

Programmeren onder RISC OS: de snelstart

in BBC-Basic

Programmeren in RISC OS: de snelstart
De programma-directory
Vensters maken
Een eenvoudige foutprocedure
Muis en toetsenbord
Een icoon op de icoonbalk plaatsen
Een menu aanbrenge
Sprites in vensters en iconen
De listing tot zover
Afwerking

Peter Scheele



Programmeren in RISC OS: de snelstart

Versie: nov. 2015

In een paar reuzenstappen maken we een werkend programma. Deze werkwijze veronderstelt dat u weet hoe RISC OS werkt. Ook verschillende programma's onder RISC OS als Edit (of een alternatieve editor), Paint en TemplEd (of een alternatieve template-editor). En dat u bekend bent met BBC-Basic en dat u een idee heeft van SoftWare Interrupts (SWI's).

Bij de verschillende stappen staan verwijzingen naar de cursus in de vorm deel.hoofdstuk.paragraaf. Daar vindt u verdere uitleg, details en achtergronden.

De programma-directory 1.3.1..6

Maak in een directoryvenster een directory door muis:menu. Kies in het menu voor New directory. Geef een naam die begint met !, bijvoorbeeld !MijnProg.

Open die directory met shift-klik.



Zet in de directory de volgende bestanden:

!Boot Start Edit, muis:menu, kies 'Create', ga naar recht en kies Obey. Voer de volgende tekst in:

```
Set MijnProg$Dir <Obey$Dir>
IconSprites <MijnProg$Dir>.!Sprites22
```

Bewaar het bestand met de naam !Boot.

De !Boot wordt tijdens de ontwikkeling van het programma verder uitgebreid.



!Run Start Edit, muis:menu, kies 'Create', ga naar rechts en kies 'Obey'. Voer de volgende tekst in:

```
Set MijnProg$Dir <Obey$Dir>
WimpSlot -min 8K -max 8K
IconSprites <Obey$Dir>.!Sprites22
```

```
Run <Obey$Dir>.!RunImage
```

Bewaar het bestand met de naam !Run.

De !Run wordt tijdens de ontwikkeling van het programma verder uitgebreid.



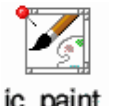
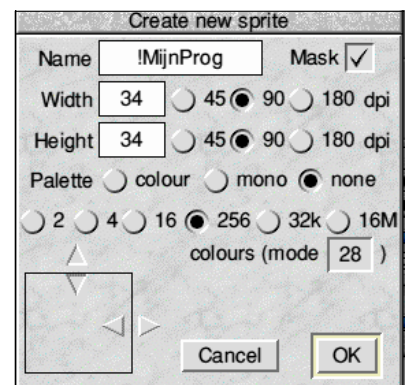
!Sprites22 Start Paint, muis:menu, kies 'Create new sprite', dit venster (of vergelijkbaar) verschijnt. Geef als naam de naam van het programma (!MijnProg), vul het verder in zoals in dit voorbeeld. Klik op OK. Er verschijnen twee vensters, bij een staat de programmaam in de titelbalk (!mijnprog). Teken een (voorlopig) plaatje.

Bewaar het bestand onder de naam !Sprites22.

Maak een verkleind icoon. Open !Sprites22, in het venster muis:menu en kies New sprite. Het venster hiernaast wordt geopend. De naam wordt sm!mijnprog. Vul bij de afmetingen twee keer 17 in en maak de afbeelding.

Bewaar het bestand onder de naam !Sprites22.

Maak een gepind icoon. Open !Sprites22, in het venster muis:menu en kies New sprite. Het venster hierboven wordt geopend. De naam wordt ic_mijnprog. De afmetingen zijn 34 x 34. Maak de afbeelding. Bewaar het bestand onder de naam !Sprites22.



!Help Start Edit, muis:menu, kies 'Create', ga naar rechts en kies Text.

Voer een passende helptekst in.

Bewaar het bestand met de naam !Help.

De !Help wordt tijdens de ontwikkeling van het programma verder uitgebreid.



!RunImage Start Edit, muis:menu, kies 'Create', ga naar rechts en kies 'BASIC'.

