

Een binair uurwerk

Jean Verswijvelen

Tijdens een kort verblijf in een klein Duits stadje kwam ik voorbij een zaak waar uurwerken verkocht werden. In het uitstalraam lagen een paar toestellen waarvan ik niet zo onmiddellijk de bedoeling zag. Bij nader onderzoek stelde ik vast dat het polsuurwerken waren met een binaire aflezing. Bij bevraging later via Internet bleken er aanbiedingen voor zulke toestellen te bestaan.

De "wijzerplaat" bestond hoofdzakelijk uit twee rijen LED's. Een rijtje van vier bovenaan en een rijtje van zes onderaan. Zij stelden binaire getallen voor die verschillen naargelang de LED oplicht of niet (dit overeenstemmend met enen en nullen).

Kort narekenen leert ons dat de vier LED's bovenaan voldoende zijn om getallen van 0 tot en met 15 ($8 + 4 + 2 + 1$) voor te stellen en dat de zes LED's onderaan volstaan voor getallen tot en met 63 ($32 + 16 + 8 + 4 + 2 + 1$). Dit is dus voldoende om een uurwerk te simuleren gebaseerd op een twaalfurige cyclus.

De uitdaging om deze uurwerken te simuleren was groot. Dus aan het werk om een programma te schrijven in QuickBasic (de programmeertaal die indertijd bij Microsoft DOS bijgeleverd werd). Ook het vrij downloadbare FreeBasic leent zich tot het maken van het programma.

De opzet lukte maar voldeed niet helemaal aan mijn verwachtingen. Het duurt namelijk één minuut vooraleer er verandering optreedt. Daarom werd het programma uitgebreid om ook seconden voor te stellen, en wel door een derde rijtje LED's (eveneens zes).

De tevredenheid was groot en daarom nog een stapje verder, namelijk een cyclus van 24 uur simuleren. Dus vijf LED's ($16 + 8 + 4 + 2 + 1 = 31$) bovenaan en ze wat verschuiven zodat de hele zaak terug symmetrisch werd.

Wat ik persoonlijk het leukste vind is dat door het bekijken van het uurwerk de verandering van de binaire getallen heel duidelijk te volgen is. Dit lijkt mij nuttig voor leerlingen die met deze materie geconfronteerd worden.

Het programma is als volgt onderverdeeld :

1. het scherm voorbereiden door de benodigde hoeveelheid cirkels te tekenen
2. de computertijd te lezen en op te splitsen in de delen uren, minuten en seconden
3. deze gevonden waarden om te zetten in binaire waarden en strings te vormen met lengte 4, 6 en nogmaals 6
4. de cirkels in te kleuren of niet naargelang de plaats van de énen en nullen in de string
5. dit alles te laten staan tot de tijd verandert

