

Station C-Control

Code : 125113

Cette notice fait partie du produit. Elle contient des informations importantes concernant son utilisation. Tenez-en compte, même si vous transmettez le produit à un tiers.

Conservez cette notice pour tout report ultérieur !

Note de l'éditeur

Cette notice est une publication de la société Conrad, 59800 Lille/France. Tous droits réservés, y compris la traduction. Toute reproduction, quel que soit le type (p.ex. photocopies, microfilms ou saisie dans des traitements de texte électronique) est soumise à une autorisation préalable écrite de l'éditeur.

Reproduction, même partielle, interdite.

Cette notice est conforme à l'état du produit au moment de l'impression.

Données techniques et conditionnement soumis à modifications sans avis préalable.

© Copyright 2001 par Conrad. Imprimé en CEE. XXX/01-05/SC

The logo for Conrad, featuring a stylized 'C' followed by the word 'ONRAD' in a bold, italicized sans-serif font.

Mode d'emploi

Remarque concernant la compatibilité électromagnétique

La station C-Control ne doit pas être reliée à un bloc d'alimentation de tension continue de 12 V. Cela développerait une sensibilité aux interférences émanant de charges électrostatiques et de crêtes de tension parasite lors du réglage du microcontrôleur, et nécessiterait une réinitialisation et un redémarrage du programme de l'application.

Cette station C-Control doit fonctionner via une connexion de tension de secteur de 230 V ou via une batterie 12 V.

Attention ! A lire absolument !

Avant de mettre en marche cette station ou tout autre appareil relié à une source électrique, veuillez lire attentivement cette notice dans son intégralité ! Elle vous permettra d'utiliser correctement l'appareil et d'appréhender les risques possibles.

La garantie ne couvre pas les dommages résultant de la non-observation des présentes instructions. Conrad Electronic décline toute responsabilité pour les dommages qui en résulteraient directement ou indirectement.

Sommaire

Attention ! A lire absolument !

Sommaire

Introduction	(p4)
Garantie	
Service	
Description du produit	(p2)
Domaine d'application	
Précautions d'emploi	
Capacités de puissance	
Utilisation	(4)
Programmation du module	
Montage	
Exécution du programme d'une application	
Eléments de réglage	(5)
Touches	
Diodes LED	
Branchements	(6)
Branchements basse tension	
Connexions réseau	
Connexion pour programmer le module	
Etat du système	(p10)
Mode reset (réinitialisation)	
Mode standby (pause)	
Mode run (marche)	
Programmation (généralités)	(10)
Généralités	
Bits, octets et mots	
Ressource du système	
Programmation graphique	(13)
Généralités	
Bloc de fonctions	
Exemple de programmation graphique	

Programmation en BASIC	(19)
Généralités	
Programmation langage CCBASIC	
L'instruction DEFINE	
Récapitulatif des commandes	
Exemples du CD	
Caractéristiques techniques	
Dimensions	

Introduction

Nous vous remercions d'avoir choisi la station C-Control. Cette station est un système de commande automatique moderne et programmable. Nous l'avons développé pour répondre aux besoins de nos clients en matière de qualité et fonctionnalité.

Garantie

Chaque station C-Control quitte nos ateliers en parfait état de marche. Conrad Electronic garantit cette station C-Control pour une durée de 12 mois à compter de la date d'achat. Pendant cette période, seront pris en charge les dommages subis lors du transport, les défauts de fabrication et les dysfonctionnements de l'appareil.

Si les capacités de la station C-Control ne satisfont pas vos besoins individuels, vous pouvez nous renvoyer l'appareil au plus tard 14 jours après la date d'achat s'il n'a pas été utilisé, dans son emballage d'origine, et obtenir son remboursement ou remplacement.

Le boîtier du module ne contient aucun composant pouvant être réparé par l'utilisateur. **Vous ne devez jamais ouvrir le boîtier !** La garantie ne couvre pas les dommages résultant d'un boîtier ouvert.

Conrad Electronic ne peut être tenu pour responsable des dommages indirects matériels ou corporels dus à l'utilisation de la station C-Control.

Description du produit

La station C-Control permet de programmer des commandes pour tout appareil électrique ou électronique. Toute autre utilisation est interdite.

Précautions d'emploi

Lisez attentivement les recommandations qui suivent. Le non-respect de ces précautions peut entraîner la mort en cas de décharge électrique ou incendie !

1. La station C-Control est montée dans un boîtier sur un rail DIN. Le module ne doit être relié à une tension d'alimentation de 230 V ~ ou une tension de relais supérieure à 24 V que via une armoire de distribution ou un coffret de commande de manière à garantir de bonnes conditions de sécurité contre les tensions dangereuses.
2. La station C-Control peut se connecter à d'autres appareils grâce aux 35 bornes de raccord. Il faut cependant distinguer les bornes de basse tensions allant de 1 à 9 et 18 à 35 et les bornes réseau allant de 10 à 17. Si vous reliez les raccords aux mauvaises bornes, vous risquez de provoquer des courts-circuits qui pourraient conduire à un incendie et endommager ainsi gravement le module et les appareils reliés.
3. La station C-Control ne doit pas être connectée à des appareils qui sont directement ou indirectement utilisés à des fins dangereux pour les hommes ou les animaux. La station C-Control doit être tenue à l'écart des environnements à risque (explosions) ou agressifs chimiquement.
4. Vous devez utiliser exclusivement le logiciel fourni avec l'appareil pour faire de la programmation.

Ne programmez le module qu'avec le programme d'application dont vous apprivoiserez les fonctions à l'aide du simulateur contenu dans le logiciel.

Capacité de puissance

La station C-Control comprend tous les groupes de composants nécessaires pour réaliser de nombreuses autres applications pour d'autres appareils, sans coût supplémentaire :

- Bloc secteur avec stabilisateur de tension
- Bornes pour raccorder une batterie tampon
- Unité de commande programmable avec microcontrôleur
- 6 ports numériques programmables
- 4 touches programmables
- 4 diodes LED programmables
- 2 relais pour connecter directement des appareils qui fonctionnent à une tension d'alimentation de 230 V (contacteur, courant continu 6 A max.)
- 2 couplages de capteurs de température (linéaire entre - 25°C et 102,5°C, résolution 0,5°C), capteurs et lignes de jonction compris, que vous pouvez rallonger selon vos besoins
- 4 ports analogiques pour mesurer la tension (entre 0 et 2,55 V)
- une interface série pour connecter un PC
- connexion pour une antenne d'horloge radio DCF77
- port pour mesurer les fréquences, par exemple lors de la connexion d'une éolienne par commande de rotation
- un haut-parleur piézoélectrique intégré

Grâce à cet équipement, vous êtes en mesure de résoudre les problèmes de commande ou réglage les plus sophistiqués en un minimum de temps.

Utilisation

Ce chapitre donne un aperçu de l'utilisation qui peut être faite du module. Vous trouverez toutes les informations détaillées dans les chapitres suivants de ce manuel.

Vous pourrez commencer à travailler avec la station C-Control après avoir suivi ces trois étapes :

1. Programmation du module
2. Montage
3. Exécution du programme d'une application

Programmation du module

Un module démonté peut être programmé via le poste de travail d'un PC. Alternativement il est possible de mettre un module préalablement monté mais non programmé dans une armoire de distribution, lorsque la connexion nécessaire au PC sera établie. Sinon le module doit être démonté.

- Commencez par installer le logiciel de programmation depuis le CD de la station qui vous est fourni. Suivez bien les recommandations d'installation du CD (fichiers install.txt et readme.txt).
- Connectez la station au PC et à la tension d'alimentation, en suivant la description du chapitre "Branchements". Branchez la tension d'alimentation.
- Ecrivez un programme d'application pour déterminer ce que doit faire la station pendant le fonctionnement.
- Compilez le programme d'application à l'aide du logiciel de programmation.
- Testez les fonctions du programme d'application avec le simulateur contenu dans le logiciel de programmation.
- Transférez le programme testé dans la station à l'aide du logiciel de programmation.

La station C-Control est maintenant programmée et peut être montée.

Montage

Pour faire fonctionner la station C-Control, il faut la monter sur un rail DIN, comme par exemple dans une boîte de distribution, au moment de l'installation électrique dans la maison de fusibles, commutateurs de sécurité et relais.

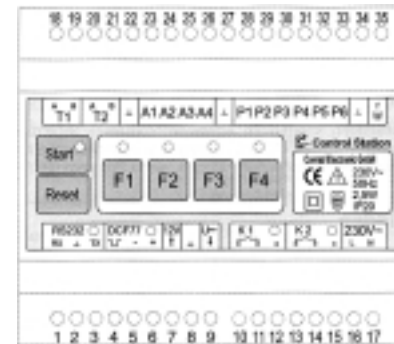
- Montez tout d'abord le système porteur, par exemple la boîte de distribution.
- Rangez la station C-Control ainsi que les autres groupes de composants de votre application dans la boîte de distribution. Le montage est terminé une fois que le module est fixé sur le rail DIN.
- Vous pouvez maintenant câbler le groupe de composants. Reportez-vous au chapitre "Branchements".
- L'ensemble de votre installation doit le cas échéant être contrôlé par un technicien qualifié.
- Terminez en recouvrant les bornes de connexion réseau.

Exécution du programme d'une application

Une fois la station C-Control montée et programmée, vous pouvez la mettre en marche pour exécuter le programme d'application.

- Mettez en route la tension d'alimentation.
- Appuyez sur la touche Start (démarrer).
- Vous (ou un autre utilisateur) pouvez maintenant utiliser la station C-Control de la manière prévue par votre programme d'application. La station suit le programme jusqu'à ce que le courant soit coupé ou que la touche Reset soit pressée. Vous pouvez redémarrer la station en appuyant à nouveau sur la touche Start.

Eléments de réglage



Touches

Le panneau avant de la station est un clavier à effleurement sur lequel sont intégrées six touches. Appuyez sur le clavier jusqu'à ce qu'un clic se déclenche.

Reset

La touche Reset permet de réinitialiser le module. On s'en sert en général pour arrêter le programme d'application ou avant d'effectuer la programmation. Lorsqu'on active cette touche, le module revient au mode initial, comme si on l'avait éteint. Toutes les variables de l'utilisateur sont alors effacées (=0), l'horloge interne est remise à 00:00:00 et une nouvelle synchronisation apparaîtra au branchement d'une antenne active DCF77.

Start

La touche Start permet de démarrer un programme d'application (voir le chapitre "Programmation") qui a été préalablement chargé dans la mémoire du module via une interface série. Le programme fonctionne tant que la commande END n'a pas été exécutée.

F1, F2, F3, F4

Ces touches peuvent être appelées depuis le programme d'application, mais aussi, par exemple, être utilisées pour choisir le type de fonctionnement (commutateur Marche / Arrêt) ou pour configurer des paramètres (voir le chapitre "Programmation").

Diodes LED

La station C-Control est constituée de neuf diodes électroluminescentes (LED), qui s'allument sur le clavier à travers des fenêtres rondes. Les LED informent l'utilisateur des différents états de fonctionnement du module. Les différentes couleurs symbolisent les différents groupes de fonction : rouge = système, vert = relais, jaune = LED de l'utilisateur.

