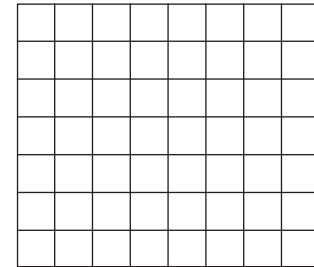
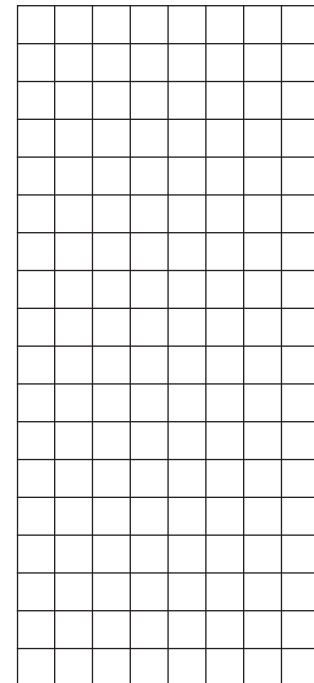


Bestnr.: 95 05 56



C-Control/BASIC stuurcomputer Type M en programmeeradapter



Impressum

Alle rechten, ook vertalingen, voorbehouden. Niets uit deze uitgave mag worden vervoelvoudigd, opgeslagen in een automatisch gegevensbestand, of openbaar gemaakt, in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door fotokopieën, of op enige andere manier, zonder voorafgaande schriftelijke toestemming van CONRAD ELECTRONIC NEDERLAND BV.
Nadruk, ook als uittreksel is niet toegestaan. Druk- en zetfouten voorbehouden. Deze gebruiksaanwijzing voldoet aan de technische eisen bij het in druk gaan. Wijzigingen in de techniek en uitvoering voorbehouden.

Omwille
van het
milieu
100%
recycling-
papier

© Copyright 1995 by CONRAD ELECTRONIC NEDERLAND BV
Windmolenweg 42, 7548 BM Boekelo
Internet: www.conrad.com

Belangrijk! Beslist lezen!

Deze bedieningshandleiding is een integraal onderdeel van dit product. Lees, voor u de C-Control/BASIC stuurcomputer of daarop aangesloten apparaten in gebruik neemt, deze handleiding zorgvuldig door! U vindt hierin uitleg over het correcte gebruik en u wordt gewezen op eventuele gevaren.

Bij schades, die ontstaan door het niet opvolgen van deze bedieningshandleiding, vervalt het recht op garantie! Wij zijn niet aansprakelijk voor schades die hiervan het gevolg zijn!

Het gebruik van de C-Control/BASIC stuurcomputer type M vraagt enige basiskennis op schakeltechnisch en computergebied. Als u niet beschikt over ervaring in de omgang met PC's, soldeerbouten en elektronische componenten, geeft onze Technische Dienst u gaarne advies betreffende meer geschikte instapmogelijkheden.

Aanwijzingen betreffende beperkte garantie en aansprakelijkheid

De C-Control/BASIC stuurcomputer, hierna aangeduid als "het apparaat", met de in de microprocessor MC68HC05B6 als ROM-masker geïntegreerde en bijbehorende PC-software wordt in de onderhavige vorm geleverd.

Conrad Electronic geeft geen garantie dat de prestatiekenmerken aan individuele eisen voldoen of dat de software in de microprocessor en de PC-software in elk geval zonder onderbrekingen en/ of fouten werken. De gebruiker draagt het volledige risico voor wat betreft de kwaliteit en het prestatievermogen van het apparaat inclusief alle software.

Conrad Electronic garandeert de functie van de meegeleverde applicatievoorbeelden onder inachtneming van de in de Technische gegevens gespecificeerde voorwaarden. Als het apparaat buiten het hiervoor vermelde niet of niet toereikend functioneert, neemt de klant alle ontstaande kosten voor service, reparatie of correctie op zich.

De garantie van Conrad Electronic beperkt zich uitsluitend tot het vervangen van het apparaat binnen de garantietijd bij duidelijke gebreken van de hardware, zoals mechanische beschadiging, ontbrekende of foutieve montage van elektronische componenten, met uitzondering van op sokkel gemonteerde geïntegreerde schakelcircuits en steekbruggen.

Er bestaat geen aansprakelijkheid voor schades, die rechtstreeks het gevolg zijn van het gebruik van de C-Control/BASIC stuurcomputer. Aanspraken die berusten op bindende wettelijke voorschriften voor productgarantie vallen hier buiten.

Gebruik waarvoor het apparaat bedoeld is

De C-Control/BASIC stuurcomputer is bedoeld voor programmeerbare aansturing van elektrische en elektronische apparaten, die werken op geaarde laagspanning. Deze apparaten kunnen in willekeurige technische systemen geïntegreerd worden, die niet direct of indirect bedoeld zijn voor medische, gezondheids- of levenbeschermende doelen of door het gebruik waarvan er gevaar zou kunnen ontstaan voor personen of goederen.

Voor het programmeren van het apparaat mag uitsluitend de originele programmeeradapter in combinatie met de meegeleverde PC-software gebruikt worden.

Inhoudsopgave

Inleiding

Aanwijzingen betreffende gebruik en veiligheid

Opbouw en manier van functioneren

Aansluiten van externe modules

Eerste ingebruikname - stap voor stap

Programmeren met CCBASIC

Tabellen en afbeeldingen

Technische gegevens

Inleiding

Wat is en wat kan de C-Control/BASIC stuurcomputer type M?

De C-Control/BASIC stuurcomputer is een compacte module voor universele toepassing bij meet-, besturings- en regelwerkzaamheden en beschikt bovendien over de capaciteiten voor seriële dataoverdracht en dataopslag.

De microprocestechniek is niet meer weg te denken uit het huidige leven. In praktisch alle moderne elektronische apparaten voeren microprocessors de regie. Hun "intelligentie" krijgen deze chips door programmering. De programmering van een microprocessor is voor een deel zeer gecompliceerd en vraagt omvangrijke specialistische kennis en duur ontwikkelingsgereedschap. Voor hobbyisten en kleine bedrijven blijft de toegang tot de microprocestechniek daarom meestal geblokkeerd. Met het C-control-System worden de mogelijkheden van deze techniek echter voor iedereen die geïnteresseerd is toegankelijk.

De C-Control/BASIC stuurcomputer als type M bestaat uit de kerncomponenten van de beproefde hardware van de C-Control/BASIC-unit (Conrad Electronic bestnr. 95 05 72). Door het weglaten van de niet voor elke

toepassing benodigde onderdelen en toepassing van de SMD-technologie ontstond een geminiaturiseerde versie, die door de geringe afmetingen speciaal voor toepassing in elektronische handapparaten of bij modelbouw aanbevolen wordt.

De programmering van de C-Control/BASIC stuurcomputer geschiedt in de algemeen bekende en makkelijk te leren programmeertaal BASIC. Zo wordt de C-Control/BASIC stuurcomputer door een paar regels BASIC-broncode een intelligente alarminstallatie, een complex dataregistratiesysteem, een besturingscentrale van een centrale verwarming of het "brein" van een klein robotmodel. Het terrein van de toepassingsmogelijkheden is praktisch onbegrensd.

Voor het contact met de buitenwereld staan acht analoge ingangen, twee analoge uitgangen alsmede 16 vrij als in- of uitgangen programmeerbare digitale poorten voor het aansluiten van sensoren, schakelaars, LEDs, transistors of relais tot uw beschikking.

De stuurcomputer bezit een ingang voor een DCF77-zendergestuurde ontvanger. Hiermee kunnen schakelfuncties tot op de seconde precies uitgevoerd worden. Als alternatief is via de DCF77-ingang een frequentiemeting mogelijk.

In vergelijking met de klassieke versie van de stuurcomputer beschikt het type M over een tweede frequentiemeetingang alsmede over een 'interrupt'-ingang. Het systeem is tevens uitgebreid met een RESET- module.

Aanwijzingen betreffende gebruik en veiligheid

Algemeen

De C-Control/BASIC stuurcomputer en de programmeeradapter werden onderworpen aan een veiligheidstest volgens de geldende wettelijke voorschriften en dienovereenkomstig gecertificeerd (CE). Bij juist gebruik bestaat er normaal-gesproken geen gevaar voor de gezondheid.

De C-Control/BASIC stuurcomputer en de programmeeradapter dienen, omdat het elektronische apparaten zijn, met de daarvoor gebruikelijke voorzichtigheid en zorgvuldigheid behandeld te worden. Het veronachtzamen van de hierin beschreven aanwijzingen of een ander gebruik dan waarvoor de apparaten bedoeld zijn, kan leiden tot beschadiging of vernieling van de stuurcomputer, van de programmeeradapter of van de daarop aangesloten apparaten.

Voorwaarden aan de omgeving

De C-Control/BASIC stuurcomputer en de programmeeradapter zijn niet beveiligd tegen lichtboogexplosies en mogen niet toegepast worden in industriële sterkstroominstallaties. De maximale ingangsgroottes volgens de specificaties in de "Technische gegevens" mogen niet overschreden worden. U mag de apparaten niet gebruiken in ruimtes of omgevingen, waarin zich brandbare of bijtende gassen, dampen of stoffen (kunnen) bevinden.

Nadat de C-Control/BASIC stuurcomputer of de programmeeradapter van een koude naar een warme ruimte gebracht zijn, mogen ze niet direct in gebruik genomen worden. Het mogelijk ontstane condenswater kan tot storingen in het functioneren of tot het uitvallen van elektronische componenten leiden. Laat de apparaten eerst op kamertemperatuur komen.

Vermijd sterke magneetvelden, zoals die in de nabijheid van machines of luidsprekers kunnen voorkomen.

Voedingsspanning

De C-Control/BASIC stuurcomputer en de programmeeradapter mogen in geen geval verbonden worden met 230 Volt netspanning!

Programmeeradapter

De programmeeradapter heeft als stroomvoorziening een **gestabiliseerde gelijkspanning** van 5V nodig. Gebruik daarvoor alleen een geteste laboratorium- of stekkernetvoeding. Let daarbij beslist op het aansluitschema! Het verkeerd polen van de voedingsspanning kan leiden tot beschadiging van de adapter.

C-Control/BASIC stuurcomputer

Voor de stroomvoorziening van dit apparaat dient u een **gestabiliseerde gelijkspanning van 5V** te gebruiken. Deze moet door de toepassingschakeling, waarin het apparaat gebruikt moet worden, ter beschikking gesteld worden en via de in het aansluitschema gekenmerkte contacten toegevoerd worden. Let hierbij beslist op het aansluitschema! Bij het verkeerd polen van de voedingsspanning kan de stuurcomputer vernield worden.

Het apparaat mag alleen in spanningloze toestand op de gebruikersschakeling gestoken of er afgetrokken worden. Anders kan het tot vernieling van de stuurcomputer of de aangesloten apparaten leiden. De eis van spanningloosheid geldt daarbij voor elk verbindingcontact van de stuurcomputer, niet alleen voor de pins voor de stroomvoorziening!

Als de stuurcomputer voor de programmering op de programmeeradapter gestoken is, dan wordt de stuurcomputer door de adapter van spanning voorzien. Voordat u de C-Control/BASIC stuurcomputer op de programmeeradapter steekt of hem daar van aftrekt, dient de stroomvoorziening van de programmeeradapter uitgeschakeld te worden.

Als stuurcomputer en adapter via een kabel (via vijfpolige stiftstrips, zie onder) met elkaar verbonden zijn, mag slechts één van beide apparaten met een spanningsbron verbonden zijn! Het andere apparaat wordt dan via de kabel van stroom voorzien.

Elektrische ontladingen

Vooraf in droge lucht kan het menselijk lichaam elektrostatisch opgeladen worden. Bij het contact met geleidende voorwerpen wordt deze lading met een kleine vonk afgevoerd. Zulke ontladingen kunnen bij het aanraken van

elektrotechnische componenten deze componenten vernielen. Voordat u de C-Control/*BASIC* stuurcomputer of de programmeeradapter aanraakt, dient u een groot, geaard voorwerp aan te raken (b.v.: een metalen PC-behuizing, een waterleiding of een verwarmingsbuis), om eventuele opladingen af te voeren.

Elektromagnetische verdraagzaamheid (EMV)

De door de C-Control/*BASIC* stuurcomputer type M afgegeven stoorstraling ligt onder de wettelijke limietwaarden (CE).