6. intussen ook de gebruiker toelaten het programma te beëindigen door een toets druk.

```
REM Programma Simulatie Binaire Klok met Seconden(12 uur)
REM Jean Verswijvelen (mei 2007)
REM Initialiseren scherm en tekenen van drie rijen LED's
REM Deze zijn initieel zwart ingekleurd
CLS
KEY OFF: SCREEN 9: COLOR 14,0: CLS
PRINT: PRINT
PRINT STRING$(20, " "); "Binaire 12-uurs-klok met secondenaanduiding"
PRINT STRING$(20, " "); "-----"
PRINT: PRINT: PRINT" Dit programma simuleert een binaire klok."
PRINT" Het bestaat uit drie rijen cirkels (simulatie van LED's)."
PRINT" De bovenste rij stelt de uren voor, de middelste de minuten"
PRINT" en de onderste de seconden."
PRINT" Voor het bovenste volstaan 4 LED's (max=12 uur)."
PRINT " Voor de beide andere rijen zijn 6 LED's nodig."
PRINT: PRINT
PRINT" Druk op een toets om het programma te starten"
PRINT" en druk nadien een toets om te eindigen!"
WHILE INKEY$ = "": WEND
CLS
FOR I = 1 TO 4
CIRCLE (200 + 40 * I, 150), 10, 7
NEXT I
FOR I = 1 TO 6
CIRCLE (160 + 40 * I, 200), 10, 7
CIRCLE (160 + 40 * I, 250), 10, 7
NEXT I
PRINT: PRINT: PRINT STRING$(30, " "); "DE BINAIRE KLOK"
REM Hoofdprogramma'
REM Zet de uren, minuten en seconden naar binaire strings met lengte 4
en 6
klok:
H = VAL(MID$(TIME$, 1, 2)):
IF H > 12 THEN H = H - 12
H1 = H: H$ = ""
DO
B=H
IF B = 0 THEN EXIT DO
R$ = LTRIM$(STR$(B MOD 2))
H$ = R$ + H$
H = INT(B / 2)
LOOP UNTIL H = 1
IF H$ <> "" THEN H$ = "1" + H$
IF H1 = 1 THEN H$ = "0001"
WHILE LEN(H$) < 4
H$ = "0" + H$
WEND
M = VAL(MID$(TIME$, 4, 2))
M1=M:M$=""
DO
B=M
IF B = 0 THEN EXIT DO
```

```

R$ = LTRIM$(STR$(B MOD 2))
M$ = R$ + M$
M = INT(B / 2)
LOOP UNTIL M = 1
IF M$ <> "" THEN M$ = "1" + M$
IF M1 = 1 THEN M$ = "000001"
WHILE LEN(M$) < 6
M$ = "0" + M$
WEND
S = VAL(MID$(TIME$, 7, 2))
S1=S:S$=""
DO
B=S
IF B = 0 THEN EXIT DO
R$ = LTRIM$(STR$(B MOD 2))
S$ = R$ + S$
S = INT(B / 2)
LOOP UNTIL S = 1
IF S$ <> "" THEN S$ = "1" + S$
IF S1 = 1 THEN S$ = "000001"
WHILE LEN(S$) < 6
S$ = "0" + S$
WEND
REM Werking van de klok
REM Het zijn 4, 6 en 6 cirkels die zwart of rood worden ingekleurd
REM De kleur is rood voor 1 en zwart voor 0
FOR I = 1 TO 4
IF MID$(H$, I, 1) = "1"
THEN PAINT (200 + 40 * I, 150), 4, 7
ELSE PAINT (200 + 40 * I, 150), 0, 7
NEXT I
FOR I = 1 TO 6
IF MID$(M$, I, 1) = "1"
THEN PAINT (160 + 40 * I, 200), 4, 7
ELSE PAINT (160 + 40 * I, 200), 0, 7
NEXT I
FOR I = 1 TO 6
IF MID$(S$, I, 1) = "1"
THEN PAINT (160 + 40 * I, 250), 4, 7
ELSE PAINT (160 + 40 * I,
250), 0, 7
NEXT I
REM Controle op tijdswijziging en in dat geval terug naar hoofdprogramma
DO
X = VAL(MID$(TIME$, 1, 2)):
Y = VAL(MID$(TIME$, 4, 2)):
Z = VAL(MID$(TIME$, 7, 2))
IF X > 12 THEN X = X - 12
REM Controleer of gebruiker wil eindigen
IF LNKEY$ <> "" THEN END
LOOP UNTIL (X <> H1) OR (Y <> M1) OR (Z <> S1)
GOTO klok

```

Dit is het programma voor de twaalfurencyclus.

De veranderingen die voor een cyclus van vierentwintig uur moeten aangebracht worden zijn :

Vijf cirkels tekenen met de lus :

```
FOR I=1 TO 5
CIRCLE (180 + 4*I , 150) , 10, 7
NEXT I
```

De opdracht

```
IF H > 12 then H = H - 12
```

laten vervallen.

Volgende wijziging aanbrengen

```
IF H1 = 1 THEN H$ = "00001"
WHILE LEN(H$) < 5
```

Ten slotte nog :

```
FOR I = 1 TO 5
IF MID$(H$, I, 1) = "1"
THEN PAINT (180 + 40 * I, 150) , 4, 7
ELSE PAINT (180 + 40 * I, 150) , 0, 7
NEXT I
```

Jean Verswijvelen <jeanverswijvelen@base.be>

Met rustpensioen

Van Eycklei 41A bus 5,
2018, Antwerpen

Bijkomende bedenkingen van de redactieraad

1° Men kan deze Basic programma's op verschillende wijze compileren.

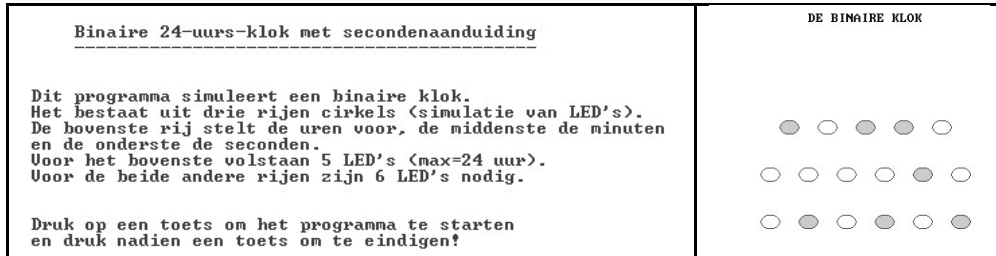
Als er nog mensen met DOS werken kan je gebruik maken van de Quickbasic of een variante. Het nadeel hiervan is dat het eigenlijk een Interpreter is en geen *.exe files aflevert, die men zou kunnen doorgeven. Dit werkt ook onder Windows 98 en zelfs onder Windows XP (misschien moet men wel een beetje sleutelen aan configuratiefiles)

2° Misschien heeft wel iemand nog een Turbo Basic (vroeger van Borland) of van op het internet Firstbasic (proefversie 30 dagen te gebruiken). Deze programma's leveren wel (16 bits) *.exe bestanden af. Bij Turbobasic zal men wel een beetje moeten sleutelen aan het programma.

