



10	K1
11	K1
12	Lege klemdraadhouder
13	K2
14	K2
15	Lege klemdraadhouder

Met de relais K1 en K2 in de module kunnen er apparaten geschakeld worden, die met netspanning van 230V~ gebruikt worden. De maximaal toegelaten contactstroom (zie "Technische gegevens") mag niet overschreden worden. De lege klemdraadhouders tussen de relais en de aansluiting van de voedingsspanning dienen voor de naleving van de nodige veiligheidsafstanden tussen de verschillende spanningspotentialen.

## Aansluiting voor de programmering van de module

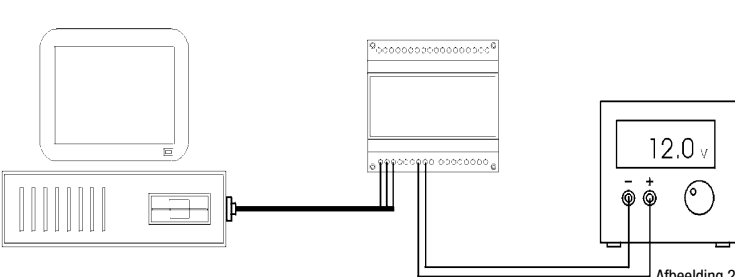
### Verbinding van de Station met de PC

Om een gebruikersprogramma van de PC naar de module over te dragen, zijn er twee aansluitingsvormen met betrekking tot de spanningsvoorziening toegelaten.

In beide gevallen dient de interfacekabel tussen PC en Station als volgt aangesloten te worden:

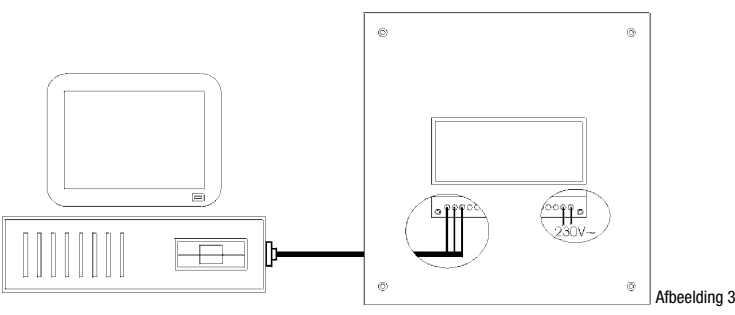
- Verbind de 9-poligen Sub-D-connector van de bij de Station bijgevoegde aansluitkabel met één van de seriële interfaces (COM1...COM4) van de PC
- Klem de **wit** gemarkeerde leidingdraden aan **klemdraadhouder 1** van de Station - **RX**
- Klem de **bruin** gemarkeerde leidingdraden aan **klemdraadhouder 2** van de Station - **GND**
- Klem de **groen** gemarkeerde leidingdraden aan **klemdraadhouder 3** van de Station - **TX**

#### Programmeren bij laagspanningsvoorziening



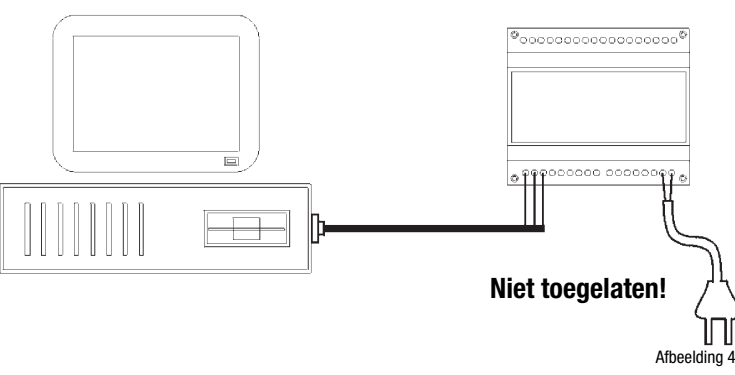
Bij de programmering met laagspanningsvoorziening dient er een gestabiliseerd 12V-voedings eenheid of een 12V-accu op de klemdraadhouders 7 (+) en 8 (-) van de module aangesloten te worden. De module kan dan vrij gehanteerd worden. Deze manier van voorziening is uiterst geschikt voor de programmering op de PC-werkplaats, in het laboratorium, in het atelier of op kantoor.

#### Programmeren bij netvoorziening



De programmering met netvoorziening dient voor de overdracht van gebruikersprogramma's op de plaats van gebruik van de module. De *C-Control Station* is daarbij op de DIN-rail gemonteerd. Om de bescherming tegen aanraking te waarborgen, moeten tenminste de netaansluitklemmen geblindeerd zijn.

Bij de aansluiting van de programmeerkabel moet de *C-Control Station* vrij van spanning zijn!



De voorziening van een vrij op tafel liggende module door middel van een standaard voedingskabel met integraal gegoten stekker (voor apparaten van de veiligheidsklasse II) is weliswaar vanuit technisch oogpunt mogelijk, maar in overeenstemming met de wettelijke bepalingen ter bescherming tegen aanraking van gevaarlijke spanningen niet geoorloofd!

## Systeemtoestanden

### Terugstelling ("Reset")

Na het doorverbinden van de bedrijfsspanning of na het indrukken van de schakelaar "Terugstelling") ("Reset") voert de module alle initialisaties door en gaat vervolgens naar de modus "Stand-by" over.

De hierna volgende initialisaties worden doorgevoerd:

- Datum en tijdstip: 1 januari 1997, 00:00:00
- Voorbereiding van de gegevensontvangst van de DCF77-radioklok
- Instellen van het seriële interface op 9600 baud, 8 gegevensbits, 1 stopbit, geen pariteit
- Wissen van al de gebruikersvariabelen (=0)

### Modus "Stand-by"

In de modus "Stand-by" worden de schakelaar "Start" en het seriële interface permanent opgevraagd om het gebruikersprogramma te starten of om te reageren op besturingscommando's vanwege een bovengeschatte computer (PC). In de modus "Stand-by" kan de module geprogrammeerd worden (download van het programma).

Indien de schakelaar "Start" als zijnde ingedrukt gedetecteerd wordt, gaat het systeem naar de modus "Run" over.

### Modus "Run"

In de modus "Run" worden de commando's van het gebruikersprogramma uitgevoerd. De modus "Run" eindigt door het indrukken van de toets "Terugstelling") ("Reset"), het uitschakelen van de bedrijfsspanning of het uitvoeren van het commando "END" ("Einde") (c.q. van de "ENDBlock") in het gebruikersprogramma. Na het commando "END" ("Einde") gaat het systeem terug naar de modus "Stand-by" over.

### Programmering (algemeen)

### Overzicht

Door programmering wordt de werkwijze van het systeem vastgelegd. Daarbij biedt de *C-Control Station* een hoge mate aan flexibiliteit en kan de *C-Control Station* zodoende voor de meest uiteenlopende doeleinden gebruikt worden. Als de *C-Control Station* geprogrammeerd is, dient ze voor u afdeling bijvoorbeeld als temperatuurregelaar, intelligente trappenlichtautomaat, alarminstallatie of rolkluisbesturingsinrichting. Indien er eens andere taken te wachten staan, kan de module een willekeurig aantal keren anders geprogrammeerd worden.

In het binnenste gedeelte van de *C-Control Station* werkt een microcontroller. Dat is een kleine computer op een chip. Op deze microcontroller is een geheugenchip aangesloten, die voor de opname van het gebruikersprogramma dient. Door de EEPROM-technologie blijft het gebruikersprogramma ook gehandhaafd wanneer de bedrijfsspanning van het systeem verbroken wordt.

De programmering van de *C-Control Station* gebeurt met behulp van comfortabele software aan de PC en dit ofwel op een grafisch ontwikkelingsoppervlak ofwel in de populaire programmeertaal BASIC.

### Bits, bytes en Words

De C-Control System staat in voor de verwerking, de berekening en de opslag van uitsluitend volledige getallen (= "Integer"). Daarbij kent de C-Control System drie verschillende gegevenstypes:

**Bit**: waardebereik 0 en -1 (niet 1!); toepassing als toestandsgeheugen (= "Flag")

**Byte**: reserveert 8 bits; waardebereik 0 tot 255; toepassing voor kleine niet-negatieve volledige getallen, ASCII-tekencodes, commandocodes

**Word (= "Integer")**: reserveert 16 bits; bestaat uit "High-Byte" en "Low-Byte"; waardebereik -32768 tot 32767; universele toepassing

## Systeemressources

### Randapparatuur

Uiteindelijk gaat het bij ieder programma voor C-Control om het schakelen en beheren van de aangesloten modules. De meeste randapparatuureenheden staan rechtstreeks in verbinding met de aansluitklemmen of de bedieningselementen op het frontpaneel.

**Digitale poorten**: de *C-Control Station* beschikt ion totaal over 16 programmeerbare digitale poorten. De poorten 1 tot 6 zijn met de klemdraadhouders P1 tot P6 verbonden en kunnen als ingang of uitgang gebruikt worden. De poorten 7 en 8 schakelen de beide relais K1 c.q. K2. Door middel van de poorten 9 tot 12 kunnen de functietoetsen F1 tot F4 opgeroepen worden. De poorten 13 tot 16 kunnen de LED's door middel van de functietoetsen in- en uitschakelen.

**Analoge poorten**: voor de detectie van analoge meetgrootten heeft de *C-Control Station* in totaal 8 analoog/digitaal-convertoerpoorten met 8 bit resolutie. Daarmee kunnen er elektrische spanningen gemeten worden, die bijvoorbeeld als uitgangssignaal van een voeler van een bepaalde fysieke grootte afhangen. Het meetbereik reikt van 0 tot 2,55V in stappen van 10 mV. Aan de *C-Control Station* zijn de analoge poorten 1 tot 4 aan de klemdraadhouders A1 tot A4 voorzien.

De analoge poorten 5 en 6 zijn met de temperatuurvoelers T1 c.q. T2 verbonden.

De analoge poort 7 meet de spanning op een intern punt van de toevoerelektronica. Indien er bij een ontbrekende netspanning een geladen 12V-accu op de klemdraadhouders 7 en 8 aangesloten is, bedraagt de meetwaarde circa 120. Nu kan door de meetwaarde te laten dalen de ontlading van de accu tijdens de werking gevolgd worden. Indien de accuspanning tot onder een kritische waarde daalt, kunnen er zodoende in het gebruikersprogramma tijdig maatregelen tegen een aankomende uitval van het systeem getroffen worden.

Aan de analoge poort 8 kan de uitval van de netspanning van 230V gedetecteerd worden. De poort is intern door middel van een spanningsverdeler met de uitgang "Plus" van de bruggeijkrichter van de geïntegreerde voedings-eenheid verbonden. Het spanningsverloop is met een condensator vereffend. Indien de meetwaarde tot op 0 daalt, is de netspanning uitgevallen. Omdat de bedrijfsspanning van de microcontroller met een grotere condensator gebufferd is, draait het systeem na een uitval van de netspanning nog een tijdje verder. In geval van een geschikte

programmering van de uitvaldetectie kunnen er zodoende nog maatregelen voor een geordende programmabeëindiging getroffen worden.

**Onvertraagde klok**: op de klemdraadhouders 4 tot 6 kan er een actieve DCF77-antenne aangesloten worden. Bij signaalontvangst wordt de interne klok van het systeem automatisch op Midden-Europese tijd/zomertijd geplaatst. De individuele informatie aangaande tijd en datum – uur, minuut, seconde, dag, weekday, maand jaar – kan in het gebruikersprogramma opgeroepen en rechtstreeks of in logische combinatie met andere voorwaarden als activator van bepaalde handelingen gebruikt worden.