LED de la touche Start (rouge)

Cette LED s'allume au démarrage du programme d'application (mode Run), après avoir appuyé sur la touche Start jusqu'à l'exécution de la commande END, l'activation de la touche Reset ou l'arrêt complet du module.

LED de l'interface série (rouge)

Cette LED signale une transmission de données sur la ligne TX de l'interface série. La LED est couplée à la ligne de données et s'allume brièvement (clignote) à chaque bit de donnée transféré. Lorsqu'un programme d'application est chargé dans le module depuis le PC, chaque octet du programme va être renvoyé au PC tel un écho. Le processus de téléchargement va donc également être signalé par la LED TX.

LED DCF77 (rouge)

Cette LED indique le statut de l'entrée de données et de la synchronisation de l'horloge interne du module, dans le cas où une antenne DCF77 serait reliée au module :

La LED clignote régulièrement toutes les secondes	Début de la réception de données. La synchronisation dure quelques minutes (au moins 2 à 3 minutes).
La LED clignote sans arrêt	La synchronisation est terminée.
La LED clignote irrégulièrement	L'antenne est mal placée ou la réception est parasitée.
La LED est éteinte	L'antenne est mal ou pas branchée, mal placée ou la réception est très fortement perturbée.

LED K1 et K2 (verte)

Ces LED renseignent sur l'état de commutation en cours des deux relais K1 et K2. Si les LED s'allument c'est que les relais correspondants sont bien détectés et que le circuit est fermé.

LED sur les touches de fonction (jaune)

Les LED des touches de fonction allant de F1 à F4 affichent l'état de fonctionnement du programme en s'allumant ou s'éteignant dans le programme d'application.

Branchements

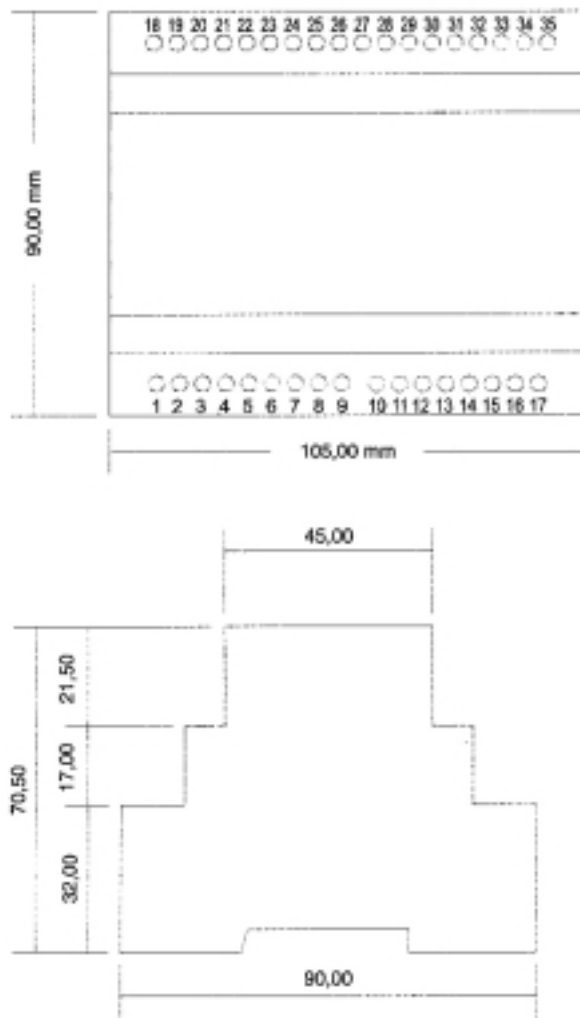
Branchements basse tension

Interface série RS232

1	Ligne de réception de données (RX), câble de l'interface, conducteur de câble blanc
2	Prese de terre (Masse, GND), câble de l'interface, conducteur de câble marron
3	Ligne d'envoi de données (TX), câble de l'interface, conducteur de câble vert

Le câble de l'interface va être relié au PC sur les bornes de 1 à 3 pendant la programmation du module. Le module peut également être relié à une transmission de données en série en cours de fonctionnement grâce au PC.

Dimensions



Antenne active DCF77

4	Ligne d'entrée des signaux, connexion de l'antenne active DCF77 au circuit d'acheminement des signaux
5	Prise de terre (Masse, GND)
6	Tension d'alimentation stabilisée (+ 5 V) pour l'antenne active DCF77

L'antenne active DCF77 (en option) va être reliée à la synchronisation du temps du module sur les bornes 4 à 6. Il est également possible de relier la tension du système stabilisée (5 V) sur la borne 6 pour pouvoir effectuer des petits circuits externes. Veillez à respecter le courant maximum prélevé (voir "caractéristiques techniques") !

Alimentation de basse tension et de l'accu tampon

7	Entrée pour un bloc d'alimentation stabilisée (12 V) ou une batterie + 12 V
8	Prise de terre (Masse, GND)
9	Sortie pour la tension du système instable, entre 12 et 24 V

Un bloc d'alimentation stabilisé (12 V) ou une batterie 12 V peut être relié à la borne 7 pour programmer le module avant le montage dans l'armoire de distribution et sans relier une tension d'alimentation de 230 V~ (sur les bornes 16 et 17).

En cours de fonctionnement normal, dans l'armoire de distribution à une tension d'alimentation de 230 V~, il est possible de relier l'accu au plomb (capacité conseillée $\geq 1,2$ Ah) sur la borne 7 pour mettre le système en mémoire-tampon contre les défaillances de la tension de secteur. l'accu doit être complètement chargée pour garantir une capacité de commutation maximum. En cas de tension disponible un courant de charge minime circule pour compenser l'autodécharge de la batterie. Dans ce cas, vous ne devez pas utiliser la station C-Control afin de permettre à l'accu vide de se recharger complètement !

Vous pouvez relier la tension continue instable du bloc d'alimentation interne à la borne 9. Elle peut être utilisée pour alimenter de petits circuits externes. Veillez à respecter le courant maximum prélevé (voir "caractéristiques techniques") ! Veillez également à ce que le voltage réel qui dépend de la charge se situe au-dessus d'une tension nominale de 12 V mais ne dépasse pas 24 V.

Ports pour les capteurs de température

18	T1 A	Capteur 1, câble noir, conducteur de câble blanc
19	T1 B	Capteur 1, câble noir, conducteur de câble marron
20	T2 A	Capteur 2, câble noir, conducteur de câble blanc
21	T2 B	Capteur 2, câble noir, conducteur de câble marron

La station C-Control contient des groupes de composants électroniques permettant de coupler deux capteurs de température avec un convertisseur A/D du microcontrôleur. Les capteurs sont des semi-conducteurs prêts à l'emploi du type AD592 qui sont fournis avec le système. Ils vont être connectés au module sur les ports T1 et T2. L'électronique interne pour chaque canal du détecteur s'ouvre à chaque fois qu'un des deux capteurs est ajusté. Lorsque vous connectez le capteur, faites attention aux symboles de couleur qui permettent de savoir quel port doit être relié à quel capteur (voir tableau). Veillez également à respecter la polarité.

Le capteur de température AD952 fonctionne comme une source de courant variant en fonction de la température et entraîne un courant de $1 \mu\text{A/K}$. La ligne de raccordement fournie peut servir de rallonge sur quelques mètres. Pour éviter toute perturbation parasite, il est indispensable d'utiliser un câble qui soit blindé. Reliez la rallonge blindée au câble d'origine blindé pour ensuite la relier à la station de contrôle via une des bornes GND. Veillez à respecter la polarité lors de la connexion de la rallonge.

Bornes GND (masse)

22	GND
27	GND
34	GND

La masse du système est reliée aux bornes GND comme potentiel de référence lors de l'utilisation des ports analogiques et numériques.

Ports analogiques

23	A1
24	A2
25	A3
26	A4

Les ports analogiques permettent de mesurer le voltage analogique qui se trouve sur le potentiel de masse de la station C-Control. La mesure se fait par le convertisseur A/D du microcontrôleur avec une résolution de 8 bits et fournit des mesures situées entre 0 et 255. La plage de tension de référence se situe entre 2,55 V et 255 V.

Ports numériques

28	P1
29	P2
30	P3
31	P4
32	P5
33	P6

Chacun des ports numériques du module peut être programmé comme entrée pour déterminer un état de commutation ou comme sortie pour déclencher un processus de commutation. Les niveaux logiques sont compris entre 0 et 0,7 V pour la sortie et entre 4,3 et 5 V pour l'entrée.

Entrée pour la mesure de fréquence

35	F
----	---

La fréquence f peut être mesurée sur la borne 35. Le signal d'entrée est soit un signal numérique de 0 / 5 V, soit une connexion cyclique via la borne GND (borne 34), comme par exemple via une éolienne qui coupe un contact à chaque rotation. La marge d'erreur de la plage de mesure est < 1% jusqu'à 30 kHz.

Connexions réseau

Bornes pour tension d'alimentation

16	Phase 230 V~ (L)
17	Conducteur neutre 230 V~ (N)

Tous les composants électroniques internes du module sont alimentés en fonctionnement normal via les bornes de tension d'alimentation. Dans le module se trouve également un bloc d'alimentation avec un transformateur, un redresseur de courant et un stabilisateur de tension.

Relais K1 et K2

10	K1
11	K1
12	Borne disponible
13	K2
14	K2
15	Borne disponible

pour revenir à plein régime.

Attention : Pour les programmes qui utilisent des transferts de données en série, vous ne devez pas activer la commande SLOWMODE, car les vitesses de transfert sont directement liées à la cadence du processeur.

En réduisant la cadence du processeur, vous allez réduire la puissance absorbée et engendrer un ralentissement de la vitesse de shuntage de la batterie. Cela n'est donc intéressant que si l'ensemble des LED et relais est éteint.

Exemples du CD

Vous trouverez des exemples vous permettant de faire de la programmation sur la station C-Control et de comprendre les étapes de nombreux cas d'utilisation dans les sous-répertoires SAMPLES contenus dans le logiciel graphique et BASIC.

Caractéristiques techniques

Tension de fonctionnement	230 V~ tension de secteur aux bornes 16 et 17 ou bloc d'alimentation 12 V~ sur les bornes 7 et 8
Puissance absorbée maximum	2,5 W
Courant max. autorisé sur la borne 6 (5 V)	$I_{6,max} = 50 \text{ mA}$
Courant max. autorisé sur la borne 9 (entre 12 et 24 V)	$I_{9,max} = 50 \text{ mA}$
Courant du port max. sur les bornes 28 à 33 (P1 à P6)	de $I_{28,max}$ à $I_{33,max} = 10 \text{ mA}$ chacun
Emission de courant général max.	$I_{6,max} + I_{9,max} + I_{28,max} + I_{33,max} = 100 \text{ mA}$
Courant de l'accu tampon sur la borne 7	5 mA env.
Alimentation des relais (K1, K2)	230 V~ max., courant de démarrage sur une période courte 8 A, courant continu 6 A max.
Gamme de température (T1, T2)	de -25,0°C à 102,5°C résolution 0,5 K, erreur max. $\pm 2,5 \%$
Gamme de mesure des ports A/D (A1 à A4)	de 0 à 2,55 V résolution 0,01 V, erreur max. ± 1 digit
Gamme de mesure de la fréquence (borne 35)	de 1 Hz à 30000 Hz
Conditions d'environnement autorisées	entre 0 et 40°C, humidité relative de l'air comprise entre 20 et 60 %

Timer

Un timer (horloge) interne de 20 millisecondes est accessible depuis l'indicateur **TIMER** prédéfini. C'est un timer spontané qui peut ne pas être mis en place ou remis à l'état initial.

