



## MCS BASIC-52 V1.3

### OPFRISKUUR VOOR EEN POPULAIRE INTERPRETER

De door Intel ontwikkelde MCS BASIC-52-interpreter is nog altijd zeer populair. Maar na 15 jaar is het hoog tijd voor een grondige facelift.

De MCS BASIC-52-interpreter V1.0 werd in 1985 door Intel speciaal ontwikkeld voor de 8052-AH microcontroller. Hij past met zijn omvang van slechts 8 Kbyte precies in het interne ROM-geheugen van deze controller. Jammer genoeg werd de productie van de 8052-AH beëindigd. Deze interpreter - de laatste 'officiële' versie is 1.1 - is nog altijd zeer geliefd, ondanks het feit dat hij een aantal fouten bevat. Tegenwoordig wordt de 8052-familie met zijn vele derivaten het meest gebruikt van alle microcontrollers. Het is dan ook jammer dat BASIC-52 V1.1 niet op de snelste leden van deze familie draait. De twee auteurs hebben de broncode grondig aangepast tot versie 1.3 en hebben daarmee het volgende bereikt:

- Alle bekende fouten zijn verholpen.
- Basic-52 V1.3 programmeert nu ook EEPROM's, zelfs als het in extern geheugen draait.
- De nieuwe instructie 'ERASE' wist een EEPROM.
- De maximale waarde voor 'XTAL' kan

nu tot 78 MHz worden ingesteld, zodat de interpreter zelfs op een Dallas 80C320 met 33 MHz draait!

- Een nieuwe reset-routine herkent de baudrate en verschillende typen controllers. BASIC-52 V1.3 draait dan ook op vele derivaten van de familie (80515, 517(A), 528, 535, 537, 575 enz.).
- De nieuwe 'OPCODE' 43H leest een waarde van de resultaten-stack in een variabele.
- De interpreter is nog altijd 8 Kbyte groot en pas dus ook in een 87C52 met een interne EPROM van 8 Kbyte.

Door Intel is de interpreter inmiddels als freeware vrijgegeven. Hij kan dus volstrekt legaal als broncode met commentaar van het Internet worden opgehaald. Ondanks het aanwezige commentaar is de broncode moeilijk te lezen en de auteurs moesten een simulator gebruiken om achter de functies van de interpreter te komen. Maar laten we niet op de zaken vooruit lopen.

H.-J. Böhling en D. Wulf

## De verbeteringen nader bekeken

Begonnen werd met een uitgebreide speurtocht op het Internet naar suggesties voor verbeteringen en beschrijvingen van fouten van MCS-BASIC-52. Jammer genoeg was de aanwezige broncode niet relocatable, m.a.w. hij kon niet worden verplaatst. Zelfs het invoegen van een paar bytes leidde meteen tot fouten. De oorzaak daarvoor lag vooral aan het feit dat de broncode uit twee aparte bestanden bestond, namelijk één voor de interpreter en één voor floating-point-berekeningen. Daarom werd begonnen om beide bestanden tot één bestand samen te voegen en alle absolute sprongen, relatieve sprongen en aanroepen zonder label van labels te voorzien. Op een paar plaatsen in de broncode waren data-regels opgenomen met instructies in machinetaal, waarschijnlijk omdat de toendertijd gebruikte assembler de nieuwe instructies voor de 8052 niet ondersteunde. Deze data-regels werden omgezet in mnemonics. Een routine in de broncode verschilde van de originele Intel-ROM-code. Deze routine werd vervangen door de mnemonics volgens de ROM-code. Na deze veranderingen was de code relocatable zodat een paar 'ORG'-instructies - met name tussen de interpreter- en de rekencode - konden worden verwijderd. De geheugenruimte die daardoor vrij kwam, kon nu worden gebruikt om andere fouten te herstellen. De gewonnen plaatsruimte was echter lang niet voldoende voor het herstellen van alle fouten en het aanbrengen van de verbeteringen. Daarom moest ook de naamsvermelding van Intel-programmeur J. Katausky worden verwijderd. Het daarbij vrijgekomen bit 40H van het interne bit-adresseerbare geheugen kon nu worden gebruikt voor de detectie van een foutje bij een underflow in de vermenigvuldigings-routine.