**Frequentiemeting**: met de frequentiemeetgang kunnen er signalen tot circa 30kHz geregistreerd en in het gebruikersprogramma opgeroepen worden.

**Serieel interface**: het seriële interface dient niet alleen voor het laden van een gebruikersprogramma in de *C-Control Station*, maar kan ook in het programma zelf voor de gegevensoverdracht gebruikt worden. Daarbij is het mogelijk om gegevens per byte of als tekst over te dragen. Het overdrachtcijfer kan op enkele niveaus (1200...9600 baud) aangepast worden.

**Klankuitvoer**: in de *C-Control Station* is er een piëzo-geluidsomzetter ingebouwd. Dat is een kleine luidspreker, waarmee er klanksignalen geproduceerd kunnen worden. Daarbij zijn klankhoogte en klankduur programmeerbaar. Zodoende zijn er kleine melodieën, beantwoordende of waarschuwende signalen programmeerbaar om een druk op een toets te bevestigen of om in bepaalde situaties bijzondere aandacht van de gebruiker naar het apparaat te laten uitgaan.

#### Programmageheugen

Het door u geprogrammeerde gebruikersprogramma wordt door de programmeersoftware naar een voor de C-Control System begrijpbaar formaat vertaald (= "gecompileerd") en naar het programmageheugen van de *C-Control Station* overgedragen. Het programmageheugen heeft een omvang van ongeveer 8000 bytes. Een commando om één van de relais in te schakelen, reserveert bijvoorbeeld 5 bytes. Ook complexere programma's hebben zelden meer dan enkele honderden bytes nodig. Zodoende staat er steeds voldoende plaats in het programmageheugen ter beschikking.

Het programmageheugen bevindt zich in de EEPROM-geheugenchip (type Microchip 24C65). Omwille van de EEPROM-technologie blijven al de gegevens in het programmageheugen ook na het uitschakelen van de spanningsvoorziening behouden.

### Geheugen voor gegevensnotities

De overige vrije bytes in de programmageugenchip staan voor het gebruikersprogramma voor de aantekening van gegevens, bijvoorbeeld meetwaarden, ter beschikking. Het maken van notities en het uitlezen van de waarden gebeurt steeds sequentieel en per Word. Een genoteerde waarde reserveert dus 2 bytes.

Het beheer van de genoteerde waarden gebeurt in overeenstemming met het bestandsprincipe. Vóór de toegang moet het bestand geopend worden. Er wordt een onderscheid gemaakt qua openen om te lezen, openen om te schrijven en openen om nieuwe gegevens bij vorige gegevens bij te voegen. Na een toegangsreeks moet het bestand gesloten worden om eventuele wijzigingen over te nemen.

Zoals het gebruikersprogramma zijn de genoteerde gegevens veilig tegen uitval van de bedrijfsspanning. De vereiste voorwaarde daarvoor is dat het bestand na de voor het laatst geschreven waarde gesloten werd.

### RAM-geheugen

Tijdens het verloop van een programma is het vaak noodzakelijk om bepaalde rekenresultaten, toestanden of parameters voorlopig op te slaan. Daarvoor biedt de *C-Control Station* een RAM-geheugen van in totaal 24 bytes, waarop er door middel van definieerbare variabelen per bit, byte of Word beroep kan gedaan worden.

In tegenstelling tot het programmageheugen blijft de inhoud van het RAM-geheugen slechts zolang behouden als de microcontroller van het systeem van bedrijfsspanning voorzien is. Na een terugstelling ("Reset") of een uitval van de spanning worden al de waarden in het RAM-geheugen terug op nul gezet (=0).

### Rekeneenheid

Zoals in het hoofdstuk "Bits, bytes en Words" reeds beschreven, gebeuren al de berekeningen met volledige getalen. Al de handelingen worden met een breedte van 16 bit uitgevoerd ("Integer"-waardebereik). De rekeneenheid van het systeem beheerst de vier basisrekenotypes (+, -, \*, /), de modulusdeling (vaststelling van de rest bij deling van volledige getallen), per bit logische verbindingen, vergelijkende handelingen, negatie, de bepaling van het absolute bedrag, de afgeronde vierkantswortel- en de signumfunctie. Bovendien is er een initialiseerbare pseudo-toevalgenerator geïmplementeerd, die volgens het principe van de priemgetalvermenigvuldiging met daarop volgende modulusdeling statistisch verdeelde getallenreeksen kan produceren.

De rekeneenheid beheerst maximaal 6 in elkaar geschoven niveaus bij cliprekeningen of vergelijkbare handelingen.

**Opgelet!** Door de begrenzing op maximaal 16 bit gegevensbreedte in het "Integer"-waardebereik kan het bij berekeningen tot het overlopen in de resultaten komen. Bovendien vindt er bij de opslag van een waarde in een variabele omwille van het principe een begrenzing op hun opgegeven waardebereik plaats. Eventuele decimalen van delingresultaten worden afgesneden.

- Voorbeeld:
  - 32767 + 1 = -32768, niet 32768
  - a = -1 = 255 voor a indien a slechts een bytevariabele is
  - 10 / 4 = 2, niet 2,5

### Grafische programmering

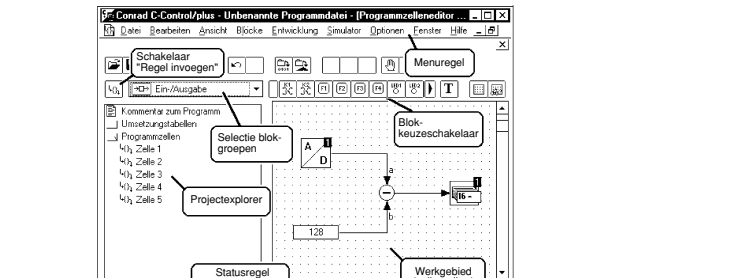
### Overzicht

Voor de grafische programmering staat er voor u een speciaal aan de *C-Control Station* aangepaste versie van het programma "C-Control/plus" ter beschikking. De **grafische programmering** is vooral **voor beginnelingen en kleine toepassingen** geschikt. De software bevindt zich op de bij de Station bijgevoegde CD. Gelieve voor de installatie de op de CD geschreven aanwijzingen in acht te nemen. De grafische programmeersoftware heeft als basisbesturingsstelsysteem minstens Microsoft Windows95 of WindowsNT vanaf 4.0 of overeenkomstige, daaropvolgende versies nodig! De software kan niet onder het intussen verouderde basisbesturingsstelsysteem Windows 3.xx geïnstalleerd worden.

Omdat de bedieningselementen van de software zelfverklarend en intuïtief te begrijpen zijn, doen wij op deze plaats afstand van een complete beschrijving van het gebruikersoppervlak. De hierna volgende paragrafen voorzien u enkel en alleen van de nodige basiskennis. Hulp bij de bediening van het programma krijgt u aan de PC door het indrukken van de toets F1 (algemene hulp) of Strg+F1 (contextgevoelige hulp). Bovendien kunt u door een klik met de

rechtse muisknop op een functieblok en door de selectie van het vervolgens aangeboden contextmenupunt "Hulp bij het bloktype" een beschrijving van de blokfunctie krijgen.

Afbeelding 5 toont een overzicht op het gebruikersoppervlak bij de grafische programmering van de *C-Control Station*. De voornaamste componenten zijn van een opschrift voorzien. Voor de bediening van de software is het noodzakelijk dat u met de basisprincipes van de bediening van standaard Windows programma's vertrouwd bent. Begrippen, zoals "menu", "statusregel", "klikken", "dubbel klikken" of "Slepen met de muis", "Bestand openen" of "Bestand opslaan" mogen voor u geen vreemde woorden zijn. Gelieve in geval van twijfel nogmaals het Windows handboek of de pagina's van de Windows on line hulp te lezen.

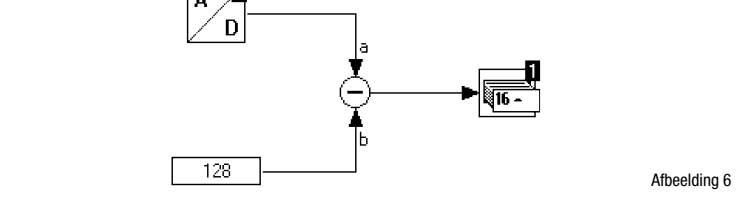


Afbeelding 5

Een grafisch programma bestaat uit meerdere zogenaamde "cellen", die de individuele werkstappen beschrijven, die de *C-Control Station* in de modus "Run" dient uit te voeren. Een overzicht van alle cellen van een project biedt de projectexplorer in de linkerzone. Nieuwe cellen kunnen bij het project bijvoorbeeld door een klik van de schakelaar "Cel invoegen" toegevoegd worden.

De werkstappen in de cellen worden blokschakelbeelden vastgelegd. Dit betekent dat individuele functieblokken door signaalstroomlijnen verbonden worden. De blokschakelbeelden worden in de celleneditor aangemaakt en bewerkt.

### Blokvoorraad



Zoals afbeelding 6 toont, zijn er functieblokken, die waarden leveren. Andere functieblokken verwerken waarden tot een resultaat en nog andere functieblokken slaan waarden op of activeren van de ingevoerde waarden afhankelijke acties. In het voorbeeld wordt er een meetwaarde door de A/D-poort 1 geregistreerd, waarvan de constante 128 afgetrokken (a-b) en het resultaat in een geheugen gedeponeerd.

De voorraad van al de in C-Control/plus ter beschikking staande functieblokken wordt in de hierna volgende groepen ingedeeld:

- In-/uitvoer** – toegangen tot relais, toetsen, LED's en de klankuitvoer
- Meetwaarden** – toegangen tot de temperatuurvoelers, de spanningmeting en de algemene analoge poorten
- Digitale poorten** – toegangen tot de digitale poortklemmen P1 tot P6
- Berekeningen** – mathematische verbindingen en functies
- Vergelijkng** – vergelijkende handelingen, die qua resultaat correct (= -1) of foutief (=0) kunnen zijn
- Logica** – verbindingen per bit na boolsche algebra
- Tijd** – toegang tot de interne onvertraagde klok (DCF77) en de timer
- Variabelen** - geheugenblokken
- Constanten** – blokken, die constante waarden leveren
- Gegevensnotitie** – aantekening van gegevens in de EEPROM-geheugenchip
- Serieel interface** – in- en uitvoer van tekst en gegevens door middel van het seriële interface van het systeem
- Programmasturing** – functieblokken voor de besturing van de programmastroom

## Voorbeeld van een grafische programmering

### Het programma aanmaken

Het hierna volgende kleine voorbeeld dient het basisprincipe van de grafische programmering te verduidelijken. De hier bijeengebrachte informatie stelt u in staat om de bijkomende voorbeelden van de CD te begrijpen en aan uw wensen aan te passen om zodoende een volledig overzicht op de grafische programmering van de *C-Control Station* te krijgen.

Veronderstelling: u wenst met de toets F1 aan de module het relais K1 Beurtelings in- en uitschakelen.

Start het programma CCSTAT.EXE. Begin met een nieuw project met het menucommando "Bestand/Nieuw". Activeer in de dialoog "Opties/omgevinginstelling/projectoverzichtvenster" de punten "Boom van de programmaccellen openklappen", "Eerste programmacel openen" en "Editor volgt in het projectvenster van geselecteerde programmacel". Deze instellingen worden bij het beëindigen van het programma opgeslagen en moeten niet steeds opnieuw ingevoerd worden.

