

# PROGRAMMEREN IN BASIC VIDEO COMPUTER SYSTEM<sup>®</sup>

VCS 800 - Modell CX 2600



SPECIAL EDITION



CX 2620

 A Warner Communications Company

## GEBRUIKSAANWIJZING

Dit is een vertaling van de eveneens  
bijgevoegde Amerikaanse gebruiksaanwijzing.  
Wij adviseren u i.v.m. de duidelijkheid de  
boekjes altijd in combinatie te gebruiken.

# PROGRAMMEREN IN BASIC

## Inleiding

**Programmeren in Basic** is een 'spel', waarmee u enig inzicht krijgt in het programmeren van computers. BASIC is de afkorting van Beginners All-purpose Symbolic Instruction Code (algemene programmeertaal voor beginners), en werd opgesteld om het leren schrijven van computerprogramma's te vereenvoudigen.

Computerprogramma's zijn eenvoudig een reeks instructies. De programma's sturen de stroom van gegevens in de computer. **Programmeren in Basic** maakt het u mogelijk, uw spelcomputer zodanig te instrueren, dat hij bepaalde, eenvoudige taken voor u kan uitvoeren.

Let erop, dat **Programmeren in Basic** slechts een klein geheugen heeft, vergeleken met 'echte' computers. Niettemin is het een uitstekende stap op weg naar het begrijpen van de computer.

**Let op:** om de levensduur van uw spelcomputer te verlengen en de elektronische onderdelen te beschermen moet de computer zijn **UITGESCHAKELD** wanneer een spelcassette wordt ingelegd of verwijderd.

Bij het gebruik van deze spelcassette kan het beeld van uw televisietoestel enigszins onstabiel worden. In dit geval dient het beeld met **VERTICAL HOLD** te worden bijgesteld.

## Het toetsenbord

Bij deze programmacassette gebruikt u het toetsenbord. Steek de stekkers goed in de aansluitingen achterop uw spelcomputer. Zie hiervoor ook de gebruiksaanwijzing.

Om de twee helften van het toetsenbord samen te voegen schuift u de lip van de linker helft in de gleuf van de rechter. De

twee helften vormen samen een 24-toetsenbord, waarmee u uw programma kunt ingeven en met de computer kunt communiceren.

Neem de sjablonen uit de enveloppe. Plaats het sjabloon met **LEFT** op de linker helft, en die met **RIGHT** op de rechterhelft van het toetsenbord.

## Het beeldscherm

Het beeldscherm kent zes niveau's:

1. Het **PROGRAMMANIVEAU** wordt gebruikt om het programma in te geven.
2. Het **REGISTERNIVEAU** geeft de tussenresultaten wanneer de computer uw programma uitvoert.
3. Het **VARIABLENIVEAU** bevat de waarde van elke programmavariabele, terwijl het programma wordt uitgevoerd.
4. Het **UITVOERNIVEAU** geeft de uitvoer van uw programma, wanneer dit wordt uitgevoerd.
5. Het **STATUSNIVEAU** geeft de beschikbare geheugenruimte aan, die op dat moment nog vrij is. Verder ziet u hier de snelheid, waarmee het programma wordt uitgevoerd.
6. Het **GRAFISCH NIVEAU** heeft twee gekleurde hokjes, die door het programma kunnen worden bestuurd.

Voordat u uw programma begint zet u de **linker moeilijkheidsschakelaar** in stand B. Nu ziet u de verscheidene niveau's op het scherm. Wanneer de schakelaar in stand B staat verdwijnen alle niveau's, behalve het grafische, dat nu op het scherm komt.

De **rechter moeilijkheidsschakelaar** is bij deze cassette buiten werking.

## De cursor

Zet de linker moeilijkheidsschakelaar in stand B en zet de computer uit en weer aan. In het programmaniveau ziet u een witte rechthoek; dit is de cursor. Midden op de onderste rij van de linkerhelft van het toetsenbord ziet de SHIFT-toets. Druk hier vier keer op. De kleur van de cursor wordt rood, dan blauw, vervolgens groen en tenslotte weer wit.