Voer de volgende tekst in of kopieer en plak die:



```
REM > !MijnProg
REM Naam Auteur
REM version 1.1

PROCinit
PROChoofd
PROCsluit
END

DEF PROCinit
  DIM blok% 255
  klaar%=FALSE:versie%=4.02*100
  app$="MijnProg":filter%=&1C33
  SYS "Wimp_Initialise",versie%,&4B534154,app$ TO versie%,taak%
ENDPROC

DEF PROChoofd
  WHILE klaar%=FALSE
    SYS "Wimp_Poll",filter%,blok% TO reason%
    CASE reason% OF
      WHEN 17,18:IF blok%!16=0 THEN klaar%=TRUE
    ENDCASE
  ENDWHILE
ENDPROC

DEF PROCsluit
  SYS "Wimp_CloseDown",taak%,&4B534154
ENDPROC
```

Bewaar het onder de naam !RunImage.

De !RunImage wordt tijdens de ontwikkeling van het programma verder uitgebreid.

U hebt nu een werkend programma.

Vensters maken 1.5.1..4

Het Info-venster (wordt getoond via muis:menu)

Start een Template-editor, maak een venster als dit en noem het 'Info'.

Bewaar de template onder de naam 'Templates' in de directory van !MijnProg.

Het hoofdvenster 'Main'.

Start een Template-editor, maak een venster als dit en noem het 'Main'.

Bewaar de template onder de naam 'Templates' in de directory van !MijnProg.

About this program	
Name	Writer
Purpose	Word Processor
Author	© Icon Technology Ltd, 1999
Version	1.01 (02-Jul-99)

Bepaal per venster de werkelijke grootte, de zichtbare grootte, de vensterattributen (titelbalk, knoppen, schuifbalken), de invoervelden (moeten via de tab-toets op elkaar volgen), de plaats van de iconen etc.

Hou bij voorkeur rekening met de adviezen in de Style guide.

Ook eventuele andere vensters moeten aangemaakt worden, ze worden alle bewaard in 'Templates'.

Vergeet niet van de vensters de statistics op te vragen want die informatie hebben we dadelijk nodig.

Voeg onderstaande regel toe in PROCinit. Op de plaats van xxx en yyy komen de waarden uit de statistics.

```
DIM wmain% xxx, winfo% yyy, naam% 11
```

Het kan zijn dat de grootte van blok% moet worden aangepast aan de grootste van xxx of yyy. Hetzelfde geldt voor de WimpSlot in !Run.

Roep beide templates aan in PROCbegin, onder de DIM en na SYS "Wimp_Initialise", met de volgende opdrachten (zet op de plaatsen xxx en yyy de juiste waarden):

```
SYS "Wimp_OpenTemplate", , "<MijnProg$Dir>.Templates"  
$naam%="main"  
SYS "Wimp_LoadTemplate", , blok%, wmain%, wmain%+xxx, -1, naam%, 0  
SYS "Wimp_CreateWindow", , blok% TO hmain%  
$naam%="info"  
SYS "Wimp_LoadTemplate", , blok%, winfo%, winfo%+yyy, -1, naam%, 0  
SYS "Wimp_CreateWindow", , blok% TO hinfo%  
SYS "Wimp_CloseTemplate"  
!blok%=hmain%:SYS "Wimp_GetWindowState", , blok%  
SYS "Wimp_OpenWindow", , blok%  
!blok%=hinfo%:SYS "Wimp_GetWindowState", , blok%  
SYS "Wimp_OpenWindow", , blok%
```

De regels met \$naam%="naam", Wimp_LoadTemplate en Wimp_CreateWindow moeten voor ieder venster apart vermeld worden.

Zet in PROChoofd net boven WHEN 17,18:

```
WHEN 2:SYS "Wimp_OpenWindow", , blok%  
WHEN 3:SYS "Wimp_CloseWindow", , blok%:klaar%=TRUE
```

Het is op tal van plaatsen mogelijk venster te openen of te sluiten. Ze komen verderop aan de orde.

Testen: Start !MijnProg, kijk of het venster zichtbaar wordt, sluit het programma met het sluiticoon en controleer bij de TaskManager of het verdwenen is.

Een eenvoudige foutprocedure 1.5.5 en 1.8.

