



Conrad Szaküzlet 1067 Budapest, Teréz krt. 23. Tel: (061) 302-3588
 Conrad Vevőszolgálat 1124 Budapest, Jagelló út 30. Tel: (061) 319-0250

"C-Control Station"

Rend.sz.: 125113

Tartalomjegyzék

A GARANCIA	1
RENDELTESETÉS	1
JELLEMZŐK	1
KEZELÉS	1
Programozás	1
Felszerelés	1
A program futtatása	1
KEZELŐSZERVEK	1
Nyomógombok	1
Világítódiodák	2
CSATLAKOZÓ KAPCSOK	2
Kisfeszültségű kapcsok	2
Hálózati kapcsok	2
A modul csatlakoztatása és programozása	3
RENDSZER ÁLLAPOTOK	3
Reset	3
Standby mód	3
Run mód	3
PROGRAMOZÁS (ÁLTALÁNOS)	3
Áttekintés	3
Bit, bájt, szó	3
Rendszer erőforrások	3
GRAFIKUS PROGRAMOZÁS	4
Áttekintés	4
Blokk-készlet	4
Példa grafikus programozásra	4
PROGRAMOZÁS CCBASIC-BEN	5
Áttekintés	5
A CCBASIC alapjai	5
Az utasítások áttekintése	7
Példa CCBASIC-ben való programozásra	10
MŰSZAKI ADATOK	11
ÁBRÁK	11

A garancia

A Conrad Electronic nem vállal felelősséget sem személyi, sem anyagi károkért, melyek a CC station használatából fakadnak. A garancia a szállítási sérülésekre, gyártási hibákra vonatkozik.

Rendeltetés

A C-Control komputer DIN sínre szerelhető. Érintésveszélyes feszültség (230V hálózati- vagy 24V fölötti relé feszültség) esetében csak zárt kapcsolószekrénybe szerelhető vagy olyan dobozba, ahol a csatlakozó kapcsok takarva vannak.

A készüléket 35 szorítós csatlakozó kapocs segítségével lehet összekötni más készülékekkel. Ezek közül a 10-17-en hálózati feszültség van, amire tekintettel kell lenni a veszélyek elkerülése érdekében.

A készülék olyan műszaki rendszerbe építhető be, amely nem gyógyászati, egészségügyi vagy életfenntartási felhasználási célt szolgál, illetve amelynek használatából személyeket vagy vagyontárgyakat érintő veszély nem keletkezhet. Robbanásveszélyes, vagy kémiai/agresszív anyagot tartalmazó környezetben nem szabad használni.

A vezérlőegység programozásához csak a vele együtt szállított PC-szoftver használható. Az alkalmazói programot először szimulátorral tesztelni kell.

Jellemzők

Stabilizált hálózati tápegység
 Csatlakozókapcsok puffer akku számára
 Programozható vezérlő egység mikrokontrollerrel
 6 programozható digitális port
 4 programozható nyomógomb
 4 programozható LED
 2 relé 230V-ról működő készülékek közvetlen kapcsolására (záró, max. 6A tartós áram)
 2 hőérzékelő áramkör (-25...102,5 C között lineáris, felbontás 0,5 C), összekötő vezetékekkel, melyek tetszőlegesen hosszabbíthatók.
 4 programozható analóg port, feszültség mérésre (0...2,55V)
 Soros interfész PC csatlakozáshoz
 DCF-77 rádióvezérlésű óra antenna csatlakozó
 Frekvenciamérő port (pl. redőny vezérlésnél szélkerék csatlakozáshoz)
 Beépített piezo hangszóró.

Kezelés

Részei: programozás, szerelés, program futtatás.

Programozás

- A szeretlen modul a PC munkahelyen lehet programozni, a már szekrénybe szerelt esetleg a helyszínen hagyva átprogramozhatjuk, amennyiben lehetőség van PC-s összeköttetés létesítésére.
- Installálja a programozó szoftvert a mellékelt CD-ről (install.txt-t és readme.txt-t elolvasni).
- Kösse össze az állomást a PC-vel és a tápfeszültséggel, a csatlakozóknál leírtak szerint. Kapcsolja be a tápfeszültséget.
- Írjon egy felhasználó programot a készülék működtetésére (ld. programozás fejezet).
- Fordítsa le a programot a szoftver segítségével.
- Tesztelje a program működését a szoftverben levő szimulátorral.
- A bevizsgált programot szoftver segítségével vigye át az állomásra.

