voor adressen (A). Wie een P2000 heeft zet hier naar D=&H60, en voor de MSX: D=&H30. Dat is de enige verandering, A is altijd één hoger.

In regel 120 worden alle 16 output-bits van UNIFACE nul gemaakt. Als we nu een STROBE-puls geven gaat er een RESTORE-puls naar de printer. Omdat we die STROBE nog veel vaker nodig hebben is daar een apart stukje programma aan gewijd. Als afsluiting van de subroutine RESTORE springen we daarheen.

Geef STROBE-puls

Door even een 1 te sturen naar UNIFACE poort A wordt de 74LS42 in onze interface geactiveerd. Afhankelijk van bit 7 en bit 6 van poort D komt er dan een RESTORE-puls (00), een CHARACTER STROBE (01), een CARRIAGE STROBE (10) of een PAPER FEED MAIN STROBE (11)

Als we in regel 120 poort D helemaal nul hebben gemaakt komt er een RESTORE. Run het programma maar eens, je ziet het READY-lampje op de printer dan even uit gaan, de kop beweegt even, want hij stond al goed, en het lampje gaat weer aan.

Twee hulproutines

We hebben nu eerst nog twee hulproutines nodig. Bij RESTORE hebben we zomaar een STROBE-puls gegeven. Bij RESTORE mag dat ook. Het is een soort reset van de printer en dat kan altijd. Tijdens het normale gebruik moeten we echter steeds wachten tot de printer gereed staat om een commando (STROBE-puls) te ontvangen. Daarom voegen we toe in het programma:

```
800 REM *** Wacht op READY ***
810 IF INP(D)>127 THEN 810
```

In regel 810 wordt gekeken naar INPUT BUFFER READY van de printer. Als op bit 7 een 1 staat is de printer niet klaar. Alle inputs en outputs van de printer

zijn in negatieve logica, dus een "0" is een "1" en een "1" is een "0". Dat is even lastig in de software, we hadden het ten koste van meer IC's (hardware) kunnen omdraaien, maar lastig gaat ook. De routine heeft geen RETURN. Hij loopt zodra de printer ready is door in de strobe-routine. Dat lijkt gek, maar in de praktijk wil je het eigenlijk steeds op deze manier.

Een andere routine die we vaak nodig hebben is het vullen van het latch-IC 74LS373. Daarom voegen we toe:

```
700 REM *** Geef LATCH-puls
***
710 OUT A,128:OUT A,0:RETURN
```

Wie vaker met UNIFACE gespeeld heeft merkt op dat de besturing van deze gesimplificeerde interface toch erg veel lijkt op de volledige UNIFACE. De STROBE blijkt op adres 1 te zitten en de latch op adres 128.

Nu is het tijd om iets echt te gaan doen.

Druk een teken af

We voegen nu de volgende regels toe aan het programma:

```
20 T=65:GOSUB 200
200 REM *** Druk teken T af ***
210 IF T=32 THEN RETURN
220 TL=T*2 MOD 128
230 OUT D,TL XOR 127
240 GOSUB 700
250 TH=INT(T/64)
260 OUT D,TH XOR 3 +64
270 GOTO 800
```

Wat doet dit stukje?

Regel 20 vertelt dat teken 65, dat is de hoofdletter A, op het papier moet worden gezet.

Regel 200 en volgende is de letter af-druk routine

Regel 210: als het een spatie is (32) dan moet je niets afdrukken. De QUME kent 96 letters, op de plaats van de spatie staat ook iets. Afhankelijk van het letterwiel kan dat van alles zijn, een pond-teken, een hekje, een doorgestreepte c, zoek maar uit. Om een spatie af te drukken heb je geen spaak aan het wiel nodig. Dus gewoon niks doen.

Regel 220 begint een splitsing van T in een laag deel (TL) en een hoog deel (TH). Voor TL moeten we T verdubbelen, want DATA 1/2 doet niet mee, en dan alleen de laagste 7 bits nemen.

Regel 230 stuurt TL naar poort D. Maar omdat de printer met negatieve logica werkt moeten alle bits met XOR 127 worden omgekeerd in polariteit. Bit 7 is RIBBON LIFT. Hier moet een "0" blijven staan. Zoniet dan blijft het lint laag

staan en komt er een onzichtbare letter op het papier.

Regel 240 roept de latch-routine aan, TL moet in de latch worden gezet.

Regel 250 doet TH door van T de laagste 6 bits te verwijderen.

Regel 260 stuurt TH naar poort D. Ook nu weer moeten de bits die resteren geïnverteerd worden. Bovendien moeten we een CHARACTER STROBE kiezen, daarom wordt er 64 bij opgeteld.

Regel 270 stuurt de routine door naar de strobe-routine. Maar wél eerst wachten op printer ready!