De C-Control/*BASIC* stuurcomputer type M kan echter zelf door stoorstraling uitzendende apparatuur of elektrostatische ontladingen in de omgeving in zijn functie beïnvloed worden. In deze gevallen wordt er automatisch een RESET gestart, om een gedefinieerd gedrag van het systeem te garanderen. Let er op, dat dan het eventueel onderbroken BASIC-programma opnieuw gestart moet worden (zie onder).

Om een veilig en zeker functioneren te garanderen, mag de C-Control/*BASIC* stuurcomputer alleen in apparaten met een afgeschermd behuizing toegepast worden. De toevoer van de voedingsspanning moet door passende maatregelen (ondersteunende condensatoren, smoorklep-pen) beschermd worden tegen spanningspieken/ -dalen. Niet beschermde ingangspoorten moeten op een gedefinieerd niveau gelegd worden.

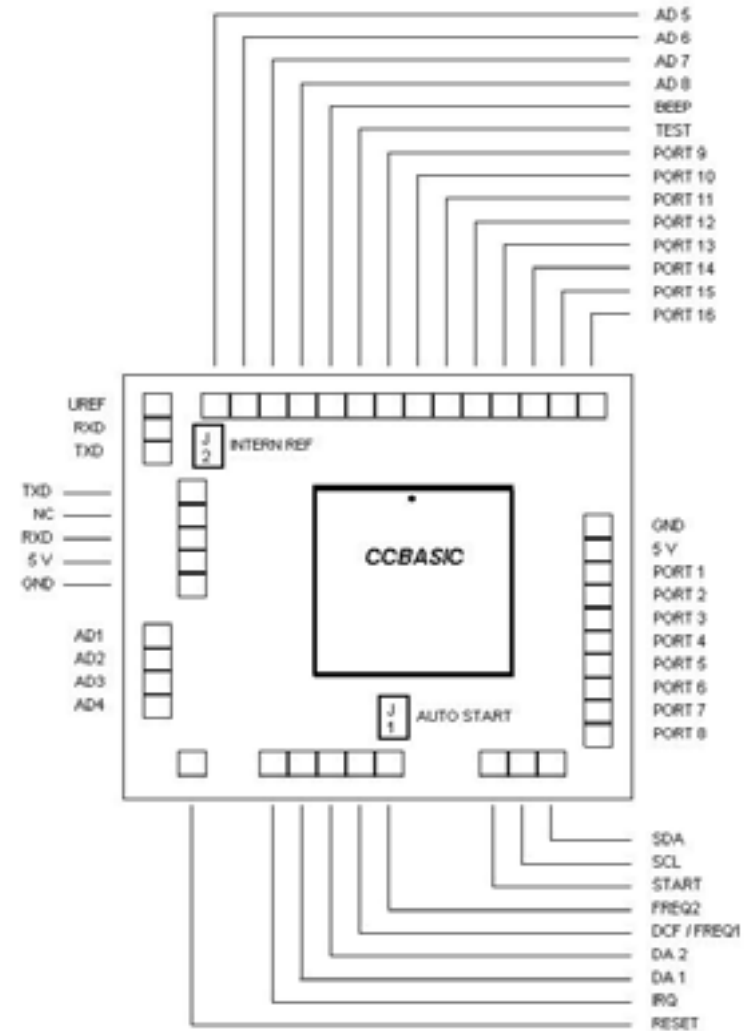
Let er svp op, dat door u gebouwde apparatuur voor ingebruikname een wet-telijke toelating (CE-certificaat) betreffende de totale functie moeten hebben!

Opbouw en functioneren

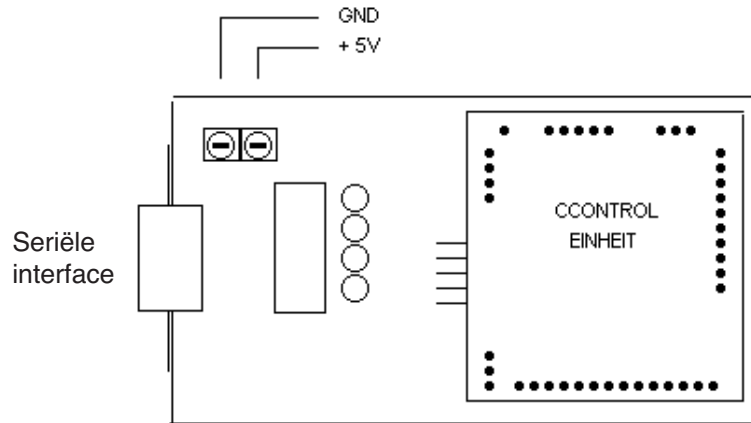
Overzicht

Aansluitschema's

Aansluitschema van de C-Control/*BASIC* stuurcomputer type M:



Aansluitschema van de programmeeradapter:



Hardware

Het hart van de C-Control/BASIC stuurcomputer is een microcontroller van het type MC68HC05B6 van MOTOROLA. De stuurcomputer is gemonteerd op een kleine printplaat. Daarop bevinden zich naast de microcontroller een geheugenchip, een RESET-module alsmede een paar passieve onderdelen (weerstanden, condensatoren). De geheugenchip heeft een capaciteit van acht kilobyte. De chip is bedoeld voor het opslaan van uw toepassingsprogramma en kan ook gebruikt worden voor het opslaan van data. Door de EEPROM-technologie blijft alle informatie ook na het uitvallen van de voedingsspanning behouden. De RESET-module garandeert een gedefinieerd uitschakelen van het systeem bij het uitvallen van de voedingsspanning.

Alle in- en uitgangen alsmede een paar systeemsignalen van de stuurcomputer zijn toegankelijk via stiftstrips. Via deze stiftstrips wordt de stuurcomputer op de applicatie resp. op de programmeeradapter gestoken. Aan de ene kant van de stuurcomputer steken vijf pins uit. Deze kant is kant 1, de nummers op de andere kant zijn doorlopend met de wijzers van de klok mee. De vijf pins aan kant 1 zijn bedoeld voor de verbinding van de stuurcomputer met de programmeeradapter via een kabel (niet bij de levering inbegrepen!), om de programmeeradapter ook te kunnen gebruiken als RS-232 niveauregelaar voor uw applicatie. In dit geval wordt de programmeeradapter via de kabel ook voorzien van 5V. Er is geen extra spanningsbron nodig.

Op de programmeeradapter bevinden zich de RS232 seriële interfacemodule met de externe componenten ervan alsmede de Sub-D-verbinder voor de verbinding met een PC via een nulmodemkabel. De stiftstrips zijn bedoeld voor het opnemen van de C-Control/BASIC stuurcomputer tijdens de programmering.

Stysteemsoftware

Hoe functioneert C-Control/BASIC ? Het door u gemaakte BASIC-programma wordt door een compiler omgezet in een volgorde van commandobytes. De commando's en de bijbehorende parameterbytes worden via de seriële interface en de programmeeradapter overgedragen naar de stuurcomputer, waar ze door het systeem daarvan in een EEPROM geheugenchip opgeslagen worden. Door het C-Control-concept kunnen uw toepassingsprogramma's in een zeer compacte vorm opgeslagen worden en beleggen meestal slechts een paar honderd van de 8000 ter beschikking bestaande bytes. Daardoor blijft een groot gedeelte van de geheugenbytes vrij en kan dat vrije gedeelte gebruikt worden voor het opslaan van data.

Zodra er op de START-ingang een low level aanwezig is, begint het systeem van de stuurcomputer de commando's na elkaar uit het geheugen te lezen en af te werken, tot aan het "einde programma" commando.

Communicatie tussen PC en de C-Control/BASIC stuurcomputer via de programmeeradapter

De programmeeradapter voert de niveauomvorming van de interface-signalen uit, om de seriële interface van de stuurcomputer aan te passen aan de RS232-standaard en een verbinding met de PC mogelijk te maken.

De seriële interface tussen PC en de stuurcomputer werkt in twee richtingen. Via deze interface worden naast de commandocodes voor de programmering van een toepassingsprogramma ook enige besturingscommando's doorgegeven. Tijdens de programmering wordt elke overgedragen byte per echo bevestigd.

In principe is een verbinding tussen PC en stuurcomputer alleen nodig voor de programmering. Aansluitend kan de stuurcomputer toegepast worden in de doelapplicatie en daar zelfstandig werken. De verbinding met de PC kan

echter ook tijdens het werken in de doelapplicatie blijven bestaan en b.v. gebruikt worden voor het overdragen van meetdata.

Jumpers

Op de stuurcomputer kunnen twee steekbruggen ("jumpers") gezet worden.

- J1 - "Auto Start" - verbindt de START-ingang permanent met GND. In dit geval begint het systeem direct na het inschakelen van de voedingsspanning resp. na een RESET met de uitvoering van een geladen BASIC-programma. Let op: om de stuurcomputer te kunnen programmeren, moet de jumper er afgehaald zijn!
- J2 - "Interne referentie" - verbindt de referentiespanningsingang met het 5V-niveau. Alle sensorsignalen op de A/D-ingangen worden dan gerefereerd aan de voedingsspanning.

Software voor het programmeren van de C-Control/BASIC stuurcomputer

De programmeeradapter wordt geleverd met software voor het programmeren van de C-Control/BASIC stuurcomputer. Deze bestaat uit een DOS-versie en een versie voor Windows 95 en Windows NT. Beide versies bieden een geïntegreerd gebruiksoppervlak voor het bewerken, compileren en overdragen van BASIC-programma's naar de stuurcomputer.

Beide versies kunnen onafhankelijk van elkaar ingezet worden. Als uw computer onder Windows 95 of Windows NT werkt, kunt u het beste met de desbetreffende softwareversie werken. Het staat u echter vrij ook de DOS-versie te installeren en te gebruiken. Als uw computer onder DOS, OS/2 of een oudere Windows-versie werkt, kunt u uitsluitend de DOS-versie gebruiken.

In het register SAMPLES vindt u een aantal voorbeelden, die het beginnen met de programmering van de C-Control/BASIC stuurcomputer vergemakkelijken.

Software voor DOS

Installatie

De DOS-software wordt simpel door het kopiëren van de bijbehorende bestanden op een vaste schijf van uw PC geïnstalleerd. Lees na het installeren beslist de Readme-file in het programmaregister! Daar vindt u nog meer actuele informatie.

Oproepen van het programma

Om de DOS-software te starten, gaar u naar het register CCE, daarna voert u CCE.EXE uit.

Programma-eigenschappen

De DOS-software wordt vooral gekenmerkt doordat deze geschikt is voor oudere computers (tot nu toe voor PC's van de XT-generatie). Bij de desbetreffende PC-hardware biedt het programma een kleurig oppervlak in techniek met meerdere vensters, met menu en muisbediening.

U vindt nog meer informatie over de omvang en bediening van de DOS-software in de Online-Help, die u kunt bereiken door op toets F1 te drukken.

Software voor Windows 95 en Windows NT

Installatie

Voor de installatie voert u SETUP.EXE van de installatiediskette uit. Lees na de installatie beslist de Readme-file in het programmaregister! Daar vindt u nog meer actuele informatie.

Oproepen van het programma

Om de Windows-software te starten, roept u na het met succes installeren het programma CCEW32D.EXE vanuit het startmenu of vanuit de Explorer op.

Programma-eigenschappen

De software voor Windows 95 of Windows NT biedt een oppervlak in het zogenaamde MDI-design (multiple document interface). Aanwijzingen voor de aparte programma-elementen alsmede voor de programmabediening vindt u wederom in de Online-help.

Als er uitspringende eigenschap in vergelijking met de DOS-software kan de geïntegreerde simulator/ debugger genoemd worden, die het mogelijk maakt BASIC- programma's te testen voor ze in de doelapplicatie toegepast worden.

Eerste ingebruikname - stap voor stap

Software-installatie

Installeer eerst de software, zoals beschreven in het vorige hoofdstuk.

Klaarmaken van stroomvoorziening

Gebruik een gestabiliseerde 5-volt-gelijkspanning, bijvoorbeeld een laboratoriumnetvoeding. Schakel de stroomvoorziening uit en klem de programmeeradapter via een korte kabel met gestripte uiteinden vast (denk aan de juiste poling).

Verbinden van de stuurcomputer met de PC

Met de programmeeradapter wordt een interfacekabel meegeleverd, een zogenaamde nulmodemkabel.