A készüléket ezután be lehet szerelni.

Felszerelés

- Szerelje fel a hordozó rendszert, pl. elosztó szekrényt.
- Tegye fel a modult és a csatlakozó egységeket a sínre.
- Huzalozza össze a rendszert. (Ld. csatlakozók).
- Szükség esetén nézesse át a rendszert szakemberrel.
- Utolsó lépésként kösse be a hálózatot/táplálást.

A program futtatása

- Kapcsolja be a tápfeszültséget.
- Nyomja meg a Start gombot.
- Ezután a készülék a program szerint kezelhető. Mindaddig működik, míg a tápfeszültség meg nem szűnik, vagy meg nem nyomják a Reset gombot. Újraindítás: Start gombbal.

Kezelőszervek (ld. 1. ábra)

Nyomógombok

Az előlapon 6 gombos fólia billentyűzet található. A gombot addig kell nyomni, míg kis kattantást nem érzünk.

Reset

Többnyire a program megállítására szolgál. Ezután a rendszer olyan inicializált állapotban van, mint a tápfeszültség bekapcsolása után. A felhasználói változók törölve vannak ("0" érték), a belső óra 00:00:00-ról indul, és DCF aktívantenna csatlakoztatása után újra lesz szinkronizálva.

Start

A soros interfészen át korábban betöltött alkalmazói program indítása (ld. programozás). A program az END parancsig fut.

F1, F2, F3, F4

A felhasználói programból lekérdezhető funkciógombok, használhatók pl. üzemmód kiválasztásra, be/ki kapcsolásra, paraméter változtatásra (ld. programozás).

Világítódiodák

9db van; színek funkció csoport szerint. Piros: rendszer, zöld: relé, sárga: felhasználói.

Start gombban levő LED (piros)

A program futtatás alatt világít. Ez rendszerint a Start gomb megnyomása után következik be.

Soros interfész LED (piros)

Jelzi az interfész TX vezetékén történő átvitelt. Kapcsolástechnikailag az adatvezetékhez van csatolva, és minden low (alacsony) bit átvitelekor röviden felvillan. Mivel a modulba való letöltéskor (download) minden bájtt echózva lesz a PC-nek, ez a folyamat is követhető a LED-del.

DCF-77 LED (piros)

Ha van DCF antenna csatlakoztatva, a LED mutatja az adat vétel állapotát és a modul belső valós idejű órájának szinkronizációját.

Rendszeres villogás másodpercenként:

Idő adatok vétele, a szinkronizáció a vételi körülményektől függően néhány percig (max. 2-3) tart.

Folyamatosan világít

A szinkronizáció sikerült

Rendszertelenül világít

Az antenna rosszul van irányozva, vagy vételi zavar van.

Nem világít

Az antenna nincsen vagy rosszul van csatlakoztatva, rosszul van irányozva, vagy erős vételi zavarok vannak.

K1 és K2 LED-ek (zöld)

A K1 és K2 relé állapotáról tájékoztatnak. Akkor világítanak, mikor a megfelelő relé be van húzva, az áramkör zárva.

Az üzemmód gombok fölötti LED-ek (sárga)

Az F1...F4 LED-ek az alkalmazói programmal ki/be kapcsolhatók, és a program állapotát jelzik.

Csatlakozó kapcsok

Kisfeszültségű kapcsok

RS232 soros interfész

A modul programozásakor az 1-3 kapcsokra csatlakozik az interfész kábel. Adatátvitelhez a modul működés közben is ráköthető a PC-re.

- 1: Adat vétel vezeték (RX), az interfész kábel fehér ere
- 2: Jel föld vezeték (test, GND), az interfész kábel barna ere
- 3: Adat küldés vezeték (TX), az interfész kábel zöld ere

DCF77 aktívantenna

Kívánság szerint DCF antennával szinkronizálható a modul (4-6 kapcsok).

- 4: Jel bemenet vezeték, ide jön az antenna null-aktív vezetéke
- 5: Jel föld vezeték (test, GND)

6: Stabilizált tápfeszültség (5V) a rendszerből az antennának (v. más külső áramkörnek) (vegye figyelembe a terhelhetőséget!).