Klik in de projectexplorer op de map "Programmaccellen". Klik 6 keer op de schakelaar "Cel invoegen". U ziet, hoe de nieuwe cellen in de projectexplorer ingevoerd worden.



Klik in de projectexplorer op de invoer "Cel 1". In de titelregel van de celleditor staat nu "Programmaceditor – cel 1". Klik in het werkgebied van de celleditor. Nu worden de tot nu toe grijze blokkeuzeschakelaars actief.

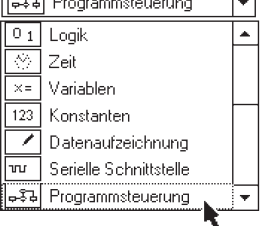
In de blokgroepselectie dient "In-/uitvoer" te staan. Klik nu op de derde blokkeuzeschakelaar van links, deze symboliseert de moduletoets F1.



Afbeelding 8

Een nieuw blok wordt in het werkgebied ingevoegd. Sleep het blok met de muis een beetje naar rechts onderaan.

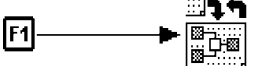
Ga nu in de blokgroepselectiel over naar de groep "Programmasturing", deze bevindt zich helemaal onderaan in de lijst.



Afbeelding 9

In het bereik van de blokkeuzeschakelaars verschijnt het pallet van de programmabesturingsblokken. Klik op de tweede schakelaar van links, "Subprogrammaoproep". Plaats het nieuwe blok een beetje rechts naast het eerste.

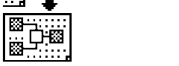
Nu dienen de blokken verbonden te worden. Klik daarvoor op het blok "F1", beweeg vervolgens de muispointer in de richting van het andere blok. Een gestippelde lijn ("rubberen band") geeft de verbindingsmodus aan. Klik nu op het tweede blok. In het werkgebied moet nu het volgende blokschakelbeeld te zien zijn:



Afbeelding 10

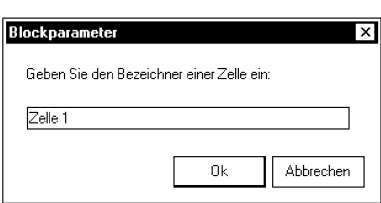
Wat betekent dat? De subfunctieoproep van het rechte blok is verbonden met de voorwaarde dat de toets F1 aan de module ingedrukt is, "...indien toets F1, dan splits af...". Vooraleer wij nu vastleggen, naar waar er afgesplitst moet worden, willen wij de andere cellen vullen.

Klik op cel 2 in de projectexplorer. In het werkgebied plaatst u het eerste blok uit de groep "Programmasturing", het blok "Afsplitsen naar cel".



Afbeelding 11

Hiermede dient de eindloze lus van het programma gerealiseerd te worden. Uit cel 2 moet er steeds terug naar cel 1 gesprongen worden. Klik met de rechtse muistoets op het blok. Er verschijnt een menu. Op de eerste plaats vindt u het menupunt "Eigenschappen". Als u dit kies, gaat de eigenschappendialoog voor het blok open.



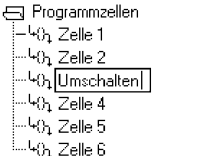
Afbeelding 12

Voer in de invoerregel de omschrijving van het doel, waarnaar er gesprongen moet worden, van het afsplitsblok in, hier cel 1. Gelieve erop te letten dat de omschrijving exact met de omschrijving in de projectexplorer overeenstemt: drukletter Z, een spatie vóór de 1. Tussen drukletters en kleine letters wordt er door de software net zozeer een onderscheid gemaakt als één, twee of geen spatie.

Indien u per vergissing "Cel1" invoert, zal de compiler een fout rapporteren!

Beëindig de dialoog met "O.K.."

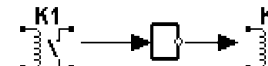
Klik nu in de projectexplorer op cel 3. Indien de cel gemarkeerd is, klikt u een tweede keer op cel 3. Er verschijnt een invoerveld, waarmee u de celnaam kunt wijzigen.



Afbeelding 13

Voer "Overschakelen" in en druk voor een bevestiging de toets ENTER in.

Ga nu opnieuw over naar het werkgebied. Plaats uit de groep "In-/uitvoer" twee keer het blok "Relais K1". Plaats de blokken op dezelfde hoogte een flink stuk uit elkaar. Plaats daartussen een "omgekeerd" blok uit de groep "Logica". Verbind de blokken als volgt:



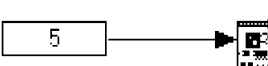
Afbeelding 14

Het nut van deze structuur dient onmiddellijk duidelijk te zijn: de schakeltoestand van het relais K1 wordt gelezen en omgekeerd opnieuw uitgevoerd.

Aangaande cel 4: klik in de projectexplorer op cel 4. Hier dient de "foutoplossing" van de toets F1 ondergebracht te worden. "Fouten" noemt men de ongewenste eigenschap van mechanische contacten, zich tot de definitieve aanmaak van de elektrische verbinding in fracties van een seconde meermaals te openen en te sluiten. Dat zou in ons voorbeeld tot een ongecontroleerde overschakeling van het relais leiden.

Het folietoetsenbord van de *C-Control Station* vertoont weliswaar een gering foutief gedrag, maar dit maar om zeker te zijn.

Plaats uit de groep "Programmasturing" een "pauze"-blok. Schuif het een beetje naar rechts. Voeg links naast het "Pauze"-blok een blok "Constante waarde" uit de groep "Constanten" in. Verbind beide blokken. Klik met de rechtse muistoets op het constantenblok. Voer door middel van de eigenschappendialoog de waarde 5 in.

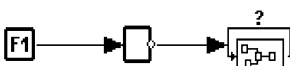


Afbeelding 15

Het "pauze"-blok wordt van de constante waarde 5 voorzien. Bij de afhandeling van de aanwijzing legt het programma een pauze van 5 x 20ms = 100 ms in. Binnen deze tijdspanne volgt er niet nog een gegevensopvraag van de schakelaar, de schakelaar is daardoor van fouten ontdaan.

Cel 5: omdat wij niet willen dat het relais meermaals overgeschakeld wordt wanneer de toets F1 langer ingedrukt wordt, moet het programma nu wachten tot de toets terug losgelaten wordt.

Daarvoor plaatst u in het werkgebied van de cel 5 uit de groep "Programmasturing" een blok "Wachten op waarde" en schuift u dit een beetje naar rechts. Plaats nu links naast het blok "Wachten op waarde" de toetsblok F1 en in het midden een Inverter. Beide bloktypes zijn immers intussen bekend. Verbind de drie blokken in serie.



Afbeelding 16

Het blok "Wachten op waarde" vertraagt de programma-afhandeling zolang tot de aan diens uitgang voorziene waarde niet gelijk aan 0 is. Omdat hier de omgekeerde toestand van de toets F1 toegevoerd wordt, wacht het programma echter zolang tot de toets losgelaten werd. Daardoor wordt voorkomen dat het langer indrukken van de toets meermaals tot een overschakeling leidt.

Cel 6 beëindigt de reactie op een druk op de toets F1. Daarvoor voegt u louter een blok "Terugkeer uit subprogramma" - uit de groep "Programmasturing" in. Na cel 6 wordt het programma in de cel na de oproep van het subprogramma voortgezet, m.a.w. in cel 2.

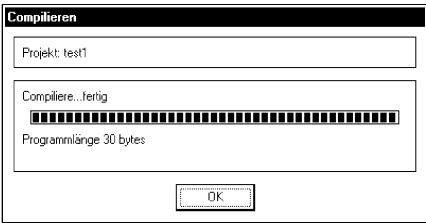
Tot slot mag niet vergeten worden, bij het blok voor de oproep van het subprogramma in cel 1 nog het doel, waarnaar er gesprongen moet worden, aan te geven. Klik op cel 1 in de projectexplorer. Open vervolgens de eigenschappendialoog (nogmaals: rechtse muistoets op het blok, "Eigenschappen" in het menu selecteren) en voer als doel, waarnaar er gesprongen moet worden, "Overschakelen" in, d.w.z. de nieuwe naam van de vroegere cel 3.

Het voorbeeldprogramma is nu compleet. Sla het project op (menu "Bestand/opslaan"), bijvoorbeeld als bestand TEST1.CPF.

### Compileren en simulatie

Het volledige project kunt u nu compileren, d.w.z. vertalen naar een voor de C-Control System begrijpbaar formaat. Kies daarvoor het menucommando "Ontwikkeling/compileren", of klik op de schakelaar "Compileren".

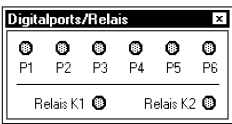
Een venster van het gebruikersoppervlak verstrekt u nu informatie over de vooruitgang van de compilatie en geeft op het einde het aantal geproduceerde codebytes of een foutmelding, indien u iets foutief ingevoerd hebt, aan.



Afbeelding 17

Het resultaat van de vertaling krijgt u niet meteen te zien, deze staat in het RAM-geheugen van uw PC en kan nu naar de module overgedragen of voor de simulatie gebruikt worden. Wanneer u het aantal codebytes leest, herinnert u zich misschien dat er in totaal ongeveer 8000 bytes ter beschikking staan. U kunt nu ongeveer inschatten, hoe ingewikkeld uw besturingstaken kunnen zijn!

De werking van het hoger vermelde voorbeeldprogramma dient nu met de simulator getest te worden. Open allereerst via het menu "Afbeelding" de simulatorvensters "Digitale poorten/relais" en "Toetsen/LED's".



Afbeelding 18



Afbeelding 19

Start nu de simulator via het menucommando "Simulator/uitvoeren" of via de corresponderende schakelaar "Toolbar". Let op de aanduiding "Simulatie in werking" in de statusregel.
Klik meermaals met de muis in het F1-veld van het venster "Toetsen/LED's". Houd daarbij de groene lichtgevende diode K1 in het venster "Digitale poorten/relais" nauwlettend in het oog. De simulatorvensters maken de randapparatuurfuncties van de *C-Control Station* grafisch na. In de simulator Gedraagt het programma zich zoals het dat ook van plan was. De functie van het voorbeeld is dus aangetoond. U kunt nu de simulator stoppen en terugstellen. Beide commando's vindt u in het menu "Simulator".

De simulator maakt het testen van gebruikersprogramma's zonder hardware mogelijk. Indien er fouten in het verloop vastgesteld worden, kunnen deze verholpen worden door een paar keer met de muis te klikken. Pas wanneer het programma gereed is, wordt het naar de *C-Control Station* overgedragen. Zodoende spaart men veel tijd en omslachtig hanteren met de module. Bovendien zou een foutief programma eventueel met onaangename gevolgen gepaard gaan. In elk geval kunnen de relais ook stroomkringen met 230V – netspanning schakelen!

### Het programma overdragen en starten

Na de test van het programma kan dit overgedragen worden.

- Zorg er beslist voor dat de Station op een betrouwbare manier van bedrijfsspanning voorzien en door middel van een serieel interface met de PC verbonden is.
- Druk de schakelaar "Terugstelling" ("Reset") aan de Station in.
- Stel door middel van "Opties / omgeving / simulator en lader" het gebruikte seriële interface in.
- Kies het menucommando "Ontwikkeling / naar C-Control-Station overdragen".