Sortie de sons avec BEEP

La station C-Control est équipée d'enceintes piézo internes qui permettent de retransmettre le son du signal. La commande BEEP est la suivante :

BEEP son, tSon, tPause

Vous pouvez utiliser les termes ou des constantes pour les trois paramètres. La hauteur de son se calcule selon la formule

son = 250000 / freq [Hz]

tSon indique la durée du son et tPause la pause après le son. L'unité des entrées de temps est de 20 millisecondes, d'où la commande

BEEP 568, 10, 3

Le calcul correspond donc : pour $10 \cdot 20 = 200$ millisecondes d'un son d'environ 440 Hz et dont la pause est de $3 \cdot 20 = 60$ millisecondes. Si après une commande BEEP, aucune autre commande BEEP ne suit, la pause peut être réglée sur 0. Si la longueur du son est 0, un son continu va être émis. Le générateur déclenche le son et continue avec le programme BASIC. Si la valeur de 'son' est 0, le générateur de son va à nouveau s'arrêter.

Mesure de fréquence avec **FREQ** et **FREQ2**

La mesure de fréquence de la station C-Control repose sur le principe de comptage d'impulsions pour une durée de son d'1 seconde. La mesure se fait continuellement en arrière plan, en parallèle à l'exécution du programme BASIC.

Les impulsions d'un cycle de seconde correspondent directement à une fréquence en Hz.

L'entrée standard d'une mesure de fréquence (borne 35) peut être appelée avec la commande **FREQ2**.

x = **FREQ2**

La plage de mesure va jusqu'à 30000 Hz environ avec une marge d'erreur < 1 %.

Si aucune antenne active n'est reliée à l'entrée DCF77, vous pouvez effectuer en alternance une mesure de fréquence sur cette entrée, dont le résultat est appelé avec la commande **FREQ**

x = **FREQ**

Ici la plage de mesure va jusqu'à 5 Kz avec une marge d'erreur < 1 %.

Mode économie d'énergie avec **SLOWMODE**

Pour activer le mode d'économie d'énergie, appelez la commande **SLOWMODE**
SLOWMODE ON

pour ralentir la cadence interne du microprocesseur (1 / 16). Si vous avez ensuite besoin d'une vitesse plus élevée, appelez alors

SLOWMODE OFF

Grâce aux relais K1 et K2 contenus dans le module vous pouvez y relier tout appareil fonctionnant à une tension de 230 V~. Le courant de contact maximum autorisé (voir "Caractéristiques techniques") ne doit pas être dépassé. Les bornes non allouées, situées entre le relais et la borne d'alimentation, permettent de respecter les distances de sécurité entre les différents potentiels de tension.

Connexion pour programmer le module

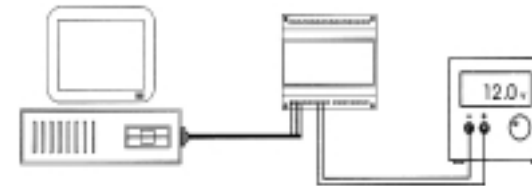
Connexion entre la station et le PC

Pour transférer un programme d'application du PC dans le module, deux types de connexion sont disponibles pour l'alimentation de tension.

Dans les deux cas, vous devez connecter le câble de l'interface entre le PC et la station en suivant les recommandations ci-après :

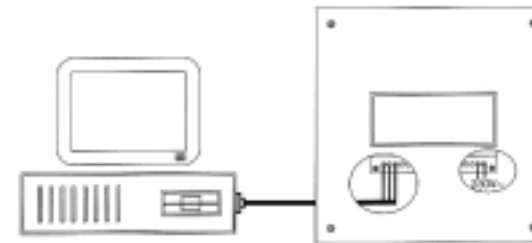
- Reliez le connecteur sub-D 9 broches du câble de connexion de la station à l'interface série (COM1 à COM4) du PC.
- Connectez le conducteur **blanc** à la **borne 1 (RX)** de la station.
- Connectez le conducteur **marron** à la borne 2 (**GND**) de la station.
- Connectez le conducteur **vert** à la borne 3 (**TX**) de la station.

Programmer avec une alimentation de basse tension

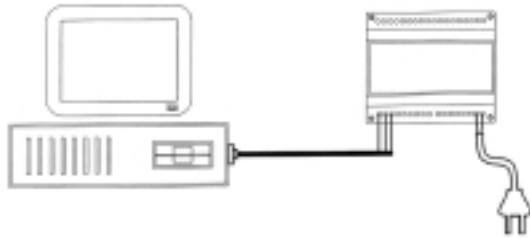


Si vous faites de la programmation avec une alimentation de basse tension, vous devez impérativement relier un bloc d'alimentation stabilisé de 12 V ou l'accu de 12 V aux bornes 7 (+) et 8 (-) du module. Le module peut ensuite être utilisé librement. Cette technique est particulièrement destinée à la programmation depuis le poste de travail d'un PC, dans un laboratoire, un atelier ou un bureau.

Programmer avec une alimentation de réseau



Faire de la programmation avec une alimentation de réseau est utile pour effectuer des transferts depuis le programme d'application dans un endroit précis du module. C'est dans ce but que la station C-Control est montée sur un rail DIN. Il est indispensable de recouvrir au moins les bornes de connexion réseau pour garantir la protection des contacts. Lorsque vous connectez le câble de programmation, la station C-Control doit être hors tension !



Il est possible techniquement d'alimenter un module posé sur une table via un câble réseau standard à l'aide d'un connecteur adéquat (destiné aux appareils appartenant à la classe de protection II). Cela n'est cependant pas autorisé par les dispositions légales de protection contre le contact avec des tensions dangereuses.

Etat du système

Mode Reset (réinitialisation)

Lorsque vous coupez l'alimentation ou pressez la touche Reset, le module procède à la réinitialisation des paramètres et se met en mode Standby.

Les paramètres sont réinitialisés comme suit :

1. Date et heure : 1er janvier 1997, 00:00:00
2. Préparation de la réception de données de l'horloge de la DCF77
3. Paramétrage de l'interface série sur 9600 bauds, 8 bits de données, 1 bit d'arrêt, aucune parité
4. Suppression de toutes les variables de l'utilisateur (= 0)

Mode Standby (pause)

En mode Standby, la touche Start et l'interface série seront sollicitées en permanence pour faire démarrer le programme d'application ou pour réagir à des commandes de contrôle d'un ordinateur supérieur. En mode Standby, vous pouvez programmer le module (téléchargement de programmes). Si vous appuyez sur la touche Start, le système passera en mode Run.

Mode Run (marche)

En mode Run, vous pouvez exécuter les commandes du programme d'application. Vous pouvez arrêter le mode Run en appuyant sur la touche Reset, en débranchant la tension de fonctionnement ou en exécutant la commande END (du bloc END) dans le programme d'application. Une fois la commande END exécutée, le système revient au mode Standby.

Programmation (généralités)

Généralités

La programmation va permettre de fixer le mode opératoire du système. C'est pourquoi la station C-Control requiert un maximum de flexibilité et peut être utilisée pour différentes fonctions. Une fois programmée, elle pourra vous servir par exemple de régulateur de température, déclencheur automatique d'éclairage, installation d'alarme ou commande de rotation. Il peut ensuite être reprogrammé librement en fonction de vos besoins.

Un microcontrôleur se trouve à l'intérieur de la station C-Control. Il s'agit d'un petit ordinateur sur une puce, à laquelle est reliée une puce mémoire qui sert à enregistrer le programme d'application.

Grâce à la technologie EEPROM, le programme d'application est ainsi conservé, même si la tension de fonctionnement du système est coupée.

Vous pouvez programmer la station C-Control à l'aide d'un logiciel facile à utiliser sur votre PC, soit par une programmation graphique, soit par le langage de programmation populaire BASIC.

TABLE tablename "tabfilename"

Les définitions des tableaux doivent toujours se situer à la fin d'un programme, après la commande END, car les données seront enregistrées juste derrière les octets de code dans la mémoire de l'EEPROM. Un programme ne doit jamais tourner sur des données de tableau, car les données pourraient être interprétées comme une commande BASIC, ce qui conduirait irrémédiablement au dysfonctionnement du système.

L'accès aux données de tableau se fait avec la commande

LOOKTAB tablename,index,variable

'tablename' désigne un tableau existant, 'index' peut être représenté par un terme et la 'variable' indique la cellule de sauvegarde, dans laquelle sera enregistrée le résultat. La valeur du terme 'index' ne doit pas être négative et ne pas dépasser N-1 si le tableau indexé possède des entrées N. Si 'index' renvoie une valeur égale à 0, C0 sera la variable qui sera enregistrée, si la valeur est égale à 1, ce sera C1, etc.

L'exemple qui suit donne le contenu d'un tableau série :

```

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

```

Les tableaux sont particulièrement utiles pour convertir des valeurs A / D en grandeurs physiques réelles. Un tableau de conversion dispose en général de 256 entrées. La valeur A / D mesurée interviendra comme index de tableau lors de la fixation de la grandeur physique.

Accès à l'horloge

Les variables utilisées pour lire et configurer l'horloge sont les suivantes :

YEAR	Année (0 à 99)
MONTH	Mois (1 à 12)
DAY	Jour du mois (1 à 31)
DOW	Jour de la semaine (0 = dimanche, 6 = samedi)
HOURL	Heure (0 à 23)
MINUTE	Minute (0 à 59)
SECONDE	Seconde (0 à 59)

Assurez-vous que l'horloge interne continue à tourner pendant l'accès. La valeur des secondes doit toujours être lue en premier. Si elle se trouve sur 59, la valeur doit à nouveau lue puis remise à 0 pour faire un nouveau test, après avoir lu la dernière information de temps (par exemple YEAR).

Dans ce cas, vous devez refaire la lecture de l'horloge, car une nouvelle minute a été perdue (cas extrême du 31 décembre avec remise à zéro de tous les paramètres concernant l'heure et la date). Le chiffre de l'année va être enregistré sous forme diadique dans le système C-Control.

et pour le déconnecter :

P = 0

Pour renverser la marche (marche vers arrêt ; arrêt vers marche) on peut aussi écrire :

P = NOT P

ou utiliser la commande

TOG P

TOG désigne l'anglais "toggle" (bascule). La commande TOG occupe très peu d'espace dans l'EEPROM et sera exécutée aussi rapidement qu'une construction NOT-P classique.

Dans la commande TOG, la variable du port P est réservée à un seul port unique, et non pas un port d'octet ou de mot.

- Désactiver un port avec la commande **DEACT**

Lorsqu'une variable de port est assignée pour la première fois, l'ordinateur de commande commute les structures du matériel correspondantes dans la puce du processeur (les transistors) sur un mode de sortie. Le courant alimente alors le processeur (10 mA autorisés maximum). La commande permet de désactiver le port connecté :

DEACT portvar

Cela signifie que le port va être commuté dans un état de grande impédance et va ensuite pouvoir travailler en mode d'entrée.