Steek deze nulmodemkabel in een vrije seriële interface van uw computer. Veel computers beschikken over een 9-polige en een 25-polige seriële interface. Als er op uw computer alleen nog een 25-polige interface vrij is, heft u een extra adapter nodig (niet bij de levering inbegrepen).

Het andere einde van de nulmodemkabel steekt u in de seriële interfacestekker van de programmeeradapter.

Opsteken van de stuurcomputer

Als de "Auto Start" - jumper er opgestoken is, verwijdert u die van de stuurcomputer.

Steek de stuurcomputer nu zo op de programmeeradapter, dat kant 1 (de kant met de 5 uitstekende pins) naar het midden van de adapter wijst. Let er op, dat alle steekverbindingen goed gemaakt zijn en dat er geen pins verbogen worden.

Een eerste programma maken

Start vanuit de MS-DOS-invoerregel het geïntegreerde ontwikkelingsoppervlak CCE.EXE. Dit programma is voorzien van een standaard-oppervlak met menu en muisbediening en maakt het invoeren van BASIC-broncode (editor), het vertalen (compiler) en het laden (lader) van de C-Control codes naar de stuurcomputer mogelijk.

Open een nieuw editorvenster via het menu "{Bestand/ nieuw". Sla het nog lege programma op onder de naam TEST1.BAS (menu "Bestand / opslaan als").

In het editorvenster voert u daarna de volgende tekst in:

```
#nogmaals  
PRINT "Hallo!"  
PAUSE 50  
GOTO nogmaals
```

Sla het programma dat nu klaar is op (menu "Bestand / opslaan").

Kies dan in het menu "Ontwikkeling" het bevel "BASIC- compiler". Het programma wordt nu vertaald (gecompileerd). Let daarbij op het beeldschermvenster "meldingen". Daar verschijnen de meldingen van de compiler over verloop en succes van de vertaling. Als er fouten in de BASIC-brontekst zitten, worden deze in het meldingsvenster opgesomd. In ons geval zouden er geen fouten mogen optreden; gebeurt dat toch, overtuig u er dan nogmaals van dat uw invoer correct is.

Het voorbeeldprogramma overdragen naar de programmeeradapter

Stel vervolgens via het "optiemenu" in, via welke seriële poort (COM1, COM2...) het C-control systeem verbonden is met de PC.

Als het meldingsvenster nog actief is (zie boven), activeert u het editorvenster opnieuw (door met de muis op het editorvenster te klikken of via toets F6).

Schakel nu de stroomvoorziening op de programmeeradapter in.

Kies nu in het menu "Ontwikkeling" het bevel "Lader". De door de compiler geproduceerde codes worden nu overgedragen naar de C-control

stuurcomputer. Meldingen over succes of fouten bij de overdracht verschijnen op het beeldscherm van het laderprogramma.

Als er een foutmelding verschijnt, controleer dan de interface-instelling, overtuig u er nogmaals van dat de "Auto Start" jumper er niet opgestoken is, dat geen ander programma de gekozen interface gebruikt of heeft gebruikt. Schakel de stroomvoorziening van de adapter eventueel meerdere keren uit en weer in. In zeldzame gevallen kan het ook nodig zijn de PC opnieuw te starten.

Als de overdracht zonder fouten heeft plaatsgevonden, bent u er nu aan toe het voorbeeldprogramma uit te voeren.

Het voorbeeldprogramma uitvoeren

Schakel de stroomvoorziening van de programmeeradapter uit. Trek de stuurcomputer van de adapter en plaats de "Auto Start" jumper.

Aansluitend steekt u de stuurcomputer weer op de programmeeradapter. De programmeeradapter functioneert nu als eenvoudige hardwarebasis voor de voorbeeldapplicatie.

Roep in het menu "Ontwikkeling" het bevel "miniterminal" op. Daardoor wordt een extern programma gestart, dat de verdere communicatie met het C-Control systeem overneemt.

Na een korte copyrightmelding verschijnt een zwart beeldscherm. Schakel nu de stroomvoorziening van de programmeeradapter in. Vanwege de opgestoken "Auto Start" jumper begint het systeem onmiddellijk met het uitvoeren van het programma.

Wat doet het testprogramma? Het PRINT- commando zendt de tekst "Hallo!" via de seriële interface naar de PC. Daar wordt de tekst door de miniterminal ontvangen en getoond. Dan volgt er een kleine pauze in het testprogramma en begint aansluitend weer van voren af aan. Het testprogramma bevindt zich in een eindeloze lus en kan alleen door het uitschakelen of reset gestopt worden. De meeste C-Control toepassingen werken volgens dit principe.

Als u genoeg "Hallo!" gelezen heeft, schakelt u de stroomvoorziening op de programmeeradapter uit. De miniterminal wordt uitgeschakeld d.m.v. de ESC-toets.

U kunt nu het testprogramma aanpassen en uw experimentele schakelingen opbouwen, om daarna nog meer ervaringen met C-Control op te doen. Vergeet niet, voor u een nieuwe programmering maakt, de "Auto Start" jumper te verwijderen.

Aansluiten van externe modules

Aan de vier kanten van de stuurcomputer zijn alle bruikbare poorten alsmede een paar systeemsignalen van de stuurcomputer op stiftstrips uitgevoerd. Het bezettingsplan van de stiftstrips vindt u in het hoofdstuk "Opbouw en functioneren", een eindje terug in deze handleiding. Aan het eind van deze handleiding vindt u een paar schakelingvoorbeelden.

Schakeling van de digitale poorten (pins PORT1 tot PORT16)

Gebruik van een digitale port als ingang

Digitale ingangen worden gebruikt voor het opvragen van schakeltoestanden.

Als een digitale port als ingang gebruikt wordt, heeft deze in ongeschakelde toestand een ongedefinieerd level. Door het schakelen van de port met een pullup- of pulldown-weerstand kan echter een bepaald level ingevoerd worden.

Gebruik van een digitale port als uitgang

Als een digitale port als uitgang gebruikt wordt, kunnen daar de volgende IC's, transistors of Low-Current-lichtdiodes direct op werken.

De maximaal toelaatbare laststroom bedraagt 10 mA per port. In elk geval moet een toereikende stroombegrenzing, bijvoorbeeld door een weerstand, gegarandeerd worden, omdat het anders tot vernieling van de microcontroller kan leiden!

Binnen de microcontroller vindt de interne schakeling van een digitale port als in- of uitgang plaats bij het de eerste keer uitvoeren van het toepassingsprogramma. Na het inschakelen van de voedingsspanning of na een reset gedragen alle digitale ports zich eerst elektrisch hoogohmig en gebruiken geen stroom.

Schakeling van de analoge poorten (pins AD1 tot AD8)

Aan de A/D-ports kunnen positieve analoge spanningen in het bereik van 0...5 Volt - bijvoorbeeld uitgangsspanningen van sensoren - toegevoerd worden. De aangelegde spanningen worden door de interne 8-bit-A/D-omvormer van de stuurcomputer omgevormd tot een waardebereik van 0 tot 255.

Een aangelegde spanning mag nooit groter zijn dan 5 Volt, omdat dat anders tot vernieling van de interne A/D-omvormer kan leiden. Als extra bescherming voor de A/D-omvormer verdienen bovendien 10k weerstanden in serie met de A/D-ports aanbeveling.

Aanleggen van de referentiespanning (Uref)

Voor de A/D-ingangen gebruikt kunnen worden, moet een referentiespanning met de referentiespanningspin (Uref) van de stuurcomputer verbonden worden. De aangelegde spanningswaarde geldt als bovengrens van het meetbereik van de A/D-omvorming en komt overeen met de omvormingswaarde 255 (\$FF hexadecimaal).

De gunstigste waarde voor de referentie hangt af van het uitgangsspanningsbereik van de toegepaste sensoren op de A/D-ingangen. Meestal kan de voedingsspanning direct als referentie gebruikt worden (jumper "interne referentie"). De referentiespanningswaarde mag echter nooit groter zijn dan 5 Volt!

Als referentie voor het onderste gedeelte van het meetbereik van de A/D-omvorming dient steeds het groundpotentiaal (massa, "minus") van de voedingsspanning.

Omvorming van de D/A-uitgangen (DA1 en DA2)

De twee 8-bit-D/A-omvormers werken volgens het principe van de pulsbreedtemodulatie. In een tijdsperiode (modulatie-interval), die uit 256 deelperioden bestaat, wordt een D/A-uitgang voor de duur van zoveel

deelperioden high-gepulst die overeenkomt met de 8-bit-waarde die bedoeld is voor uitvoer. De duur van een deelperiode bedraagt 2(s, die van de totale modulatie-interval 512 (s (1953 Hz).

Voor de demodulatie, dus de omvorming naar een echt analogoog signaal is meestal een simpel RC-element voldoende. Let daarbij echter op de restrimpel en de haalbare maximumwaarde van het uitgangssignaal. Beiden zijn afhankelijk van de last, die na het RC-element volgt.

Aansluiting van een DCF77-actieve antenne (Pin DCF/FREQ1)

Het uitgangssignaal van een DCF-actieve antenne (low-active) wordt aan pin DCF/FREQ1 toegevoerd. De DCF-ingang evalueert de signalen en neemt, als de ontvangen data geldig zijn, de tijd en datum over in de interne real-time klok.

Als alternatief voor de tijdsontvangst kunnen met de pin DCF/FREQ1 frequenties van CMOS/TTL-compatible rechthoeksignalen tot ca. 5000 Hz (bij een resolutie van 1Hz en een fout van max. 1%) gemeten worden.

Frequentiemeting op pin FREQ2

Via pin FREQ2 kunnen frequenties van CMOS/TTL-compatible rechthoeksignalen tot ca. 32000 Hz gemeten worden.

Geluidsafgifte op de BEEP-pin

Op de BEEP-pin kunnen per BEEP-commando (zie onder) rechthoeksignalen (0/5V) afgegeven worden, die b.v. door het aansluiten van een piëzo-elektrische geluidomvormer (geluidomvormer zonder interne elektronica tussen BEEP en GND) als geluiden hoorbaar gemaakt worden.

Om luidere geluidssignalen te krijgen kan een NF-versterkermodule aangesloten worden op de BEEP via een spanningsdeler (500k...1M-potmeter) en een koppelcondensator (10µ elko) en daarop een passende luidspreker.

Er mag in geen geval een laagohmige luidspreker direct met de BEEP-pin verbonden worden, omdat dit tot vernieling van de stuurcomputer kan leiden!

De interrupt- ingang IRQ

De IRQ-pin is op de stuurcomputer met een 10k pullup-weerstand high getrokken. Als bij het uitvoeren van een BASIC-programma een low-flank op de IRQ-pin herkend wordt, dan vertakt de verwerking van het programma voor het volgende BASIC-commando naar een door de gebruiker gedefinieerde plek (zie INTERRUPT- commando).

De RESET- ingang

De RESET-pin is op de stuurcomputer met een 10k pullup-weerstand high getrokken. Als op de RESET-pin een low level herkend wordt, dan worden alle activiteiten van de stuurcomputer direct onderbroken en alle ports en interne geheugens worden teruggezet (= 0).

De START- ingang

De START-pin is op de stuurcomputer met een 10k pullup-weerstand high getrokken. De bewerking van een in de stuurcomputer geladen programma na het inschakelen van de voedingsspanning of RESET begint, als de START- ingang low getrokken wordt.

Let er op, dat een geplaatste "Auto Start"- jumper de START- ingang op low level houdt.

Gebruik van de seriële interface (pins RXD en TXD)

Zender TXD

Op de TXD- pin wordt het digitale signaal van de zender van de seriële interface afgegeven. De levels zijn CMOS/TTL- compatible. Voor het aansluiten op een PC moet er een niveauomvorming plaatsvinden, b.v. met de programmeeradapter of met een separate schakeling met een MAX232-IC.

Ontvanger RXD

Op de RXD- pin kunnen seriële data met CMOS/TTL level ingevoerd worden. Deze signalen worden door de ontvanger van de seriële interface geëvalueerd.

I²C- bus interface (pins SDA en SCL)

Op de pins SDA en SCL is de I²C-bus geplaatst, die de microprocessor van de stuurcomputer met de seriële EEPROM verbindt. Deze beide pins zijn bedoeld voor latere uitbreidingen en dienen niet voor schakelingen gebruikt te worden!