Op het beeldscherm wordt de vooruitgang van de programmaoverdracht met een balk weergegeven. Tijdens de overdracht knippert de lichtgevende TX-diode aan de RS232-klemdraadhouders van de Station.

De *C-Control Station* is nu geprogrammeerd.

- Druk de schakelaar "Start" in om het kleine testprogramma uit te voeren.

U ziet dat – zoals geprogrammeerd en gesimuleerd – met iedere druk op de toets F1 aan de module het relais K1 in- en uitgeschakeld wordt. Voor het antwoord over de schakeltoestand van het relais zorgt de lichtgevende diode aan de relaisklemdraadhouder.

Met de met dit kleine testprogramma opgedane ervaringen en de andere voorbeelden van de CD bent u in staat, stapsgewijs complexere toepassingen te realiseren.

## Programmeren in BASIC

### Overzicht

Voor de **realisatie van complexe projecten** of voor gebruikers met gepaste kennis vooraf is de programmering van de *C-Control Station* in de **programmeertaal BASIC** in C-Control/BASIC (CCBASIC) aanbevelenswaardig. Daarvoor vindt u twee versies van het programmeeroppervlak op de bij de Station bijgevoegde CD. De hoofdversie heeft als basisbesturingssysteem minstens Microsoft Windows95 of WindowsNT vanaf 4.0 of overeenkomstige, daaropvolgende versies nodig! De hoofdversie van C-Control/BASIC kan niet onder het intussen verouderde basisbesturings-systeem Windows 3.xx geïnstalleerd worden.

Op de CD bevindt zich echter ook een verminderde versie (zonder simulator). Deze kan onder MS-DOS draaien en kan bijgevolg ook op de meeste oudere PC's en Notebooks gebruikt worden tot en met 8086/88-systemen (XT). Goedkope oudere Notebooks, die de huidige kantoorsoftware niet meer aankunnen, zijn zodoende heel geschikt voor een wijziging van de programmering van een *C-Control Station* ter plaatse en zonder demontage van de module uit de schakelkast.

Gelieve voor de installatie van beide versies opnieuw de op de CD terug te vinden aanwijzingen te volgen.

### Essentiële elementen bij de programmeertaal CCBASIC

#### Algemeenheden

Een BASIC-programma bestaat uit meerdere regels met zogeheten brontekst. Daarbij bevat iedere regel één of meerdere *reken-* of besturingsinstructies. CCBASIC is het BASIC-dialect, dat voor de programmering van de *C-Control Station* gebruikt wordt. De syntaxis komt ongeveer overeen met het standaard BASIC. Bij een aantal commando's zijn er afwijkingen of uitbreidingen, die speciaal op de hardware van de *C-Control Station* afgestemd zijn.

Indien een programmaregel meerdere aanwijzingen bevat, dienen deze door dubbele punten ":" gescheiden te worden.

Regelnummers, zoals in oudere BASIC-dialecten gebruikelijk, zijn niet noodzakelijk. Indien er toch regelnummers aangegeven worden, kunnen deze als doel, waarnaar er gesprongen moet worden, gebruikt worden.

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

Een invloed op de volgorde van de programmeerhandelingen hebben de nummers daarenboven niet. Indien er bijvoorbeeld in de brontekst op een met 200 genummerde regel een regel 100 volgt, wordt desondanks de regel 200 vóór de 100 afgehandeld.

Commentaar kan er als toelichting bij het geschreven programma mee in de brontekst opgenomen worden zodat de leesbaarheid en de onderhoudsvriendelijkheid daarvan verbeteren. Commentaar in CCBASIC begint steeds met een komma bovenaan en verklaart de rest van de regel over de niet tot het programma behorende tekst.

```
 a = b + c      ^...  Kommentar ...
```

##### Omschrijvende elementen

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

Er wordt geen onderscheid gemaakt tussen drukletters en kleine letters. Een omschrijvend element begint steeds met een letter of met een onderlijning. Spaties binnen een omschrijvend element zijn niet geoorloofd.

##### Variabelen en constanten

Variabelen en constanten zijn objecten van het informatieverwerkende procédé. In CCBASIC slaan beide een numerieke waarde op. Tervijl de waarde van een constante één keer aangegeven wordt en vervolgens ongewijzigd blijft, kan de waarde van een variabele tijdens het verloop van het programma willekeurig vaak veranderen. Constanten kunnen in CCBASIC in decimale, hexadecimale en binaire vorm aangegeven worden. De syntaxis voor hexadecimale en binaire getallen wordt hier aan de hand van het voorbeeld van het getal 46 (decimaal) getoond:

```
 &H2E      \hexadeximal
 &B101110  \binaer
```

Tot variabelen wordt er steeds door middel van hun omschrijvend element toegang gekregen. Dit omschrijvende element moet vóór de eerste toepassing van de variabele in het programma in een DEFINE-regel gedefinieerd worden.

##### Labels

Labels markeren bepaalde punten in de volgorde van de programmahandelingen. Labels zijn doelen van spronghandelingen binnen een algoritme. In CCBASIC staan er labels aan het begin van een regel en beginnen ze steeds met een dubbel kruis, dan volgt – zonder spaties) – het omschrijvende element van het label.

Het voorbeeld toont de definitie van het label "Label1" en de toepassing in een sprongcommando:

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

##### Termen

Een term ontstaat onmiddellijk (als variabele of constante) of door berekening van een bepaalde waarde. Termen zijn onderdelen van aanwijzingen en staan bijvoorbeeld bij de toewijzing van een waarde aan een variabele rechts van het toewijzingstekens "=". Termen worden door combinaties van operanden en operatoren gevormd.

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

##### Operanden en operatoren

Een operand is in de basisvorm ofwel een constante, een variabele of een functieoproep, maar kan ook zelf een uit operanden en operatoren samengestelde term zijn. Operatoren duiden op rekenhandelingen, die met de omgevende operanden uit te voeren zijn. Daarbij is er een gedefinieerde hiërarchie (zie commandobeschrijving), die de volgorde van de berekeningen bepaalt.

##### Functies

Een functie voert een gedefinieerde handeling – bijvoorbeeld een berekening – door en levert door de oproep ervan een resultaatwaarde. De meeste functies verwachten één of meerdere argumenten, die in ronde haakjes na het omschrijvende element van de functie "(" overgedragen werden en door komma's gescheiden zijn. Enkele functies worden zonder argument opgeroepen. In dit geval worden er geen ronde haakjes geschreven.

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

In CCBASIC zijn al de ondersteunde functies vooraf gedefinieerd. Hun omschrijvende elementen behoren tot de gereserveerde woorden.

##### Toewijzingen

De toewijzing is de eenvoudigste vorm van een programma-aanwijzing. Na het omschrijvende element van een variabele, dat een waarde toegewezen krijgt, volgt het toewijzingsteken "=" en vervolgens de term, die de toegewezen waarde bepaalt. Een toewijzing stemt daardoor met een eenvoudige mathematische formule overeen.

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

##### Commando's

Naast de eenvoudige toewijzingen zijn commando's aanwijzingen voor de uitvoering van programmahandelingen door de *C-Control Station*. Commando's beginnen steeds met een gereserveerd woord. Enkele commando's verwachten één of meerdere parameters voor een nauwkeurige specificatie van de uit te voeren programmahandeling. Deze parameters worden na het omschrijvende element van het commando en een spatie vermeld en daarbij door komma's gescheiden (uitzondering PRINT, zie overzicht van de commando's). In tegenstelling tot de argumenten bij de oproep van een functie staan de commandoparameters niet binnen ronde haakjes!

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

##### Aanwijzingen voor de besturing van de programmastroom

Deze aanwijzingen laten het toe, de volgorde van de op zich streng sequentieel afgehandelde programmahandelingen te besturen en aan ingangswaarden van het informatieverwerkende procédé aan te passen. Ze bieden een hoge flexibiliteit en vormen voor de oplossing van sommige toepassingstechnische problemen zelfs de vereiste waarde.

Aanwijzingen voor de besturing van de programmastroom bestaan uit één of meerdere gereserveerde woorden en vereisen op een telkens speciale manier eventueel bijkomende gegevens.

```
GOTO label1
IF a > b THEN GOSUB label2
```

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

##### Compileraanwijzingen

Aanvullend op de programma-aanwijzingen bevat een oorspronkelijke CCBASIC-tekst compileraanwijzingen, die bijvoorbeeld voor het aanleggen van gegevensblokken (tabellen) of voor de definitie van variabelen en constanten dienen.

Voor compileraanwijzingen geldt de regel van het dubbele punt voor het scheiden van meerdere aanwijzingen in een regel niet. Er mag telkens slechts één compileraanwijzing in een regel staan.

## De DEFINE-aanwijzing

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

##### Definitie van symbolische constanten

Er 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 de toewijzing van significante omschrijvende elementen voor constanten wordt de leesbaarheid van de brontekst verbeterd. Indien al de constanten globaal gedefinieerd worden, is een programma ook gemakkelijker te onderhouden. Dat geldt in het bijzonder wanneer één en dezelfde constante meermaals in het programma benodigd wordt.

De definitie van een symbolische constante gebeurt op de volgende manier:

```
DEFINE bezeichner wert
```

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

Zo dient het voorbeeld hierboven beter

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

Te luiden.

##### Definitie van variabelen

Het basisbesturingssysteem van de *C-Control Station* stelt 24 byte-opslagcellen. Van het interne geheugen (RAM) ter beschikking van de gebruiker in zijn/programma's. In dit opslagbereik worden al de variabelen van een BASIC-programma opgeslagen. De 24 bytes kunnen al naargelang nodig ook per bit of als 16 bit "Integer" (Word) gebruikt worden.

In tegenstelling tot standaard BASIC moeten in CCBASIC al de door het programma gebruikte variabelen vóór hun eerste gebruik gedefinieerd worden. Daarbij dient het gegevenstype gespecificeerd te worden (bit, byte of Word) en kan (voor bits moet!) er een opslagcelnummer aangegeven worden. De gebruiker moet er zelf op letten dat er geen ongewenste overlappings bij de toewijzing van de geheugenplaatsen ontstaan omdat er in het andere geval over de variabelen onderling geschreven kan worden. Zo bijvoorbeeld reserveren bit[18], byte[2] en word[1] ieder een gedeelte van de cel 2 van het opslagbereik.

- Definitie van een bitvariabele:

```
DEFINE bezeichner BIT[nr]
```

Daarbij zijn voor nr waarden van 1 tot 192 (24 bytes met ieder 8 bit) toegelaten.

- Definitie van een bytevariabele met celnummer:

```
DEFINE bezeichner BYTE[nr]
```

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

- Definitie van een "Integer"-variabele met celnummer:

```
DEFINE bezeichner WORD[nr]
```

Daarbij zijn voor nr waarden van 1 tot 12 (één Word reserveert 2 bytes) toegelaten.

Indien bij byte- en Worddefinities de celvermelding [nr] weggelaten wordt, neemt de compiler de verdeling over het opslagbereik op zich.

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

Gelieve er dan op te letten dat er niet afwisselend bytes en Words gedefinieerd worden. De hierna volgende aanwijzingen leiden tot twee ongebruikte (weggegeven kostbaren!) bytes, tussen a en b evenals tussen c en d omdat Words in principe op de bytes 1,3,5,7,... enz. van de 24 bytes afgestemd worden.

Beter zou zijn,

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

Te schrijven.