- La commande **PULSE**

PULSE portvar

Cette commande va permettre d'envoyer une pulsation d'une largeur de quelques millisecondes sur le port désigné 'portvar'. Si le port de la commande PULSE est bas (=0), cela donnera une pulsation haute (0-1-0), sinon une pulsation basse (1-0-1).

Définition et utilisation des tableaux de données

En langage BASIC standard, les lignes de données servent à stocker des blocs de données constants, auxquels on peut ensuite accéder par un accès séquentiel. CCBASIC ne supporte aucune ligne de données, c'est pourquoi il requiert des moyens flexibles permettant de définir et d'accéder aux blocs de données. Les données constantes peuvent être enregistrées sous forme de tableaux. Chaque tableau possède un indicateur (nom de tableau) et peut contenir autant de données que vous le souhaitez, tant qu'il y a de la place dans la mémoire du programme. Chaque entrée de données (Cx) va être stockée comme une valeur d'entier et occuper un espace de 2 octets. Vous pouvez exécuter les données en texte source comme suit :

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

ou les importer d'un fichier texte externe depuis le compilateur CCBASIC

Bits, octets et mots

Le système C-Control travaille, calcule et enregistre des nombres entiers ("Entiers"). Il existe trois types de données différentes :

Bit : deux valeurs représentées par 0 et -1 (non pas 1 !) ; utilisé comme mémoire d'état ("indicateur").

Octet : ensemble de 8 bits ; représenté de la valeur 0 à la valeur 255 ; utilisé pour les petits nombres entiers non négatifs, pour la table de codage ASCII et les codes de commande.

Mot (= "entier") : ensemble de 16 bits ; composé d'octets hauts et bas ; la valeur est comprise entre -32768 et 32767 ; utilisation universelle.

Ressource du système

Périphériques

La plupart des périphériques sont connectées à la station C-Control via les bornes de connexion ou les commandes du panneau avant.

Ports numériques : La station C-Control dispose en tout de 16 ports numériques programmables. Les ports 1 à 6 sont reliés aux bornes P1 à P6 et peuvent être utilisés comme entrée ou sortie. Les ports 7 et 8 commutent les deux relais K1 et K2. Sur les ports 9 à 12, il est possible d'appeler les touches de fonction F1 à F4. Les ports 13 à 16 permettent aux LED situées sur les touches de fonction de s'allumer ou de s'éteindre.

Ports analogiques : Pour mesurer des grandeurs analogiques, la station C-Control possède en tout 8 ports permettant une conversion numérique-analogique ayant une résolution de 8 bits. Ils vous permettent ainsi de mesurer des tensions électriques qui peuvent, par exemple, servir de signal de sortie à un capteur en fonction d'une grandeur physique déterminée. La gamme de mesure s'étend de 0 à 2,55 V par palier de 10 mV. Les ports analogiques 1 à 4 sont reliés aux bornes A1 à A4 sur la station C-Control.

Les ports analogiques 5 et 6 sont connectés aux capteurs de température T1 et T2.

Le port analogique 7 mesure la tension sur un point interne de l'électronique de l'alimentation. Si une batterie chargée de 12 V est reliée aux bornes 7 et 8, alors que la tension de secteur défaille, la valeur de mesure sera d'environ 120. La batterie pourra alors se décharger lorsque la valeur de mesure baissera. Si la tension de la batterie chute en dessous d'une valeur critique, il est possible de prendre des mesures directement dans le programme d'application avant que le système ne tombe en panne.

Sur le port analogique 8 il est possible de détecter la défaillance de la tension de 230 V~. Le port est interne et connecté sur la sortie "+" du redresseur à pont du bloc secteur intégré via un diviseur de tension. La courbe de tension est filtrée par un condensateur. Si la valeur de mesure chute à 0, cela signifie que la tension de secteur est tombée. Comme la tension de fonctionnement du microcontrôleur est mise dans la mémoire tampon à l'aide d'un condensateur puissant, le système continue à tourner pendant un court moment après la chute de la tension. Si vous programmez habilement la reconnaissance de baisse de tension, il est possible d'éviter la défaillance du système.

Horloge : Sur les bornes 4 à 6, vous pouvez connecter une antenne active DCF77. Dès qu'un signal sera reçu, l'horloge interne va ajuster automatiquement l'heure du système au fuseau horaire européen. Vous pouvez déclencher les informations concernant la date, à savoir les heures, minutes, secondes, jours, semaines, mois et années directement dans le programme de l'application et en combinaison logique avec d'autres opérations que vous souhaitez lancer.

Mesure de la fréquence : Grâce à l'entrée de mesure de la fréquence, les signaux allant jusqu'à 30 kHz environ apparaissent et peuvent être déclenchés directement dans le programme d'application. Interface série : L'interface série ne sert pas uniquement à charger un programme d'application dans la station C-Control, mais aussi peut également être utilisée dans le programme lui-même pour transférer des données. Il est donc également possible de transférer des données sous forme d'octets ou de textes. Le taux de transfert peut être très rapidement adapté (entre 1200 et 9600 bauds).

Sortie de son : Une enceinte piézoélectrique se trouve dans la station C-Control. Les signaux

sonores pourront ainsi sortir par un petit haut-parleur. La durée et la hauteur du son peuvent être programmées. Vous pouvez aussi programmer des petites mélodies, signaux d'alerte ou de confirmation pour certaines touches ou pour signaler des situations précises lors de l'utilisation de l'appareil.

Mémoire du programme

Votre programme d'application va être traduit ("compilé") par le logiciel de programmation dans un format compatible avec le système de la station C-Control. La mémoire du programme a une capacité de 8000 octets environ. La commande qui permet de déclencher l'allumage d'un des relais, par exemple, nécessite 5 octets. Les programmes plus complexes n'en utilisent généralement pas plus d'une centaine. Vous avez donc à votre disposition une mémoire suffisante.

La mémoire du programme se situe dans la puce de la mémoire EEPROM (modèle de micropuce 24C65). Grâce à la technologie EEPROM, toutes vos données sont conservées dans la mémoire du programme, même si la tension d'alimentation est coupée.

Mémoire pour l'enregistrement de données

Les octets disponibles qui restent sur la puce de la mémoire du programme peuvent être utilisés pour enregistrer des données, comme par exemple des valeurs de mesure. L'enregistrement et le tri des valeurs sont toujours séquentiels et représentés en mots. Une valeur enregistrée occupe 2 octets.

La gestion des valeurs enregistrées se fait d'après le principe de fichier. Le fichier doit être ouvert avant de pouvoir y accéder. Il existe différents type d'ouverture, celle pour lire, écrire et stocker de nouvelles données sur les précédentes. Après une séquence d'accès, le fichier doit être fermé, pour enregistrer éventuellement les modifications qui ont été faites.

Tout comme le programme d'application, les données enregistrées sont à l'abri d'une éventuelle chute de la tension de fonctionnement. C'est pour cette raison qu'il faut fermer le fichier après l'enregistrement de la dernière valeur.

Mémoire principale

Dans le cas où le système tomberait en panne, il est souvent nécessaire de sauvegarder des paramètres, états ou calculs. C'est pourquoi la station C-Control dispose en tout d'une mémoire principale de 24 octets, qui peut être représentée par des variables prédéfinies (bits, octets ou mots). Contrairement à la mémoire du programme, le contenu de la mémoire principale n'est conservé que tant que le microcontrôleur du système est alimenté par la tension de fonctionnement. Après une réinitialisation (Reset) ou une panne, toutes les valeurs contenues dans la mémoire principale sont remises à zéro (= 0).

Unité de calcul

Comme expliqué dans la partie "Bits, octets et mots", les valeurs qui s'affichent sont représentées par des chiffres entiers. Toutes les opérations vont s'afficher sur une largeur de 16 bits (gamme de valeurs des chiffres entiers). L'unité de calcul du système contrôle les quatre signes de calcul de base (+, -, *, /), la division modulo (recherche du reste par la division des nombres entiers), la fonction logique exprimée en bits, les opérations de comparaison, la négation, la détermination de la somme absolue, la fonction de la racine carré et la fonction signe. Un générateur pseudo-aléatoire initialisable est également intégré : il peut générer des suites de nombres réparties statistiquement grâce la division modulo d'après le principe de la multiplication des nombre premiers.

L'unité de calcul domine au maximum 6 plans entrelacés si vous effectuez des opérations de calculs entre parenthèses ou de comparaison.

Attention ! En délimitant la largeur de données à 16 bits maximum dans la gamme de valeurs du nombre entier, il peut arriver que le résultat du calcul dépasse la valeur délimitée. De plus, lorsque vous enregistrez une valeur dans une variable, la gamme de valeurs déclarée va être limitée. Tout

PRINT# terme

qui va permettre d'enregistrer le résultat,
ou

INPUT# variable

pour laquelle la 'variable' représentera une variable définie d'un entier du programme.

L'écriture et la lecture dans / de la donnée sont strictement séquentielles. C'est pourquoi un pointeur de fichier est géré en interne, lequel va augmenter de 1 après chaque accès.

Vous devez vous assurer qu'il reste suffisamment d'espace sur l'EEPROM pour enregistrer vos données écrites. Pour ce faire, vous pouvez appeler la fonction **FILEFREE**, qui vous donnera la taille de l'espace restant sur la mémoire (en mots). L'exemple suivant vous montre comment procéder :

```
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
```

Vous devez également vérifier si d'autres données sont encore affichées avant de lire des données, à l'aide de la fonction **EOF** ("end of file" / fin de fichier). S'il n'y a plus aucune données disponibles dans le fichier, le résultat affichera 1, sinon 0.

En reprenant l'exemple ci-dessus, voici ce que ça donne :

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

Une fois l'accès aux données terminé, vous devez refermer immédiatement le fichier, car ce n'est qu'après la fermeture, que les données seront protégées d'une éventuelle panne ou réinitialisation (Reset) du système. La commande est la suivante (elle ne contient aucun paramètre) :

CLOSE#

Commandes de port

- La commande de commutation TOG

En principe l'accès aux ports de l'ordinateur de commande se fait comme pour celui aux variables. Pour une commutation sur un port numérique P, on écrit

```
P = 1
```

Prédiviseur	NP1	NP0
1	0	0
3	0	1
4	1	0
13	1	1

NT (bits 3 à 5) et NR (bits 0 à 2) déterminent d'autres valeurs de diviseur, séparées pour l'envoi (NT) et la réception (NR), qui sont codées comme suit :

Diviseur	NT2 et NR2	NT1 et NR1	NT2 et 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

Le taux de transfert pour l'envoi se calcule selon la formule suivante :

$$\text{Taux d'envoi} = 125000 / (\text{NP} * \text{NT})$$

et le taux pour la réception

$$\text{Taux de réception} = 125000 / (\text{NP} * \text{NR})$$

Les autres paramètres de l'interface, 8 bits de données, aucun bit de parité, 1 bit d'arrêt, sont fixes et ne peuvent pas être modifiés.

Fonctions de données

Les fonctions de données permettent d'enregistrer des valeurs de mesure ou d'autres données, ou peuvent également être utilisées pour enregistrer des informations qui seront utilisées ultérieurement dans les variables du programme, notamment en cas de panne du système.