Zet op een van de eerste regels de volgende code:

```
ON ERROR PROCerror(REPORT$+" (in regel "+STR$ ERL+" )",3):END
```

En onderaan de listing:

```
DEF PROCerror(M$,vlag%)  
!blok%=ERR  
$(blok%+4)=M$+CHR$0  
SYS "Wimp_ReportError", blok%,vlag%,app$ TO ,M%  
ENDPROC
```

Als er een probleem is met het programma, verschijnt er een venster met de foutmelding en de desbetreffende programmaregel. Er is een knop om het programma af te sluiten.

Als het programma onverklaarbaar vastloopt, of zelfs de hele machine, zonder foutmelding, controleer dan de grootte van de WimpSlot en van blok% in PROCinit. Maak die groter en test.

Muis en toetsenbord

1.6.1 en verder

De muis

Zet in PROHoofd op de juiste plaats:

```
WHEN 6:PROCmuis(blok%!0,blok%!4,blok%!8,blok%!12,blok%!16)
```

Nu wordt door RISC OS bijgehouden en doorgegeven welke acties er met de muis gedaan worden.

En maak een nieuwe procedure:

```
DEF PROCmuis(x%,y%,knop%,venster%,icoon%)  
  REM Wat te doen met de data die nu in x%, y%, knop%, venster% en icoon% zit?  
ENDPROC
```

In x% en y% staan de coördinaten van de muispil in het venster, knop% bevat het nummer van de muisknop waarmee geklikt is, venster% bevat het nummer van het venster en icoon% het nummer van het icoon waarop geklikt is.

Op de plaats van de REM komen de volgende regels (voorbeeld, in eigen programma gedetailleerd uitwerken):

```
IF (venster%=hmain%) AND (knop%=4) AND (icoon%=11) THEN klaar%=TRUE:REM icoon 11  
is een knopje met Quit
```

Het kan ook zo:

```
CASE (venster%=hmain%) AND icoon% OF  
  WHEN 4:PROCprint:REM icoon 4 is een knopje met Print  
  WHEN 11:klaar%=TRUE:REM icoon 11 is een knopje met Quit  
ENDCASE
```

Voor ieder venster, iedere muisknop en ieder icoon moet de actie beschreven worden. Dat kan bij grote programma's redelijk complex worden.

Het toetsenbord

Voeg in PROHoofd op de juiste plaats de regel toe:

```
WHEN 8:PROctoets(venster%,icoon%,toets%)
```

Nu wordt door RISC OS bijgehouden en doorgegeven welke acties er met het toetsenbord gedaan worden.

En maak een nieuwe procedure:

```
DEF PROctoets(venster%,icoon%,toets%)  
IF (icoon%=5) AND (toets%=13) THEN PROCinvoer(venster%,icoon%) ELSE SYS  
"Wimp_ProcessKey",toets%  
ENDPROC
```

De Wimp_ProcessKey is om algemene toetsaanslagen als functietoetsen door te geven aan RISC OS.

De invoervelden kunnen geformatteerd worden, daarmee formatteert men de invoer. Dat moet gebeuren bij het ontwerpen in de template-editor bij Validation. Denk aan A0-9 voor alleen cijfers, Aa-z voor alleen letters of A~X alle tekens behalve X. Met Pspritename kan een eigen pointer worden gebruikt (die moet wel als sprite getekend worden en staan in !Sprites22).

Een icoon op de icoonbalk plaatsen 1.7.1

Zet deze twee regels in PROCinit, net boven ENDPROC.

```
blok%!0=-1:blok%!4=0:blok%!8=0:blok%!12=68: blok%!16=68:blok%!20=&3002:$(blok%+24)="!MijnProg"  
SYS "Wimp_CreateIcon",,blok% TO ihandle%
```

Verder moet dat icoon klikbaar zijn met muis:select om een venster te openen en muis:menu om een menu te tonen. Dat gebeurt in PROCmuis met de volgende twee regels:

```
IF (venster%=-2) AND (knop%=4) !blok%=hmain%:SYS "Wimp_GetWindowState",,blok%:SYS  
"Wimp_OpenWindow",,blok%
```

En:

```
IF (venster%=-2) AND (knop%=2) PROCmenu(x%,y%)
```

PROCmenu moet nog wel beschreven worden, dat gebeurt in de volgende paragraaf. De volgende regel kan nu uit PROCbegin verwijderd worden:

```
!blok%=hmain%:SYS "Wimp_GetWindowState",,blok%  
SYS "Wimp_OpenWindow",,blok%
```

Een menu aanbrengen 1.7.2

Een simpel voorbeeld: het menu vanaf de icoonbalk. Voor uitgebreidere menu's, menu's vanuit een venster en geneste menu's, kijk in de cursus.