Kisfeszültségű tápellátás és akku puffereles

Ha a 7. pontra stabilizált feszültséget vagy 12V akkut kötünk, a programozást a kapcsolószekrénybe való szerelés és a 16 és 17 pontokra a 230V-os hálózat bekötése nélkül el lehet végezni. A normál üzem során (kapcs. szekrény, 230V) is lehet 12V ólomzselés akkut csatlakoztatni (ajánlott kapacitás min. 1,2Ah), a rendszer pufferelesére, amikor a hálózat kimarad. Az akkunak jól töltve kell lenni a relék táplálásához. Amikor a hálózat jelen van, kísértékű töltőáram folyik az akku önkisülésének ellensúlyozására. Ezt azonban nem szabad üres akkuk feltöltésére használni!

A kivezetett rendszerfeszültség kisebb külső áramkörök táplálására szolgál. Vegye figyelembe a terhelhetőséget, valamint hogy a valóságos feszültség a névleges 12V-ot a terheléstől függően jelentősen túllépheti, és akár 24V is lehet!

7: Stabilizált 12V vagy akku bemenet plusz

8: test (GND)

9: Stabilizálatlan rendszer feszültség kimeneti pont

Hőmérséklet érzékelő portok

A mikrokontroller A/D átalakítójára két hőérzékelőt lehet kötni. A rendszerrel két AD 592 típusú érzékelőt szállítunk, melyeket vezetékkel a T1 és T2 portokra lehet kötni. A belső elektronika külön-külön van az egyes érzékelőkhöz illesztve a pontos mérés érdekében. A csatlakoztatásnál ügyelni kell az összejelölő színekre (ld. táblázat), valamint a polaritásra.

Az AD 592 érzékelő egy hőfüggő áramgenerátor, 1uA/K érzékenységgel. A szállított vezeték néhány méterrel meg lehet hosszabbítani, a zavarvédelem érdekében árnyékolt kábel használatával. A hosszabbító vezeték árnyékolását hozzá kell kötni az eredetihez, és a CControl egyik GND kapcsához (polaritásra ügyelve).

18: T1A, 1-es szenzor, fekete kábel, fehér ér

19: T1B, 1-es szenzor, fekete kábel, barna ér

20: T2A, 2-es szenzor, szürke kábel, fehér ér

21: T2B, 2-es szenzor, szürke kábel, barna ér

GND kapcsok (test)

A 22, 27 és 34-es kapcsokon van a rendszer föld, a digitális és analóg portok viszonyítási pontja.

Analóg portok

Testpotenciálhoz viszonyított feszültség mérésekhez szolgálnak. A CControl-nak 8 bites A/D átalakítója van, amely 0...255 közötti értékeket tud kiadni. A referencia feszültség 2,55V, ami a 255-ös átalakítási értéknek felel meg.

Kapcsok: 23...26, nevek: A1...A4.

Digitális portok

Ezek be- és kimenetként is használhatók, kapcsolási folyamatok vezérlése során. Logikai szintek: KI: 0...0,7V, BE: 4,3...5V.

Kapcsok: 28...33, nevek: P1...P6.

Frekvenciamérő bemenet

A 35-ös kapcsan digitális pulzusok f frekvenciája mérhető. A bemeneti jel lehet 0/5V-os digitális jel vagy a 34-es GND ponthoz képesti ciklikus kontaktus létesítés (pl. szélkerék) segítségével. A mérési tartomány 1%-os hibával kb. 30kHz-ig terjed.

Hálózati kapcsok

Tápfeszültség csatlakozás

Normál üzemben innen lesz táplálva a modul minden belső elektronikus része. Ehhez rendelkezésre áll egy hálózati tápegység transzformátorral, egyenirányítóval és stabilizátorral.

A 16 pontra a fázis, a 17-re a nulla kerül.

K1 és K2 relék

Hálózati feszültségről működő készülékek kapcsolására szolgálnak. A műszaki adatokban megadott áramértékeket nem szabad túllépni. A tápfeszültség és a relé kapcsok közötti üres kapcsok a biztonságos távolság betartására vannak elhelyezve.

A modul csatlakoztatása és programozása

Az állomás és a PC összekötése

Az alkalmazói program áttöltésére tápfeszültségtől függően két módszer van. Az interfész kábelt mindkét esetben a következőképpen kell kötni:

-Kösse az állomás 9 pólusú interfész kábelét a PC egyik soros interfészére (COM1...COM4).

-Kösse a fehérrel jelzett eret az állomás 1-es kapcsára - RX

-Kösse a barnával jelzett eret az állomás 2-es kapcsára - GND

-Kösse a zölddel jelzett eret az állomás 3-as kapcsára - TX

Programozás kisfeszültségű táplálásnál

A 7(plusz) és 8(mínusz) pontokra 12V-os stabilizált hálózati tápegység kimenetet vagy egy akkumulátort kell kötni. Ez a megoldás jól használható PC munkahelyeken, laborban, műhelyben és irodában.