La zone de sauvegarde contenue dans la puce EEPROM est destinée à cette utilisation. Elle va être gérée comme un fichier, après avoir été ouverte avec les attributs correspondants. La commande d'ouverture de fichier est la suivante :

OPEN# FOR WRITE

ou

OPEN# FOR APPEND

ou

OPEN# FOR READ

WRITE représente l'ouverture du fichier pour recouvrir des anciens enregistrements, APPEND l'ouverture pour ajouter de nouvelles données sur des anciennes données et READ l'ouverture pour lire les données.

Vous ne pouvez enregistrer et lire que des valeurs d'entier. Chaque valeur représente 2 octets dans l'EEPROM. Pour écrire et lire, vous devez utiliser la commande

calcul de division ayant une valeur avec une virgule sera coupé.

Exemple :

- 32767 + 1 font - 32768, et non 32768
- a = -1 font 255 pour a, uniquement s'il s'agit d'une seule variable d'octet
- 10 / 4 font 2, en non 2,5

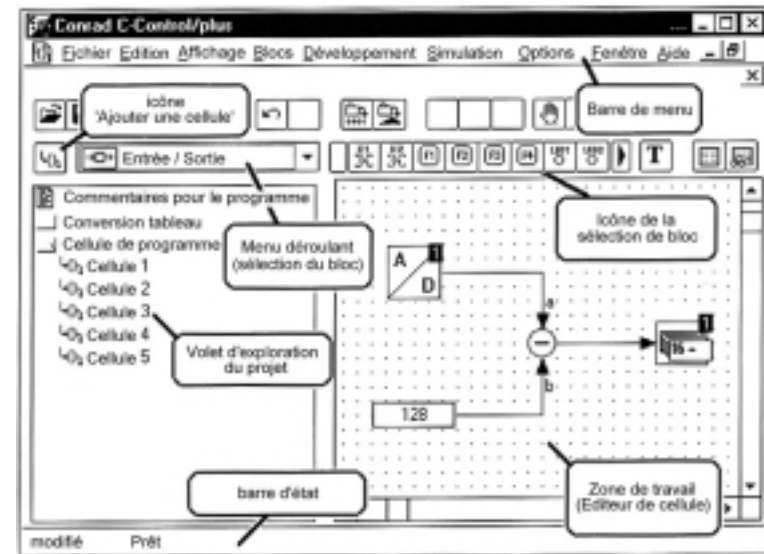
Programmation graphique

Généralités

Vous avez à votre disposition une version adaptée du programme "C-Control/plus" qui vous permet d'effectuer une programmation graphique sur la station C-Control. La **programmation graphique** est destinée **aux débutants** pour une **utilisation simple**. Vous trouverez le logiciel sur le CD fourni avec la station. Pour l'installer, suivez les recommandations qui se trouvent sur le CD. Le logiciel de programmation fonctionne sur Windows 95 ou Windows NT dès 4.0 ou version ultérieure, mais ne peut pas être installé sur une version Windows 3.xx.

Comme les commandes du logiciel sont intuitivement faciles à comprendre, nous renonçons à vous faire une description complète de l'interface. Vous pouvez prendre connaissance des bases indispensables en utilisant le programme d'aide du logiciel. Appuyez sur la touche F1 (aide générale) ou sur Ctrl + F1 (aide personnalisée). Vous pouvez également faire un clic droit sur le bloc de fonction souhaité et choisir dans le menu contextuel "Aide sur le type de bloc" pour obtenir une description des fonctions du bloc.

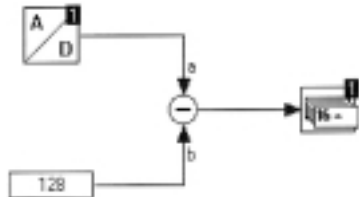
Le schéma vous permet d'avoir une vue d'ensemble de l'interface de programmation graphique de la station C-Control. Les composants les plus importants sont mis en valeur. Avant de pouvoir utiliser le logiciel, il est indispensable de connaître le fonctionnement standard de Windows et de connaître des termes tels que "Menu", "Barre d'état", "Cliquer", "Double-clicquer", "Pointer la souris sur", "Ouvrir le fichier" ou "Enregistrer les données". Vous pouvez également consulter la notice d'utilisation ou les pages d'aide en ligne de Windows.



Un programme graphique repose sur plusieurs cellules qui décrivent les étapes de fonctionnement devant être exécutées en mode Run. A gauche de la zone de travail, vous avez un aperçu de toutes les cellules d'un projet grâce au volet d'exploration du projet. Vous pouvez ajouter une nouvelle cellule en cliquant sur l'icône "Zelle einfügen (Ajouter une cellule)".

Vous pouvez déterminer les étapes de fonctionnement dans les cellules grâce au bloc d'icônes. Cela signifie que chaque bloc de fonction sera connecté grâce aux signaux des lignes de liaison.

Bloc de fonctions



Comme le montre le schéma, il existe des blocs de fonctions qui délivrent des valeurs dont certaines sont produites pour obtenir un résultat tandis que d'autres sont enregistrées ou qui déclenchent des actions depuis les valeurs alimentées.

Dans l'exemple donné, la valeur de mesure détectée depuis le Port 1 A / D est la constante 128 (a-b). Ce résultat est ensuite enregistré.

Les blocs de fonctions de l'interface C-Control/Plus se répartissent en différents groupes :

- Ein-/Ausgabe (Entrée / Sortie) : accès aux relais, touches, LED et configuration du son.
- Meßwert (Valeur de mesure) : accès aux capteurs de température, mesures de tension et ports analogiques.
- Digitalports (Ports numériques) : accès aux bornes des ports numériques P1 à P6.
- Berechnungen (Calculs) : combinaisons et fonctions mathématiques.
- Vergleich (Comparaison) : opérations de comparaison, dont le résultat peut être vrai (= -1) ou faux (= 0).
- Logik (Logique) : combinaisons en bits d'après l'algèbre de Boole.
- Zeit (Horloge) : accès à l'horloge interne et au timer (DCF77).
- Variablen (Variables) : Blocs mémoire.
- Konstanten (Constantes) : Blocs qui délivrent des valeurs constantes.
- Datenaufzeichnung (Affichage de données) : Affichage des données contenues dans la puce mémoire EEPROM.
- Serielle Schnittstelle (Interface série) : introduction / extraction de textes et données sur l'interface série du système.
- Programmsteuerung (Commande de programme) : Blocs de fonctions permettant de commander les flux de programme.

Exemple de programmation graphique

Paramétrage du programme

L'exemple ci-après vous explique les principes de bases de la programmation graphique. Celui-ci vous permet d'avoir un autre exemple en plus de ceux contenus sur le CD et d'adapter vos idées aux informations qui vous sont communiquées, de manière à avoir progressivement un aperçu complet de la programmation graphique de la Station C-Control.

Vous voulez pouvoir démarrer / arrêter alternativement le relais K1 avec la touche F1 du module. Commencez en lançant le programme CCSTAT.EXE. Initialisez un nouveau projet en cliquant sur "Datei/Neu (Fichier / Nouveau). Activez dans la boîte de dialogue "Optionen/Umgebungseinstellungen/Projektübersichtsfenster (Options / Paramètres de

en appuyant sur la touche ENTER de la station C-Control via le câble d'interface.

La commande INPUT patientera jusqu'à ce que le transfert de données soit complètement fait depuis le terminal. Si vous appelez la commande INPUT avant que le transfert soit terminé, le programme va rester bloqué à cette étape ! Vous n'aurez plus qu'à appuyer sur la touche Reset pour redémarrer la station.

- Communication octet par octet via l'interface série

Il est possible de transférer en série quelques octets, pendant que les commandes PRINT et INPUT attendent ou envoient des chaînes de caractères courtes pour la représentation d'une valeur numérique. CCBASIC utilise alors les commandes PUT et GET.

PUT terme

La commande PUT envoie la valeur prédéfinie d'un terme. Si cela est nécessaire, le résultat sera réduit au préalable sur une plage de valeurs en octets comprise entre 0 et 255.

GET variable

Cette commande patiente sur un octet de réception en série et enregistre la valeur dans la variable souhaitée.

- Autres commandes et fonctions de l'interface

Comme nous l'avons mentionné, INPUT et GET vont, dans certaines circonstances, suspendre le programme indéfiniment sur les données série de réception. Si vous souhaitez empêcher la "suspension" du programme, vous pouvez avant chaque INPUT et GET vérifier si les données reçues sont disponibles, en appelant la fonction de statut **RXD**. La fonction renvoie dans ce cas la valeur 1. Si la mémoire tampon de l'interface est vide, le résultat de la fonction sera égal à 0.

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

Le taux de transfert de l'interface série en réception et envoi est prédéfini à 9600 bits par seconde (baud). Vous pouvez configurer un taux différent en utilisant la commande BAUD. CCBASIC contient pour cela quelques constantes prédéfinies : **R1200**, **R2400**, **R4800** et **R9600** adaptées aux taux compris entre 1200 et 9600 bits par seconde.

BAUD R2400

commute les données entrantes et sortantes sur un taux de 2400 bits par seconde. En principe il est aussi possible d'avoir d'autres taux que ce soit pour la réception ou l'envoi. Les taux de transfert de l'interface série vont être dérivés par une division sur la cadence interne du microprocesseur de l'ordinateur de commande C-Control / BASIC. La valeur d'octet à transmettre par la commande BAUD contient les valeurs de diviseurs nécessaires Nxx.

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

Les bits 7 et 6 contiennent des prédiviseurs NP pour l'envoi et la réception. NP peut avoir les valeurs les valeurs 1, 3, 4 et 13. Le tableau suivant montre les paramètres requis pour NP1 et NP0 :

PAUSE terme

Dans l'exemple ci-dessous,

PAUSE 50

le programme sera suspendu pendant environ 50*20 millisecondes = 1 seconde. La valeur de la suspension du programme est en principe de +/- 20 millisecondes au maximum.

Communication via une interface série

• Extraction de données

Vous pouvez extraire des données via l'interface série de la station C sous forme de texte. Par exemple, si un PC est relié à un programme de terminal via un câble d'interface, les données concernées pourront s'afficher dessus.

PRINT terme

Dans cet exemple le calcul est renvoyé depuis le terme.

PRINT "texte"

Ici il est transféré dans le texte mis entre guillemets.

Dans les deux cas, la prochaine sortie va se mettre sur la prochaine ligne de l'écran, car le transfert dépend d'un interligne causé par le programme du terminal. L'interlignage peut être éliminé, si vous ajoutez un point virgule après le paramètre (terme ou "texte").

PRINT terme;

ou

PRINT "texte";

CCBASIC supporte plusieurs sorties avec une commande PRINT, pourvu que chaque paramètre soit séparé par une virgule ou un point virgule. Une virgule ajoute une tabulation dans la sortie, qui apparaît sur l'écran sous forme d'un nombre d'espaces. Si deux sorties venaient à se suivre sans être séparées par un espace, vous devez impérativement introduire un point virgule dans la commande PRINT.

PRINT "a= ", a

PRINT "a= "; a

Une commande PRINT séparée sans paramètre ne contient qu'un seul interligne.

PRINT

• Introduction de données

Grâce à la commande

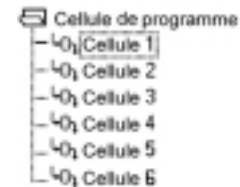
INPUT variable

l'interface série va lire la valeur de l'entier et l'enregistrer dans une variable pour une utilisation prochaine.

