

# GRAPHPAK: GRAFISCHE, MAPPER-, EN GELUIDSROUTINES IN TURBO PASCAL

**Handige uitbreidingen voor Turbo Pascal**  
**MSX Computer Magazine**  
**nummer 37 - maart 1990**

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

De meeste MSX-gebruikers zullen hun eerste programma's op hun computer geschreven hebben in Basic. Wie dan later de overstap maakt naar Turbo Pascal, zal het geluid en de fraaie grafische mogelijkheden van MSX moeten missen. We hebben al aandacht besteed aan het gebruik van de MSX-BIOS onder Turbo Pascal, maar dat vereist nogal diepgaande kennis van machinetaal. Toch is er ook hoop voor de pure Pascal-programmeurs: GraphPak is een verzameling nuttige MSX-procedures en -functies voor gebruik onder Turbo Pascal. Is dat een oplossing?

## **Turbo Pascal: geen MSX**

Wie ooit in Turbo Pascal op de MSX bezig is geweest, zal weten dat het bijna onbegonnen werk is om bijvoorbeeld meerkanaals geluid te maken. Voor de vergevorderde programmeurs zijn er altijd wel ingewikkelde oplossingen om de mooie zaken uit de MSX-computer naar boven te halen, maar het kost tijd om de routines die hiervoor nodig zijn te schrijven en te testen.

Voor het aanroepen van een aantal routines die een MSX-computer ingebouwd heeft — denk aan een BEEP of het COLOR-commando — is er een grote verscheidenheid aan standaardbibliotheken voorhanden. Dit zijn programmadelen die voor het eigenlijke programma komen te staan — als zogenaamde 'include files' — en die door de programmeur aangeroept kunnen worden. De procedures en functies voegen als het ware extra mogelijkheden aan Turbo Pascal toe. De meeste van deze modules zijn geschreven door hobbyisten. Ze werken over het algemeen prima, maar bij de mogelijkheden die MSX echt sterk maken, zoals bijvoorbeeld het gebruik maken van een memory-mapper, of het PAINTen van een cirkel, moeten helaas veel programmeurs afhaken. Nu was het begin 1989 al bekend, dat er in Duitsland iemand bezig was met het schrijven van een veelbelovende trুকjes- en oefjesbibliotheek voor de MSX, maar meer nieuws bleef helaas uit. Tot voor kort althans...

## **Uhh? Zozo!**

Dat was ongeveer onze eerste reactie toen we GraphPak zagen. We waren verbaasd over de grote verscheidenheid aan routines en de gemakkelijke toepasbaarheid ervan. Daarna zagen we de

mogelijkheden die opeens open lagen voor alle denkbare toepassingen in Turbo Pascal. Wie heeft er niet ooit een muis of joystick willen uitlezen in Pascal, of een grote kaartenbak willen opslaan in het geheugen? Nu is het een kwestie van de juiste module erbij pakken en klaar is de routine.

GraphPak bestaat uit een groot aantal bestanden op één enkelzijdige diskette, opgedeeld in drie groepen: de .GIC\$-, .MIC\$- en .SIC\$-bestanden. Deze extensies zijn afkortingen van achtereenvolgens Graphics-, Mapper- en SystemInClude.

In totaal zijn er dertig modules die samen 167 functies en procedures bevatten. Op het eerste lijkt het wat onzinnig om hiervoor zoveel kleine bestanden aan te maken; waarom niet één grote include-file? Maar een groot voordeel van het opsplitsen is, dat alleen de routines die werkelijk van belang zijn worden mee-gecompileerd, wat niet alleen de grootte van het uiteindelijke programma reduceert, maar ook de compileertijd bekort. Alhoewel het niet helemaal duidelijk naar voren komt in de handleiding, zijn er toch een paar routines die voor iedere groep altijd aangeroepen moeten worden. Dit zijn CrtlNit, CrtExit en MapInit. Deze routines initialiseren geheugen en variabelen voor de rest van de functies. De test-vlag MSX2 bijvoorbeeld, wordt in CrtlNit gezet als het programma op een MSX2 — of een 2 +loopt. CrtlNit en CrtExit zijn al bestaande functies in Turbo Pascal, maar ze worden hier opnieuw gedefinieerd omdat veel schermaanstellingen nét even anders verlopen.