Programozás hálózati táplálás mellett

(3. ábra)

A programnak a PC-ről a modul telephelyére való áttöltése során használatos. A CControl állomás DIN sínre van szerelve. Érintésvédelmi célból legalább a hálózati csatlakozó pontokat szigetelni kell.

A programozó kábel csatlakoztatásakor a CControl állomásnak feszültségmentesnek kell lennie!

Noha lehetőség van az asztalon szabadon elhelyezett modulnak hálózatról való megtáplálására II. érintésvédelmi osztályú készülékekhez szolgáló kábellel, ezt érintésvédelmi megfontolásokból mégsem szabad megtenni! (4. ábra)

Rendszer állapotok

Reset

A tápfeszültség rákapcsolása vagy a Reset gomb megnyomása után a modul végrehajtja az inicializálási műveleteket, majd stbby (nyugalmi) állapotba lép.

Az inicializálás lépései:

1. Dátum és pontos idő: 1997, jan. 1. 00:00:00
2. A DCF 77 vétel előkészítése
3. Soros interfész beállítás: 9600 baud, 8 adatbit, 1 stopbit, paritás nincs
4. Felhasználói változók törlése (=0).

Standby mód

Ebben az üzemmódban folyamatosan le lesz kérdezve a Start gomb és a soros interfész, a felhasználói program indítására vagy a főlérendelt PC parancsaira való reagálásra. Ebben a módban lehet a modult programozni (letöltés, "download").

A Start gomb megnyomásának észlelése után a rendszer program futtatás (run) módba lép.

Run mód

Ebben az üzemmódban a felhasználói program parancsai lesznek végrehajtva. Kilépés: Reset gombbal, tápfeszültség kikapcsolással, vagy az END paranccsal/blokkal. END után a rendszer ismét stbby módba lép.

Programozás (általános)

Áttekintés

A modul rugalmasan programozható és átprogramozható (pl. hőmérséklet szabályozás, redőny vezérlés, riasztó, lépcsőház világítás kapcsoló stb.).

Tartalmaz egy egychipes számítógépet (mikrokontrollert), melyhez egy memória chip kapcsolódik a felhasználói programok tárolására. Az EEPROM a programot a tápfeszültség kikapcsolása után is megőrzi.

A programozás PC -n keresztül, grafikus felület segítségével vagy a népszerű BASIC nyelven lehetséges.

Bit, bájt, szó

A rendszer csak egész ("integer") számokat ismer:

Bit: lehetséges értékei 1 és -1 (nem 1!), állapot tárolásra használatos ("flag").

Byte: 8 bit, értéktartománya 0...255, kis nemnegatív egészek, ASCII karakter kódok, parancs kódok.

Word (integer): 16 bitet foglal, high és low bájtra osztva, értéktartománya -32768...32767; általános célokra.

Rendszer erőforrások

Perifériák

A csatlakoztatott eszközökkel való kapcsolattartásra szolgáló perifériák többsége közvetlenül az előlap csatlakozó kapcsainál vagy kezelőszerveinél érhető el.

Digitális portok: összesen 16 db van. Az 1...6 portok a P1...P6 kapcsokra vannak kötve, be- és kimenetként egyaránt használhatók. A 7 és 8 portok a K1 és K2 relét kapcsolják. A 9...12 portról az F1...F4 funkció gombok kérdezhetők le, a 13...16-tal a funkció gombok fölötti LED-eket lehet be- és kikapcsolni.

Analóg portok: az analóg mennyiségek (pl. hőérzékelő jele) számára 8 db 8 bites felbontású analóg-digitális átalakító port van. Mérési tartomány: 0-2,55V, 10mV-os lépésekben. Az A1...A4 portok az A1...A4 kapcsokra vannak kivezetve.

Az 5 és 6 analóg portok a T1 ill. T2 hőérzékelőhöz tartoznak.

A 7-es port a tápfeszültség elektronikát méri: ha pl. kimarad a hálózat, és a 7-8 kapcsokra 12V-os akku van kötve, a mérési érték kb. 120. Az érték csökkenéséből figyelemmel kísérhető az akku feszültség csökkenése, és kritikus esetre az alkalmazói programmal óvintézkedéseket lehet végrehajtatni.