Systeembronnen

Onder het begrip "systeembronnen" worden hier alle interne functie-eenheden samengevat, die niet onmiddellijk uit de eigenschappen van de microcontroller afgeleid worden, maar door het op de chip maskergeprogrammeerde functioneringssysteem ter beschikking gesteld worden. Hoe deze systeembronnen in het BASIC- programma aangesproken worden, wordt verderop in het commando-overzicht beschreven.

Timer

Op de achtergrond van het systeem loopt een 16-Bit-timer met een impuls van 20 milliseconden, waarvan de waarde uitgelezen en voor het maken van tijdreferenties in het BASIC- programma gebruikt kan worden.

Real-time -klok

De via DCF77 ontvangen tijd- en datum informatie wordt door het systeem naar zeven interne geheugencellen (jaar, maand, dag, weekdag, uur, minuut, seconde) overgebracht en tot aan de volgende synchronisatie in porties van 20 milliseconden verhoogd. De precisie van de real-time -klok tussen de synchronisatietijdstippen wordt bepaald door de afwijking van de 4MHz- kwarts van de normale frequentie van maximaal 0,1 promille, afhankelijk van strooiingen in de serieproductie en van de temperatuur. Dit komt overeen met een afwijking van maximaal 0,36 seconden per uur.

Na het inschakelen van de voedingsspanning en na een reset start de klok met 01.01.97, 00:00:00 uur.

De interne geheugencellen voor datum en tijd kunnen vanuit het BASIC- programma gelezen en beschreven worden. Door het beschrijven van de geheugencellen kan de klok dus ook zonder DCF77- ontvangst ingesteld worden. Voor programmatests of bij geringe aanspraak op de precisie kan zo afgezien worden van de DCF77- antenne.

Userbytes

De microcontroller MC68HC05B6 beschikt over in totaal 240 bytes RAM. De C-Control stuurcomputer bezet daarvan het grootste gedeelte voor bedrijfssysteemfuncties (stack, timer, klok, DCF77- vensterbuffer, interface buffer, tussengeheugen voor berekeningen enz.). Er staan de gebruiker 24 bytes ter beschikking voor gebruik in BASIC- programma's. Het gebruik van deze userbytes wordt in het hoofdstuk over het DEFINE- commando verderop beschreven.

Programmeren met CCBASIC

CCBASIC is het BASIC- dialect, dat wordt gebruikt voor het programmeren van de C-Control/*BASIC* stuurcomputer. De syntax komt ongeveer overeen met die van het standaard BASIC. Bij sommige commando's zijn er afwijkingen of uitbreidingen, die speciaal toegesneden zijn op de hardware van de stuurcomputer.

Wat is een programma?

Een programma is de beschrijving van een informatieverwerkingsproces. In de loop van een dergelijk proces wordt uit een veelheid van variabele of constante ingangswaarden een hoeveelheid uitgangswaarden berekend. De uitgangswaarden zijn of zelf het doel van de informatie-inwinning of dienen indirect als reactie op de ingangswaarden. Naast de eigenlijke berekeningen kan een programma aanwijzingen bevatten voor toegang tot de hardware van de computer of voor sturing van de programmastroom.

Een BASIC- programma bestaat uit meerdere regels van een zogenaamde brontekst. Daarbij bevat elke regel een of meerdere reken- of besturingsaanwijzingen. Behalve uit deze aanwijzingen zelf bepaalt de volgorde ervan zeer wezenlijk de aan het begin beschreven informatieverwerking. De uitvoering van de met de aanwijzingen overeenkomende operaties door de stuurcomputer geschiedt in volgorde, dus na elkaar. Een opeenvolging van aanwijzingen met een bepaald doel noemt men ook een algoritme.

Datatypes

Data zijn de objecten van het informatieverwerkingsproces, ze vertegenwoordigen de opgeslagen informatie. De C-Control/*BASIC* stuurcomputer verwerkt uitsluitend hele numerieke data - zogenaamde "integere getallen" van 1, 8 of 16 Bit en slaat die op.

Een variabele van 8 Bit (Byte) kan alleen niet negatieve waarden van 0 tot 255 opnemen. Het waardebereik van een integere variabele van 16 bit (Word) gaat van -32768 tot +32767. Let er bij alle berekeningen op, dat de resultaten deze limietwaarden niet over- of onderschrijden, omdat dit anders tot "Overflow" leidt.

$$a = 255 + 1$$

geeft bijvoorbeeld voor a de waarde 0 en niet 256, als a slechts één bit vertegenwoordigt!

$$a = -32768 - 1$$

geeft 32767 en niet -32769, als a een Word vertegenwoordigt!

Fundamentele elementen van de programmeertaal CCBASIC

Algemeen

Elke programmaregel bevat een of meerdere aanwijzingen, die door dubbele punten : gescheiden zijn.

Regelnummers, zoals gebruikelijk in oudere BASIC- dialecten, zijn niet noodzakelijk. Als er toch regelnummers aangegeven worden, dan kunnen die als sprongdoel gebruikt worden.

```
10 ...  
GOTO 10
```

De nummers hebben daarbij geen invloed op de volgorde van de programma- operaties. Als er bijvoorbeeld in de brontekst op een met regel 200 genummerde regel een regel 100 volgt, dan wordt desondanks regel 200 voor regel 100 afgewerkt.

Als uitleg van het geschreven programma kunnen in de brontekst ook commentaren opgenomen worden, waardoor de leesbaarheid en gebruiksvriendelijkheid vergroot worden. Een commentaar in BASIC begint

steeds met een apostrof en verklaart de rest van de regel tot de niet bij het programma behorende tekst.

```
a = b + c ' ... commentaar ...
```

Markers

Markers zijn programma-elementen bestaande uit alfanumerieke tekens (A tot Z, 0 tot 9), die op een door de programmeur vastgelegde manier objecten, zoals variabelen en constanten, aanduiden. Labelnamen en de zogenaamde "gereserveerde woorden" zijn eveneens markers.

Er is geen verschil tussen hoofd- en kleine letters. Een marker begint steeds met een letter of met een liggende streep. Spaties binnen een marker zijn niet toegestaan.

Variabelen en constanten

Variabelen en constanten zijn objecten van het informatieverwerkingsproces. In CCBASIC slaan beide een numerieke waarde op. Terwijl de waarde van een constante eenmaal aangegeven wordt en dan onveranderd blijft, kan de waarde van een variabele willekeurig vaak veranderen.

Constanten kunnen in CCBASIC in decimale, hexadecimale en binaire vorm aangegeven worden. De syntax voor hexadecimale en binaire getallen wordt hier met het voorbeeld van het getal 46 (decimaal) getoond:

```
&H2E  
&B10110
```

Bovendien kunnen per DEFINE- regel (zie onder) symbolische constanten afgesproken worden.

Toegang tot variabelen wordt steeds gekregen d.m.v. de desbetreffende marker. Deze marker moet voor de eerste keer dat de variabele gebruikt wordt in het programma in een DEFINE- regel gedefinieerd worden.

Label

Labels markeren bepaalde punten in de volgorde van de programma-operaties. Labels zijn doelen van een sprongoperatie binnen een algoritme.

In CCBASIC staan labels aan het begin van een regel en beginnen steeds met een dubbel kruis, dan volgt - zonder spatie - de marker van het label.

Het voorbeeld laat de definitie van label "label 1" en het gebruik in een sprongcommando zien:

```
#label1 ...  
GOTO label1
```

Termen

Een term geeft direct (als variabele of als constante) of door berekening een bepaalde waarde. Termen zijn onderdelen van aanwijzingen en staan bijvoorbeeld bij de toewijzing van een waarde aan een variabele rechts van de toewijzingsmarker "=". Termen worden gevormd door combinaties van opdrachtelelementen en bewerkingstekens.

```
100  
c  
a + b  
(ABS(x) - 13) * 10
```

Opdrachtelelementen en bewerkingstekens

Een opdrachtelelement is in principe of een constante, een variabele of een functieoproep, maar kan zelf echter ook weer een uit opdrachtelelementen en bewerkingstekens samengestelde term zijn.

Bewerkingstekens duiden rekenbewerkingen aan, die met de omstaande opdrachtelelementen uitgevoerd moeten worden. Daarbij bestaat een gedefinieerde rangorde van de bewerkingselementen (zie commando- beschrijving), die de volgorde van de berekeningen bepaalt.

Functies

Een functie voert een gedefinieerde operatie - bijvoorbeeld een berekening - uit en levert door het oproepen ervan een resultaat. De meeste functies verwachten een of meerdere argumenten, die in haakjes "()" na de functiemarker gegeven worden en door komma's gescheiden zijn. Sommige functies worden zonder argument opgeroepen. In dit geval worden er geen haakjes gebruikt.

ABS(x)
MAX(a,b)
RAND
EOF

In CCBASIC zijn alle ondersteunde functies vooraf gedefinieerd. De markers ervan horen bij de gereserveerde woorden. De formulering van per toepassing gedefinieerde functies is in CCBASIC niet voorzien.

Toewijzingen

De toewijzing is de eenvoudigste vorm van een programmeeraanwijzing. Na de marker van een variabele, waaraan een waarde toegewezen moet worden, volgt het toewijzingsteken "=" en dan een term, die de toe te wijzen waarde bepaalt. Een toewijzing komt daardoor overeen met een eenvoudige wiskundige formule.

```
a = 10
b = x - y
c = SQR (a*a + b*b)
```

Commando's

Naast de eenvoudige toewijzingen zijn commando's aanwijzingen voor het uitvoeren van programmeeroperaties door de C-Control/*BASIC* stuurcomputer. Commando's beginnen steeds met een gereserveerd woord. Sommige commando's verwachten een of meerdere parameters voor een exacte specificatie van de uit te voeren programmaoperatie. Deze parameters worden na de commandomarker en een spatie opgevoerd en daarbij door komma's gescheiden (met uitzondering van PRINT, zie commando-overzicht). In tegenstelling tot de argumenten bij het oproepen van een functie staan de commandoparameters niet tussen ronde haakjes!

```
RANDOMIZE
PAUSE 100
BEEP 440, 50, 50
```

Aanwijzingen voor het sturen van de programmastroom

Deze aanwijzingen maken het mogelijk de volgorde van de op zich streng in volgorde afgewerkte programmaoperaties te sturen en aan te passen aan ingangswaarden van het informatieverwerkingsproces. Ze bieden een grote flexibiliteit bij de algoritmeformulering en zijn voor de oplossing van vele toepassingstechnische problemen zelfs een basisvoorwaarde.

Aanwijzingen voor het sturen van de programmastroom bestaan uit een of meer gereserveerde woorden en verlangen op een steeds speciale manier eventueel nadere aanduidingen.

```
GOTOlabel1
IF a > b THEN GOSUB label 2

FOR i = 0 TO 10 STEP 2
...
NEXT
```

Compileraanwijzingen

Naast de programma-aanwijzingen bevat een CCBASIC- brontekst compileraanwijzingen, die bijvoorbeeld bedoeld zijn voor het aanleggen van datablokken (tabellen) of voor de definitie van variabelen en constanten.

Voor compileraanwijzingen geldt de dubbele puntregel voor het scheiden van meerdere aanwijzingen in een regel niet. Er mag steeds maar één compileraanwijzing in één regel staan.

De DEFINE- aanwijzing

De **DEFINE**-aanwijzing is een compiler- aanwijzing.

Definitie van symbolische constanten

Het is een goede programmeerstijl om in plaats van "magische" getallen in het programma

```
IF x > 1234 THEN GOTO alarm
```

beter symbolische constanten te gebruiken. Door het uitgeven van significante markers voor constanten wordt de leesbaarheid van de brontekst beter. Als alle constanten globaal gedefinieerd worden, kan er ook makkelijker onderhoud aan een programma uitgevoerd worden. Dit geldt vooral, als een en dezelfde constante meermaals in het programma nodig is.

De definitie van een symbolische constante geschiedt als volgt:

```
DEFINE gemarkeerde waarde
```

Daarbij is de waarde of een decimaal, hexadecimaal of een binair getal.

Zo zou het hiervoor genoemde voorbeeld beter kunnen luiden:

```
DEFINE limit 1234  
...  
IF x > limit THEN GOTO alarm
```

Definitie van variabelen

De C-Control/BASIC stuurcomputer stelt de gebruiker 24 byte-geheugenregels van het interne geheugen ter beschikking. In dit geheugenbereik worden alle variabelen van een BASIC- programma opgeslagen. De 24 bytes kunnen naar behoefte ook bitsgewijs of als 16-bit integer (Word) gebruikt worden.