Al in 1993 had D. Karmann de Interpreter herschreven naar versie 1.2 voor een 8031-controller en daarbij twee bekende fouten hersteld:

- Een BASIC-autostart-EPROM werkte nu ook samen met uitgebreide opdrachten en
- Variabelen met de naam 'F' konden nu zonder enige beperking worden gebruikt.

Dan Karmann was zo vriendelijk om de broncode van zijn verbeteringen aan de auteurs ter beschikking te stellen.

## BASIC-52 V1.3 en Elektuur-boards

In het artikel BASIC-537 in Elektuur 2/2000 werd al uitvoerig beschreven welke hardware-aanpassingen aan de Elektuur-boards '80C537' en '80535' nodig waren om de variant BASIC-537 te kunnen gebruiken. Wat daar werd geschreven, geldt ook voor V1.3, echter met de volgende bijzonderheden:

- De waarde MTOP hoeft niet meer te worden opgegeven, omdat V1.3 het aanwezige RAM automatisch herkent.
- BASIC-52 V1.3 herkent de gebruikte baudrate automatisch en daarom moet na iedere nieuwe start een spatieteken naar het board worden gestuurd.

De BASIC-besturingsit Elektuur 2-3/1998 kan met BASIC-52 V1.3 ook met een Dallas 80C320 Speed it  $\mu$ P worden gebruikt. De Dallas-controller is hard- en software-matig compatibel met de 8052. De CPU werd echter geheel nieuw ontworpen en heeft voor de uitvoering van één machinecyclus nog maar vier klok-slagen nodig in plaats van twaalf. De maximale klokfrequentie bedraagt 33 MHz. Voor de ombouw wordt het aanwezige kristal vervangen door een exemplaar van 22.118.400 Hz, zodat de uitvoeringstijd ongeveer zesmaal hoger wordt dan met een 80C32-controller. Weliswaar zou de controller ook met 33 MHz geklokt kunnen worden, maar dan ontstaan er problemen met de verkrijgbaarheid van een voldoende snelle EPROM (minimaal 40 ns) en een geschikt kristal. Om technische redenen worden grondtoon-kristallen namelijk maximaal tot een frequentie van 22 MHz gemaakt. Daarboven worden boven-ton-kristallen gebruikt. Door een externe resonantiekering wordt het kristal gedwongen op de desbetreffende boventoon te resoneren. Maar bij 22 MHz moet ook het programhet programmeergeheugen voor de interpreter voldoende snel zijn (70 ns). Voor het datageheugen kan een type met een normale toegangstijd van 150 ns worden gebruikt. Bij toegang tot dit geheugen worden door de controller zelf wacht-cycli ingelast. De twee TTL-IC's (74HC573, 74HC00) moeten worden vervangen door snellere exemplaren uit de 74AC- of 74F-serie. Als EEPROM wordt een 28C64 (IC5 op de print) gebruikt. Daarbij moeten wel de pennen 1 en 27 losgemaakt worden van +5 V. Pen 1 kan los blijven, pen 27 moet met /WR van de controller worden verbonden (pen 27 van IC3).

## Terminal-MCS-51

Als men BASIC-52 V1.3 op diskette EPS 000121-11 bestelt, komt men automatisch in het bezit van de nieuwste versie van Terminal-MCS-51, een terminalprogramma voor DOS-computers dat natuurlijk ook in een DOS-venster van Windows werkt. Terminal-MCS-51 werd geschreven om de bekende tekortkomingen van BASIC-52 op te heffen. Zo beschikt BASIC-52 jammer genoeg niet over een 'RENUMBER'-opdracht om de BASIC-regels te hernummeren. Ook is BASIC-52 zeer beperkt in zijn mogelijkheden en het aanpassen van een BASIC-programma is bewerkelijk. Terminal-MCS-51 biedt als extra o.a. de volgende mogelijkheden:

- Eenvoudig laden en opslaan van BASIC-52-programma's.
- Hernummeren van de regelnummers en nummers van BASIC-52-programma's.
- Een geïntegreerde regel-editor voor BASIC-52-regels.
- Mogelijkheid om een willekeurige ASCII-editor te koppelen.

De shareware-versie van Terminal-MCS-51 kan van de Elektuur-homepage worden gedownload. De shareware is gelijk aan de versie op de diskette, op één uitzondering na: het hernummeren van BASIC-regelnummers is niet mogelijk.