La valeur va être envoyée dans un programme de terminal sur un PC et vous pourrez la transférer

l'environnement / Fenêtre de visualisation du projet)" les points "Baum der Programmzellen aufklappen (Ouvrir l'arborescence des cellules de programme)", "Erste Programmzelle öffnen (Ouvrir la première cellule de programme)" et "Editor folgt im Projektfenster ausgewählter Programmzelle (Suivre l'éditeur dans la fenêtre du projet de la cellule de programme choisie)". Ces paramètres vont s'enregistrer lorsque vous quitterez le programme et n'ont pas besoin d'être reconfigurés à chaque fois.

Cliquez sur le dossier "Programmzellen (cellules du programme)" dans le volet d'exploration du projet. Cliquez ensuite 6 fois sur l'icône "Zelle einfügen (Ajouter une cellule)". Vous devez maintenant apercevoir l'arborescence des cellules, telle que ci-dessous :



Cliquez sur l'entrée "Zelle 1 (Cellule 1)" dans le volet d'exploration du projet. Dans la barre de titre de l'éditeur de cellule s'affiche maintenant "Programmzelleneditor - Zelle 1 (Editeur de cellules de programme - Cellule 1)". Cliquez alors sur la zone de travail de l'éditeur de cellules. L'icône de sélection de bloc qui était jusqu'à présent grisé est désormais active.

Dans le menu déroulant de la sélection de bloc, vous devez choisir "Ein-/Ausgabe (Entrée / Sortie)". Cliquez ensuite sur la troisième icône (en partant de la gauche) de la sélection de bloc ; elle représente la touche du module F1.



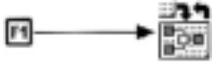
Un nouveau bloc apparaît dans la zone de travail. Déplacez-le vers la droite avec la souris.

Choisissez maintenant dans le menu déroulant le bloc du groupe "Programmsteuerung (Commande du programme)", qui se trouve tout en bas de la liste.



Dans la zone des icônes de la sélection de bloc apparaît la palette du bloc de commande du programme. Cliquez sur la deuxième icône en partant de la gauche, "Unterprogrammaufruf (Appel du sous-programme)". Placez le nouveau bloc un peu à droite à côté du premier.

Les deux blocs doivent maintenant être connectés. Cliquez sur le bloc "F1", bougez le pointeur de la souris en direction de l'autre bloc. Une ligne de tirets ("élastique") indique le mode de connexion. Cliquez ensuite sur le second bloc. L'icône suivante doit apparaître dans la zone de travail :



Qu'est-ce que cela signifie ? L'appel de la sous-fonction du bloc droit correspond à la pression de la touche F1, "à la pression de F1, subdiviser...". Avant d'aller plus loin, vous devez remplir les autres cellules.

Cliquez sur la cellule 2 dans le volet d'exploration du projet. Placez le premier bloc du groupe "Programmsteuerung", le bloc "Verzweige zu Zelle (subdivision vers la cellule)", dans la zone de travail.



Vous avez ainsi réalisé la boucle continue du programme. Depuis la cellule 2, vous devez toujours revenir à la cellule 1. Cliquez maintenant sur le bloc avec le clic droit de la souris. Choisissez dans le menu qui apparaît le point "Eigenschaften (Propriétés)". La boîte de dialogue suivante va s'afficher :



Blockparameter = paramètre du bloc

Geben Sie den Bezeichner einer Zelle ein : = Attribuez une cellule à l'indicateur :

Zelle 1 = Cellule 1

OK = Ok

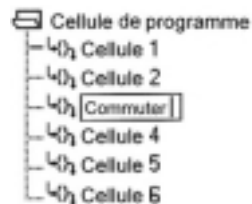
Abbrechen = annuler

Indiquez l'objet de transfert du bloc de branchement dans le champ correspondant, soit la cellule 1 ("Zelle 1"). Veillez à respecter exactement la désignation de la cellule avec celle du volet d'exploration du projet : Z majuscule et un espace avant le 1. Le logiciel va distinguer les majuscules des minuscules, comme un, deux ou trois espaces.

Si vous avez entré "zelle 1", l'ordinateur va afficher un message d'erreur.

Refermez la fenêtre en cliquant sur Ok.

Cliquez maintenant sur la cellule 3 ("Zelle 3") dans le volet d'exploration du projet. Une fois la cellule sélectionnée, cliquez une nouvelle fois sur Zelle 3 : vous pouvez maintenant remplir le champ et changez le nom de la cellule.



Par conséquent, les calculs ou les recherches d'état récurrents ou plus considérables peuvent être encapsulés dans une sous-fonction.

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

```
...
x = calc
...
```

```
#calc RETURN y * z / (y + z)
```

- Aiguillage du programme

ON variable **GOTO** étiquette0,étiquette1,...étiquettes

ou

ON variable **GOSUB** étiquette0,étiquette1,...étiquettes

Dans la fonction de la valeur du sélecteur 'variable' produit un aiguillage de programme ou un appel de sous-routine vers les points d'entrée. Si la valeur est 0, il sera aiguillé vers étiquette0, si elle est égale à 1, ce sera vers l'étiquette1, etc. Si la valeur de la variable est négative ou supérieure au nombre de destinations de branchement, la prise en charge du programme se fera sans aiguillage.

- Fin du programme

END

Lorsque l'ordinateur de commande arrive à l'instruction END, le programme qui était en cours d'exécution se termine. Le système devient ensuite inactif. Vous pouvez maintenant transférer un nouveau de programme d'application ou lancer à nouveau le programme en appuyant sur la touche Start.

- Temporisation du flux de programme

L'instruction

WAIT conditionterme

interrompt l'exécution du programme jusqu'à ce que le calcul de 'conditionterme' renvoie une valeur égale à 0.

```
define F1 port [9]
```

```
...
```

```
WAIT F1
```

Dans cet exemple, le programme va être suspendu jusqu'à ce que le port numérique 9 (touche F1 sur la station) détecte un niveau haut.

La commande PAUSE suspend l'exécution du programme pour un temps donné. La durée de la pause est déterminée par une valeur est calculée grâce à une multiplication du terme avec l'unité de base de 20 millisecondes.

- Instruction de branchement

GOTO étiquette

L'instruction GOTO permet de contraindre l'ordinateur de commande à prendre en charge le programme à un endroit donné. La destination du branchement va être un indicateur étiquette et peut se trouver avant ou après l'instruction GOTO dans le texte source.

- Appel et retour d'une sous-routine

Vous pouvez appeler une sous-routine en utilisant une instruction.

GOSUB étiquette

étiquette est le point de départ de la sous-routine.

Les segments de programme, qui sont réunis dans les sous-routines, peuvent être utilisés plusieurs fois lors de la prise en charge du programme. Une sous-routine commence toujours par une étiquette, contient une ou plusieurs instructions et se termine par

RETURN

Après RETURN, la prise en charge du programme avec l'instruction va continuer vers GOSUB. La prise en charge du programme ne pourra pas accéder à une instruction RETURN si elle n'est pas précédée d'un GOSUB.

Pour appeler une sous-routine depuis une sous-routine, la profondeur d'imbrication maximale autorisée est de quatre.

```
#programmepincipal
GOSUB sub1
...

#sub1
GOSUB sub2
...
RETURN

#sub2
GOSUB sub3
...
RETURN

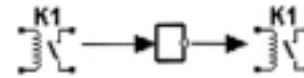
#sub3
GOSUB sub4
...
RETURN

#sub4
...
RETURN
```

Vous pouvez également appeler RETURN avec une valeur chiffrée.

RETURN terme

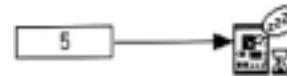
Entrez le terme "Umschalten (Commuter)" et appuyez sur la touche ENTREE de votre clavier. Passez maintenant à nouveau à la zone de travail. Placez le bloc "Relais K1" deux fois sur le groupe "Ein-/Ausgabe (Entrée / Sortie)". Mettre les blocs à la même hauteur. Placez un bloc "Invertieren (Inverser)" du groupe "Logik (Logique)" entre les deux. Connectez les blocs comme ci-après :



L'état de commutation du relais K1 va alors être lu pour être ensuite redistribué mais dans le sens opposé.

Passez à la cellule 4 ("Zelle 4") : cliquez sur Zelle 4 dans le volet d'exploration du projet. L'antirebond de la touche F1 doit maintenant être placé. Le bondissement désigne le caractère importun des contacts mécaniques qui consiste à s'ouvrir et se fermer plusieurs fois avant de pouvoir produire du courant électrique. Dans notre exemple, cela conduirait à la mise hors tension indésirable du relais. Certes le clavier à effleurement de la station a un comportement de rebond mais il vaut mieux faire attention.

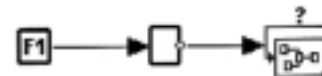
Placez ensuite un bloc "Pause" du groupe "Programmsteuerung (commande de programme)" légèrement vers la droite. Mettez à sa gauche un bloc "konstanter Wert (Valeur constante)" du groupe "Konstanten (Constantes)". Connectez les deux blocs. Cliquez sur le block de constantes avec le clic droit de la souris. Entrez la valeur 5.



La valeur 5 du bloc "Pause" va être enregistrée. Lors de la prise en charge de la consigne, le programme fait une pause de $5 \times 20 \text{ ms} = 100 \text{ ms}$. Pendant cette durée, il n'est plus possible d'appeler aucune autre touche, car le dispositif antirebond est activé.

Cellule 5 ("Zelle 5") : Comme il n'est pas utile que le relais change de sens plusieurs fois, suite à une pression prolongée de la touche F1, le programme doit maintenant patienter jusqu'à ce que la touche soit relâchée.

Pour cela placez dans la zone de travail de la cellule 5 ("Zelle 5") un bloc "Warten auf Wert (en attente de valeur)" du groupe "Programmsteuerung (commande de programme)" légèrement sur la droite. Mettez à sa gauche un bloc F1 et au milieu un inverseur. Vous connaissez maintenant les deux types de blocs. Connectez-les avec le troisième bloc à la même hauteur.



Le bloc "Warten..." (en attente...) ralentit la prise en charge du programme jusqu'à ce que la valeur alimentée de l'entrée soit différente de 0. Cela va inverser l'état de la touche F1, entraînant la mise en attente du programme jusqu'à ce que la touche soit relâchée.

La cellule 6 ("Zelle 6") met fin à la réactivité de la touche F1. Pour cela utilisez un bloc "Rückkehr aus Unterprogramm (Retour depuis le sous-programme)" du groupe "Programmsteuerung (Commande de programme)". En fonction de la cellule 6, le sous-programme va être appelé dans la cellule 2.

Enfin, n'oubliez pas de préciser l'objet de transfert du bloc vers le sous-programme de la cellule 1. Cliquez sur la cellule 1 du volet d'exploration du projet. Ouvrez ensuite la boîte de dialogue des propriétés du bloc de branchement (en faisant un clic droit sur la souris après avoir pointé le bloc, et en ouvrant "Eigenschaften (Propriétés)"). Inscrivez l'objet "Umschalten (Commuter)" comme

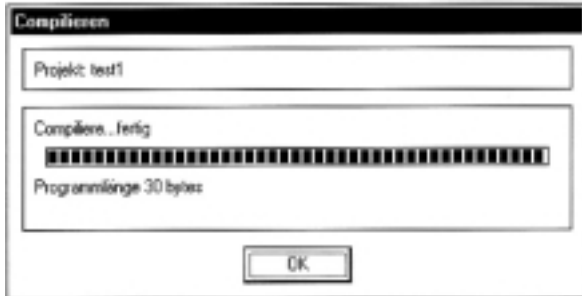
nouveau nom de la cellule 3.