Er moet een menublok gebruikt worden van 28+(2x24) bytes. Zet daarvoor bovenin PROCinit:

```
DIM menublok% 76
```

Het menu moet ook geïnitieerd worden in PROCinit, net boven ENDPROC, met de volgende regels:

```
menublok%!28=0:menublok%!32=hinfo%:menublok%!36=&7000021  
$(menublok%+40)="Info"  
menublok%!52=&80:menublok%!56=-1:menublok%!60=&7000021  
$(menublok%+64)="Einde"
```

Dan moet PROCmenu(x%,y%) gevuld worden:

```
DEF PROCmenu(x%,y%)  
$menublok%=app$:menublok?!12=7:menublok?!13=2  
menublok?!14=7:menublok?!15=0  
menublok%!16=96:menublok%!20=44:menublok%!24=0  
menublok%!28=0:menublok%!32=hinfo%  
menublok%!36=&7000021  
$(menublok%+40)="Info"  
menublok%!52=&80:menublok%!56=-1:menublok%!60=&7000021  
$(menublok%+64)="Einde"  
SYS "Wimp_CreateMenu",,menublok%,x%,y%  
ENDPROC
```

Om op de menukeuze te kunnen reageren, moet in PROChoofd event 9 aangezet worden:

WHEN 9:PROCmenukeuze
En de procedure moet geplaatst worden:

```
DEF PROCmenukeuze
  keuze%=!blok%
  IF keuze%=0 !blok%=hinfo%
  SYS "Wimp_GetWindowState",,blok%:SYS "Wimp_OpenWindow",,blok%
  IF keuze%=1 klaar%=TRUE
ENDPROC
```

Sprites in vensters en iconen

Maak een of meer sprites en bewaar die in het bestand Sprites.

Open het venster in de template-editor. Sleep eerst het bestand Sprites naar het icoon van de template-editor op de icoonbalk; nu weet de template-editor dat er sprites in komen.

Sleep dan een icoon in de template, die met de afbeelding voldoet al, of kies een button. Klik op het desbetreffende icoon met muis:menu en via het menu Icon # naar Edit... In het venster staat Validation. Maak die leeg. Vink ook Text uit en zet bij Sprite de naam van de sprite. Bewaar het geheel.

In PROCbegin komt, voor het laden van de template: *IconSprites<MijnProg\$Dir>.Sprites

De listing tot zover

```
REM > MijnProg
REM Peter Scheele
REM Version 0.1

ON ERROR PROCerror(REPORT$+" (in regel "+STR$ ERL+" )",3):END

PROCbegin
PROChoofd
PROCeinde
END

DEF PROCbegin
  DIM blok% 2000
  DIM wmain% 1037, winfo% 421, naam% 11, menublok% 76
  klaar%=FALSE:versie%=5.21*100
  app$="MijnProg":filter%=&0
  *IconSprites<MijnProg$Dir>.Sprites

  SYS "Wimp_Initialise",versie%,&4B534154,app$ TO versie%,taak%
  SYS "Wimp_OpenTemplate",,"<MijnProg$Dir>.Templates"
  $naam%="main"
  SYS "Wimp_LoadTemplate",,blok%,wmain%,wmain%+1037,-1,naam%,0
  SYS "Wimp_CreateWindow",,blok% TO hmain%
  $naam%="info"
  SYS "Wimp_LoadTemplate",,blok%,winfo%,winfo%+421,-1,naam%,0
  SYS "Wimp_CreateWindow",,blok% TO hinfo%
  SYS "Wimp_CloseTemplate"

  blok%!0=-1:blok%!4=0:blok%!8=0:blok%!12=68
  blok%!16=68:blok%!20=&3002:$ (blok%+24)="!MijnProg"
  SYS "Wimp_CreateIcon",,blok% TO ihandle%
```

```

menublok%!28=0:menublok%!32=hinfo%:menublok%!36=&7000021
$(menublok%+40)="Info"
menublok%!52=&80:menublok%!56=-1:menublok%!60=&7000021
$(menublok%+64)="Einde"
ENDPROC