| <b>Naam:</b>   | <b>Doel:</b>  |
|----------------|---|
| IsMouse        | kijkt of er een muis aangesloten is                     |
| GetMouse       | haalt de muispositie op                                 |
| HighVideo      | 4-kleuren weergave op scherm O, 80 karakter             |
| GetDate        | haal datumstring op                                     |
| HiSpeedCopyPic | high speed VRAM naar VRAM copy                          |
| LoadOverlay    | laadt (een deel van) een overlay in memory Mapper       |
| MapAvail       | geeft aantal bytes dat in memory mapper aanwezig is     |
| Play           | laat 3 kanaals geluid horen                             |
| QuickCircle    | tekent een cirkel                                       |
| IntervalOn     | laat Turbo Pascal na een bepaald Tijdsinterval reageren |
| VDPstatus      | leest één van de VDP statusregisters                    |

Enkele procedures en functies in GraphPak

## Handleiding

Omtrent de handleiding durven we niets definitiefs te melden, er schijnen zowel Engelstalige als Duitstalige versies in omloop te zijn. Ons testexemplaar werd geleverd met een Duitstalige handleiding, met een omvang van bijna 200 pagina's A5. De drukwaliteit is matig; de lettertjes zijn net een maatje te klein en te vet. Hierdoor is de inkt af en toe iets doorgelopen, wat het geheel een beetje vlekkerig doet overkomen. Jammer, temeer daar de handleiding qua inhoud goed in elkaar steekt. De index is duidelijk en uitgebreid, de koppen bovenaan de bladzijde zijn makkelijk na te lopen en het geheel zit in een handige ringmap; makkelijk als naslagwerk. Alle mogelijkheden zijn alfabetisch gerangschikt, met gemiddeld iets meer dan een pagina per commando. Iedere beschrijving bevat de naam, de vorm waarin de functie gedeclareerd is, welke modules er eerder ingeladen moeten zijn voordat deze functie gebruikt kan worden, de parameters en het resultaat van het geheel. Ook zit er overal een klein voorbeeld bij en een handige verwijzing naar aanverwante

mogelijkheden. Kijken we bijvoorbeeld bij High Video, zien we verwijzingen naar: BlinkColor, BlinkOff, Norm Video, SetAttribute en GetAttribute. Voldoende informatie om een onderwerp duidelijk te maken.

De handleiding komt verder wat magertjes over: hij is gericht op de programmeur die al redelijk tot goed met Turbo Pascal overweg kan en die alleen wil leren hoe hij snel routines bij elkaar kan vergaren. Op de diskette staat ook nog de brontekst van een demonstratieprogramma dat alle mogelijkheden van het pakket laat zien, wat ook weer de nodige informatie biedt.

Zoals reeds gesteld, de door ons bekeken handleiding is Duitstalig. Nu zal dat voor de technische termen geen probleem zijn — die spreken voor zichzelf — maar in de bronteksten zelf is het nog wel eens verwarrend om tussen de Engelse Pascal-termen opeens Duits commentaar te zien staan. De modules zelf zijn redelijk voorzien van commentaar; de moeilijke stukken wat vaker dan de makkelijke — zoals het hoort, dus. Omtrent de Engelstalige handleiding durven we niets te zeggen, we hebben deze niet onder ogen gehad. Weliswaar is er als alles goed is een Engelstalig pakket naar ons onderweg, maar gezien de sluitingsdatum van dit nummer hebben we daar niet op kunnen wachten. In het volgende nummer van MCM zullen we daar aandacht aan schenken.

## **Events**

Een sterk punt van MSX-Basic ten opzichte van veel andere Basic-versies is de mogelijkheid om bepaalde onverwachte zaken af te kunnen vangen: 'event trapping', op z'n Engels gezegd. Voorbeelden hiervan zijn een timer-interrupt, het afvangen van de Control-Stop toetscombinatie of het reageren op de vuurknop van een joystick. Door heel erg slim gebruik te maken van de interne structuur van Turbo Pascal, is de auteur van GraphPak erin geslaagd om deze functies ook op te nemen in het pakket.