De automatische verdeling van de variabelen over het geheugen door de compiler begint bij celnummer 1. Het hoger vermelde (beter) voorbeeld reserveert 6 bytes. Bij definitie van nog andere bits, bytes en Words met vermelding van celnummer moet er opnieuw op een ongewenste overlapping gelet worden.

Een reeds gedefinieerd omschrijvend element van variabele mag geen tweede keer gedefinieerd worden.

**Definitie van digitale poorten**

In CCBASIC wordt er op poorten en op variabelen beroep gedaan. Ook hier moet iedere gebruikte poort vooraf gedefinieerd zijn. In totaal zijn kan er op 16 poorten beroep gedaan worden.

- Definitie van een één van de 16 digitale poorten:

```
DEFINE bezeichner PORT[nr]
```

Daarbij zijn voor nr waarden van 1 tot 16 toegelaten.

Zoals in het hoofdstuk "Randapparatuur" beschreven, zijn de poorten vast met verschillende systeemcomponenten verbonden. Zo wordt meestal het hierna volgende definitieblok aangeboden:

```
DEFINE P1 PORT[ 1]
DEFINE P2 PORT[ 2]
DEFINE P3 PORT[ 3]
DEFINE P4 PORT[ 4]
DEFINE P5 PORT[ 5]
DEFINE P6 PORT[ 6]
```

```
DEFINE K1 PORT[ 7]
DEFINE K2 PORT[ 8]
```

```
DEFINE F1 PORT[ 9]
DEFINE F2 PORT[10]
DEFINE F3 PORT[11]
DEFINE F4 PORT[12]
```

```
DEFINE LED1 PORT[13]
DEFINE LED2 PORT[14]
DEFINE LED3 PORT[15]
DEFINE LED4 PORT[16]
```

**Definitie van analoge poorten**

```
DEFINE bezeichner AD[1]
```

Daarbij zijn voor nr waarden van 1 tot 8 toegelaten.

Ook de analoge poorten zijn in het systeem vast verdeeld. De gebruikelijke definitie dient als volgt te luiden:

```
DEFINE A1 AD(1)
DEFINE A2 AD(2)
DEFINE A3 AD(3)
DEFINE A4 AD(4)
```

```
DEFINE T1 AD(5)
DEFINE T2 AD(6)
```

```
DEFINE U1 AD(7)
DEFINE U2 AD(8)
```

## Overzicht van de commando's

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

## Mathematische en logische operatoren

- Basisrekentypes: + - \* /
- De modulusoperator **MOD** levert de rest van een "Integer"-deling,

```
a = 10 MOD 3
```

Levert bijvoorbeeld voor a de waarde 1 op.

- Vergelijkende operatoren: > (groter dan), < (kleiner dan), >= (groter of gelijk), <= (kleiner of gelijk), = (gelijk), <> (ongelijk)

Het resultaat van een vergelijkende handeling is ofwel -1 (niet 1!, vergelijking correct) of 0 (vergelijking foutief).

```
a = 10 < 3
```

Levert bijvoorbeeld voor a de waarde 0 op.

- Logische operatoren: **NOT** (negatie), **AND** ("En"-verbinding), **NAND** ("En"-verbinding met daaropvolgende negatie), **OR** ("Of"-verbinding), **NOR** ("Of"-verbinding met daaropvolgende negatie), **XOR** (exclusieve "Of"-verbinding)

De logische operatoren kunnen behalve voor de formulering van voorwaarden (meestal in verbinding met vergelijkende handelingen) ook voor binaire byte- of Wordmanipulaties gebruikt worden.

- Schuifoperatoren: **SHL** (naar links schuiven), **SHR** (naar rechts schuiven) worden voor een aritmetische verschuiving van bitstalen in byte- of Wordvariabelen gebruikt. Links van de operator staat de te verschuiven waarde, rechts het getal, hoeveel bits er verschoven dient te worden. Bij het schuiven naar links stemt iedere individuele verschuiving overeen met een vermenigvuldiging met 2, bij het schuiven naar rechts met een deling door 2.

```
a = 10 SHL 3
```

Stemt dus overeen

```
a = 10 * 2 * 2 * 2
```

En levert bijvoorbeeld voor a de waarde 80 op.

Gelieve erop te letten dat SHR aritmetisch schuift, m.a.w. met voortekens!

```
a = &B1000000000000000 SHR 1
```

Levert (alle waarden in binaire notatie) voor a de waarde &B1100000000000000 niet &B0100000000000000 op!

## Mathematische functies en commando's

De argumenten x en y, al naargelang de functie of het commando, zijn steeds termen (definitie zie hoger).

- De wortelfunctie **SQR**(x) levert een benadering voor de vierkantswortel uit het argument x. Daarbij worden de cijfers achter de komma afgesneden.
- De signumfunctie **SGN**(x) levert 1 op wanneer de waarde van het argument x groter dan 0 is en resulteert in -1 wanneer de waarde kleiner dan 0 is. Voor x = 0 is ook het resultaat van de SGN-functie gelijk aan 0.
- De maximumfunctie **MAX**(x,y) levert x op wanneer x > y is, anders y.
- De minimumfunctie **MIN**(x,y) levert x op wanneer x < y is, anders y.
- Het commando **RANDOMIZE** x initialiseert de interne pseudo-toevalgenerator van de besturingscomputer met de waarde van x. Eén en dezelfde initialisatiewaarde leidt steeds tot een identieke volgorde van getallen. De speciale vorm **RANDOMIZE TIMER** laadt de waarde van de vrijlopende timer in de generator.
- De toevalsfunctie **RAND** levert de volgende "Integer"-toevalswaarde van de pseudo-toevalgenerator. De toevalsgetallen worden na het vermenigvuldigende procédé met daaropvolgende modulusdeling uit de telkens vorige waarde tot stand gebracht.

## Hiërarchie van operatoren en functieoproepen

Bij de berekening van termen met operatoren en functies is hun hiërarchie van doorslaggevend belang. Gedeeltelijke afdrukken met operatoren van een hoge rang worden vóór afdrukken met operatoren van een lagere rang berekend (vergelijk met de rekenregel: "Puntrekening vóór streeprekening"). Bij operatoren van dezelfde rang volgt de berekening van links naar rechts.

Zoals in de mathematica gebruikelijk, kan er echter door het plaatsen van haakjes bijkomende invloed op de volgorde van de berekening uitgeoefend worden. CCBASIC ondersteunt maximaal 6 niveaus van haakjes.

In het kader van de overzichtelijkheid van een programma dienen echter "wilde" haakjesafdrukken vermeden en dienen complexe berekeningen over meerdere BASIC-regels verdeeld te worden.

De hierna volgende lijst toont de CCBASIC operatorenhiërarchie:

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

## Aanwijzingen voor de besturing van de programmastroom

- Lus

```
FOR variable = anfang TO ende STEP schrittweite
...
NEXT
```

De FOR-lus voert de aanwijzingen tot aan NEXT zolang uit tot de waarde van de variabele identiek aan de waarde van de term is. Vóór de eerste cyclus wordt de waarde van de term in het begin berekend en aan de lusvariabele toegewezen. In iedere fase wordt de waarde van de term "Stapwijdte" met de lusvariabele opgeteld. In de vorm

```
FOR variable = anfang TO ende
...
NEXT
```

Bedraagt de stapwijdte constant 1.

De waarden van de term "Einde" en van de term."Stapwijdte" worden met iedere luscyclus opnieuw berekend. Dat laat een uitgebreide controle van het programmaverloop toe.

FOR-lussen kunnen in elkaar geschoven worden. De diepte van het in elkaar schuiven is slechts door de lus-variabelen noodzakelijke geheugenplaats beperkt.

```
FOR v1 = anfang1 TO ende1
FOR v2 = anfang2 TO ende2
FOR v3 = anfang3 TO ende3
...
NEXT
NEXT
NEXT
```

Iedere FOR-lus mag tijdens het verloop van het programma uitsluitend via haar eigen NEXT-aanwijzing lopen. Volgende brontekst kan weliswaar gecompileerd en in de *C-Control Station* geladen worden, maar zal niet zoals mis-schien verwacht functioneren:

```
FOR v1 = anfang1 TO ende1
...
GOTO anothernext
...
NEXT
```

```
FOR v2 = anfang2 TO ende2
...
#anothernext
NEXT
```

Gelieve bovendien op het waardebereik van lusvariabele en term "Einde" te letten!

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

Wordt een eindloze lus omdat v als bytevariabele nooit de waarde 1000 kan bereiken, maar al na 255 terug naar 0 rolt.

- Voorwaardelijke uitvoering

```
IF bedingung THEN anweisung1
Of
IF bedingung THEN anweisung1 ELSE anweisung2
```

De IF...THEN...ELSE-constructie maakt de aanpassing van de programmastroom aan voorwaarden aangaande de inschakelduur van het programma mogelijk. Als voorwaarde dient een willekeurige term gebruikt te worden. Indien de berekening ervan een waarde, die niet gelijk aan 0 is, oplevert, geldt de voorwaarde als zijnde vervuld en wordt de aanwijzing 1 uitgevoerd. Indien er bijkomend een ELSE en een tweede aanwijzing aangegeven worden, wordt deze aanwijzing alternatief uitgevoerd wanneer de berekende term een waarde oplevert, die gelijk aan 0 is.

De volledige IF...THEN...ELSE-constructie moet in een regel van de brontekst staan. Aanwijzingsblokken (meerdere aanwijzingen) na THEN en ELSE zijn niet geoorloofd.

- Sprongaanwijzing

```
GOTO label
```

Met de GOTO-aanwijzing kan de besturingscomputer verzocht worden, met de programma-afhandeling op een bepaald punt door te gaan. Als doel van de sprong wordt er een omschrijvend element van het label aangegeven. Het sprongdoel kan zich vóór of na de GOTO-aanwijzing in de brontekst bevinden.

- Oproep en terugkeer uit een subroutine

De oproep van een subroutine volgt met de aanwijzing

```
GOSUB label
```

Daarbij is het label het beginpunt van de subroutine.

In de zogenaamde subroutines zijn er programmahoofdstukken samengevat, die meervoudig tijdens het verloop van de programma-afhandeling benodigd worden. Een subroutine begint steeds met een label, bevat dan één of meerdere aanwijzingen en tot slot een

```
RETURN
```

Na de RETURN wordt de programma-afhandeling met de aanwijzing na de GOSUB voortgezet. De programma-afhandeling mag zonder een voorafgaande GOSUB nooit aan een RETURN-aanwijzing geraken.

De maximaal toegelaten diepte bij het in elkaar schuiven bij het oproepen van subroutines uit subroutines is vier.

```
#hauptprogramm
GOSUB subl
```

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

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

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

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

Door middel van RETURN kan er ook een getallenwaarde aan de oproeper teruggegeven worden.

```
RETURN term
```

Zodoende kunnen wederkerende of omvangrijkere statusbepalingen in een subfunctie geïntegreerd worden.

```
define x word
define y word
define z word
```

```
...
x = calc
...
#calc RETURN y + z / (y + z)
```

- Waardegestuurde programma-afsplitsing

```
ON variable GOTO label0,label1.....labeln
Of
ON variable GOSUB label0,label1.....labeln
```

Afhankelijk van de waarde van de selectorvariabele volgt er een programma-afsplitsing of een oproep van subrou-tines bij de opgesomde sprongpunten. Indien de waarde 0 is, wordt er naar label 0 gesprongen, bij een waarde gelijk aan 1 naar label 1 enz. Indien de variabelenwaarde negatief of groter dan het aantal vermelde sprongdoelen is, wordt de programma-afhandeling zonder afsplitsing voortgezet.