De cursor dient voor het ingeven van uw programma. Elk van de vier kleuren van de cursor correspondeert met de kleuren bij de toetsen. Wit wordt gebruikt voor het geven van programmainstructies, de andere kleuren voor de diverse tekens.

### Verplaatsen van de cursor

Zorg ervoor dat de linker moeilijkheidsschakelaar in stand B staat en zet de computer uit en weer aan. Druk op de toets FORWARD. De cursor springt van PROGRAMMA- naar REGISTERNIVEAU. Druk nogmaals op FORWARD, en de cursor gaat naar het VARIABELNIVEAU. Door nogmaals op FORWARD te drukken komt u in het UITVOERNIVEAU terecht. Met de toets BACKWARD springt u een niveau terug. De toetsen FORWARD en BACKWARD kunnen éénmaal of continu worden ingedrukt. Zet de linker moeilijkheidsschakelaar nu in stand A. De naam van elk niveau verdwijnt, en een gedeelte van het GRAFISCH NIVEAU verschijnt. Spring nu opnieuw met de cursor door de niveau's; de cursor komt niet in het grafisch niveau terecht.

### Niveau's verwijderen van het scherm

Zet de linker moeilijkheidsschakelaar in stand B en zet de computer uit en weer aan. Links op het toetsenbord ziet u een reeks instructies (wit), overeenkomend met de diverse niveau's:

STATUS, PROGRAM, STACK (register), VARIABLES, OUTPUT (uitvoer) en GRAPHICS. We beginnen met het STATUSNIVEAU. Druk éénmaal op de toets STATUS, en het statusniveau verdwijnt van het scherm; het programmaniveau schuift nu tot geheel bovenaan.

Druk op de toets PROGRAM. Het programmaniveau verdwijnt, en nu is het registerniveau (STACK) het hoogst. Druk op STACK. Het registerniveau verdwijnt, en het variabelenniveau schuift op tot bovenaan het scherm. Druk op VARIABLES en het variabelenniveau verdwijnt; nu staat het uitvoerniveau bovenaan.

Verwijder het uitvoerniveau door op OUTPUT te drukken. Nu wordt het gehele grafische niveau zichtbaar op het scherm. Dit niveau verwijdert u met de toets GRAPHICS. Nu is uw beeldscherm leeg.

Druk nu op de toets STACK. Het registerniveau komt nu weer op het beeldscherm. Verwijder het, en zoek de UITVOER en VARIABELEN niveau's op. U kunt elk niveau verwijderen of terugbrengen op het scherm, onafhankelijk van de stand van de moeilijkheidsschakelaar.

Het is belangrijk te onthouden, dat het voor de uitvoer van het programma geen verschil maakt, of een bepaald niveau al dan niet op het scherm staat.

## Uitvoeren van een programma

We schrijven een simpel programma. Zet de linker moeilijkheidsschakelaar in stand B en zet de computer uit en weer aan. (Een andere manier om alle programma's uit te wissen is door op GAME SELECT te drukken. Met de toets GAME RESET gaat u terug naar het begin van het programma,

waarbij alleen alle variabelen — niet het programma zelf — worden gewist.)

Verwijder de register-, variabelen-, uitvoeren grafische niveau's van het beeldscherm. De cursor is wit en staat rechts van het getal 1 in het programmaniveau. Maak de cursor blauw en druk op toets A. Naast de 1 verschijnt nu de letter A op het scherm. (Elke regel is genummerd, waardoor u kunt zien waar de ene regel ophoudt en de volgende begint.) Maak de cursor rood en druk op de toets —. Naast de A verschijnt een kleine pijl. Ga terug naar blauw en druk op B. Maak de cursor rood en druk op + en het getal 2. Nu gaat u terug naar wit en druk op NEW LINE; voor het overspringen naar een nieuwe regel moet de cursor altijd wit zijn.

Het beeldscherm ziet er nu zó uit:

Wanneer u een typefout maakt bij de ingave kunt u deze weghalen met de toets ERASE. Deze toets heeft GEEN kleurencode; hij kan bij elke kleur van de cursor worden gebruikt.