BASIC-52 V1.3 is ook als geprogrammeerde EPROM EPS 000121-21 via de EPS-service verkrijgbaar.

**Tabel 1. SFR's van de 80535/537 A/D converter**

SFR	Adres	Bit	Naam	Functie
ADCON	0D8H	2-0	MX2,MX1,MX0	keuze van de analoge ingang
ADCON	0D8H	3	ADEX	start via soft- of hardware
ADCON	0D8H	4	ADM	enkelvoudige of continue conversie
ADCON	0D8H	5	BSY	conversie bezig
DAPR	0DAH	3-0	DAPR.3-0	onderste referentiespanning
DAPR	0DAH	7-4	DAPR.7-4	bovenste referentiespanning
ADDAT	0D9H	7-0		8-bits meetresultaat

## EPROM's programmeren

MCS-BASIC-52 V1.1 kan EPROM's programmeren, maar jammer genoeg niet als het in een extern programmeergeheugen draait. Een voorstel van R. Skowronek leidde tot de volgende elegante oplossing:

BASIC-52 V1.3 programmeert nu hardware-EEPROM's, zelfs als het in een extern programmeergeheugen draait (instructies 'PROG(1-6)' en 'PGM'). De EEPROM wordt net als een RAM geschreven, alleen navenant langer. De eerder gebruikte Poort-pennen 1.3 (/ALEDIS), 1.4 (/PRGPLS), 1.5 (/PRGEN), bit 51H (INTELB) in het interne geheugen en de externe RAM-adressen 12AH...12BH (IPROGS) zijn niet meer nodig en nu voor de gebruiker weer onbeperkt beschikbaar. De 'FPROG'-instructies waren niet meer nodig en zijn verwijderd. Dat leverde niet alleen plaatsruimte op, maar daardoor kwam ook het BASIC-token 0F9H vrij. Dit werd gebruikt voor de nieuwe 'ERASE'-functie. 'ERASE' wist een EEPROM (het geheugengebied 08000H...0C000H wordt met 0FFH gevuld). 'ERASE' wordt in de commando-mode opgeroepen en het duurt circa 2:45 minuten om het 16 Kbyte grote gebied te wissen.

Een bijzonderheid van BASIC-52 is dat het aantal opdrachten met maximaal 16 eigen opdrachten kan worden uitgebreid. Deze uitgebreide opdrachten moeten in assembler worden geprogrammeerd. Door middel van zogenaamde 'OPCODES' kunnen in deze opdrachten ook systeemroutines worden aangeroepen. Jammer genoeg was het tot nu toe niet mogelijk om een op de resultaten-stack gezette waarde toe te wijzen aan een BASIC-variabele. Dit moest worden opgelost door omslachtig gebruik van de 'POP'-opdracht. In BASIC-52 V1.3 is voor dit doel nu OPCODE 43H opgenomen (deze wordt bijvoorbeeld gebruikt in de uitgebreide opdracht 'RDSFR(adres)variabele').

De opdracht 'TIME=0' zette de milliseconden-timer jammer genoeg niet op

nul, waardoor TIME nooit precies nul werd. In V1.3 is deze fout hersteld.

De functie 'ASC(x)' leverde voor tekens die overeenkwamen met BASIC-operatoren (\*, +, -, /, <, =, > en ?) verkeerde retourwaarden. De interpreter vertaalde deze tekens bij de omzetting in het interne token-formaat in het token van de BASIC-operator zelf. Het vraagteken werd bijvoorbeeld het token voor 'PRINT'. Dus werd niet de juiste ASCII-waarde geretourneerd, maar de waarde van het token. BASIC-52 V1.3 levert nu juiste waarden voor alle ASCII-tekens.

Om BASIC-52 V1.3 op zoveel mogelijk 'afstammelingen' van de 8052-familie te laten draaien en vooral ook op de 80C320 Speed it  $\mu$ P van Dallas (maximum klokfrequentie 33 MHz) werd de reset-routine flink onder handen genomen:

- Een nieuwe baudrate-herkenning gebruikt timer 2 (in plaats van een software-meetroutine) om de baudrate te bepalen. De meting is daardoor onafhankelijk geworden van het type controller en de kristalfrequentie.
- De voor tijdmeting gebruikte timer 0 (BASIC-software-klok) wordt nu in 16-bit-mode gebruikt in plaats van 13 bits zoals voorheen. Daardoor is de maximale waarde voor XTAL = 78627473 Hz geworden. De standaardwaarde na inschakelen is nog wel 11059200 Hz, maar met 'XTAL=waarde' kan hij opnieuw worden ingesteld. Als alternatief kan de nieuwe waarde direct op de volgende adressen in de EPROM worden gepatched:

```
17F1H = 11
17F0H = 05
17EFH = 92
17EEH = 00
```