Ce programme est maintenant terminé. Enregistrez votre projet (sous le menu "Datei (Fichier) / Speichern (Enregistrer)) en lui donnant le nom TEST1.CPF.

Compilation et simulation

Vous pouvez maintenant compiler le projet, c'est à dire le convertir dans un format compatible au système de la station C-Control. Sélectionnez le menu "Entwicklung/Compilieren (Développer / Compiler) ou cliquez sur l'icône "Compilieren (Compiler)".

Une fenêtre va s'ouvrir et vous indiquer la progression de la compilation. Une fois terminée, la barre d'octets doit être pleine ou un message d'erreur s'affiche.



Compilieren = compiler

Projekt : test1 = projet : test 1

Compilieren... fertig = compilation... terminée

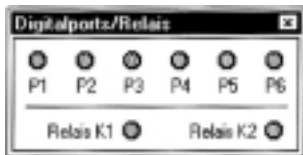
Programmlänge 30 bytes : longueur du programme : 30 octets

Ok = OK

Vous ne trouverez pas de trace de la conversion du projet immédiatement, car elle se situe dans la mémoire vive de votre PC. Le projet converti peut maintenant être transmis au module ou être utilisé pour la simulation. Si vous lisez le nombre d'octets de code, vous vous souvenez sans doute qu'il en reste 8000 disponibles. Vous avez maintenant un aperçu de la complexité de vos tâches de commandes !

Vous pouvez à présent tester la fonction du programme présenté plus haut avec le simulateur.

Cliquez tout d'abord sur le menu "Ansicht (Affichage)" et ouvrez la fenêtre du simulateur "Digitalports/Relais (Ports numériques / Relais)" et "Tasten/LEDs (Touches / LED)".



Digitalports/relais = ports numériques / relais

Relais K1 = relais K1

```
...  
NEXT
```

Dans cet exemple, le pas de progression constant est de 1.

Les valeurs du terme de fin et du terme progressif vont être recalculées à chaque itération de la boucle. Ce qui permet d'avoir un contrôle permanent de la progression du programme.

Les boucles d'itération FOR peuvent s'imbriquer les unes dans les autres. La profondeur des imbrications est limitée par les variables des boucles en fonction de l'espace de la mémoire autorisé.

```
FOR v1 = début1 TO fin1  
    FOR v2 = début2 TO fin2  
        FOR v3 = début3 TO fin3  
    ...  
    NEXT  
NEXT  
NEXT
```

Chaque boucle d'itération FOR ne peut tourner que sur sa propre instruction NEXT en cours de programme. Le texte source qui suit peut en effet être compilé et chargé dans la station C-Control, et pourrait ne pas fonctionner comme prévu :

```
FOR v1 = début1 TO fin1  
...  
    GOTO autretexte  
...  
NEXT  
FOR v2 = début2 TO fin2  
...  
#autretexte  
NEXT
```

Veillez à respecter la plage de valeurs de la variable de la boucle d'itération et le terme de fin !

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

• Exécution conditionnelle

```
IF condition THEN instruction1
```

ou

```
IF condition THEN instruction1 ELSE instruction2
```

La construction IF... THEN... ELSE permet d'adapter le flux du programme par rapport à la condition, pendant l'exécution du programme. Comme condition nous vous conseillons d'utiliser un terme. Si le calcul renvoie une valeur différente de 0, c'est que la condition est remplie et que l'instruction 1 sera exécutée. Si vous utilisez en même temps ELSE et une seconde instruction, cette dernière va être exécutée alternativement, une fois que le terme calculé aura atteint la valeur 0. La construction IF... THEN... ELSE doit se trouver dans une ligne de texte source. Les blocs d'instruction (plusieurs instructions) ne peuvent pas être ajoutés après THEN et ELSE.

Fonctions et commandes mathématiques

Les arguments x et y, qu'ils soient fonction ou commande, sont toujours des termes (voir la définition plus haut).

- La fonction racine **SQR(x)** renvoie une approximation pour la racine carrée depuis l'argument x. Les virgules doivent être supprimées.
- La fonction signe **SGN(x)** renvoie 1 (si la valeur de l'argument x est supérieure à 0) ou -1 (si la valeur est inférieure à 0). Pour x = 0, le résultat de la fonction SGN est égale à 0.
- La fonction maximum **MAX(x,y)** renvoie x (si x > y), sinon y.
- La fonction minimum **MIN(x,y)** renvoie x (si x < y), sinon y.
- La commande **RANDOMIZE** lance le générateur pseudo-aléatoire interne de l'ordinateur de commande avec la valeur de x. Une seule et même valeur d'initialisation renvoie toujours la même suite de chiffres. Le format spécial RANDOMIZE TIMER charge la valeur de l'horloge spontanée dans le générateur.
- La fonction aléatoire **RAND** renvoie la dernière valeur aléatoire de l'entier du générateur pseudo-aléatoire. Les chiffres aléatoires sont le résultat du processus multiplicatif et de la division modulo correspondante, appliqué à la valeur précédente.

Priorité des opérateurs et appels de fonction

Lors du calcul d'un terme avec des opérateurs, leur priorité est décisive. Les opérations ayant une priorité élevée seront exécutées avant les autres. Pour les opérateurs ayant la même priorité, le calcul se fera de la gauche vers la droite.

Comme en mathématiques, les parenthèses dans une expression modifient l'ordre d'évaluation déterminé par la priorité des opérateurs. CCBASIC supporte au maximum 6 parenthèses.

Pour obtenir un programme clair et précis, vous devez éviter les expressions entre parenthèses et écrire sur plusieurs lignes BASIC les calculs compliqués.

La liste suivante répertorie les priorités des opérateurs CCBASIC :

Priorité	Opérateur
9	()
8	appel de la fonction
7	signe négatif
6	* / MOD SHL SHR
5	+ -
4	> >= < <= <>
3	NOT
2	AND NAD
1	OR NOR XOR

Instructions pour commander les flux de programme

- Boucle d'itération

```
FOR variable = début TO fin STEP pas de progression
...
NEXT
```

La boucle d'itération FOR mène les instructions jusqu'à NEXT jusqu'à ce que la valeur de la variable atteigne la valeur du terme du début. La valeur du terme du début calcule et la variable de la boucle est assignée avant la première itération. A chaque passage, la valeur du terme progressif va être ajoutée à la variable de la boucle d'itération.

```
FOR variable = début TO fin
```



Tasten/LEDs = touches / LED

Démarrez le simulateur depuis la commande du menu "Simulator/Ausführen (Simulateur / Démarrer)" ou en cliquant sur l'icône correspondante dans la barre d'outils. Vérifiez bien que le message "Simulation läuft (simulation en cours)" s'affiche dans la barre d'état.

Cliquez ensuite plusieurs fois avec la souris sur F1 dans la fenêtre "Tasten/LED (Touches / LED)". Assurez-vous que la diode verte K1 s'allume dans la fenêtre "Digitalports/Relais (Ports numériques / Relais)". La fenêtre du simulateur fait une copie graphique des fonctions du périphérique de la station C-Control. Le programme va se comporter dans le simulateur comme prévu. La fonction de l'exemple est maintenant démontrée. Vous pouvez arrêter le simulateur et le remettre à l'état initial. Vous trouverez le deux commandes directement dans le menu du simulateur.

Le simulateur vous permet de tester le programme d'application sans matériel informatique supplémentaire. Si des erreurs apparaissent, vous pouvez les supprimer à l'aide de quelques clics de souris. Le programme va d'abord être transféré dans la station. Grâce au module vous économisez ainsi un temps précieux et des manipulations compliquées. Un programme défectueux pourrait engendrer des conséquences désagréables, c'est pourquoi vous pouvez aussi connecter les relais et circuits à une tension de secteur de 230 V~.

Transfert et démarrage du programme

Une fois le programme testé, vous pouvez le transférer.

- Assurez-vous que la station est reliée au PC à une tension de fonctionnement se situant dans la gamme de tension autorisée, via une interface série.
- Appuyez sur la touche Reset de la station.
- Paramétrez l'interface série en allant sur "Optionen (Options) / Umgebung (Environnement) / Simulator und Lader (Simulateur et chargeur)".
- Choisissez la commande du menu "Entwicklung (Développement) / In C-Control übertragen (Transférer dans la station C-Control)".

La progression du transfert va s'afficher sur l'écran sous forme de barre. La diode électroluminescente TX va clignoter pendant le transfert sur la borne RS232 de la station.

La station C-Control est maintenant programmée.

- Appuyez sur la touche Start pour lancer le petit programme de test.

Vous remarquerez que le relais K1 s'allume / s'éteint par une simple pression sur la touche F1 du module, que ce soit pour la programmation ou la simulation. La diode transmet le rétrosignal aux bornes du relais, via l'état de commutation du relais.

Grâce à ce petit programme de test et les autres exemples contenus sur le CD, vous êtes maintenant en mesure de les appliquer à des utilisations plus complexes.

Programmation en BASIC

Généralités

Si vous souhaitez réaliser des projets complexes ou si vous avez des connaissances préalables en langage de programmation, nous vous conseillons de programmer la station C-Control en langage BASIC, en C-Control/BASIC (CCBASIC). Vous trouverez pour ce faire deux versions de la surface du programme sur le CD fourni avec la station. La dernière version peut être utilisée sur un système d'exploitation à partir de Microsoft Windows 95 ou Windows NT version 4.0 minimum. Elle ne peut en

revanche pas être installée sur Windows 3.xx.

Vous trouverez aussi sur le CD une autre version allégée (sans simulateur) qui fonctionne sous MS-DOS et peut être utilisée sur la plupart des PC ou ordinateurs de poche, voire même sur des systèmes 8086/88 (XT).

Il est également possible de faire de la programmation sur la station C-Control depuis les plus vieux ordinateurs portables bon marché, même si les logiciels de bureautique ne sont pas mis à jour, et sans avoir à monter le module dans une armoire de distribution.

Si vous souhaitez installer les deux versions, veuillez suivre les recommandations contenues sur le CD.

Programmation en langage CCBASIC

Généralités

Un programme en langage BASIC est constitué de plusieurs lignes appelées textes sources. Chaque ligne contient un ou plusieurs nombres ou ordres de gestion. CCBASIC est le dialecte BASIC qui doit être utilisé pour programmer la station C-Control. La syntaxe rejoint un peu celle du langage BASIC standard. Il existe pour certaines commandes des différences ou élargissements qui sont spécifiques au matériel de la station C-Control.

Chaque ligne du programme contient plusieurs consignes qui sont séparées par un : double point.

Il n'est pas utile de numéroter les lignes comme dans les anciens dialectes BASIC. Si vous souhaitez néanmoins en introduire, ils pourront être utilisés comme objet de transfert.