In tegenstelling tot standaard- BASIC moeten in CCBASIC alle door het programma gebruikte variabelen voordat ze voor het eerst gebruikt worden gedefinieerd worden. Daarbij dient het datatype gespecificeerd te worden

(bit, byte of Word) en kan (voor bits moet!) een regelnummer van het geheugen aangegeven worden. De gebruiker moet er zelf op letten dat er bij het uitgeven van geheugenplaatsen geen overlappingen ontstaan, omdat dat anders tot tegengesteld overschrijven van de variabelen zou kunnen leiden. Zo beleggen bijvoorbeeld bit[18], byte[2] en Word[1] elk een deel van cel 2 van het geheugenbereik.

- Definitie van een bitvariabele:

```
DEFINE marker BIT[nr]
```

Daarbij zijn voor nr waarden van 1 tot 192 (24 bytes met elk 8 bits) toegestaan.

- Definitie van een bytevariabele met regelnummer:

```
DEFINE marker BYTE[nr]
```

Daarbij zijn voor nr waarden van 1 tot 24 (24 bytes) toegestaan.

- Definitie van een integrale variabele met celnummer:

```
DEFINE marker WORD[nr]
```

Daarbij zijn voor nr waarden van 1 tot 12 (een Word belegt 2 bytes) toegestaan.

Als bij byte- en woorddefinities de regelaanduiding [nr] weggelaten wordt, neemt de compiler de verdeling in het geheugenbereik over. Let er dan op, dat niet afwisselend bytes en Words gedefinieerd worden. De volgende aanwijzingen:

```
DEFINE a BYTE  
DEFINE b WORD  
DEFINE c BYTE  
DEFINE d WORD
```

leiden tot twee ongebruikte (weggegeven kostbare) bytes, tussen a en b

alsmede tussen c en d, omdat Words principieel aan de bytes 1, 3, 5, 7.... enz. van de 24 bytes gericht worden.

Het zou beter zijn om te schrijven

```
DEFINE b WORD  
DEFINE d WORD  
DEFINE a BYTE  
DEFINE c BYTE
```

De automatische opdeling van de variabelen op het geheugen door de compiler begint bij regelnummer 1. Het bovenstaande (betere) voorbeeld belegt 6 bytes. Bij het definiëren van nog meer bits, bytes en Words met aanduiding van de regelnummer dient u steeds op ongewenste overlapping te letten.

Een reeds gedefinieerde variabelenmarker mag niet een tweede keer gedefinieerd worden.

Definitie van digital ports

In CCBASIC worden ports net zo behandeld als variabelen. Ook hier moet iedere gebruikte port vooraf gedefinieerd zijn.

- Definitie van een van de 16 digitale ports:

```
DEFINE marker PORT[nr]
```

Daarbij zijn voor nr waarden van 1 tot 16 toegestaan.

- Definitie van een 8 bit brede port:

```
DEFINE marker BYTEPORT[nr]
```

Daarbij zijn voor nr alleen de waarden 1 (ports 1 tot 8 als byteport) en 2 (ports 9 tot 16 toegestaan).

- Definitie van een marker voor de gemeenschappelijke toegang tot alle 16 digital ports als een 16 bit port:

```
DEFINE marker WORDPORT[nr]
```

Voor nr is alleen de waarde 1 toegestaan.

Definitie van analoge ports

- Definitie van een van de 8 A/D- ports:

```
DEFINE marker AD[nr]
```

Daarbij zijn voor nr alleen waarden van 1 tot 8 toegestaan.

- Definitie van een van de 2 D/A- ports:

```
DEFINE marker DA[nr]
```

Daarbij zijn alleen de waarden 1 en 2 toegestaan.

Commando-overzicht

Dit hoofdstuk geeft een compleet overzicht van de CCBASIC bewerkingstekens, functies en aanwijzingen.

Mathematische en logische bewerkingstekens

- Basisrekensoorten: +-*/
- De module-operator **MOD** levert de rest van een integerverdeling.

```
a = 10 MOD 3
```

levert bijvoorbeeld voor a de waarde 1.

- Vergelijkingstekens: > (groter dan), < (kleiner dan), >= (groter dan of gelijk aan), <= (kleiner dan of gelijk aan), = (gelijk aan), <> (niet gelijk).

Het resultaat van een vergelijkingsoperatie is of -1 resp. 255 (vergelijking waar) of 0 (niet juist).

$$a = 10 < 3$$

geeft bijvoorbeeld voor a de waarde 0.

- Logische bewerkingstekens: **NOT** (negatie) **AND** (en- koppeling), **NAND** (en- koppeling met aansluitende ontkenning), **OR** (of- koppeling), **NOR** (of- koppeling met aansluitende ontkenning), **XOR** (exclusieve of- koppeling).

De logische bewerkingstekens kunnen behalve voor de formulering van voorwaarden (meestal in samenhang met vergelijkingsoperaties) ook voor binaire byte- of Wordmanipulaties gebruikt worden.

- Schuiftkens: **SHL** (naar links schuiven), **SHR** (naar rechts schuiven) worden gebruikt voor het bitsgewijs verschuiven van bitpatronen in byte- of wordvariabelen. Links van de operator staat de te verschuiven waarde, rechts het getal om hoeveel bits er verschoven moet worden. Bij het schuiven naar links komt elke aparte verschuiving overeen met een vermenigvuldiging met 2, bij het schuiven naar rechts met een deling door 2.

$$a = 10 \text{ SHL } 3$$

komt dus overeen met

$$a = 10 * 2 * 2 * 2$$

en levert bijvoorbeeld voor a de waarde 80.

Wiskundige functies en commando's

De argumenten x en y, afhankelijk van functie of commando, zijn steeds termen (voor definitie zie boven).

- De wortelfunctie **SQR(x)** levert een benadering voor de vierkantswortel uit argument x. Daarbij worden de cijfers achter de komma afgekapt.
- De signumfunctie **SGN(x)** geeft 1, als de waarde van het argument x > 0, en -1, als de waarde < 0 is. Voor x = 0 is ook het resultaat van de SGN-functie gelijk aan 0.

- De maximumfunctie **MAX(x,y)** geeft x, als x > y, anders y.
- De minimumfunctie **MIN(x,y)** geeft x, als x < y, anders y.
- Het commando **RANDOMIZE x** initieert de interne pseudo-toevalsgenerator van de stuurcomputer met de waarde van x. Een en dezelfde initiëeringswaarde leidt steeds tot een identieke volgorde van getallen. De speciale vorm **RANDOMIZE TIMER** laadt de waarde van de vrijkomende timer in de generator.
- De toevalsfunctie **RAND** levert de volgende integer-toevalswaarde van de pseudo- toevalsgenerator. De toevalsgetallen worden volgens de vermenigvuldigende procedure met aansluitende modulusverdeling (zie een goed wiskundeboek) uit de betreffende voorafgaande waarde berekend.

Rangvolgorde van bewerkingstekens en functieoproepen

Bij de berekening van termen met bewerkingstekens en functies is de volgorde ervan van beslissende betekenis. Deelprotocollen met bewerkingstekens met een hoge rang worden berekend voor de protocollen met een lagere rang (vergelijk de rekenregel: vermenigvuldigen gaat voor delen). Bij bewerkingstekens van gelijke rang geschiedt de berekening van links naar rechts.

Net als in de wiskunde kan echter door het plaatsen van haakjes extra invloed op de volgorde van de berekening verkregen worden. CCBASIC ondersteunt maximaal 3 niveau's met haakjes.

Voor de overzichtelijkheid van een programma dienen echter "wilde" haakjes vermeden te worden en moeten complexe berekeningen onderverdeeld worden over meerdere BASIC- regels.

De volgende lijst toont de CCBASIC- operatorvolgorde:

Rang	Bewerkingstekens
9	()
8	Functieoproep
7	Negatief voorteken
6	* / MOD SHL SHR
5	+ -
4	< >= < <= = <>
3	NOT
2	AND NAND
1	OR NOR XOR

Aanwijzingen voor het sturen van de programmastroom

- Lus
FOR variabele = begin **TO** eind **STEP** stapsgewijs
 ...
NEXT

De FOR- lus voert de aanwijzingen tot aan NEXT zo lang uit, tot de waarde van de variabele gelijk is aan die van het einde van de term. Voor de eerste doorloop wordt de waarde van het begin van de term berekend en aan de lusvariabelen toegewezen. Bij iedere doorloop wordt de waarde van de stapsgewijze term opgeteld bij de lusvariabele. In de vorm

```
FOR variabele = begin TO einde
...
NEXT
```

bedraagt de stapbreedte constant 1.

De waarden van de eindterm en de stapbreedte term worden bij elke lusdoorloop opnieuw berekend. Dit maakt een uitgebreide controle van het programmaverloop mogelijk.

FOR- lussen kunnen in elkaar geschoven worden. De schuifdiepte wordt alleen beperkt door de voor de lusvariabelen benodigde geheugenruimte.

```
FOR v1 = begin1 TO einde1
  FOR v2 = begin2 TO einde2
    FOR v3 = begin3 TO einde3
      ...
    NEXT
  NEXT
NEXT
```

Iedere FOR- lus mag in het verloop van het programma alleen via haar eigen NEXT- aanwijzing lopen. De volgende brontekst kan weliswaar gecompileerd en in de stuurcomputer geladen worden, maar zal echter niet zoals misschien verwacht functioneren:

```
FOR v1 = begin1 TO einde1
...
  GOTO anothernext
...
NEXT

FOR v2 = begin2 TO einde2
...
#anothernext
NEXT
```

Let bovendien op het waardebereik van de lusvariabele en de eindterm!
 DEFINE v BYTE

```
FOR v = 1 TO 1000
...
NEXT
```

wordt een doorlopende lus, omdat v als bytevariabele nooit de waarde van 1000 kan bereiken, maar al na 255 weer doorrolt naar 0.

- Beperkte uitvoering
IF voorwaarde **THEN** aanwijzing1

of

IF voorwaarde **THEN** aanwijzing1 **ELSE** aanwijzing2

De IF...THEN...ELSE- constructie maakt de aanpassing mogelijk van de programmastroom aan voorwaarden aan de looptijd van het programma. Als voorwaarde dient een willekeurige term ingezet te worden. Als deze berekening een waarde oplevert die niet gelijk is aan 0, dan geldt de voorwaarde als voldaan, en aanwijzing 1 wordt uitgevoerd. Als er bovendien een ELSE en een tweede aanwijzing aangegeven worden, dan wordt deze aanwijzing alternatief uitgevoerd, als de berekende term een waarde gelijk aan nul geeft.

De totale IF...THEN...ELSE- constructie moet in een brontekstregel staan. Aanwijzingsblokken (meerdere aanwijzingen) na THEN en ELSE zijn niet toegestaan.

- Sprongaanwijzing

GOTO label

Met de GOTO- aanwijzing kan de stuurcomputer genoopt worden het bewerken van een programma op een bepaalde plaats voort te zetten. Als doel van de sprong wordt een labelmarker aangegeven. Het sprongdoel kan zich voor of na de GOTO- aanwijzing in de brontekst bevinden.

- Oproep en terugkeer uit een subroutine

De oproep van een subroutine geschiedt met de aanwijzing

GOSUB label

Daarbij is label het beginpunt van de subroutine.

In de zogenaamde subroutines zijn programmastappen samengevat, die meerdere keren in het verloop van de programmabewerking nodig zijn. Een subroutine begint steeds met een label, bevat dan een of meer aanwijzingen en afsluitend een

RETURN

Na de RETURN wordt de programmabewerking voortgezet met de aanwijzing na de GOSUB. De programmabewerking mag zonder een voorafgaand GOSUB nooit bij een RETURN- aanwijzing komen.

De maximaal toelaatbare inschuifdiepte bij het oproepen van subroutines uit subroutines is vier.

```
#hoofdprogramma
  GOSUB sub1
  ...
```

```
#sub1
  GOSUB sub2
  ...
RETURN
```

```
#sub2
  GOSUB sub3
  ...
RETURN
```

```
#sub3
  GOSUB sub4
  ...
RETURN
```

```
#sub4
  ...
RETURN
```

- Waardegestuurde programmavertakking

ON variabele **GOTO** label0, label1, labeln

of

ON variabele **GOSUB** label0, label1, labeln

Afhankelijk van de waarde van de variabele van de selector vindt een vertakking van het programma of een oproep van een subroutine naar de getoonde inspringpunten plaats. Als de waarde 0 is, dan wordt afgetakt naar

label0, bij een waarde gelijk aan 1 naar label1 enz. Als de waarde van de variabele negatief is of groter dan het aantal opgevoerde sprongdoelen, dan wordt de bewerking van het programma voortgezet zonder vertakking.