```

```

DEF PROChoofd
  WHILE klaar%=FALSE
    SYS "Wimp_Poll",filter%,blok% TO reason%
    CASE reason% OF
      WHEN 2:SYS "Wimp_OpenWindow",,blok%
      WHEN 3:SYS "Wimp_CloseWindow",,blok%:klaar%=TRUE
      WHEN 6:PROCmuis(blok%!0,blok%!4,blok%!8,blok%!12,blok%!16)
      WHEN 9:PROCmenukeuze
      WHEN 17,18:IF blok%!16=0 THEN klaar%=TRUE
    ENDCASE
  ENDWHILE
ENDPROC

```

```

DEF PROCeinde
  SYS "Wimp_CloseDown",taak%,&4B534154
ENDPROC

```

```

DEF PROCerror(M$,vlag%)
  !blok%=ERR
  $(blok%+4)=M$+CHR$0
  SYS "Wimp_ReportError",blok%,vlag%,app$ TO ,M%
ENDPROC

```

```

DEF PROCmuis(x%,y%,knop%,venster%,icoon%)
  IF (venster%=hmain%) AND (knop%=4) AND (icoon%=5) THEN klaar%=TRUE
  IF (venster%=-2) AND (knop%=4) !blok%=hmain%:SYS "Wimp_GetWindowState",,blok%:SYS "Wimp_OpenWindow",,blok%
  IF (venster%=-2) AND (knop%=2) PROCmenu(x%,y%)
ENDPROC

```

```

DEF PROCmenu(x%,y%)
  $menublok%=app$:menublok?!12=7:menublok?!13=2
  menublok?!14=7:menublok?!15=0
  menublok?!16=96:menublok?!20=44:menublok?!24=0
  menublok?!28=0:menublok?!32=hinfo%
  menublok?!36=&7000021
  $(menublok%+40)="Info"
  menublok%!52=&80:menublok%!56=-1:menublok%!60=&7000021
  $(menublok%+64)="Einde"
  SYS "Wimp_CreateMenu",,menublok%,x%-100,y%+100
ENDPROC

```

```

DEF PROCmenukeuze
  keuze%=!blok%
  IF keuze%=0 !blok%=hinfo%:SYS "Wimp_GetWindowState",,blok%:SYS "Wimp_OpenWindow",,blok%
  IF keuze%=1 klaar%=TRUE
ENDPROC

```


Afwerking

Maak in !Run de WimpSlot zo klein mogelijk. Het gaat in stappen van 4. Een schatting is mogelijk: bepaal de grootte van Templates, Sprites, RunImage en de arrays (DIM) en rond het geheel af naar boven, het getal moet deelbaar zijn door 4. Zie deel 1, hoofdstuk 3.3.

In PROchoofd staat een filter. Dat moet precies ingesteld worden, zodat het programma alleen aandacht vraagt voor dat wat het nodig heeft. Daarmee belast MijnProg het geheugen en de processortijd niet meer dan nodig is. Het filter (filter%) staat nu op 0. Voor uitleg zie deel 1, hoofdstuk 4.5.

Men doet er goed aan voor een overzichtelijk listing te zorgen. Inspringen, documenteren met REM's, lege regels tussen procedures en goed gekozen namen.

Deze beschrijving en het programma zijn lang niet compleet, maar ze bieden een start. Aan de volgende zaken moet nog aandacht worden besteed:

- (Help)berichten
- Uitgebreide sprites en afbeeldingen
- Drag and drop
- Uitgebreide menu's, ook aan te roepen vanuit een venster
- Andere fonts
- De afhandeling van radiobuttons
- Bestanden in de !Boot

Een deel van bovenstaande wordt behandeld in deel 2 van de cursus.

Peter Scheele
november 2015
p.scheele@xs4all.nl