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

La numérotation n'aura aucune influence sur l'ordre des opérations du programmes. Par exemple, si vous numérotez une ligne dans le texte source avec le chiffre 200 et que la ligne suivante est 100, la ligne 200 sera exécutée la première.

Vous pouvez insérer des commentaires dans le texte source du programme et en augmenter sa lisibilité et sa facilité d'entretien. Un commentaire en CCBASIC commence avec un apostrophe ' et explique le reste de la ligne qui n'est pas afférant au programme.

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

Indicateurs

Les indicateurs sont des éléments du programme qui émanent des objets programmés et qui apparaissent en caractères alpha numériques (A à Z, 0 à 9). Les noms d'étiquettes et autres "mots réservés" sont également des indicateurs.

Aucune différence n'est faite entre les majuscules et les minuscules. Un indicateur commence toujours par une lettre ou par un curseur trait "_". Les espaces à l'intérieur d'un indicateur sont interdits.

Variables et constantes

Les variables et constantes sont des objets du processus de traitement de l'information. En CCBASIC, elles enregistrent toutes les deux des valeurs numériques. La valeur d'une constante apparaît une fois pour ensuite ne pas être modifiée, tandis que la valeur d'une variable peut changer indéfiniment en cours de programme.

Les constantes peuvent apparaître en CCBASIC sous forme décimale, hexadécimale et binaire. La syntaxe pour les chiffres hexadécimaux et binaires est la même que dans l'exemple ci-après avec le chiffre 46 (décimal) :

```
&H2E      'hexadécimal
&H101110 'binaire
```

```
DEFINE A4 PORT[4]
```

```
DEFINE T1 PORT[5]
```

```
DEFINE T2 PORT[6]
```

```
DEFINE U1 PORT[7]
```

```
DEFINE U2 PORT[8]
```

Récapitulatif des commandes

Ce chapitre donne un récapitulatif complet sur les opérateurs, fonctions et instructions CCBASIC.

Opérateurs mathématiques et logiques

- Signes de calcul de base : + - * /

- L'opérateur modulo MOD donne le reste d'une division d'un entier, avec par exemple, a égale 1.

```
a = 10 MOD 3
```

- Opérateurs de comparaison : > (supérieur à), < (inférieur à), >= (supérieur ou égal à), <= (inférieur ou égal à), = (égal), <> (différent)

Le résultat d'une opération de comparaison est soit - 1 (et non 1!, comparaison vraie) et 0 (comparaison fausse),

```
a = 10 < 3
```

avec ici a égale 0.

- Opérateurs logiques : **NOT** (négation), **AND** ("et"), **NAND** ("et" accompagné d'une négation), **OR** ("ou"), **NOR** ("ou" accompagné d'une négation), **XOR** ("ou" exclusif).

Les opérateurs logiques peuvent être utilisés lors de manipulation d'octets ou de mots sauf si vous souhaitez formuler des commandes (en particulier dans une connexion avec des opérations de comparaison).

- Opérateurs de décalage : **SHL** (décalage vers la gauche) et **SHR** (décalage vers la droite) sont utilisés pour faire un décalage arithmétique d'un échantillon de bit dans une variable d'octet ou de mot. A gauche de l'opérateur se trouve la valeur à déplacer et à droite le chiffre qui désigne le nombre de bits qui doivent être déplacés. Le décalage vers la gauche correspond à chaque décalage d'une multiplication par 2 et au décalage de calcul d'une division par 2.

```
a = 10 SHL 3
a = 10 * 2 * 2 * 2
```

Avec, dans cet exemple, est égal à 80.

Veuillez à faire précéder le décalage SHL d'un signe !

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

Ici, est égal à &B1100000000000000 et non &B0100000000000000 (toutes les valeurs sont binaires) !

DEFINE a BYTE
DEFINE c BYTE

Le compilateur qui s'occupe de la répartition automatique des variables sur la mémoire commence par le numéro de cellule 1. L'exemple ci-dessus contient 6 octets. Si vous définissez d'autres bits, octets et mots en spécifiant le numéro de la cellule, veillez également à ne pas engendrer un chevauchement indésirable.

Un indicateur de variable défini ne doit pas être défini une seconde fois.

Définition des ports numériques

En langage CCBASIC, il est possible d'accéder aux ports et aux variables. Ici aussi, les ports utilisés doivent être définis au préalable. En tout, 16 ports sont occupés.

- Définition d'un des 16 ports numériques :

DEFINE indicateur **PORT**[nr]

Pour 'nr' les valeurs disponibles vont de 1 à 16.

Comme expliqué dans la partie consacrée aux périphériques, les ports sont connectés à différents composants du système. Ce qui donne le bloc de définition suivant :

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]

Définition des ports analogiques

DEFINE indicateur **AD**[nr]

Pour 'nr' les valeurs disponibles vont de 1 à 8.

Les ports analogiques sont également répartis dans le système. La définition la plus convenable est la suivante :

DEFINE A1 PORT[1]
DEFINE A2 PORT[2]
DEFINE A3 PORT[3]

Pour les variables, on y accède via leur indicateur, lequel doit être défini dans une ligne **DEFINE** avant la première utilisation de la variable dans le programme.

Étiquette

Les étiquettes marquent des points précis dans l'ordre des opérations du programme. Les étiquettes sont des lignes d'opérations de rupture de séquence qui se trouvent à l'intérieur d'un algorithme.

Dans le langage CCBASIC, les étiquettes se trouvent au début d'une ligne et commencent toujours par une double croix (diésis) #, suivi de l'indicateur de l'étiquette, sans espace.

L'exemple montre la définition de l'étiquette "Label1" et l'utilisation dans une commande de saut.

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

Terme

Un terme donne une valeur déterminée (sous forme de variable ou constante) ou via un calcul. Les termes sont des parties de consignes qui se trouvent, par exemple lors de l'assignation d'une valeur, sur une variable à droite du symbole "=". Les termes sont le résultat de combinaisons entre les opérateurs et opérands.

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

Opérands et opérateurs

Un opérande est en format de base soit une constante, soit une variable, soit un appel de fonction, mais peut être aussi un terme combinant opérands et opérateurs.

Les opérateurs permettent de faire des opérations de calcul qui sont exécutés avec les opérands présents. Les opérateurs sont classés par ordre de priorité, ce qui définit ensuite l'ordre des calculs.

Fonctions

Une fonction réalise une opération prédéfinie, comme par exemple un calcul, et délivre une valeur de résultat via son appel. La plupart des fonctions reçoit un ou plusieurs arguments qui se trouvent entre parenthèses, après l'indicateur de fonction, et sont séparés par des virgules. Certaines fonctions peuvent être appelées sans argument, et dans ce cas aucune parenthèse n'apparaît.

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

En langage CCBASIC, toutes fonctions soutenues sont prédéfinies. Ses indicateurs font partie des mots réservés.

Assignations

L'assignation est la forme la plus simple d'une consigne de programme. L'indicateur d'une variable, qui doit devenir une valeur assignée, est suivi du signe d'assignation "=" et d'un terme, qui correspond à la valeur qui a été allouée.

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

Commandes

A côté des assignations simples, il existe des commandes / consignes qui permettent d'exécuter des opérations de programme via la station C-Control. Les commandes commencent toujours par un mot réservé. Certaines commandes sont constituées d'un ou plusieurs paramètres qui permettent de préciser l'opération de programme à effectuer. Ces paramètres qui se trouvent après l'indicateur de commande, sont suivis d'un espace et séparés par une virgule (exception pour PRINT, voir l'ensemble des commandes). Contrairement aux arguments qui apparaissent à l'appel d'une fonction, les paramètres de commande ne se trouvent pas entre parenthèses !

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

Instructions pour commander les flux de programme

Ces instructions permettent de commander la suite des opérations de programme qui sont prises en charge séquentiellement et ajustées aux valeurs d'entrée du processus de traitement de l'information. Elles demandent un haut niveau de flexibilité lors de la formulation d'algorithmes et permettent de résoudre la plupart des problèmes d'utilisation technique et même des hypothèses de base.

Les instructions permettant de commander les flux de programme sont composées d'un ou plusieurs mots réservés auxquels il est possible d'ajouter d'autres spécifications.

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

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

Instructions de compilation

En plus des instructions de programme, un texte source CCBASIC comprend des instructions de compilation, qui servent par exemple à construire des blocs de données (tableaux) ou à définir des variables et des constantes.

Vous ne pouvez pas séparer des instructions de compilation qui se trouvent sur la même ligne selon la règle du double point. Chaque ligne doit comprendre une seule instruction de compilation.

L'instruction DEFINE

L'instruction DEFINE est une instruction de compilation.

Définition des constantes symboliques

C'est un bon style de programmation qui utilise des constantes symboliques à la place des chiffres "magiques" dans les programmes.

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

En attribuant des indicateurs significatifs aux constantes, la lisibilité des textes sources devient plus accessible. Lorsque toutes les constantes ont été globalement définies, il est ensuite plus facile de surveiller le programme. C'est particulièrement valable lorsqu'une seule et même constante est utilisée à répétition dans le programme.

La définition d'une constante symbolique est la suivante :

```
DEFINE indicateur valeur
```

La valeur est soit un nombre décimal, hexadécimal ou binaire. Ce qui sur notre exemple donne :

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

Définition des variables

Le système d'exploitation de la station C-Control possède une mémoire interne (RAM) de 24 octets dédiés aux cellules de la mémoire qui est à votre disposition. Toutes les variables d'un programme BASIC sont enregistrées dans cette zone de sauvegarde. Les 24 octets peuvent être utilisés aussi bien dans des opérations bit à bit que comme entier de 16 bits (mot).

Contrairement au langage BASIC standard, les variables doivent être prédéfinies avant d'être utilisées dans un programme sous CCBASIC. C'est pourquoi il est impératif de spécifier le type de données (bit, octet ou mot) et de numéroter les cellules de la mémoire (pour les bits, la numérotation est obligatoire !). Vous devez vous assurer que l'attribution de l'espace de la mémoire ne crée aucun chevauchement indésirable, sinon cela risque d'écraser les variables déjà existantes.

Par exemple, bit[18], octet[2] et mot[1] recouvrent chacun une partie de la cellule 2 de la zone de la mémoire.

- Définition d'une variable d'un bit :

```
DEFINE indicateur BIT[nr]
```

Pour 'nr' les valeurs disponibles vont de 1 à 192 (24 octets par 8 bits).

- Définition d'une variable d'un octet avec un numéro de cellule :

```
DEFINE indicateur BYTE[nr]
```

Pour 'nr' les valeurs disponibles vont de 1 à 24 (24 octets).

- Définition d'une variable d'un entier avec un numéro de cellule :

```
DEFINE indicateur WORD[nr]
```

Pour 'nr' les valeurs disponibles vont de 1 à 12 (un mot contient 2 octets).

Si les données de la cellule [nr] sont supprimées lors des définitions d'octet ou de mot, c'est le compilateur qui se charge de la répartition sur la zone de sauvegarde. Veillez à ce que les octets et mots ne soient pas définis alternativement. Les instructions suivantes

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

permet d'avoir deux octets non utilisés, un entre a et b, et un entre c et d, car en principe les mots ne peuvent pas être ajustés aux octets 1,3,5,7... etc. des 24 octets.

Il conviendrait mieux d'écrire

```
DEFINE b WORD
DEFINE d WORD
```