A 8-as port a 230V-os hálózat kiesését érzékeli. Belül a hálózati hídegyenirányítóról leosztott feszültséggel van összekötve, amelyet kondenzátor simít. Ha a mért érték 0, hálózatkimaradás van. Mivel a mikroprocesszor tápfeszültség nagyobb pufferral van ellátva, van még egy kis idő egy megfelelően megírt program segítségével történő rendezett lezárásra.

Valósidejű óra: A 4...6 kapcsokra DCF aktívantennát lehet kötni. Sikeres vétel esetén a rendszer közép-európai téli ill. nyári időre áll. Az idő- és dátum adatokat a programból le lehet kérdezni, és - esetleg logikai kombinációban egyéb feltételekkel együtt - műveletek feltételes végrehajtására használni.

Frekvenciamérés: kb. 30kHz-ig lehet frekvenciát mérni és a programból lekérdezni.

Soros interfész: alkalmazói program betöltésére, valamint a programfutás során bájtos vagy szöveges adatátvitelre szolgál. Az adatátviteli sebesség 1200...9600 baud közötti fokozatokból választható.

Hangjelzés: a készülék programozható magasságú és időtartamú hangokat tud kiadni piezo átalakítóval, pl. rövid dallamok, nyugtázó és figyelmeztető jelzések, billentyű megnyomás jelzés.

Hangjelzés: a készülék piezo átalakítójával programozható magasságú és időtartamú hangokat lehet közvetíteni, pl. rövid dallamok, nyugtázó és figyelmeztető jelzések, billentyű megnyomás jelzés.

Program tár

Az Ön által írt programot a "compiler" lefordítja a CControl számára feldolgozható formába, és behelyezi annak kb. 8000 bájtos tárába. Pl. egy relé bekapcsolása 5 bájtot igényel; valamivel bonyolultabb programok sem használnak többet néhány száz bájtjánál, így elegendő hely áll rendelkezésre. A tárolás EEPROM chipben (Microchip 24C65) történik, amely tápfeszültség lekapcsolás után is megtartja a programot.

Adat tár

A megmaradó szabad bájtokba a mérési eredmények stb. kerülhetnek. Kiolvasás: soros, szavas formában. Egy rögzített érték tehát 2 bájtot foglal.

Az adatfeldolgozás fájlok szerkezetben történik. A fájlokat meg lehet nyitni kiolvasás, írás, hozzáírás céljából, majd be kell zárni. Tápfeszültség kikapcsolás idejére az adatok is védettek, feltéve, hogy a fájl be lett zárva.

Munka memória

A számítások közben szükség van átmeneti adatok tárolására is. Ehhez 24 bájtos munkatár áll rendelkezésre, melyet definiálható változókon keresztül bites-, bájtos- vagy szavas formában használhatunk.

A tár tartalma csak bekapcsolt tápfeszültség mellett őrződik meg. Reset (visszaállítás) vagy tápfeszültség kiesés esetén a tárolt értékek 0-ra állnak.

Aritmetikai egység

A műveletek egész számokkal végezhetők, 16 bites formában. Lehetséges műveletek: a négy alpművelet, modulo osztás (egész osztás maradék értékének meghatározása), bitenkénti logikai kapcsolat, összehasonlítás, negálás, abszolútérték meghatározás, kerekített négyzetgyökvonás, előjel (signum) függvény. Van egy inicializálható pszeudo véletlen generátor is. Az egymásba skatulyázás max. mélysége zárójelezésnél ill. hasonló műveleteknél 6.

Mivel a 16 bites szó -32768...+32767 lehet, ügyelni kell, nehogy ehhez képest "túlcsordulás" lépjen fel - pl. az $a=255+1$ művelet eredménye 256 helyett 0 lesz, ha az a csak egy bájtot reprezentál, az $a=-32768-1$ pedig -32769 helyett 32767 lesz, ha az a egy szót (=2 byte) reprezentál. 10/4 eredménye 2, nem 2,5.

Grafikus programozás

Áttekintés

Az itt használt programozási mód elsősorban kezdők számára és egyszerűbb programokhoz szolgál. A szoftvert a mellékelt CD-ről lehet installálni. Rendszer követelmény a grafikus programozáshoz: min. Windows95, vagy Windows NT4.0-tól. Win 3.xx-el nem lehet telepíteni.