- Programmavertakking na een interruptsignaal op pin IRQ

Als er tijdens het afwerken van een programma een interruptsignaal (lage flank) gedetecteerd wordt op de IRQ- pin, dan wordt de op dit moment bewerkte BASIC- aanwijzing op het eerstvolgende mogelijke tijdstip onderbroken en de verwerking van het programma wordt voortgezet op de vooraf met het commando

INTERRUPT label

vastgelegde plaats. Het terugspringen naar de uitgangspositie geschiedt door het commando

RETURN INTERRUPT

In het volgende voorbeeld wordt door iedere interruptie een lichtdiode in-/uitgeschakeld:

```
DEFINE led PORT[8]
led = OFF

` vastleggen van de interruptroutine

INTERRUPT switch_it

` eindeloze lus
#loop
GOTO loop

` interruptroutine
#switch_it
    tog led
RETURN INTERRUPT
```

Als er nog meer IRQ- signalen binnenkomen, terwijl de stuurcomputer net bezig is met de bewerking van een interrupt, dan wordt er maximaal één IRQ- gebeurtenis opgeslagen en in aansluiting op RETURN INTERRUPT uitgevoerd. Alle andere signalen gaan verloren.

De interrupt- routine kan door meervoudige oproep van het INTERRUPT- commando willekeurig vaak anders gedefinieerd worden, ook tijdens de uitvoering van een interrupt- routine.

- Einde programma

END

Als de stuurcomputer in de loop van de bewerking van een programma bij de END- aanwijzing komt, wordt de bewerking van het programma beëindigd. Het systeem blijft dan in inactieve toestand. Nu kan een nieuw toepassingsprogramma overgedragen of de uitvoering via de starttoets weer gestart worden.

- Vertraging van de programmastroom

De aanwijzing

WAIT conditionterm

onderbreekt de uitvoering van het programma zo lang, tot de berekening van de conditionterm een waarde niet gelijk aan 0 geeft.

```
Define key port[9]
...
WAIT key
```

In dit voorbeeld wordt zo lang gewacht, tot er van digital port 9 een HIGH-level (= logisch 1) gelezen wordt.

Het PAUSE commando onderbreekt de programma-uitvoering voor een bepaalde tijd. De berekende waarde van de parameterterm gaat als

vermenigvuldigingsfactor met de basiseenheid van 20 milliseconden de vastlegging van de pauzetijd in.

PAUSE term

Bijvoorbeeld wordt door het commando

PAUSE 50

de uitvoering van het programma gedurende ca. $50 \cdot 20$ milliseconden = 1 seconde onderbroken. De maximale tijdafwijking van de daadwerkelijke pauze van de aangegeven waarde bedraagt daarbij door het principe bepaald +/- 20 milliseconden.

Communicatie via de seriële interface

- Data- output

De data- output geschiedt als tekst via de seriële interface van de C-control/*BASIC* stuurcomputer. Als via de programmeeradapter en een interfacekabel bijvoorbeeld een PC met een terminalprogramma is aangesloten, dan kunnen de uitgegeven data daar getoond worden.

PRINT term

geeft het resultaat van de berekening vanuit term.

PRINT "text"

brengt de tussen aanhalingstekens staande tekst over.

In beide gevallen wordt er aan de overdracht een regeldoorschuipteken gehangen, dat voor het terminalprogramma aanleiding is, de volgende uitvoer op de volgende beeldschermregel te tonen. Het doorschuiven van de regel kan onderdrukt worden, als aan het PRINT- commando na de parameter (term of "text" een puntkomma toegevoegd wordt.

PRINT term;

of

PRINT "text";

CCBASIC ondersteunt bovendien meerdere uitvoeren met een PRINT-commando, waarbij de aparte parameters door komma of puntkomma gescheiden worden. Een komma voegt in de uitvoer een tab- teken in, dat in overeenstemming met de instellingen in het terminalprogramma als een aantal spaties op het beeldscherm verschijnt. Als er twee uitvoeren zonder spatie op elkaar moeten volgen, dient u deze twee in het PRINT- commando door een puntkomma te scheiden.

```
PRINT "a= ", a
PRINT "a= "; a
```

Een enkel PRINT- commando zonder parameters laat slechts één regel doorschuiven.

PRINT

- Data-input

Met het commando

INPUT variabele

kan een integerwaarde door de seriële interface gelezen en voor de aansluitende bewerking in een variabele opgeslagen worden.

De waarde wordt b.v. in een terminalprogramma op een PC ingevoerd en na het indrukken van de ENTER- toets per interfacekabel en programmeeradapter doorgegeven aan de C-control/*BASIC* stuurcomputer.

Het INPUT- commando wacht zo lang, tot er een complete dataoverdracht van de terminal ontvangen is. Als het INPUT- commando opgeroepen wordt, zonder dat er een dataoverdracht van de terminal volgt, zal het programma eindeloos op deze plek blijven staan! Hier helpt dan alleen nog maar de reset- toets en het aansluitend opnieuw starten van de C-control/*BASIC* stuurcomputer.

- Bytesgewijze communicatie via de seriële interface

Terwijl PRINT en INPUT korte ketens van tekens voor het weergeven van een numerieke waarde zenden respectievelijk verwachten, kan het wenselijk zijn aparte bytes serieel over te brengen. Daarvoor biedt CCBASIC de commando's PUT en GET.

PUT term

zendt de berekende waarde van een term. Indien nodig wordt het resultaat eerst tot het byte-waardebereik (0...2550 gereduceerd).

GET variabele

wacht op een serieel ontvangen byte en slaat de waarde dan op in de aangegeven variabelen.

- Andere interfacecommando's en -functies

Zoals beschreven wachten INPUT en GET onder bepaalde omstandigheden eindeloos op de ontvangst van seriële data. Als het "ophangen" van het programma op deze manier verhinderd moet worden, dan kan voor zowel INPUT als GET door het oproepen van de statusfunctie **RXD** bepaald worden, of er ontvangen data ter beschikking staan. De functie levert in dit geval de waarde -1. Als de interfacebuffer leeg is, dan is het functieresultaat gelijk aan 0.

```
...
if RXD then GET thebyte
...
```

De vooraf ingestelde overdrachtsnelheid van de seriële interface bedraagt voor zender en ontvanger 9600 bit per seconde (baud). Met het BAUD commando kunnen echter andere snelheden ingesteld worden. CCBASIC bevat daarvoor een paar vooraf gedefinieerde constanten: **R1200**, **R2400**, **R4800**, **R9600** voor de snelheden van 1200 tot 9600 bits per seconde.

BAUD R2400

schakelt bijvoorbeeld zender en ontvanger over op de snelheid van 2400 bit per seconde. In principe zijn ook andere dan de vooraf gedefinieerde snelheden, ook voor zender en ontvanger verschillend, mogelijk. De overdrachtsnelheden van de seriële interface worden door deling uit een interne puls van de microprocessor van de C-control/BASIC stuurcomputer afgeleid. De aan het BAUD commando over te geven bytewaarde bevat de noodzakelijke deelwaarden Nxx.

b7	b6	b5	b4	b3	b2	b1	b0
NP1	NP0	NT2	NT1	NT0	NR2	NR1	NR0

Bit 7 en 6 bevatten een voor zender en ontvanger gemeenschappelijke voordeler NP. NP kan de waarden 1, 3, 4 en 13 aannemen. De volgende tabel geeft de daarvoor noodzakelijke instellingen voor NP1 en NP0:

Voordeler	NP1	NP0
1	0	0
3	0	1
4	1	0
13	1	1

NT (bit 3 en 5) en NR (bit 0 tot 2) bepalen verdere delerwaarden, gescheiden voor zender (NT) en ontvanger (NR), volgens onderstaande codering:

Deler	NT2 resp. NR2	NT1 resp. NR1	NT2 rep. NR2
1	0	0	0
2	0	0	1
4	0	1	0
8	0	1	1
16	1	0	0
32	1	0	1
64	1	1	0
128	1	1	1

De overdrachtsnelheid van de zender wordt berekend met de formule:

$$\text{Zendsnelheid} = 125000 / (\text{NP} * \text{NT})$$

Die van de ontvanger overeenkomstig:

Ontvangstsnelheid = 125000 / (NP * NR)

De overige interfaceparameters - 8 databits, geen pariteitbit, 1 stopbit - zijn vast en kunnen niet veranderd worden.

Bestandsfuncties

De bestandsfuncties maken het opslaan van meetwaarden of andere data mogelijk of kunnen gebruikt worden voor het opslaan van informatie, die na het uitvallen van de voedingsspanning weer in de programmavariabelen ingevoerd moeten worden.

Het geheugenbereik in de EEPROM- chip na het gebruikersprogramma - meestal het grootste gedeelte - staat voor dit doel tot uw beschikking. Het geheugenbereik wordt als een bestand beheerd, waartoe u lezend of schrijvend toegang heeft, nadat het met het desbetreffende attribuut is opgeroepen. Het commando voor het oproepen van een bestand luidt als volgt:

OPEN# FOR WRITE

Of

OPEN# FOR APPEND

Of

OPEN# FOR READ

Daarbij betekent WRITE het openen voor schrijven met overschrijving van eventuele oude registraties, APPEND het openen voor schrijven met het aanhangen van de nieuwe aan de oude registraties en READ het openen voor het lezen van het opgeslagene.

Er kunnen alleen integrale waarden opgeslagen en gelezen worden. Elke waarde belegt dus 2 bytes in de EEPROM. Het schrijven en lezen geschiedt met de commando's

PRINT# term

waarbij het berekende resultaat van de term opgeslagen wordt, respectievelijk

INPUT# variabele

waarbij variabele een gedefinieerde integrale variabele van het programma aanduidt.

Het schrijven in en lezen van het bestand gebeurt strikt sequentieel. Daartoe wordt intern een bestandwijzer beheerd, die na elke toegang met 1 verhoogd wordt.

Voor iedere keer dat er geschreven wordt dient u te controleren of er nog genoeg ruimte in de EEPROM aanwezig is voor het opslaan van data. Daartoe kunt u de functie **FILEFREE** opvragen, die als resultaat de grootte van het nog vrije geheugen levert (in words). Het volgende voorbeeld laat de toepassing van de functie zien.

```
DEFINE a      WORD
DEFINE b      WORD
DEFINE c      WORD
DEFINE blocksize 3
...
IF FILEFREE >= blocksize THEN GOSUB writeblock
...
#writeblock
    PRINT# a
    PRINT# b
    PRINT# c
RETURN
```

Voor elke keer dat er gelezen wordt, dient u te controleren of er nog meer data geregistreerd zijn. De functie daarvoor luidt EOF ("End Of File"). Het resultaat is -1, als er in het bestand geen data meer aanwezig zijn, anders 0. Het opvragen van de EOF- functie moet het uitlezen van de datablocks op dezelfde manier aanpakken als bij het schrijven van de data. Bij het

bovenstaande voorbeeld zou dus

```
IF NOT EOF THEN GOSUB readblock
...
#readblock
  INPUT# a
  INPUT# b
  INPUT# c
RETURN
```

moeten horen.

Als u klaar bent met het bestand, dan dient dit direct weer gesloten te worden. Pas dan zijn de data beveiligd tegen het uitvallen van de spanning of een reset van het systeem. Het commando daarvoor luidt:

CLOSE#

en heeft geen parameters.

Poortcommando's

- het omschakelcommando TOG

In principe vindt de toegang tot de ports van de stuurcomputer op dezelfde manier plaats als voor variabelen. Om een digital port P in te schakelen, schrijft u

```
P = 1
```

en

```
P = 0
```

om de port uit te schakelen.

Om de port om te schakelen (van EIN naar AUS; AUS naar EIN), kunt u schrijven

```
P = NOT P
```

of het commando

```
TOG P
```

gebruiken. TOG staat voor het Engelse "toggle". Het TOG- commando gebuikt minder plaats in de EEPROM en wordt sneller uitgevoerd dan de klassieke NOT-P- constructie.