- Programma-einde

```
END
```

Indien de besturingscomputer tijdens het verloop van de programma-afhandeling aan de aanwijzing END geraakt, wordt de programma-afhandeling beëindigd. Het systeem blijft dan in een inactieve toestand stilstaan. Nu kan er een nieuw gebruikersprogramma overgedragen worden of kan de uitvoering door middel van de schakelaar "Start" gestart worden.

- Vertraging van de programmastroom

De aanwijzing

```
WAIT conditionterm
```

Onderbreekt de uitvoering van het programma zolang tot de berekening van de term "condition" een waarde, die niet gelijk aan 0 is, oplevert.

```
define F1 port[9]
...
WAIT F1
```

In dit voorbeeld wordt er zolang gewacht tot er door de digitale poort 9 (toets F1 aan de *Station*) een HIGH-niveau gelezen wordt.

Het commando PAUZE onderbreekt de uitvoering van het programma gedurende een bepaalde tijdspanne. De berekende waarde van de parameterterm gaat als vermenigvuldigende factor met de basiseenheid van 20 milliseconden in de vastlegging van de pauzetijd in.

```
PAUSE term
```

Bijvoorbeeld door het commando

```
PAUSE 50
```

De uitvoering van het programma gedurende circa 50\*20 milliseconden = 1 seconde onderbroken. De maximale tij-dafwijking van de effectieve pauze van de aangegeven waarde bedraagt daarbij omwille van het principe +/- 20 milliseconden.

## Communicatie via het seriële interface

- Gegevensuitvoer

De gegevensuitvoer gebeurt als tekst door middel van het seriële interface van de *C-Control Station*. Indien er door middel van een interfacekabel bijvoorbeeld een PC met een terminalprogramma aangesloten is, kunnen de uitgevoerde gegevens daar weergegeven worden.

```
PRINT term
```

Voert het resultaat van de berekening van term uit.

```
PRINT "text"
```

Draagt de in aanhalingstekens staande tekst over.

In beide gevallen wordt er aan de overdracht een regeltoevoerteken bijgevoegd, dat het terminalprogramma zoekt, de volgende uitvoer in de volgende beeldschermregel door te voeren. De regeltoevoer kan onderdrukt worden wanneer er aan het commando PRINT na de parameter (term of "tekst") een puntkomma toegevoegd wordt.

```
PRINT term
Of
PRINT "text"
```

CCBASIC ondersteunt bovendien nog andere uitvoer met een commando PRINT, waarbij de individuele parameters door een komma of een puntkomma gescheiden worden. Een komma voegt in de uitvoer een tabulatorteken in, dat in overeenstemming met de instellingen in het terminalprogramma als een aantal spaties op het beeldscherm verschijnt. Indien er twee keer uitvoer zonder tussenruimte op elkaar volgt, dienen deze in het commando PRINT door een puntkomma gescheiden te worden.

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

Eén enkel commando PRINT zonder parameter voert slechts één regeltoevoer uit.

<b>PRINT</b>
<span></span>
• Gegevensinvoer
<span></span>
Met het commando
<b>INPUT</b> <i>variable</i>

kan er een "Integer"-waarde door het seriële interface gelezen en voor de daaropvolgende verdere bewerking in een variabele opgeslagen worden.

De waarde wordt in een terminalprogramma aan een PC ingevoerd en na het indrukken van de toets ENTER via een interfacekabel naar de *C-Control Station* overgedragen.

Het commando INPUT wacht zolang tot er een complete gegevensoverdracht door de terminal ontvangen werd. Indien het commando INPUT opgeroepen wordt, zonder dat er een gegevensoverdracht door de terminal volgt, zal het programma zonder einde op deze plaats blijven stilstaan! Hier helpt dan alleen nog maar de schakelaar "Terugstelling" ("Reset") en de daaropvolgende nieuwe start van de Station.

- Commando per byte door middel van het seriële interface

Terwijl PRINT en INPUT korte tekenkettingen voor de weergave van een numerieke waarde zenden c.q. verwachten, kan het nuttig zijn om individuele bytes serieel over te dragen. Daarvoor biedt CCBASIC de commando's PUT en GET.

<b>PUT</b> <i>term</i>
<span></span>
Zendt de berekende waarde van een term. Zo nodig, wordt het resultaat voordoen op het byte-waardebereik (0..255) gereduceerd.
<b>GET</b> <i>variable</i>

Wacht op een serieel ontvangen byte en slaat de waarde vervolgens in de aangegeven variabelen op.

- Bijkomende interfacecommando's en -functies

Zoals beschreven, wachten INPUT en GET soms oneindig op de ontvangst van seriële gegevens. Indien het "ophangen" van het programma op deze manier voorkomen wordt, kan vóór iedere door INPUT of GET door oproep van de statusfunctie **RXD** vastgesteld worden, of er ontvangen gegevens ter beschikking staan. De functie levert in dit geval de waarde 1. Indien het interfacebuffer leeg is, is het functieresultaat gelijk aan 0.

<pre>... if <b>RXD</b> then GET thebyte ...</pre>
<span></span>
Het vooraf ingestelde overdrachtcijfer van het seriële interface bedraagt voor zender en ontvanger 9600 bit per seconde (transmissiesnelheid). Met het commando BAUD kunnen er echter ook andere cijfers ingesteld worden. CCBASIC Bevat daarvoor een aantal vooraf gedefinieerde constanten: <b>R1200</b> , <b>R2400</b> , <b>R4800</b> , <b>R9600</b> voor de cijfers 1200 tot 9600 bit per seconde.
<b>BAUD</b> <b>R2400</b>

Schakelt bijvoorbeeld zender en ontvanger op het cijfer van 2400 bit per seconde over. In principe zijn er ook andere (ook voor zender en ontvanger uiteenlopende) cijfers dan de vooraf gedefinieerde cijfers mogelijk. De overdrachtcijfers van het seriële interface worden door deling uit een interne cyclus van de microprocessor van de C-Control/*BASIC* besturingscomputer afgeleid. De aan het commando BAUD over te dragen bytewaarde bevat de noodzakelijke delerwaarden 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 gezamenlijke voordeler NP. NP kan de waarden 1, 3, 4 en 13 aannemen. De hierna volgende tabel toont de daarvoor noodzakelijke instellingen voor NP1 en NPO:

<b>Voordeler</b>	<b>NP1</b>	<b>NPO</b>
1	0	0
3	0	1
4	1	0
13	1	1

NT (bit 3 tot 5) en NR (bit 0 tot 2) bepalen nog andere delerwaarden, gescheiden voor zender (NT) en ontvanger (NR) en dit in overeenstemming met de hierna volgende codering:

<b>Deler</b>	<b>NT2 c.q. NR2</b>	<b>NT1 c.q. NR1</b>	<b>NT2 c.q. NR2</b>
1	0	0	0
2	0	0	1
4	0	1	0
8	0	1	1
16	1	0	0
32	1	1	0
64	1	1	0
128	1	1	1

Het overdrachtcijfer van de zender wordt aan de hand van de hierna volgende formule berekend

**Zendercijfer = 125000 / (NP \* NT),**

Overeenkomstig met die van de ontvanger

**Ontvangercijfer = 125000 / (NP \* NR).**

De andere interfaceparameters - 8 gegevensbits, geen pariteitbit, 1 stopbit – zijn vast en kunnen zijn gewijzigd worden.

### Bestandfuncties

De bestandfuncties laten de notitie van meetwaarden of andere gegevens toe of kunnen gebruikt worden voor de opslag van informatie, die na uitval van de bedrijfsspanning opnieuw in de programmavariabelen geladen dienen te worden. Het opslagbereik in de EEPROM-chip na het gebruikersprogramma – meestal het grootste gedeelte – staan voor dit doeleinde ter beschikking. Het opslagbereik wordt als een bestand beheerd, waarop er al lezend of schrijvend

beroept gedaan kan worden nadat het met het gepaste hulpmiddel geopend werd. Het commando om het bestand te openen luidt als volgt:

<b>OPEN# FOR WRITE</b>
Of
<b>OPEN# FOR APPEND</b>
Of
<b>OPEN# FOR READ</b>

Daarbij betekent WRITE het openen voor het schrijven, waarbij er over eventuele oude notities geschreven wordt. APPEND staat voor het openen voor het schrijven met bijlagen van de nieuwe aan de oude notities en READ betekent het openen om de notities uit te lezen.

Er kunnen uitsluitend "Integer"-waarden opgeslagen en gelezen worden. Iedere waarde reserveert dus 2 bytes in het EEPROM. Het schrijven en lezen gebeurt met de commando's

<b>PRINT# term</b>
<span></span>
, waarbij het berekende resultaat van de term opgeslagen wordt c.q.
<b>INPUT# variable</b>
<span></span>
, waarbij "Variabele" een gedefinieerde "Integer"-variabele van het programma betekent.

Het schrijven in het bestand en het lezen uit het bestand gebeurt strikt sequentieel. Daarvoor wordt er intern een bestandpointer beheerd, die – telkens wanneer er beroep op gedaan wordt – met 1 verhoogd wordt.

Telkens vooraleer er geschreven wordt, moet er nagegaan worden, of er nog voldoende plaats in het EEPROM voor de opname van de gegevens beschikbaar is. Daarvoor kan de functie **FILEFREE** opgeroepen worden. Deze levert als resultaat de grootte van het nog vrije geheugen (in Words). Het hierna volgende voorbeeld toont de toepassing van de functie

```
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
```

Telkens vooraleer er gelezen wordt, dient er nagegaan te worden, of er nog andere gegevens genoteerd zijn. De functie daarvoor luidt **EOF** ("end of file"). Het resultaat ervan is 1 wanneer er in het bestand geen andere gegevens beschikbaar zijn, in het andere geval 0. De oproep van de functie EOF dient het uitlezen van gegevensblokken op dezelfde manier als bij het schrijven van de gegevens in te lijsten.

Bij het hoger vermelde voorbeeld zou als
<pre><b>IF</b> NOT <b>EOF</b> THEN GOSUB readblock ... #readblock <b>INPUT#</b> a <b>INPUT#</b> b <b>INPUT#</b> c <b>RETURN</b></pre>

behoeren.

Na het beëindigen van het gebruik van een bestand dient het bestand onmiddellijk terug gesloten te worden. Pas dan zijn de gegevens tegen een spanningsuitval of een terugstelling ("Reset") beveiligd. Het commando daarvoor luidt

<b>CLOSE#</b>
<span></span>
en heeft geen parameter.
<b>Poortcommando's</b>
• Het overschakelcommando <b>TOG</b>

In principe volgt de toegang tot de poorten van de besturingscomputer zoals de toegang tot variabelen. Om een digitale poort P in te schakelen, schrijft men

<b>P = 1</b>
En
<b>P = 0</b>

, om deze uit te schakelen.

Om de poort over te schakelen (AAN na UIT; UIT na AAN), kan men schrijven

<b>P = NOT P</b>
<span></span>
Of het commando
<b>TOG</b> <b>P</b>

Gebruiken. TOG staat voor het Engelse woord "toggle". Het commando TOG heeft weinig plaats in het EEPROM nodig en wordt sneller dan de klassieke NOT-P constructie uitgevoerd. De poortvariabele P mag bij het commando TOG uitsluitend voor één enkele digitale poort en niet voor een byte- of Wordpoort staan.