Mivel a szoftver kezelés szemléletes, könnyen követhető, itt csak a legalapvetőbb ismereteket adjuk meg. A PC-n az F1 gombbal lehet behívni az általános segítő, Ctrl-F1-gyel az adott témához kapcsolódót. Ezen kívül a funkcióblokkra való jobboldali egérgomb kattintással és a felajánlott "Hilfe zum Blocktyp" menüpont kiválasztásával leírást kaphatunk a blokk funkcióról.

Az 5. ábra áttekintést ad a felhasználói felületről. A legfontosabb elemeken felirat van. A Windows alapfogalmak ismeretét feltételezzük. Az 5. ábra felirata:

Menüleiste	menüsor, részei: file, szerkesztés, nézet, blokkok, fejlesztés, szimulátor, opciók, ablak, segítő
Zelle einfügen	cella beiktatás

Ein/ausgabe	be/kivitel
Blockgruppen auswahl	blokkcsoport készlet
Blockauswahl schalter	blokk kiválasztó kapcsoló
Arbeitsbereich	munkafelület
Statuszeile	státusz sor

A grafikus program ún. cellákból áll, melyek a C-Control által végrehajtandó műveleteket jelképezik. Baloldalt láthatók a projektünk cellái. Új cella beiktatás: pl. rákattintva a "Zelle einfügen" kapcsolóra.

A cellákban levő lépéseket blokkra mutatja, ahol az egyes funkció blokkokat jelfolyam ábrázoló vonal köti össze. A blokkkapcsolási rajzokat a cella editorban lehet elkészíteni és alakítani.

Blokk-készlet

A 6. ábra szemlélteti, hogy egyes blokkok értéket adnak ki, mások feldolgoznak, vagy tárolnak ill. az értéktől függő műveletet hajtanak végre. A C-Control blokk-készlete:

- **Be/kivitel** - relék, gombok, LED-ek, hangjelzés kezelés
- **Mérési eredmények** - hőérzékelők, feszültségmérés, általános analóg portok kezelése
- **Digitális portok** - P1...P6 port kapcsok kezelése
- **Számítások** - matematikai kapcsolatok és függvények
- **Összehasonlítás** - a művelet eredménye igaz (=1) vagy hamis (=0) lehet.
- **Logika** - bitenkénti összehasonlítás Boole-algebra szerint
- **Idő** - belső valósídejű óra (DCF77), valamint timer (időzítő)
- **Változók** - tároló blokkok
- **Konstansok** - állandó értékeket szolgáltató blokkok
- **Adatrögzítés** - adatok beírása az EEPROM chipbe
- **Soros interfész** - szöveg és adatok be- és kivitele a rendszer interfészen keresztül
- **Program vezérlés** - programfutást vezérlő funkcióblokkok

Példa grafikus programozásra

Program készítése

A példa nyomán a CD-n található példákat is lehet használni saját programhoz.

Tegyük fel, hogy az F1 gombbal kívánja a modul K1 reléjét be/kivitel kapcsolni.

Indítsa el a CCSTAT.EXE programot. Kezden új projektet a "Datei/Neu" (fájl/új) menüparanccsal. Aktiválja az "Optionen/Umgebungseinstellungen/Projektübersichtsfenster" dialógusban a következő pontokat: "Baum der Programzellen aufklappen" (programcellák fájlat nyitni), "Erste Programzelle öffnen" (első programcellát nyitni), "Editor folgt im Projektfenster ausgewählter Programzelle" (szerkesztés a kiválasztott programcellában). Ezen beállítások a program befejezéséig tárolódnak, nem kell őket mindig újból bevinni.

Kattintson a "Projektexplorer"-ben a "Programzellen" (programcellák) mappára, majd 6-szor kattintson a "Zelle einfügen" (cella beiktatása) kapcsolóra. Látni fogja, ahogyan bekerülnek az új cellák. (7. ábra.)

Kattintson a "Projektexplorer"-ben a "Zelle 1" bevitelre. A címsorban megjelenik a "Programzelleneditor - Zelle1" felirat. Kattintson a cella editor munkafelületére - az eddig szürke blokkválasztó kapcsolók aktívá válnak.

A blokkcsoport választóban "Ein/Ausgabe" (be/kivitel) kell álljon. Kattintson balról a 3.-ra, amely az F1-et jelképezi. (8. ábra)

A munkafelületbe egy új blokk lesz beillesztve. Vonszolja a blokkot az egérrel kicsit jobbra le.

Váltson a blokkcsoport választóban (Blockgruppenauswahl) a lista alján található "Programmsteuerung" (program vezérlés) csoportra. (9. ábra.)