Met de eerder ingegeven regel gaan we nu de toets ERASE proberen. Druk éénmaal op de toets. De cursor springt van regel 2 naar direct rechts van regel 1. Druk nogmaals op ERASE. De laatste 2 van regel 1 verdwijnt. Geef nu (rode cursor) een 1 in. Maak de cursor wit en druk op BACKWARD, totdat de cursor direct rechts van de B in deze regel staat. Druk op ERASE. De letter B verdwijnt. Zet er nu een A voor in de plaats (rode cursor). Maak de cursor wit, ga met FORWARD tot het regeleinde en druk op NEW LINE.

Let op: de cursor moet niet OP het te wissen teken staan, maar pal rechts ervan. Uw beeldscherm ziet er nu zó uit:

De cursor staat nu pal rechts van regelnummer 2. Geef nu in PRINT met de groene cursor, en met de blauwe cursor een A. Daarna geeft u NEW LINE. Uw scherm ziet er nu zó uit:

Nu geeft u met de groene cursor GOTO in, gevolgd door (rode cursor) 1. Voor we verder gaan kijken we nog eens naar het scherm:

U ziet dat het statusniveau aangeeft, dat er nog 52 'bytes' geheugenruimte over zijn. De snelheid staat op 30 (SPEED = 30). Maak de cursor wit en druk op de toets SLOWER. Elke keer wanneer u deze toets indrukt ziet u SPEED afnemen. Wanneer u op de toets FASTER drukt neemt SPEED toe.

Voor we het programma beginnen geeft u in SPEED = 1. Nu drukt u op de toets STACK. Druk nu tweemaal op de toets RUN/HALT om het programma te starten. U ziet hoe de computer uw programma verwerkt.

Om het programma te stoppen drukt u op RUN/HALT. Zoals al eerder vermeld wordt met de toets GAME RESET op de computer elke variabele gewist (het programma zelf NIET), en teruggesprongen naar het programmabegin.

Laten we stap voor stap doornemen hoe de computer uw programma verwerkt. Zet de SPEED op 60. Maak de cursor wit en druk op de toets STEP; hierdoor wordt u programma regel voor regel uitgevoerd.

In regel 1 staat de opdracht A—A + 1. De computer leest dit als 'A wordt A + 1. Let op het registerniveau wanneer regel 1 wordt uitgevoerd. In regel 2 gaf u de opdracht PRINT A. Hiermee geeft u de computer de instructie, de waarde van A in het uitvoerniveau af te beelden.

Maar dat zien we later. In regel drie leest de computer GOTO 1. De computer gaat terug naar regel 1, en geeft A een nieuwe waarde. Elke keer dat het programma wordt gedraaid krijgt A een nieuwe waarde, die wordt afgedrukt, waarna het programma opnieuw wordt afgedraaid. De computer gaat hiermee door tot alle geheugenruimte is gebruikt. De resterende geheugenruimte kunnen we vinden achter SYMBOLS in het statusniveau.

Breng nu het variabelenniveau op het scherm. Wijzig SPEED in 1, en wis het geheugen met GAME RESET. Druk op RUN/HALT en kijk hoe de computer uw programma afwerkt. Bij elke stap van het programma ziet u, welke waarde A heeft. Het scherm ziet er zó uit:

Stop het programma en wis alle variabelen (GAME RESET). Verwijder het variabelenniveau en haal het uitvoerniveau op het scherm. Wijzig SPEED in 60. Het scherm ziet er nu zó uit:

Druk op RUN/HALT en start het programma. In regel 2 krijgt de computer opdracht, de waarde van A af te beelden (PRINT A). Deze waarde verschijnt in het uitvoerniveau. Let op SYMBOLS in het statusniveau. Ook al staat '1' in het uitvoerniveau op het scherm, de computer gaat door met het veranderen van A. Zet de linker-moeilijkheidsschakelaar in stand A. Uw beeldscherm ziet er ongeveer zó uit:

Wanneer we méér van de uitvoer op het scherm willen zien, halen we de status-, programma- en registerniveau's weg. Nu ziet het scherm er zó uit:

Maar stel, dat we het scherm niet willen overbelasten. Stop het programma en wis de variabelen met GAME RESET. Verwijder het uitvoerniveau, en breng het programmaniveau op het scherm. Met de

toets FORWARD zetten we de cursor vlak achter regel 1, zodat het scherm er zó uit-ziet:

Nu drukken we op NEW LINE. We kunnen nu een nieuwe instructie invoegen. Met de groene cursor wordt dat CLEAR. Zo ziet het scherm eruit:

Haal de register-, variabelen- en uitvoerniveau's op het scherm. Laat de linker-moeilijkheidsschakelaar in stand A, en start het programma. Zodra de computer de opdracht CLEAR krijgt maakt hij het uitvoerniveau schoon, en geeft (volgens regel 3) de nieuwe waarde van A.

PROGRAMMEREN IN BASIC kan slechts werken met getallen van twee cijfers. Dus zodra '99' wordt bereikt wordt de waarde 'overbelast' en zal vervolgens '00' verschijnen.

## De instructie note

Telkens wanneer het programma een waarde opbergt in NOTE (rode cursor) geeft de luidspreker van uw televisie een toon weer.

We zullen dit met een programma demonstren. Wanneer u al een programma had ingegeven, druk dan nu op GAME SELECT.

Geef het volgende in:

1	Note	-	Note	+	1
2	Goto				1

Start het programma. Laat het langzaam uitvoeren, en let op hoe het werkt (registerniveau). U ziet dat de waarde van NOTE wordt afgedrukt in het variabelenniveau.

Stop het programma en voeg een nieuwe regel 2 in (met de witte cursor naar het eind van regel 1, en op NEW LINE drukken). De oude regel 2 wordt nu regel 3.

## 2 IF Note 6 Then Note — 0

Het beeldscherm (zonder register- en variabeleniveaus) ziet er nu zó uit:

Met de instructies IF en THEN kunnen we de computer opdragen, WANNEER (IF) iets het geval is, DAN (THEN) moet dat gebeuren. In dit geval moet, wanneer Note groter is dan 6, Note de waarde 0 krijgen. Start het programma, en kijk in het registerniveau hoe het werkt. U ziet dat wanneer Note 7 wordt (groter dan zes), Note de waarde 0 krijgt toegekend.

Met een ander programma is hetzelfde resultaat te krijgen. Geef het volgende in:

Draai het programma. U merkt dat hetzelfde gebeurt, maar de tussenruimte tussen de tonen gelijk is. Wanneer u in het uitvoerniveau de waarde van C wilt kunnen zien, dan geeft u ergens in het programma de instructies 'Clear' en 'Print C' in. Om het flikkeren van het beeld in het uitvoerniveau te verminderen geeft u na de waarde in de Print C-instructie met de groene cursor een komma in (Print C,).

Voor de variabelen in uw programma kunt u elke letter van het alfabet gebruiken. Alle tot dusver geleerde toetsenfuncties zien we in het nevenstaande programma; het gebruikt vrijwel alle beschikbare geheugenruimte. Nadat u het programma heeft ingegeven verwijdert u de status- register- en uitvoerniveaus en draait u het programma.

In dit programma staan twee IF/THEN-instructies. Wanneer de voorwaarde in de eerste (regel 4) niet zijn vervuld, dan gaat de computer verder met regel 5. Wanneer

er genoeg geheugenruimte is kan er een willekeurig aantal instructies tussen de twee IF/THEN-instructies staan.

In sommige gevallen kunnen we twee opdrachten in één regel geven. Hierdoor kunnen we wat geheugenruimte besparen.

## De instructies Key en Print

De instructie KEY (rode cursor) wordt gebruikt om een variabele met het toetsenbord een waarde te geven, terwijl het programma loopt. Wanneer u geen waarde ingeeft, dan neemt het programma '0' aan.

Hier is een eenvoudig voorbeeld van de KEY-instructie:

Start het programma en druk op een van de toetsen op de rechterhelft van het toetsenbord. Na wat oefening kunt u een melodie spelen.