3° Zoals Jean zegt kan je ook het programma Freebasic gratis downloaden van internet. Het programma werkt zowel onder DOS als onder Windows en is uitgetest tot Windows XP. De

Windows versie geeft 32 bits *.exe files die draaien onder Vista en dus normaal ook onder Windows 7.

Hieronder een paar schermafdrucken.



Ikzelf geef de voorkeur aan de 24-uren versie met daarbij de seconden.

- 4° Wie denkt dat Basic ouderwets is, kan natuurlijk het basisconcept programmeren in een “moderne” taal. Men heeft tegenwoordig de keuze uit Visual Basic, Visual C en C# (lees C Sharp). Deze talen hebben uitgebreide grafische mogelijkheden en ook de voorziening om de tijd per seconde op te vragen. Men kan de thuisgebruikers-versies gratis aanvragen bij Microsoft mits eenvoudige registratie.
- 5° Een zakrekentoestel van de familie TI 84 heeft een ingebouwde tijdsfunctie, die men kan opvragen. Het is mogelijk een programma te maken voor de binaire voorstelling van de tijd. Wegens de beperkte processorsnelheid geven we de voorkeur aan een voorstelling met 0 en 1.



Program Bklok.8xp

```
Repeat A=1
GetTime→L1
L1(1)→U
5→N
seq(0,X,1,N)→L2
For(I,1,N)
If U≥2^(N-I)
Then
U-2^(N-I)→U
1→L2(I)
End
End
```

```

L1(2)→M
6→N
seq(0,X,1,N)→L3
For(I,1,N)
If M≥2^(N-I)
Then
M-2^(N-I)→M
1→L3(I)
End
End
L1(3)→S
6→N
seq(0,X,1,N)→L3
For(I,1,N)
If S≥2^(N-I)
Then
S-2^(N-I)→S
1→L4(I)
End
End
ClrHome
Output(1,5,"BINAIRE KLOK")
Output(3,1,"H")
Output(3,5,L2)
Output(5,1,"M")
Output(5,4,L3)
Output(7,1,"S")
Output(7,4,L4)
1→B
While B=1
If sum(GetTime=L1)<3
0→B
End
End

```

6° Ook de Voyage 200 van TI beschikt over een ingebouwde tijdsfunctie. Bovendien is de processorsnelheid hoog genoeg om een voorstelling met cirkeltjes te maken. Het is wel niet in kleur. Een opgevuld cirkeltje stelt een 1 voor, een leeg cirkeltje een 0.

Hieronder een paar schermafdrukken.



Hierbij het programma bklok.v2p

```
bklok()
Prgm
© regis ockerman
© naar een idee
© van jean verswijvelen
ClrIO
Output 1,80,"DIGITALE KLOK"
Output 20,1,"H"
Output 40,1,"M"
Output 60,1,"S"
For i,1,100
EndFor
Clrdraw
Zoomsqr
For i,1,5,1
Circle i-2,1,.1
Endfor
For i,1,6,1
Circle i-3,0,.1
Circle i-3,-1,.1
Endfor
Loop
getTime()->list1
list1[1]->u
5->n
seq(0,x,1,n)->list2
For i,1,n
If u>2^(n-i) Then
u-2^(n-i)->u
1->list2[i]
EndIf
EndFor
list1[2]->m
6->n
seq(0,x,1,n)->list3
For i,1,n
If m>2^(n-i) Then
m-2^(n-i)->m
1->list3[i]
EndIf
EndFor
list1[3]->s
6->n
seq(0,x,1,n)->list4
For i,1,n
If s>2^(n-i) Then
s-2^(n-i)->s
1->list4[i]
EndIf
EndFor
For i,1,5,1
Circle i-2,1,.05,list2[i]
```

```
Endfor
For i,1,6,1
Circle i-3,0,.05,list3[i]
Circle i-3,-1,.05,list4[i]
Endfor
1-a
While a=1
If (getTime()=list1)[3]=false
0-a
EndWhile
EndLoop
EndPrgm
```

7° Omdat het intikken van programma's een tijdrovende bezigheid is, kan je beter de bronfiles van iemand overnemen. Voor de Basic kan je terecht bij Jean. Voor de TI-files kan je contact nemen met mij.

Namens de redactieraad
Regis Ockerman <regisockerman@gmail.com>

U komt toch ook naar ons congres op zaterdag 30 juni en zondag 1 juli 2012



VVWL congres 2010 : Congressisten in de ban van de spreker.

Foto : A. Mus