A blokkválasztó kapcsoló tartományban megjelenik a programvezérlő blokkok palettája. Kattintson balról a második, "Unterprogramm aufrufen" (alprogram behívás) kapcsolóra. Helyezze el az újabb blokkot az előbbiétől kissé jobbra.

Most következik a blokkok összekötése. Kattintson az F1 blokkra, majd mozgassa az egérmutatót a másik blokk irányába.

Szagatott vonal ("gumi-szalag") mutatja az összekötési módot. Kattintson a második blokkra, a munkafelületen megjelenik a blokkábra (ld. 10. ábra).

Jelentése: a jobboldali blokk alfunkció behívása ahhoz a feltételhez van kötve, hogy a modul F1 gombját megnyomják, vagyis "...ha F1, akkor elágazás...". Az elágazás tisztázása előtt töltsük fel a többi cellát.

Kattintson a "Projektexplorer"-ben a 2-es cellára. A munkafelületbe helyezze el a program vezérlés első, "Verzweigen zu zelle" (elágazás cellához) blokkját. (11. ábra.)

Végtelen ciklus létesítés: a 2-es cellából mindig az 1-esbe kell ugratni. Kattintson a jobb egérgombbal a blokkra - megjelenik egy menü. Válassza az első, "Eigenschaften" (tulajdonságok) pontot, amire egy blokk paraméter párbeszédablak következik (12. ábra). Adja be a cella azonosítóját ("Geben sie den Bezeichner...").

Írja be a beviteli sorba az elágazás blokk ugrási célját, itt az 1-es cellát. Ügyeljen a formára: nagy Z, az 1-es előtt szóköz. A szoftver a kis- és nagybetűket, valamint az 1, 2 vagy nulla szóközt is megkülönbözteti, nem megfelelő forma esetén hibát jelez.

Ok-val zárja a dialógust.

Kattintson a 3-as cellára, és amikor be van jelölve, ismét kattintson rá. A megjelenő beviteli mezőben cellanevet lehet módosítani. Írja be: "Umschalten" (bekapcsolás), nyomjon enter-t.

Lépjen ismét a munkafelületre. A be/kivitel (Ein/ausgabe) csoportból kétszer rakja le a "Relais K1" blokkot. Rendezze lazán egymás mellé a blokkokat, és tegyen közéjük egy inverter blokkot ("Invertieren") a "Logik" csoportból. Kösse össze őket a 14. ábrán látható módon. A struktúra működése: a K1 relé állapota ki lesz olvasva, majd ellentétesre váltva.

Kattintson a 4-es cellára. Itt az F1 pergésmentesítésével foglalkozunk. (A pergés ("prell") a mechanikus átkapcsolásnak többszöri rövid megszakadás után történő létrejöttét jelenti.)

Készítsünk prellmentesítést a C-Control fólia billentyűzetére.

Vegyen elő a programvezérlés csoportból egy szünet, "Pause" blokkot, tolja egy kicsit jobbra. Tőle balra helyezzen el egy konstanst (konstanter Wert) a "Konstanten" csoportból. Kösse össze a két blokkot. Kattintson a jobb egérgombbal a konstans blokkra. A tulajdonságoknál (Eigenschaftsdialog) 5-öt adjon be. (15. ábra.)

A "Pause" blokk az 5-ös értéket fogja tárolni. Az utasítás végrehajtásakor a program $5 \times 20 \text{ms} = 100 \text{ms}$ szünetet iktat be. Ez alatt a gomb nem lesz lekérdezve, a pergést elkerüljük.

5-ös cella: hogy az F1 nyomása alatt a relé ne legyen többször átkapcsolva, a programnak meg kell várnia a gomb elengedését. Ehhez tegyen a programvezérlés csoport 5-ös cellájának munkafelületére egy értékre várás (Warten auf wert) blokkot, és tolja kissé jobbra. Helyezze baloldalt mellé az F1 gomb blokkot, közéjük egy invertert, és kösse őket össze (16. ábra).

A várakozás blokk addig késlelteti a program végrehajtást, míg a bemenetére betáplált érték nem nulla. Mivel ide az F1 állapotának invertáltja (fordítottja), kerül, a várakozás addig tart, amíg a gomb le van nyomva.

A 6-os cella lezárja az F1 gomb megnyomásának kezelését. Ehhez mindössze egy a főprogramba visszatérő (Rückkehr aus unterprogramm) blokkot kell betenni, a program vezérlés csoportból. A 6-os cella után a program az alprogram hívás után, vagyis a 2-es cellában folytatódik.