Zoals we later zullen zien kunnen we de instructie KEY ook gebruiken voor programmeren met het grafische niveau.

Probeer dit programma, met KEY, NOTE en PRINT eens:

Let bij dit programma eens op het uitvoerniveau. Telkens als u een getal ingeeft met de rechterhelft van het toetsenbord speelt het programma die noot, en drukt de waarde af op het beeldscherm.

Nu brengen we een kleine wijziging in het programma aan. In regel 2 zetten we een komma (,) na Print A. Wanneer we het programma draaien zien we, dat de computer zoveel mogelijk waarden van A afdrukt, als maar op een regel passen.

## Gebruik van de instructie Print

Zoals we in vorige programma's zagen kunt u de instructie PRINT gebruiken, om een waarde van een variabele op het uitvoerniveau te laten afbeelden. U kunt ook woorden laten afdrukken. Die woorden moeten dan tussen haakjes staan ('. . .'). Stel SPEED in op B en geef dit programma in:

Draai het programma en let op het uitvoerniveau. Het programma drukt elk woord af . . . maar boven elkaar,

Door een simpele verandering gaat het beter. Aan het eind van regel 1 en regel 2 zetten we een komma:

Die komma geeft aan, dat, wat er ook op regel 2 staat, het NA regel één, op dezelfde regel in het uitvoerniveau, moet worden afgedrukt. Voor de komma na regel 2 in het programma geldt hetzelfde, maar nu met regel 3. We kunnen de zaak echter vereenvoudigen als volgt:

En het kan zelfs nóg korter:

Door de instructie zo kort mogelijk te houden besparen we bovendien geheugenruimte.

## Gebruik van het grafisch niveau

Het grafische niveau is het blauwe, rechthoekige veld op het scherm. Op het eerste gezicht lijkt het één rood hokje te bevatten in de linker bovenhoek. Maar 'daaronder' zit een wit hokje. Beide kunnen onafhankelijk van elkaar worden verplaatst door uw programma.

Om de hokjes te verplaatsen moet u hun coördinaten wijzigen. Het rode hokje is hokje 1. De horizontale plaats kan worden

gewijzigd met opdracht Hor 1, de verticale plaats met Ver 1. Hor 2 en Ver 2 verplaatsen het witte hokje 2.

Beide hokjes gaan uit van de linker bovenhoek van het blauwe veld, waar alleen het rode hokje zichtbaar is. De plaats linksboven heeft de coördinaten 0,0; de waarden voor Hor en Ver zijn beide dus 0, zolang deze variabelen geen andere waarde hebben gekregen.

Wanneer zij een andere waarde krijgen toegewezen (bijvoorbeeld Hor 1 — 10), dan springt het hokje naar de betreffende plaats op het blauwe veld, zodra het programma wordt uitgevoerd.

Geef het volgende programma in:

Dit programma verplaatst het rode hokje naar rechts, telkens met één stap. Wanneer u het programma laat draaien ziet u het rode hokje langzaam naar rechts gaan. In het variabelenniveau ziet u de waarde van Hor 1 steeds met 1 toenemen, tot '99', de hoogste waarde die mogelijk is. Daarna wordt de waarde 9, en het rode hokje springt geheel rechts naar geheel links.

De horizontale coördinaten (Hor 1 en Hor 2) lopen dus van 0 tot 99. Ditzelfde geldt voor de verticale coördinaten Ver 1 en Ver 2; 0 is geheel boven, en 99 geheel onder. Geef het volgende programma nu in:

Verwijder het programmaniveau van het scherm en zorg dat het gehele grafische niveau zichtbaar is. Dit programma laat zien, hoe de hokjes over het scherm kunnen worden verplaatst. Bovendien geeft het een indruk van de mogelijkheden van HIT en ELSE.

In regel 5 krijgt de computer opdracht om noot 2 te laten horen, wanneer de hokjes botsen (HIT), wat nu en dan gebeurt. Hiervoor moeten de hokjes dezelfde coördinaten (plaats) hebben. Anders (ELSE) moet

de computer toon 7 laten horen, wat dus gebeurt tot de hokjes weer botsen.