Applaus voor de oplossing van dit netelige probleem! Voortaan staan hiervoor de volgende procedures aan de Pascal programmeur ter beschikking: OnInterval, OnKey, OnSprite, OnStop en OnStrig. Dit komt overeen met de:ON event GOSUB commando's in Basic, waarbij event dan Interval, Key, Sprite, Stop of Strig kan zijn. Natuurlijk zijn ook de opdrachten aanwezig om bepaalde gebeurtenissen tijdelijk of definitief te stoppen.

Bij het gebruiken van een event moet een procedure opgegeven worden die opgeroepen zal worden als het event optreedt. Het is alleen niet mogelijk om parameters mee te geven aan die subroutine. Lastig, maar door globale variabelen te veranderen is het toch mogelijk het hoofdprogramma te laten weten dat er iets belangrijks gebeurd is.

## **MSX-bestanden**

Een nadeel bij het verwerken van grote hoeveelheden gegevens is, dat Turbo Pascal er boven een bepaalde limiet niet meer mee kan omgaan. Ook dat probleem is ondervangen; er is een reeks routines bijgekomen om files op de MSX-manier te openen, te lezen, te schrijven en te sluiten. Hierbij blijft de record-structuur van Pascal volledig intact, dus het maken van een Random Access-File is nog steeds goed mogelijk. Het enige verschil met standaard Turbo Pascal is eigenlijk dat een record nummer niet langer een integer is, maar ook een getal van het type 'real' mag zijn. Hierdoor zijn datafiles tot 4 Gigabyte mogelijk! Tezamen met de uitgebreide aansturing van de memory-mapper vormen Turbo Pascal en GraphPak samen dus een handige combinatie voor het ontwikkelen van databases voor MSX.

Helaas is een aantal commando's waarin Basic wél voorziet, niet opgenomen in de instructieset van

GraphPak. Elke verwijzing naar het klok-RAM ontbreekt — de SET-commando's onder Basic — maar ook de RND-functie, DSKI, DSKO en DSKF missen. Weliswaar zijn dit commando's die voor een goede programmeur niet al te lastig zelf toegevoegd zouden kunnen worden, maar dat is nu net niet de bedoeling!

## In de praktijk

Toen we GraphPak werkelijk gingen gebruiken, kwamen we al snel een probleem tegen: de demonstratie-programma's op de diskette konden we niet zo compileren dat ze het ook werkelijk deden. De al gecompileerde versies deden het zonder problemen, maar onze eigen probeersels liepen vast. Geen nood, de kennissenkring van onze geachte hoofdredacteur is groot, dus was contact met de schrijver van GraphPak, Uwe Schröder, zo gelegd. Na telefonisch overleg werd duidelijk dat Turbo Pascal in Nederland altijd al verkocht is met een bug erin. Die bug houdt in dat de 'User Break' van Pascal niet werkt onder MSX-DOS. Dit was bekend en staat zelfs in een apart hoofdstuk in de Turbo Pascalhandleiding, maar lastig is het wel. Temeer daar alle interrupt-routines — de events — gebaseerd zijn op die userbreak. Gelukkig kon de oplossing ons telefonisch worden meegedeeld, evenals een mogelijkheid om voortaan grafische karakters in de Pascal editor te kunnen gebruiken! Om TURBO.COM aan te passen kan het programma GP.PAS uit listing 1 gebruikt worden. Dit geeft ook meteen een klein overzicht van de mogelijkheden van GraphPak. En inderdaad, na deze kleine aanpassing werkt de User Break zonder problemen, op elke plaats in een programma. Ook GraphPak geeft geen problemen meer. Tijdens de test bleek, dat de modules van GraphPak aardig wat geheugen kosten. Met de mapper-routines — LoadOverlay bijvoorbeeld — zal dit wel op te vangen zijn. Voor de volledigheid hebben we tabel 2 samengesteld, met een overzicht van de omvang van de groepen modules. Genoemd zijn achtereenvolgens het aantal code-bytes, het aantal vrije bytes bij compileren in het geheugen en naar een .COM file. Deze aantallen gelden voor een Philips NMS 8245 met 2 drives. Duidelijk is te zien, dat de grafische groep het meeste geheugen in beslag neemt. Nu lijkt het weinig waarschijnlijk dat iemand alle modules van deze groep gebruikt, dus ook dit zal geen probleem vormen. En dan is het nog altijd mogelijk om naar een COM-file te compileren.