Als het programma nu ge-RUN-d wordt drukt de printer een A op het papier. Draai de rol maar even omhoog dan kun je hem zien.

Printkop verplaatsing

Nu moet de printerkop verplaatst worden. Dat doet de printer flitsend, maar doe je daarbij iets fout dan knalt hij links of rechts tegen het einde van de rail. Daar kan hij tegen, maar je schrikt enorm. Het ready-lampje gaat dan uit, maar door de RESTORE aan het begin van het programma komt alles weer goed. We voegen de volgende programmaregels toe:

```
30 H=12:GOSUB 300
300 REM *** Printerkop
verplaatsen ***
310 IF H=0 THEN RETURN
320 R=48:IF H<0 THEN
H=-H:R=32
330 HL=H MOD 128
340 OUT D,HL XOR 127
350 GOSUB 700
360 HH=INT(H/128)
370 OUT D,HH XOR 15 +R+128
380 GOTO 800
```

Deze routine zit bijna net zo in elkaar als de karakter-routine. De verplaatsing in horizontale richting wordt gegeven in H. Positief naar rechts, negatief naar links. De stapgrootte is 1/120 inch, en DATA 1/2 doet nu wél mee. Blijkbaar hadden de oudere printers stapjes van 1/60 inch.

In regel 320 wordt de richting afgescheiden. De R is nodig bij HH, omdat ook deze in negatieve logica zit lijkt het wat merkwaardig zo.

In regel 370 wordt HH samengevoegd met de richting-bits op DATA 1024 en 2048 en de keuze voor een CARRIAGE STROBE.

Als alles goed is gaat schuift de printerkop na een A een stukje naar rechts. Probeer ook maar eens:

```
30 H=600:GOSUB 300
en de printerkop vliegt naar het eind
van de regel. Nu ook negatieve waar-
```

den nog even proberen:

```
30 H=600:GOSUB 300:
H=-600:GOSUB 300
nu vliegt hij ook weer terug.
```

Rol verplaatsing

Nu nog de papierrol besturing. Deze gaat in stapjes van 1/48 inch, meegegeven in de variabele V. Voeg weer de volgende regels toe:

```
40 V=10:GOSUB 400
400 REM *** Papier verplaatsing
***
410 IF V=0 THEN RETURN
420 R=48:IF V<0 THEN
V=-V:R=32
430 VL=V*2 MOD 128
440 OUT D,VL XOR 127
450 GOSUB 700
460 VH=INT(V/64)
470 OUT D,VH XOR 15 +R+192
480 GOTO 800
```

Deze routine wijkt nauwelijks af van de H-routine. Zie bij het RUN-en dat de printer zijn papier verplaatst tegelijk met het verplaatsen van de printerkop. Ja, het papier kan ook weer naar beneden, kijk maar:

```
40 V=40:GOSUB 400:V=-40:
GOSUB 400.
```

Een regel tekst

Nu hebben we alle ingrediënten om de printer geheel te besturen. De regels 20, 30 en 40 kunnen nu weg, we maken er andere van:

```
20 A$="Kijken of hij het nu doet."
30 FOR I=1 TO LEN(A$)
40 T=ASC(MID$(A$,I,1)):
GOSUB 200
50 H=12:GOSUB 300
60 NEXT I
70 V=8:GOSUB 400
```

Bij RUN tikt de printer keurig de tekst regel op het papier. Aan het einde rolt het papier een regel omhoog. Omdat het maar één regel is hebben we nog geen terugloop wagen nodig. De RESTORE op regel 10 zorgt daar wel voor. Toch moet dat nog even echt geprobeerd worden. Daarom maken we wat meer regels tekst in DATA aan:

```
1000 DATA "Als het goed is
werkt hij nu"
1010 DATA "met een snelle
terugloop van"
1020 DATA "de printerkop."
9000 DATA "EINDE"
```

Nu in het hoofdprogramma de volgen-

de regels wijzigen:

```
10 GOSUB 100:HT=0
20 READ A$:IF A$="EINDE"
  THEN 90
50 H=12:GOSUB 300:HT=HT+H
80 H=-HT:GOSUB 300:HT=0:
  GOTO 20
```

HT is een totalisator, die bijhoudt hoeveel de kop horizontaal verplaatst is. In regel 80 verplaatsen we dat aantal terug. En zie, op deze manier is een heel veel vol te printen.

Tips

Deze BASIC-routines lopen op een XT nogal traag. Maar u maakt ze toch in een andere taal, nietwaar? Daarom nog wat tips voor als u definitief aan de slag gaat.