Voor de baudrate-herkenning moet de gebruiker, net als voorheen, over de seriële interface een spatie naar BASIC-52 sturen. De tekenlengte wordt dan met behulp van timer 2 door de herkenningroutine gemeten. In principe wordt timer 2 door BASIC-52 altijd als

baudrate-generator gebruikt. Dat geldt voor 8052-controllers en ook voor de Dallas 80C320. Controllers van het type 80535, 80515 en 80517 kunnen timer 2 echter niet als baudrate-generator gebruiken. Ze hebben een speciale, volledige van timer 2 onafhankelijke baudrate-generator die slechts twee snelheden heeft, uiteraard afhankelijk van de gebruikte kristalfrequentie. Bij een kristalfrequentie van precies 12 MHz zijn baudrates van 4800 Bd en 9600 Bd mogelijk. Als de instelroutine van BASIC-52 V1.3 een dergelijke processor herkent, wordt de generator gestart met een waarde die past bij het ingegeven spatieteken (4800/9600 Bd).

De 80C517A-controller die ook veel wordt toegepast, heeft weer een uitgebreidere speciale baudrategenerator. Deze is volledig instelbaar op alle baudrates en is desondanks volledig onafhankelijk van timer 2. Ook deze controller wordt door BASIC-52 V1.3 herkend en ingesteld op de herkende baudrate. Bij de controllers met een aparte baudrategenerator kan de gebruiker vrij over timer 2 beschikken.

## I<sup>2</sup>C en SFR

Zoals we hebben gezien kan BASIC-52 eenvoudig worden uitgebreid met (in assembler geschreven) extra BASIC-opdrachten. Deze uitbreidingen passen echter niet meer in het 8 Kbyte grote codegebied van de BASIC-interpreter. Door de auteurs zijn zes nieuwe opdrachten geschreven, waarvan er vier betrekking hebben op communicatie met I<sup>2</sup>C-componenten en twee op het Special Function Register (SFR).

### I2CSTART

Stuurt de startconditie naar de I<sup>2</sup>C-bus.

### I2CSTOP

Stuurt de stopconditie naar de I<sup>2</sup>C-bus.

### I2CPUT(waarde)

Stuurt een byte naar de I<sup>2</sup>C-bus.

### I2CGET(variabele)

Leest een byte van de I<sup>2</sup>C-bus in een variabele.

De twee andere opdrachten maken het mogelijk om het Special Function Register (SFR) te schrijven en te lezen. Sommige derivaten van de familie bevatten toegevoegde SFR's. Om ook deze via BASIC toegankelijk te maken, zijn de volgende opdrachten geïmplementeerd:

### WRSFR(adres) waarde

Schrijft de waarde in het opgegeven SFR-adres.

```

4 REM *****
5 REM * Programma voor sturing van de A/D-omzetter in de 80535/537 *
6 REM *****
10 RANGE=5 : REM referentiespanning van de controller
20 ADCO=0D8H : REM register met ADC-statusinformatie e.d.
30 ADDAT=0D9H : REM register voor meetwaarde-overdracht
40 DAPR=0DAH : REM register voor meetbereik en -start
50 RDSFR : REM lees statusbyte
60 BUSY=BUSY.AND.0C0H : REM isoleer functiebits
70 WRSFR (ADCO)BUSY : REM kies meetingang 0 (Port 7.0)
80 WRSFR (DAPR)0 : REM start meting in het grootste meetbereik
90 RDSFR (ADCO)BUSY : REM lees toestand van de A/D-omzetter
10 IF (BUSY.AND.020H)>0 THEN GOTO 90 : REM wacht indien bezig
110 RDSFR (ADDAT)VOLTAGE : REM lees meetwaarde
120 PRINT "De spanning op ingang 0 bedraagt:",
130 PRINT VOLTAGE*RANGE/256,"volt" : REM pas waarde aan meetbereik aan
140 PRINT "Wilt u met een meting in het 1-V-bereik doorgaan (J/N)?",
150 I=GET.AND.0DFH
160 IF I=ASC(J) THEN GOTO 190
170 IF I=ASC(N) THEN GOTO 260
180 GOTO 150
190 RANGE=1 : REM meetbereik = 1 volt
200 WRSFR (DAPR)040H : REM bereik instellen en meting starten
210 RDSFR (ADCO)BUR (ADCO)BUSY : REM lees toestand van de A/D-omzetter
220 IF (BUSY.AND.020H)>0 THEN GOTO 210 : REM wacht indien bezig
230 RDSFR (ADDAT)VOLTAGE : REM lees meetwaarde
240 PRINT "De spanning op ingang 0 bedraagt precies:",
250 PRINT VOLTAGE*RANGE/256,"volt" : REM pas waarde aan meetbereik aan
260 END