- Deactiveren van een poort met **DEACT**

Zodra er aan een poortvariabele voor het eerst een waarde toegewezen wordt, schakelt de besturingscomputer de bijbehorende hardwarestructuren in de processorchip (transistoren) naar uitgangsmodus. In overeenstemming met de aangesloten schakeling vloeit er dus respectievelijk stroom uit en naar de processor (max. 10 mA toegelaten). Het commando

<b>DEACT</b> <i>portvar</i>
<span></span>
Deactiveert de aangegeven poort. Dat betekent dat de poort in een toestand met veel ohm geschakeld wordt en in de ingangsmodus werkt.

- Het commando **IMPULSEN**

Met het commando
<b>PULSE</b> <i>portvar</i>

wordt er een impuls met een breedte van enkele milliseconden aan de als "portvar" bestempelde poort uitgevoerd. Indien de poort vóór uitvoering van het commando IMPULSEN op low (=0) staat, wordt er een High-Puls (0-1-0), in het andere geval een Low-Puls (1-0-1) uitgevoerd.

### Definitie en toepassing van gegevenstabellen

In standaard BASIC dienen DATA-regels voor het deponeren van constante gegevensblokken, waarop er dan sequentieel beroep gedaan kan worden. CCBASIC ondersteunt geen DATA-regels, biedt echter een in ruime mate flexibel werktuig voor de definitie en voor de toegang tot gegevensblokken. Constante gegevens kunnen in de vorm van tabellen gearchiveerd worden. Iedere tabel krijgt een omschrijvend element (tablename) toegewezen en kan een willekeurig aantal boekingen bevatten voor zover het programmegeheugen plaats biedt. Iedere gegevensinvoer (Cx) wordt als "Integer"-waarde gearchiveerd en neemt zodoende twee bytes in beslag. Daarbij kunnen de gegevens rechtstreeks in de brontekst vermeld worden

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

Of door de CCBASIC-compiler uit een extern tekstbestand geïmporteerd worden.

<b>TABLE</b> <i>tablename</i> "tabfilename"
<span></span>
De tabeldefinities moeten steeds op het einde van een programma, achter het commando END staan omdat de gegevens doorlopend achter de vorige codebytes in de EEPROM-geheugenchip gearchiveerd worden. De programma-afhandeling mag nooit via tabelgegevens gebeuren omdat de gegevens anders als BASIC commando's geïnterpreteerd zouden worden, het geen zeker tot het uitvallen van het systeem zou leiden. De toegang tot de tabelgegevens volgt met het commando
<b>LOKTAB</b> <i>tablename,index,variable</i>

"tablename" staat synoniem voor een geldige tabel, vóór "index" kan een willekeurige term staan en de variabele staat synoniem voor de opslagcel, in dewelke het resultaat opgeslagen dient te worden. De berekende waarde van de indexterm mag niet negatief zijn en maximaal N-1 bedragen wanneer de aangeduide tabel N boekingen heeft. Indien de index de waarde 0 vertoont, wordt C0 in de aangegeven variabele opgeslagen, voor index gelijk aan 1 C1 en zo verder. Het hierna volgende voorbeeld voert de inhoud van een tabel serieel uit

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

Op het beeldscherm van het terminalprogramma dient te verschijnen

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

Uiterst nuttig blijken tabellen bij de omzetting van A/D-waarden in echte fysieke grootten te zijn. Een omzettingsta-bel heeft dan doorgaans 256 boekingen. De gemeten A/D-waarde gaat dan als tabelindex in de bepaling van de fysieke grootte op.

### Toegang tot de onvertraagde klok

Om de stand van de interne onvertraagde klok uit te lezen en te bepalen, zijn de hierna volgende 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...23)
MINUTE	minuut (0...59)
SECOND	seconde (0...59)

Gelieve in acht te nemen dat de interne klok tijdens de toegang verder loopt.

De secondewaarde dient daarom steeds eerst uitgelezen te worden. Indien deze op 59 staat, moet na het lezen van de laatste relevante tijdinformatie (bijvoorbeeld YEAR) de secondewaarde nogmaals gelezen en op =0 getest worden. In dit geval dient het lezen van de onvertraagde klok herhaald te worden omdat er een nieuwe minuut aangebroken is (extreem geval oudejaarsdag met het verder schakelen van alle cijfers qua tijdstip en datum).

Het jaartal wordt in de C-Control System slechts met twee cijfers opgeslagen. Indien de analyse van het jaartal qua algoritmes een rol speelt, gelieve dan de nakende overgang van 1999 naar, die in de C-Control van 99 naar 0 blijkt te zijn, in acht te nemen.

### Timer

De interne 20-milliseconden timer kan door middel van het vooraf gedefinieerde, omschrijvende element **TIMER** uit-gelezen worden. De timer is vrijlopend en kan niet ingesteld of teruggesteld worden.

### Uitvoer van klanken met BEEP

De *C-Control Station* is intern met een piëzo-geluidsomzetter uitgerust, via dewelke er signaalklanken uitgevoerd kunnen worden. Het commando daarvoor luidt

<b>BEEP</b> <i>ton, tTon, tPause</i>
<span></span>
Voor de drie parameters kunnen er constanten of termen gebruikt worden. Daarbij bepaalt "ton" de toonhoogte aan de hand van de formule
<b>toon = 250000 / frequentie [Hz]</b>
<span></span>
"tTon" bepaalt de duur van de klank en "tPause" de pauze na de klank. De eenheid voor de tijdaanduiding bedraagt 20 milliseconden. Het commando
<b>BEEP</b> <i>568, 10, 3</i>

Voert dus voor 10\*20=200 milliseconden een klank van ongeveer 440 Hz (kamerklank A) uit en maakt daarna een pauze van 3\*20=60 milliseconden. Indien er na een BEEP niet nog een andere BEEP volgt, kan de pauze ook op 0 geplaatst worden. Indien er voor de klanklengte 0 aangegeven is, wordt er een continue klank tot stand gebracht. De klankgenerator schakelt de klank in en gaat met de afhandeling van het BASIC-programma verder. Met de waarde 0 voor "ton" kan de klankgenerator terug uitgeschakeld worden.

### Frequentiemeting met FREQ en FREQ2

De frequentiemeting van de *C-Control Station* is gebaseerd op het impulstelprincipe bij een poorttijd van 1 seconde. De meting gebeurt voortdurend op de achtergrond en dit evenwijdig met de afhandeling van het BASIC-programma.

De in de secondecyclus getelde impulsen stemmen aldus rechtstreeks met de frequentie in Hz overeen.

De standaardingang voor de frequentiemeting (klemdraadhouder 35) kan met FREQ2 opgeroepen worden.

<b>x = FREQ2</b>
<span></span>
Deze heeft een meetbereik tot circa 30000 Hz bij een fout <1%.

Indien er op de DCF77-ingang geen actieve antenne aangesloten is, kan er met deze ingang alternatief een frequentiemeting plaatsvinden, waarvan het resultaat met de functie FREQ opgeroepen kan worden.

<b>x = FREQ</b>
-----------------

Het meetbereik reikt tot ongeveer 5 kilohertz met een meetfout van minder dan één procent. Daarna wordt het resultaat alsmaar onnauwkeuriger.

- Stroombesparende modus met SLOWMODE

Toepassingen, die geen hoog rekenvermogen nodig hebben, kunnen door de oproep van het commando
<b>SLOWMODE ON</b>
<span></span>
de interne cyclus van de microprocessor vertragen (1/16). Indien tijdens het verloop van het verloop opnieuw een hogere snelheid noodzakelijk is, kan met
<b>SLOWMODE OFF</b>

de oorspronkelijke toestand terug hersteld worden.

Opgelet: programma's, die seriële gegevensoverdrachten gebruiken, mogen de SLOWMODE niet activeren omdat de ingestelde overdrachtcijfers met de processorcyclus gereduceerd worden.

De reductie van de processorcyclus leidt tot een vermindering van de krachtontneming en kan de overbruggings-duur van de accu bij het uitvallen van de netspanning verlengen. Dat heeft echter alleen maar zin indien tegelijkertijd al de LED's en relais uitgeschakeld zijn.

### Voorbeeld bij de programmering met CCBASIC

Het volgende kleine voorbeeld dient het basisprincipe van de programmering met CCBASIC te verduidelijken. De hier verstrekte informatie stelt u in staat om de op de CD terug te vinden, bijkomende voorbeelden te begrijpen en aan uw wensen aan te passen om zodoende stapsgewijs een volledig inzicht in de BASIC-programmering van de *C-Control Station* te krijgen.

Verondersteld wordt dat u met de toets F1 aan de module het relais K1 afwisselend wenst in en uit te schakelen. Een druk op de toets dient met een pieptoon beantwoord te worden. Evenwijdig daarmee dient op het tempo van minuten het relais K2 in- en uitgeschakeld te worden.

Start C-Control/BASIC (CCEW32D.EXE) onder Windows. Open een nieuw bestand met het menucommando "Bestand/nieuw". Sla het bestand op onder de naam "TEST1.BAS".

### De brontekst

Begin nu met de bewerking van de brontekst.

Allereerst moeten de gebruikte randapparatuureenheden gedefinieerd worden:

<pre>define K1 port[7] define K2 port[8] define F1 port[9]</pre>
<span></span>
Voor het overige definiëren wij een variabele, waarvan de toepassing nog getoond wordt.
<b>define</b> oldsec byte

Een programma bestaat meestal uit initialisatie, hoofdilus en subfuncties. In ons voorbeeld willen wij als initialisatie-toestand het relais K1 inschakelen en het relais K2 uitschakelen.

<b>K1 = ON</b>
<b>K2 = OFF</b>

De hoofdilus in ons voorbeeld dient het label "loop" te dragen. Deze omvat de oproep van de toets F1 en de minutucyclusfunctie voor relais K2. In het begin van een nieuwe minuut is de secondewaarde steeds 0 en dit één secon-de lang. Om het relais tijdens deze seconde niet meermaals om te schakelen, wordt de vaststelling van het begin van de minuut aan de wissel van de secondewaarde gekoppeld.

```
#loop
if not F1 then gosub toggleK1
if (0 = second) and (second <> oldsec) then tog K2
goto loop
```

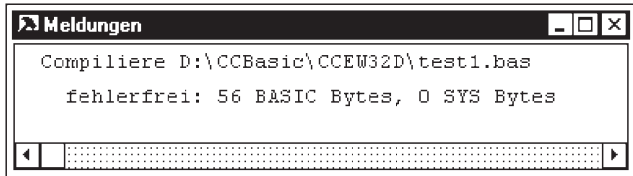
Let op de negatie van de toetswaarde. Omdat de toetsen qua hardware met "Pullup"-weerstand na 5V bedraad zijn, levert de oproep ervan in een niet-ingedrukte toestand "High"- en in een gedrukte toestand "Low"-niveaus (opmerking: de toetsoproep in het voorbeeld bij de grafische programmering bevat reeds de negatie al!)

Indien F1 als zijnde ingedrukt gedetecteerd wordt, splitst het programma naar de subfunctie "toggleK1" af. De subfunctie schakelt aanvankelijk het relais K1 over. Om de toets te vergrendelen wordt er dan op 100ms (5x20ms) en dan op het loslaten van de toets gewacht. Vervolgens keert de subfunctie terug naar de hoofdlus.

```
#toggleK1
tog K1
pause 5
wait F1
return
```

## Compileren en simulatie

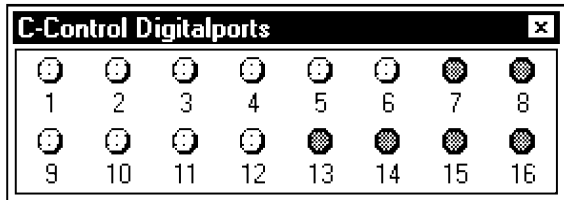
Wanneer u de brontekst ingevoerd hebt, compileert u het programma door middel van het menucommando "Ontwikkeling/compilieren". In het meldingvenster krijgt u de status van de transmissie, eventueel een lijst met foutmeldingen. Ons voorbeeld dient foutloos gecompileerd te worden, zoals afbeelding 20 toont.