- Als code-ruimte krap is combineer dan de routines zoveel mogelijk. Er is erg veel overeenkomst

- Er zijn printwielen met verschillende letterbreedte. Nummer 10 betekent 10 letters per inch, de H moet dan 12 zijn (12/120). Er zijn ook wielen 12 en 15, maar ook wielen met proportionele letters. Maak dan een tabel H-waardes op ASCII volgorde.

- Verplaatsen van de printerkop over kleine afstandjes kost relatief veel tijd. Nodeloze verplaatsing is te voorkomen door in het programma bij te houden waar de kop staat en waar hij zou moeten staan. Zolang er geen tekens geprint worden, houden we alleen bij in het programma waar de kop zou moeten staan. Pas als het echt nodig is, dus vóór het printen van een teken, wordt de kop verplaatst in de juiste richting en over de juiste afstand. Zo slaat de printer spaties domweg over. Dat is helemaal de moeite bij het "printen" van de linker marge van een tekst, de kop doet dat dan helemaal niet, maar springt direct naar het begin van de regel.

- Heen en weer printen kan ook. Maar mij is gebleken dat de printerkop dan minder nauwkeurig op zijn plaats komt. Blijkbaar maakt het uit of hij van links komt of van rechts. Heen en weer printen geeft overigens nauwelijks winst in tijd. Grote verplaatsingen gaan relatief snel. Een veel mooiere bladspiegel is het resultaat.

- De printer kijkt zelf niet naar het papier en het inktlint. Daarvoor zijn wel sensors en status-bits. In het programma zijn daar mooie foutmeldingen voor te maken. De papiersensor zit echter bij de invoer. Een los A4 blad heeft nog plaats voor vele regels als hij uit de papiersensor loopt. Uw programma

zou daar rekening mee kunnen houden.

Naschrift

Het aan de praat brengen van een merkwaardig soort printer als deze is een soort puzzel. Dat gaat niet meteen goed en regelmatig vraag je jezelf af waarom het ding niet werkt zoals zou moeten. Maar het ligt vrij zeker aan jouw programma en niet aan de prin-

ter. Dat spitst de geest en het vergroot de voldoening als het eindelijk toch nog werkt. Wij hopen met dit soort artikelen meer mensen te prikkelen ook eens van de gebaande paden af te wijken en eens "iets gek" te doen. En als het dan lukt, laat het de andere lezers van PTC-print dan alstublieft eens weten. Want samen komen we verder. Dit artikel is tenslotte ook geboren uit de brief van Donald Prins uit Brummen.

Klaas Robers

```
10 GOSUB 100:HT=0
20 READ A$:IF A$="EINDE" THEN 90
30 FOR I=1 TO LEN(A$)
40 T=ASC(MID$(A$,I,1)):GOSUB 200
50 H=12:GOSUB 300:HT=HT+H
60 NEXT I
70 V=8:GOSUB 400
80 H=-HT:GOSUB 300:HT=0:GOTO 20
90 END
100 REM *** Restore ***
110 D=6H0:A=D+1
120 OUT A,0:OUT D,0
150 GOTO 900
199 REM
200 REM *** Druk teken T af ***
210 IF T=32 THEN RETURN
220 TL=T*2 MOD 128
230 OUT D,TL XOR 127
240 GOSUB 700
250 TH=INT(T/64)
260 OUT D,TH XOR 3 +64
270 GOTO 800
299 REM
300 REM *** Printkop verplaatsen ***
310 IF H=0 THEN RETURN
320 R=48:IF H<0 THEN H=-H:R=32
330 HL=H MOD 128
340 OUT D,HL XOR 127
350 GOSUB 700
360 HH = INT(H/128)
370 OUT D,HH XOR 15 +R+128
380 GOTO 800
399 REM
400 REM *** Papier verplaatsing ***
410 IF V=0 THEN RETURN
420 R=48:IF V<0 THEN V=-V:R=32
430 VL=V*2 MOD 128
440 OUT D,VL XOR 127
450 GOSUB 700
460 VH=INT(V/64)
470 OUT D,VH XOR 15 +R+192
480 GOTO 800
499 REM
700 REM *** Geef LATCH-puls ***
710 OUT A,128:OUT A,0:RETURN
799 REM
800 REM *** Wacht op READY ***
810 IF INP(D)>127 THEN 810
899 REM
900 REM *** Geef STROBE-puls ***
910 OUT A,1:OUT A,0:RETURN
1000 DATA "Als het goed is werkt hij nu"
1010 DATA "met een snelle terugloop van"
1020 DATA "de printerkop."
9000 DATA "EINDE"
```

Dit is de listing van het testprogramma afgedrukt door de QUME zelf. Het programma loopt in deze vorm goed op de P2000. Voor de PC en MSX is alleen regel 110 iets anders. Alle routines zijn hierin terug te vinden. Kijk in geval van twijfel eerst in deze listing, niet in de tekst.