De portvariabele P mag bij het TOG commando alleen voor een enkele digital port staan, niet voor een byte- of wordport.

- Deactiveren van een port met **DEACT**

Zodra aan een portvariabele voor het eerst een waarde toegewezen wordt, schakelt de stuurcomputer de bijbehorende hardwarestructuren in de processorchip (transistoren) in uitgangsfunctie. Er gaat dus in overeenstemming met de aangesloten schakeling stroom van resp. naar de processor (max. 10 mA toegestaan!). Het commando

DEACT portvar

deactiveert de aangegeven port. Dat betekent dat de port in een hoogohmige toestand wordt geschakeld en in ingangsfunctie werkt.

Het DEACT- commando mag op enkele digital ports of byteports toegepast worden.

- Het **PULSE**- commando

Met het commando

PULSE portvar

wordt een puls van enige milliseconden breed naar de met portvar aangeduide port gestuurd. Dit is bijvoorbeeld nuttig voor het schakelen van extern aangesloten flankgetriggerde logische schakelcircuits. Als de port

voor de uitvoering van het PULSE- commando op low (= 0) staat, dan wordt er een high-pulse gegeven (0-1-0), anders een low-pulse (1-0-1).

De portvariabele mag bij het PULSE commando alleen voor een enkele digital port staan, niet voor een byte- of wordport.

Definitie en toepassing van datatabellen

In standaard- BASIC dienen DATA- regels voor het opslaan van constante datablokken, waar dan in volgorde toegang tot verkregen kan worden. CCBASIC ondersteunt geen DATA- regels, maar biedt echter een veel flexibeler gereedschap voor de definiëring van en toegang tot datablocks. Constante data kunnen in de vorm van tabellen opgeslagen worden. Elke tabel krijgt een marker (tablename = tabelnaam) toegewezen en kan willekeurig veel invoeren bevatten, voor zover het programmageheugen plaats heeft. Elke data-invoer (Cx) wordt als integere waarde opgeslagen en belegt dus 2 bytes. Daarbij kunnen de data direct in de brontekst opgenomen worden:

```
TABLE tablename C0 C1 C2 C3 ...  
C4 C5 ...  
... Cn  
TABEND
```

of door de CCBASIC- compiler uit een extern tekstbestand geïmporteerd worden

```
TABLE tablename "tabfilename"
```

De tabeldefinities moeten steeds aan het eind van een programma, na het END commando staan, omdat de data naadloos achter de er aan voorafgaande codebytes in de EEPROM geheugenchip opgeslagen worden. De programmabewerking mag nooit via datatabellen lopen, omdat de data anders als BASIC- commando's zouden worden geïnterpreteerd, wat beslist tot een crash van het systeem leidt.

De toegang tot de tabeldata geschiedt met het commando

```
LOOKTAB tablename, index, variabele
```

tablename geeft een geldige tabel aan, voor index kan een willekeurige term staan en de variabele geeft de geheugencel aan, waarin het resultaat opgeslagen moet worden. De berekende waarde van de indexterm mag niet negatief zijn en mag maximaal N-1 bedragen, als de geïndiceerde tabel N invoeren heeft. Als de index de waarde 0 heeft, dan wordt C0 in de aangegeven variabelen opgeslagen, voor index gelijk aan 1 C1 enzovoort. Het volgende voorbeeld geeft de inhoud van de tabel serieel aan:

```
DEFINE value WORD  
DEFINE i BYTE  
  
FOR i = 0 to 3  
    LOOKTAB mytab, i, value  
    PRINT "mytab [" ; i; "] = "; value  
NEXT  
  
END  
  
TABLE mytab 12 -20 0 1000  
TABEND
```

Op het beeldscherm van het terminalprogramma moet verschijnen

```
mytab[0] =12  
mytab[1]=-20  
mytab[2]=0  
mytab[3]=1000
```

Tabellen zijn bijzonder nuttig bij het omzetten van A/D- waarden in echte natuurkundige grootheden. Een omzettingstabel bevat dan in de regel 256 invoeren. De gemeten A/D- waarde gaat dan als tabellenindex over in de bepaling van de natuurkundige grootheid.

Toegang tot de real-time klok

Om de stand van de interne real-time klok uit te lezen en in te stellen, zijn de volgende globale variabelen gedefinieerd:

YEAR	jaar (0...99)
MONTH	maand (1...12)
DAY	dag van de maand (1...31)
DOW	weekdag (0 = zondag...6 = zaterdag)
HOUR	uur (0...24)
MINUTE	minuut (0...59)
SECOND	seconde (0...59)

Let u er a.u.b. op, dat tijdens de toegang de interne klok doorloopt. De waarde van de seconde moet daarom steeds als eerste afgelezen worden. Als deze waarde op 59 staat, dan moet na het lezen van de laatste interessante tijdinformatie (b.v. YEAR) de waarde van de seconden opnieuw gelezen en op =0 gezet worden. In dit geval dient het aflezen van de interne klok herhaald te worden, omdat een nieuwe minuut aangebroken is (als extreem geval oudjaar met het doorschakelen van alle posities in tijd en datum).

Het jaartal wordt in het C-control systeem slechts met twee cijfers opgeslagen. Als het evalueren van het jaartal in uw algoritmen een rol speelt, let dan s.v.p. op de sprong van 1999 naar 2000, die in C-Control weergegeven wordt als wisseling van 99 naar 0.

Timer

De interne 20-milliseconden-timer kan via de vooraf gedefinieerde marker **TIMER** afgelezen worden. De timer loopt vanzelf en kan niet ingesteld of teruggezet worden.

Overige commando's

- Het geven van geluidssignalen met **BEEP**

De C-control/*BASIC* stuurcomputer kan op een van zijn pins (BEEP-pin, komt overeen met processoruitgang TCMP1) geluiden als rechthoekschommelingen afgeven. Het commando daarvoor luidt:

BEEP ton, tTon, tPause

Voor de drie parameters kunnen constanten of termen toegepast worden. Daarbij bepaalt 'ton' de geluidshoogte volgens de formule

ton = 250000 / freq [Hz]

tTon bepaalt de duur van het signaal en tPause de pauze na het signaal. De eenheid voor de tijdsaanduiding bedraagt 20 milliseconden. Het commando

BEEP 568, 10, 3

geeft dus voor $10 \cdot 20 = 200$ milliseconden een signaal van ca. 440 Hz (kamersignaal A) en heeft daarna een pauze van $3 \cdot 20 = 60$ milliseconden. Als er na een BEEP geen andere BEEP meer volgt, kan de pauze ook op 0 gezet worden. Als voor de lengte van het signaal 0 aangegeven wordt, wordt er een permanent signaal geproduceerd. De toongenerator schakelt het geluid in en gaat door met de bewerking van het BASIC-programma. Met de waarde 0 voor ton kan de toongenerator weer uitgeschakeld worden.

- Frequentiemeting met de functies FREQ, FREQ1 en FREQ2

Als op de DCF77- ingang geen actieve antenne aangesloten is, kan met deze ingang alternatief een frequentiemeting plaatsvinden, waarvan het resultaat met de functie FREQ op elk moment opgevraagd kan worden.

x = FREQ

De frequentiemeting is gebaseerd op het principe van pulsen tellen bij een poorttijd van 1 seconde. De meting vindt steeds op de

achtergrond plaats, parallel aan het afwerken van het BASIC-programma. Het meetbereik gaat tot ca. 5 Kilohertz met een meetfout van minder dan één procent. Daarna wordt het resultaat toenemend minder precies.

Het oproepen van `FREQ1` is identiek aan dat van `FREQ`.

Het oproepen van `FREQ2` levert de meetwaarde van de tweede frequentiemeetingang (tot 32000Hz).

- Stroomsparmode met **SLOWMODE**

Toepassingen die geen hoge rekencapaciteit nodig hebben, kunnen door het oproepen van het commando

SLOWMODE ON

de interne maat van de microprocessor vertragen (1/16). In combinatie met de jumpers "LED" en "232" kan op deze manier de energiebehoefte van de stuurcomputer nogmaals verlaagd worden. Als er in de loop van het programma weer een hogere snelheid nodig is, dan kan met

SLOWMODE OFF

de uitgangstoestand weer ingesteld worden.

Let op: programma's, die seriële dataoverdracht gebruiken, mogen de SLOWMODE niet activeren, omdat de ingestelde overdrachtsnelheden samen met de processormaat verlaagd worden.

Inbinden van assemblerroutines

De volgende informatie richt zich tot de professionele gebruiker van de C-Control/*BASIC* stuurcomputer en is voor de eigenlijke BASIC-programmering niet noodzakelijk!

Er wordt kennis van de interne opbouw van de microcontroller MC68HC05B6 en van de assemblerprogrammering van deze controller ver-

ondersteld. Bovendien is er een assembler voor 68HC05-processoren nodig. Hier wordt het boek "Motorola 68HC05" van Zakeriya Zengin van uitgeverij Heise (ISBN 3-88229-034-X, Conrad bestnr. 91 91 79) aanbevolen, dat de microcontroller volledig beschrijft en op een meegeleverde diskette o.a. een assembler en talrijke voorbeelden levert.

De meeste toepassingstechnische problemen kunnen zeker uitsluitend door een BASIC- programma opgelost worden. Het kan echter toch nog voorkomen, dat voor een speciale opgave een hogere verwerkingsnelheid of speciale toegang tot bepaalde hardware nodig is. Voor dit geval staan naast de externe EEPROM- geheugenchip in de microprocessor zelf nog eens 2555 EEPROM bytes tot uw beschikking voor het opnemen van in de assembler geprogrammeerde routines. Deze routines kunnen vanuit het BASIC- programma opgeroepen worden. Het commando daarvoor luidt

`SYS adr`

waarbij `adr` een constante is en het adres bepaalt, bijvoorbeeld `&H101`, omdat het interne EEPROM- bereik op dit adres begint. De assemblercode moet dus per `ORG-` commando op het adres `&H101` opgeslagen worden. De terugkeer vanuit de assembleroutine naar BASIC geschiedt per `RTS-` commando. Het uitwisselen van data tussen BASIC en assembler kan gebeuren via de in het bestand `SYSADR.INC` opgenomen RAM- adressen.

Hoe komt de assemblercode in de C-Control/*BASIC* stuurcomputer? De in de assembler geschreven extra routines worden in een separaat brontekstbestand (b.v. `ADDONS.ASM`) opgeslagen. Aansluitend wordt de assembler opgeroepen om daaruit een objectcodebestand in S19-format te maken (b.v. `ADDONS.S19`). Lees daartoe de documentatie van de tot uw beschikking staande assembler. Uw BASIC- programma, dat het `SYS-` commando bevat, moet met het commando

SYSCODE "ADDONS.S19"

De gegenereerde code inbinden. Het `SYSCODE` commando mag slechts eenmaal in een CC*BASIC*- programma verschijnen en moet aan het einde, nog achter eventuele tabellendefinities staan.

Tabellen en afbeeldingen

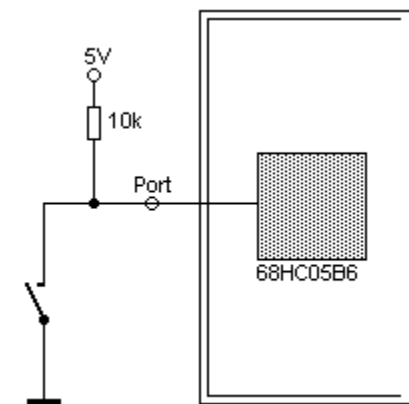
Technische gegevens (stuurcomputer)

Voedingsspanning U_b	5V gestabiliseerde gelijkspanning, +0,5V
Stroomverbruik	ca. 6 mA
Afmetingen	ca. 80 mm x 50 mm
Microcontroller	Motorola MC68HC05B6 4 MHz maatfrequentie 6 Kilobyte maskergeprogrammeerd systeem
Geheugenchip voor toepassingsprogramma en -data	microchip 24C65, seriële EEPROM met I2C-interface, 8k x 8 bit
A/D ports	8 x 8 bit A/D, 0...5 Volt tegen gemeenschappelijke massa Referentiespanning U_{ref} instelbaar (normaal $U_b = U_{ref}$) Ingangsstroom ca. 10 μ A bij omvorming absolute fout \pm digit (= 1/256 van de waarde van het einde van het meetbereik) vermeerderd met de fout van de referentiespanning
Digital ports	16 stuks, vrij programmeerbaar als in- of uitgang Niveau (0,2 mA last op uitgangen): $(U_b - 0,3V) < U_{out,high} < (U_b - 0,1V)$ $0,1V < U_{out,low} < 0,3V$ $(0,7 * U_b) < U_{in,high} < U_b$ $0V < U_{in,low} < (0,2 * U_b)$ max. toelaatbare laststroom: \pm 10mA Let op: een overschrijding door ontbrekende stroombegrenzing bij het aansluiten van

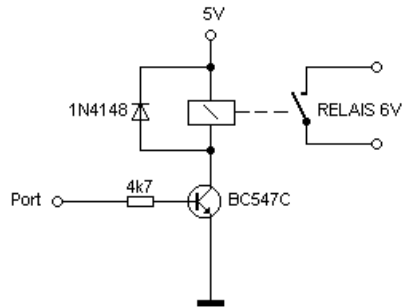
spanningen op de digital ports kan leiden tot directe vernieling van de microcontroller!