```

De A/D-omzetter in de 80517A heeft een resolutie van 10 bits en een grotere meetnauwkeurigheid. Bij deze controller-typen kan echter de interne referentiespanning niet via stuurregisters worden beïnvloed. In het volgende voorbeeld kan de te meten ingang in het BASIC-programma worden gekozen:

```

4 REM *****
5 REM * Programma voor sturing van de A/D-omzetter in de 80517A *
6 REM *****
10 RANGE=5 : REM referentiespanning van de controllers
20 ADCO=0D8H : REM register voor keuze van de meetingang e.d.
30 ADDATL=0DAH : REM meetwaarde LSB
40 ADDATH=0D9H : REM meetwaarde MSB
50 PRINT "A.u.b het nummer van de meetingang (0-7) opgeven:",
60 INPUT CHANNEL
70 CHANNEL=CHANNEL.AND.7 : REM overbodige bits maskeren
80 RDSFR (ADCO)BUSY : REM andere functiebits inlezen
90 BUSY=(BUSY.AND.0C0H).OR.CHANNEL : REM kanaalkeuze toevoegen
100 NRSFR (ADCO)BUSY : REM kanaalkeuze in controller ingeven
110 WRSFR (ADDATL)0 : REM meting starten
120 RDSFR (ADCO)BUSY : REM A/D-busy-toestand afvragen
130 IF (BUSY.AND.020H)>0 THEN GOTO 120 : REM wacht indien bezig
140 RDSFR (ADDATL)UOLTPART : REM LS-byte lezen
150 RDSFR (ADDATH)VOLTAGE : REM MS-byte lezen
160 VOLTAGE=VOLTAGE+UOLTPART/256 : REM meetwaarde berekenen
170 PRINT "De spanning op ingang:",CHANNEL,"bedraagt:",
180 PRINT VOLTAGE*RANGE/256,"Volt" : REM pas waarde aan meetbereik aan
190 END

```

RDSFR(adres) variabele  
Leest de waarde uit het met adres opgegeven SFR in de variabele.

Door middel van deze opdrachten is het nu op eenvoudige wijze mogelijk om met de speciale hardwarefuncties waarover sommige controllers beschikken, te communiceren. We noemen bijvoorbeeld extra I/O-poorten of A/D-omzetters. Een goed voorbeeld van het gebruik van de SFR-commando's is te vinden in de BASIC-programma's 'AD-535.LIS' en 'AD-517A.LIS' (zie kader).

BASIC-52 V1.3 draait op een groot aantal controllers, waaronder de types 80535, 537, 517 en vele andere. De toegevoegde SFR's kunnen nu direct vanuit BASIC worden benaderd en het is niet meer nodig om de besturings- c.q. regelsoftware voor de speciale hardwarefuncties in assembler te programmeren. Dat dit een enorme besparing aan overhead geeft en ook het programmeren aanmerkelijk vereenvoudigt, zal duidelijk zijn. In de voorbeelden worden de A/D-omzetters van respectievelijk een 80535 en een 80517(A) gebruikt. De A/D-omzetter in de 80535/537 heeft een resolutie van 8 bits. De externe referentiespanning ligt gewoonlijk aan +5 V. Het oplossend vermogen is dus 20 mV. Maar omdat de interne referentiespanning met de SFR 'DAPR' kan worden geprogrammeerd, kan met een tweede meting - in een kleiner meetbereik - een hogere resolutie tot circa 4 mV worden verkregen. De in tabel 1 vermelde SFR's worden daarvoor gebruikt.

(000121)