Afbeelding 20

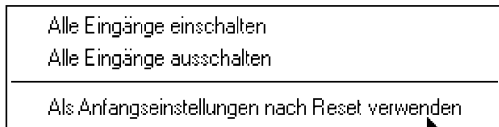
Om het programma te simuleren moeten wij aanvankelijk terug naar het venster met de brontekst overschakelen en dit ofwel door het sluiten van het meldingvenster door middel van het menu "Venster" ofwel door het indrukken van "Ctrl+Tab".

Open het simulatorvenster "C-Control Digitalports" via het menu "Afbeelding". In tegenstelling tot de grafische programmering biedt de BASIC-programmeersoftware geen directe afbeelding van de *C-Control Station* hardware. Het venster "C-Control Digitalports" stelt al de 16 digitale poorten voor. Zoals reeds beschreven, zijn de poorten 1 tot 6 met de klemdraadhouders P1 tot P6 verbonden en door middel van "Pullup"-weerstand in de schakeling op "High"-niveaus gelegd net zoals de poorten 9 tot 12, die met de folietoetsen F1...F4 verbonden zijn. Schakel in het simulatorvenster "CControl Digitalports" allereerst deze poorten in doordat u bij een ingedrukte toets "Shift" op de LED's 1...6 en 9...12 klikt zodat deze in een lichtgroene kleur verschijnen.



Afbeelding 21

Klik dan met de rechtse muistoets op het venster en kies in het verschijnende menu "Als beginpositie na terugstelling gebruiken" (zie afbeelding 22). Nu stemt het simulatorvenster met de Hardwareverhoudingen van de *C-Control Station* overeen. Dit instellingsprocédé moet u slechts één keer doorvoeren. Uw instellingen worden bij het einde van het programma automatisch opgeslagen.

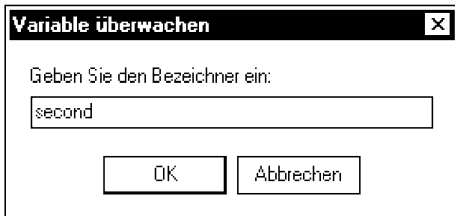


Afbeelding 22

Nu kunt u de simulator starten. Activeer het venster "Editorvenster". Kies in het menu "Simulator" het commando "Uitvoeren".

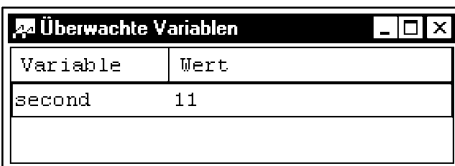
De LED's 7 en 8 krijgen een rode rand, de LED 7 verschijnt in het lichtrood. Dat betekent dat beide poorten als uitgang gebruikt worden, poort 7 is ingeschakeld. Zoals reeds beschreven, zijn de poorten 7 en 8 met de relais K1 en K2 verbonden. Klik nu op de LED 9, die met de folietoets F1 overeenkomt. Zoals gepland, kan de toestand van relais K1 omgeschakeld worden.

Open nu via het menu "Afbeelding" het venster "Geobserveerde variabelen". Druk de toets "Einf" in en voer in de volgende dialoog "second" in:



Afbeelding 23

Na de "O.K." moet het hierna volgende venster te zien zijn:



Afbeelding 24

In het venster "Geobserveerde variabelen" kunt u het waardeverloop van systeemvariabelen en eigen variabelen volgen, in dit voorbeeld de secundewaarde van de onvertraagde klok. Laat de simulator een tijdje in werking. Observeer bij de wissel van de seconden van 59 naar 0, m.a.w. in het begin van de minuut, de overschakeling van de LED 8, die de schakeltoestand van het relais K2 vormt.

Nu is de tweede functie van ons voorbeeldprogramma aangetoond. Houd de simulator aan en stel deze terug. De nodige commando's vindt u in het menu "Simulator".

## Overdragen en starten van het programma

Na de functionele test kan het voorbeeldprogramma naar de Station overgedragen worden.

- Zorg er beslist voor dat de Station op een betrouwbare manier van bedrijfsspanning voorzien en door middel van een seriële interface met de PC verbonden is.

- Druk de schakelaar "Terugstelling" ("Reset") aan de Station in.

- Stel via "Opties / omgeving / simulator en lader" het gebruikte seriële interface in.

- Activeer het editorvenster "test1.bas".

- Kies het menucommando "Ontwikkeling / In C-Control-Unit overdragen".

Tijdens de overdracht knippert de lichtgevende TX-diode aan de RS232-Klemdraadhouders van de Station. In het meldingvenster op het beeldscherm wordt u van het succes van de overdracht of van eventuele fouten op de hoogte gebracht. Indien er een fout gerapporteerd wordt, gelieve dan de verbindingsteiding, de instelling van het seriële interface en de spanningsvoorziening van de Station te controleren. Vóór een nieuwe overdracht poging moet u het editorvenster opnieuw activeren.

Indien de overdracht succesvol was, is de *C-Control Station* nu geprogrammeerd.

- Druk de schakelaar "Start" in om het kleine testprogramma uit te voeren.

Zoals geprogrammeerd en gesimuleerd, ziet u dat het relais K1 in- en uitgeschakeld wordt telkens wanneer de toets F1 aan de module ingedrukt wordt. Op het tempo van minuten schakelt het relais K2 om.

## Voorbeeld van de CD

In de SAMPLES-subregisters van de BASIC-software en van de grafische software vindt u een aantal voorbeelden, die voor u de programmering van de *C-Control Station* duidelijk maken en een aanzetting van een oplossing voor tal van toepassingen bieden.

## Technische gegevens

Bedrijfsspanning	230V – netspanning aan klemdraadhouders 16 en 17 of gestabiliseerd 12V-voedingsapparaat aan klemdraadhouders 7 en 8
Max. krachtontneming	2,5W
Max. toegelaten stroom uit klemdraadhouder 6 (5V)	$I_{6,max} = 50\text{mA}$
Max. toegelaten stroom uit klemdraadhouder 9 (12...24V)	$I_{9,max} = 50\text{mA}$
Max. poortstroom aan klemdraadhouders 28 tot 33 (P1...P6)	$I_{28,max} \dots I_{33,max} = \text{ieder } 10\text{mA}$
Max. totale stroomafgifte	$I_{6,max} + I_{9,max} + I_{28,max} + \dots + I_{33,max} = 100\text{mA}$
Accubufferstroom uit klemdraadhouder 7	Circa 5mA
Relaisschakelvermogen (K1, K2)	Max. 230V~, kortstondig 8A inschakelstroom, max. 6A continue stroom
Temperatuurmeetbereik (T1,T2)	-25,0°C...102,5°C 0,5K resolutie, max. fout $\pm 2,5\%$
Meetbereik A/D-poorten (A1...A4)	0...2,55V 0,01V resolutie, max. fout $\pm 1$ digit
Frequentiemeetbereik (klemdraadhouder 35)	1Hz...30000Hz
Toegelaten omgevingsomstandigheden	0...40°C, 20...60% relatieve luchtvochtigheid

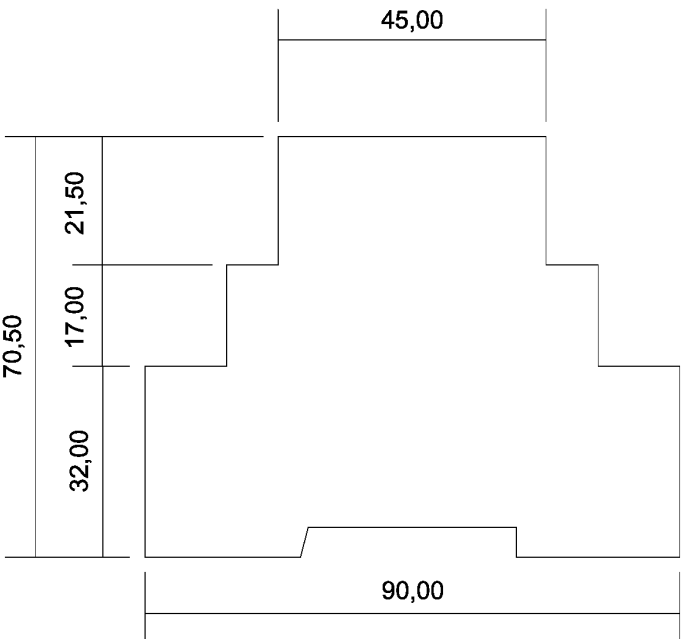
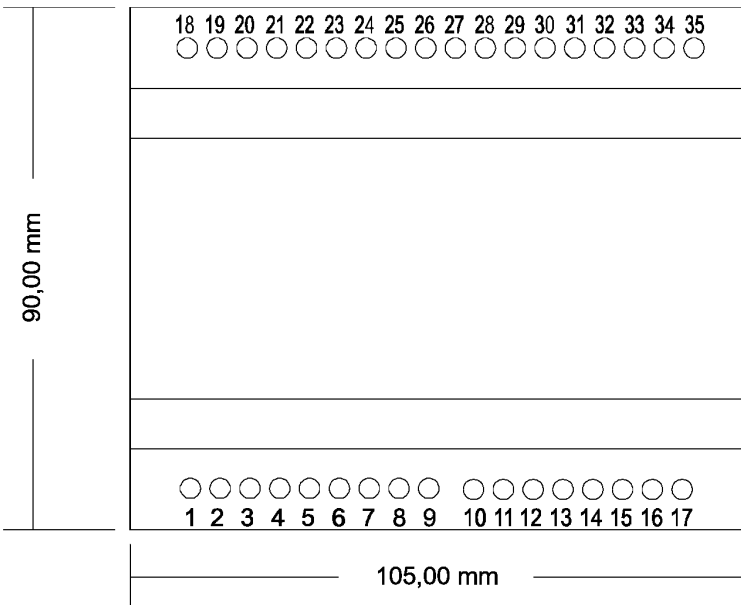
## Opmerking over de elektromagnetische verdraagbaarheid

De *C-Control Station* mag bij een ononderbroken werking niet door middel van een gelijkspanningsvoedingsapparaat van 2V gevoed worden. In deze configuratie ontstaat er een verhoogde gevoeligheid ten opzichte van elektrostatische ontladingen en net-storingsspanningspieken, die tot de uitschakeling van de microcontroller kunnen leiden en vervolgens een terugstelling ("Reset") en een nieuwe start van het gebruikersprogramma noodzakelijk maken.

De *C-Control Station* moet bij een ononderbroken werking steeds door middel van een netspanningsaansluiting van 230V of een accu van 1 2V gevoed worden.

Conrad Electronic GmbH  
D-92240 Hirschau

## Buitenafmetingen



Deze gebruiksaanwijzing is een publicatie van Conrad Electronic GmbH, Klaus-Conrad-Straße 1, D-92240 Hirschau/Duitsland  
Deze gebruiksaanwijzing voldoet aan de technische eisen bij het ter perse gaan. Wijzigingen in techniek en uitrusting voorbehouden.