D/A- omvormer	2 pulsbreedte- gemoduleerde uitgangen, PWM(PBM)-snelheid 1953 Hz
DCF/FREQ1- ingang,	digital port met 10k pull-up voor het aansluiten van een DCF77- actieve antenne met open-collector uitgang
Seriële interface	RS232 met niveau-omvormer MAX232 of soortgelijk type op programmeeradapter Overdracht met 9600 baud, 8 bit, 1 startbit, 1 stopbit, geen pariteitsbit, geen handshake Verbinding met de PC via nulmodemkabel

Aansluiting van een schakelcontact op een digitale ingang

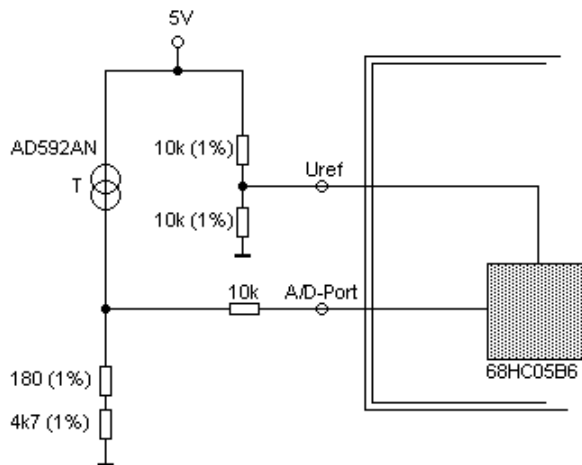


Aansluiting van een transistor voor het aansturen van een relais



Aansluiten van een temperatuursensor op een A/D- port

Het volgende eenvoudige voorbeeld toont de aansluiting van een temperatuursensor op de control unit. De sensor AD592 werkt als temperatuurafhankelijke stroombron en levert 1(A per Kelvin, bij 25 °C (ca. 298 Kelvin), dus 298 (a. Het voorbeeld biedt een resolutie van ca. 2 Kelvin en een theoretisch meetbereik van 0 tot 512 Kelvin (ca. -273 °C tot 239 °C), wat voor eenvoudige temperatuurregelingen voldoende is.



Door de keuze van de referentiespanning ($= V_{cc} / 2 = 2,5V$) en van de meetweerstand ($2,5V / 0.000512A = 4882,8$ (= ca. $4,7k + 180$) is het omrekenen van de A/D- omvormingswaarde naar een temperatuur in graden Celcius bijzonder eenvoudig: $T[^\circ C] = AD * 2 - 273$. Het volgende voorbeeld laat een eenvoudige aansturing van een verwarming zien:

```

define sensor ad[1]
define verwarming port[1]
define temperatuur word

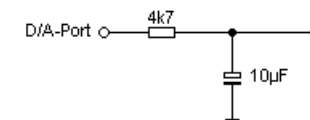
define ingangsdrempel 20
define uitgangsdrempel 30

#mess
    temperatuur = sensor*2 - 273
    if temperatuur < ingangsdrempel then verwarming = 1
    if temperatuur > uitgangsdrempel then verwarming = 0
goto mess
    
```

Voor hogere eisen aan resolutie, precisie en andere meetbereiken zijn er voor wat de hardware betreft een grotere hoeveelheid bouwelementen (operatie-versterker, referentiespanningsbron) en voor de software een tabellenbestand met overeenkomstige karakteristiek nodig.

Aansluiten van een RC-lid op de D/A-port

Met de volgende eenvoudige schakeling kan uit het pulsbreedte-gemoduleerde signaal van een D/A port een analoge gelijkspanning gewonnen worden.



Voor hogere eisen aan restdrempel en belastbaarheid of het omzetten in andere spanningsbereiken is een grotere hoeveelheid aan elektronische bouwelementen nodig.

Overzicht van de commando's en gereserveerde woorden van CCBASIC

Aansluitend worden op deze plaats nogmaals alle gereserveerde woorden van CCBASIC in de vorm van een tabel en in alfabetische volgorde samengevat, om bij syntaxische vragen een snel overzicht te hebben.

	Betekenis	Toepassingsvoorbeeld
ABS(x)	Absolute bedrag van x, x: term	y = ABS(-1)
AD[i]	Specificeert een A/D-port bij DEFINE, I: 1...8	DEFINE potmeter AD[1]
AND	Logische en bitsgewijze UND- koppeling	If(x<0) AND (y<0)... y = x AND &H7F
APPEND	Bestand openen om data aan te hangen	OPEN# FOR APPEND
BAUD rate	Overdrachtsnelheid van de seriële interface instellen snelheid: constante	BAUD R2400
BEEP t,d,p	Een geluid afgeven, t: geluid = 250000/freq. d = duur = d * 20ms p: pauze = p * 20ms	BEEP 400,25,10
BIT[i]	Specificeert een userbit bij DEFINE, i: 1...192	DEFINE flag BIT[16]
BYTE[i]	Specificeert een userbyte bij DEFINE, i: 1...24	DEFINE x BYTE[1]
BYTEPORT[i]	Specificeert een 8-bit-digitale port bij DEFINE, i: 1...2	DEFINE leds BYTEPORT[1]
&B...	Aanduiding van een binair getal	&B1010000
CLOSE#	Bestand sluiten	CLOSE#
DAY	Dag-waarde van de real-time klok	DAY = 1 PRINT DAY

DA[i]	Specificeert een D/A-port bij DEFINE, I:1...2	DEFINE v DA[2]
DEACT p	Deactiveert een digital port, p: port variabele	DEACT led
DEFINE	Definitie van een marker voor een port, een variabele of een constante	DEFINE sensor AD[1] DFINE flag BIT[1] DEFINE limit 1000
DOW	Weekdag-waarde van de real-time klok	IF DOW=1 THEN PRINT "MAANDAG"
ELSE	Alternatieve aftakking bij een bepaalde programma-uitvoering	IF x=0 THEN y=0 ELSE y=5
END	Einde programma	IF x THEN END
EOF	Test op einde bestand bij het uitlezen van het bestand	IF EOF THEN GOTO xxx
FILEFREE	Test op vrije geheugen-plaats bij het schrijven van bestand	IF FILEFREE THEN GOTO xxx
FOR	Programmalus	FORi=1 TO 10
FREQ FREQ1 FREQ2	Opvragen van de frequentie- meetingen	X = FREQ
GET v	Byte van de seriële interface lezen, v: variabele van marker	GET teken
GOSUB I	Sprong naar een subroutine, I: label-marker	GOSUB proc
GOTO I	Programmasprong, I: label-marker	GOTO start
&H...	Aanduiding van een hexgetal	&H1FFE
HOUR	Uur-waarde van de realime klok	PRINT HOUR

IF	Bepaalde programma-uitvoering, na IF volgt de bepalingsterm	IF $x < y$ THEN GOTO ready
INPUT v	Inlezen van een getals-waarde van de seriële interface, v: variabele-marker	INPUT x
INPUT# v	Inlezen van een getals-waarde uit het bestand, v: variabele-marker	INPUT# x
INTERRUPT I	Definiëren van een interrupt- routine I: geldig label	INTERRUPT event
LOOKTAB t,i,v	Laden van een waarde uit een tabel, t: tabellenmarker i: index v: variabele-marker	LOOKTAB kty, meet, temp
MAX(x,y)	Bepaling van de grootste waarde, x,y: termen	$Z = \text{MAX}(x, 10)$
MIN(x,y)	Bepaling van de kleinste waarde, x,y: termen	$Z = \text{MIN}(x, 10)$
MINUTE	Minuten-waarde van de real-time klok	PRINT MINUTE
MOD	Operator voor moduledivisie	$X = 10 \text{ MOD } 3$
MONTH	Maand-waarde van de real-time klok	PRINT MONTH
NAND	Logische en bitsgewijze UND-koppeling met aansluitende ontkenning	$X = a \text{ NAND } b$
NOR	Logische en bitsgewijze OF-koppeling met aansluitende ontkenning	$X = a \text{ NOR } b$

NOT	Logische en bitsgewijze ontkenning	$Y = \text{NOT } x$
OFF	Vooraf gedefinieerde constante voor 0	Led = OFF
ON	Vooraf gedefinieerde constante voor &HFFF	Led = ON
ON x GOSUB	Subroutine- oproep afhankelijk van x	ON sel GOSUB p1, p2, p3
ON x GOTO	Programmasprong afhankelijk van x	ON sel GOTO p1, p2, p3
OPEN#FOR m	Bestand met mode m openen, M: APPEND, WRITE of READ	OPEN# FOR WRITE
OR	Logische en bitsgewijze OR- koppeling	$x = a \text{ OR } b$
PAUSE t	Programma-onderbreking gedurende $t \cdot 20$ milliseconden	PAUSE 50
PORT[i]	Specificeert een digitale port bij DEFINE, i: 1...16	
PRINT x	Uitgifte van waarden en teksten via de seriële interface, x: term of tekst tussen aanhalingstekens	PRINT "A=", a, "B="; b
PRINT# x	Registreren van een waarde in het bestand, x: term	PRINT# a
PULSE p	Uitgifte van een puls naar een digitale port, p: port variabele	PULSE clock
PUT x	Uitgifte van een byte via de seriële interface, x: term	PUT ch
RAND	Integer toevalsgetal genereren	$X = \text{RAND}$

RANDOMIZE x	Toevalsgetallengenerator opnieuw initialiseren	RANDOMIZE TIMER
RETURN	Terugkeer uit een subroutine	RETURN
RETURN INTERRUPT	Terugkeer uit een interrupt- routine	RETURN INTERRUPT
RXD	Test, of een byte serieel ontvangen is	IF RXD THEN GET ch
READ	Bestand openen voor het uitlezen van data	OPEN# FOR READ
SECOND	Seconde-waarde van de real-time klok	PRINT SECOND
SHL	Operator voor bitsgewijs naar links schuiven	A = a shl 2
SHR	Operator voor bitsgewijs naar rechts schuiven	A = a shr 2
SLOWMODE m	Processortempo/16, m: ON of OFF	SLOWMODE ON
STEP	Aanduiding van de stapbreedte bij de FOR-lus	FOR i=100 TO 0 STEP -5
SYS a	Oproepen van een systeemprogramma, a: adres	SYS &H101
SYSCODE	Aanduiding van systeemcode of van een S-19 bestand	SYSCODE "TEST.S19"
SYSEND	Afsluiten bij met SYSCODE direct ingevoerde bytes	SYSCODE &H81 SYSEND
TABLE	Definitie van een datatabel, direct of via tabellenbestand	TABLE kty "KTY10.TAB"
TABEND	Afsluiting bij met TABLE direct ingevoerde waarden	TABLE tab 10 5-3 TABEND
TIMER	Opvragen van de 20-ms timer	x = timer

TO	Aanduiding van de eindwaarde bij de FOR-lus	FOR i=1 TO 10
TOG p	Omschakelen van een digital port, p: port variabele	TOG led
WAIT condition	Wachten op intreden van een voorwaarde (condition <>0) condition: term	WAIT key
WORD[i]	Specificeert een userword bij DEFINE, i: altijd 1	DEFINE x WORD[1]
WORDPORT[i]	Specificeert een 16-bit digital port bij DEFINE, l: altijd 1	DEFINE all WORDPORT[1]
WRITE	Bestand openen voor het schrijven van nieuwe data	OPEN# FOR WRITE
YEAR	Jaar-waarde van de real-time klok	PRINT YEAR