| Groep | Bytes in groep | Bytes vrij | Bytes vrij (.COM) |
|-------|----------------|------------|-------------------|
| .SIC  | 3572           | 17606      | 41507             |
| .MIC  | 3965           | 17494      | 41358             |
| .GIC  | 18821          | 0*         | 23986             |

Tabel 2 \*: compiler overflow

## Auteursrecht

Op de bronteksten van GraphPak rust uiteraard gewoon het auteursrecht van de maker. Hij geeft echter expliciet toestemming zijn modules in (mee-)gecompileerde vorm te verspreiden. Kopiëren van de originelen blijft dus gewoon verboden, maar zelfgeschreven programma's die delen van GraphPak bevatten mogen verspreid worden alsof het om volkomen eigen programmatuur gaat. Om het verspreiden van de bronteksten tegen te gaan, bevat elk exemplaar van GraphPak een serienummer, dat waarschijnlijk tevens ergens onleesbaar tussen de machinecode is geplaatst! De

diskette is verder dan ook niet beveiligd tegen kopiëren. Enerzijds is dit alles zeer toe te juichen. Zo kan men met het pakket ontwikkelde programma's verder vrij gebruiken en ook een veiligheidskopie maken voor eigen gebruik. Maar we vragen ons af hoe snel het geheel gepiratiseerd zal worden

### **Conclusie**

GraphPak is een zeer fraai pakket. De beginnende Turbo Pascal-programmeur zal er veel commando's in vinden die hij gewend is van MSX-Basic; de gevorderde programmeur zal waarschijnlijk een kleine vreugdesnik niet kunnen onderdrukken bij het zien van de routines voor de memory-mapper.

De handleiding is in het Duits, maar door de duidelijkheid ervan zal dat meestal geen probleem zijn. Bovendien is er ook een Engelstalige versie, die we echter niet onder ogen gehad hebben. GraphPak is zowel op MSX1 als MSX2 te gebruiken, met dien verstande dat de memory-mapper routines op de MSX1 natuurlijk niet mee zullen doen. Al met al, een aanrader voor iedereen die Turbo Pascal serieus wil gebruiken.

```

program GP_demo;
{ Patcht TURBO.COM om zodat de 'events' interval, stop,
strig, key en sprite zonder problemen gebruikt kunnen
worden. Tevens kan na het runnen van deze patch
de grafische set in de Turbo editor gebruikt worden.

v1.0.0 1990 by PtB }

{$i msxfile.sic}

type
    line = string[9];
    patchrec = record
        adr: integer ;
        chk, new: byte;
    end;

const
    patch: array[0..4] of patchrec = (
        (adr: $272; chk: $38;new: 8),
        (adr: $278; chk: $39;new: 9),
        (adr: $2491,chk: $E6;new: 0),
        (adr: $2492,chk: $7F;new: 0),
        (adr: $5295,chk: $FF;new: $CF));

var
    objectfile: MSXfile;
    i: integer;

procedure IOcheck; external $201B;

procedure error (tekst: line);

begin
    write('Fout tijdens ', tekst, ' van TURBO.COM');
    halt
end;

procedure checkbyte (offset: real; check: byte);

var
    records: integer;
    result: byte;

begin
    MSXseek(objectfile, offset);
    MSXblockread(objectfile, result, SizeOf(result), records)
    if (records = 0) OR (result o check) then error('lezen')
end;

procedure schrijfbyte (offset; real; value: byte);

```

```
var
  records: integer;

begin
  MSXseek(objectfile, offset);
  MSXblockwrite(objectfile, value, SizeOf(value));
  if lOresult <> 0 then error('schrijven');
end;

begin { hoofdprogramma }
  MSXassign(objectfile, 'TURBO.COM') ;
  MSXreset(objectfile, 1);
  IOcheck;
  for i := 0 to 4 do begin
    checkbyte(patch[i].adr, patch[i].chk) ;
    schrijfbyte(patch[i].adr, patch[i].new)
  end;

end.
```