(Wanneer u ELSE NOTE — 7 uit regel 5 schrapt wordt alleen toon 2 gespeeld bij botsing; anders blijft het stil.)

## Gebruik van de instructie MOD

Mod is een rekenkundige instructie, die veel lijkt op delen (:). Programmeren in Basic kent alleen gehele getallen; breuken zijn niet mogelijk. Voorbeeld: 14:5. De uitkomst moet zijn 2 4/5, of 2, rest 4. Bij programmeren in Basic is de uitkomst echter 2. Hoewel na de deling een rest van 4 overblijft, kunnen breuken (en resten) niet worden verwerkt. Dus de uitkomst is 2.

Daardoor zijn de uitkomsten van 12:4 en 13:4 hetzelfde. In beide gevallen is het antwoord 3, want 4 past drie keer helemaal in 12 zowel als 13. De rest van 1 in het laatste geval kan de computer niet verwerken.

Behalve met de Mod-functie. Mod geeft aan, welke rest overblijft na de deling. Dus 14 Mod 5 is 4; 14:5 is 2 (waarvoor 10 werd gebruikt), waarna 4 overblijft. Deze rest wordt verkregen met Mod, dus Mod = 4. Wat is 14 Mod 4? Het antwoord is 1, want na de deling 13:4 blijft een rest van 1 over (4x3 = 12, +1 = 13). Daarentegen is 12 Mod 4 0, want deze deling komt uit, en laat geen rest over.

Normaal gesproken is delen door 0 niet mogelijk. In Programmeren in Basic echter kan het wel, alleen is de uitkomst 0 (5:0 = 0).

## Voorrang bij het rekenen

Bij een vergelijking worden de instructies (+, -, x, :, Mod, —) volgens bepaalde 'voorrangsregels' bewerkt. In Programmeren in Basic is de voorrang als volgt:

1. x : (eerst)
2. + -
3. Mod
4. =
5. — (laatst)

## Rekenen met haakjes

Door haakjes te gebruiken (links op het toetsenbord, groene cursor) kunnen de voorrangsregels in een vergelijking worden gewijzigd.

In de volgende vergelijking bijvoorbeeld:

A 5 + 3 x 2 Mod 7

is de eerste stap  $3 \times 2 = 6$   
de tweede stap  $5 + 6 = 11$   
en de derde stap  $11 \text{ Mod } 7 = 4$ ,  
dus A = 4.

Met haakjes kan de berekening echter heel anders lopen:

A (5 + 3) x 2 Mod 7

De eerste stap is  $5 + 3 = 8$   
de tweede stap  $8 \times 2 = 16$   
en de derde stap  $16 \text{ Mod } 7 = 2$ ,  
dus A = 2.

## Programmavoorbeelden

Hieronder vindt u enkele programma's. Hiermee kunt u de mogelijkheden van Programmeren in Basic ontdekken.

Let goed op het gebruik van de instructies (IF, THEN, HIT, ELKE, PRINT, KEY, etc.) in het programma. Bedenk ook, dat een KEY-instructie afhangt van wat u rechts op uw toetsenbord geeft.

Wanneer u een programma niet goed begrijpt, stop het dan, druk op GAME RESET, en draai het met de toets STEP. Door in het registerniveau te kijken wat er ge-

beurt ontdekt u het hoe en waarom van elke instructie al gauw.

Nadat u de programma's heeft onderzocht en de beginselen van het programmeren onder de knie heeft, bent u al een heel eind op weg om uw eigen programma's te schrijven.

**Hit**

**Music**

#### **Klok-achtig programma**

Kijk bij dit programma alleen naar het uitvoerniveau. De snelheid (SPEED) kan worden ingesteld op 30 of 60.

#### **Pong-spel (bal en racket)**

##### **Pong-spel**

Zonder geluid



ATARI CONSUMER DIVISION  
ONDERDEEL VAN WEA-RECORDS B.V.  
POSTBUS 1882, 1200 BW HILVERSUM, PAYS BAS  
A Warner Communications Compagny 