Ne felejtse el az ugrási célt sem megadni. Kattintson az 1-es cellára a projekt explorer-ben, nyissa meg az elágazás blokk jellemzők dialógust (jobb egérgomb a blokkra, menüből tulajdonságok választás), és adja meg ugrási célnak az átkapcsolást (Umschalten, a korábbi 3-as cella új neve).

Ezzel a példaprogram kész. Mentse el ("Datei" menü/"Speichern" pont), pl. TEST1.CPF fájlneven.

Kompilálás, szimulálás

Következhet a projekt lefordítása, az "Entwicklung/Compilieren" menü paranccsal vagy a "Compilieren" kapcsolóra való kattintással.

A felhasználói felületen ablak tájékoztat a fordítás lefutásáról, az esetleges hibákról és a végén a kódbájtok számáról. (17. ábra.) A fordítás eredménye nem látható, az a PC munkamemóriába kerül, és át lehet küldeni a modulra, vagy szimulációra használni. Ha megnézi a kódbájtok számát, fogalmat alkothat, hogy milyen bonyolultságú program fog elférni az összesen rendelkezésre álló 8000 bájton.

A példa program működését tesztelje le a szimulátorral. Az Ansicht (Nézet) menüből nyissa meg a "Digitalports/Relais" és a "Tasten/LEDs" pontokat. (18, 19 ábra.)

Indítsa a szimulátort a "Simulator/Ausführen" menüparanccsal vagy a megfelelő eszköztár kapcsolóval. Figyeljen a "Simulation lauff" (szimuláció fut) kiírásra a státusz sorban.

Kattintson az egérrel többször a "Tasten/LEDs" ablakban az F1-mezőre. A C-Control szimulátor ablakai grafikusan szemléltetik a perifériák működését. Ha a program rendben van, a szimulátort leállítjuk, majd visszaállítjuk. Mindkét parancs a "Simulator" menüben van.

A szimulátorral hardver nélkül, és így kisebb kockázattal tesztelhetünk programokat, és a hibákat néhány egérekattintással kijavíthatjuk. A C-Control egységbe csak a kész programot kell átküldeni.

Program átvitele és indítása

- Győződjön meg, hogy az állomás megfelelő módon össze van kötve a tápfeszültséggel és soros interfészen keresztül a PC-vel.
- Nyomja meg az állomáson a Reset gombot.
- Állítsa be az alkalmazott interfészt az Optionen / Umgebung / Simulator und Lader (Opciók, környezet, szimulátor és betöltő) pontokon keresztül.
- Válassza ki az "Entwicklung / In C-Control Station übertragen" (átvitel) menü parancsot.

A képernyőn grafikusan követhető a program átvitel. Az állomás RS232 kapcsain ez alatt világít a TX világítódioda.

Ezután az állomás be van programozva.

- A Start gombbal indítsa el a kis tesztprogramot.

Láthatja, hogy a modul F1 gombjának minden megnyomására a K1 relé be/ki kapcsol. A kapcsolási állapotot a relé kapocs jelződiodája mutatja.

A tesztprogram valamint a CD mintaprogramok alapján fokozatosan áttérhet bonyolultabb munkákra.

Programozás CCBASIC-ben

Áttekintés

Bonyolultabb feladatokhoz vagy tapasztaltabb felhasználók számára lehetőség van BASIC programozásra (CCBASIC). A CD-n két verzió található. A fő verzió számára legalább Win95 vagy NT stb. szükséges, Win 3xx nem elég.

A CD-n van csökkentett kiépítésű (szimulátor nélküli) verzió is, amely MS-DOS alatt is fut, így használható régebbi PC-ken és notebook-okon, egészen 8086/88-ig (XT). A régi notebook-ok kiválóak az állomás helyszínen történő átprogramozására, a kapcsolószekrényből való kiszereles nélkül.

Installáláshoz kövesse a CD utasításait.

A CCBASIC alapjai

Általános

A Basic ún. forrásszöveg sorokból áll, melyek egy vagy több számolási vagy vezérlési utasítást tartalmaznak.

A CCBASIC a C-Control számára kidolgozott Basic programvariáns. Szintaxisa nagyjából megfelel a standard nyelvének, de néhány parancs kifejezetten az itteni hardverhez illeszkedik.

A végrehajtás a sorok egymás utáni sorrendje szerint történik. Minden programsor egy vagy több utasítást tartalmaz, melyek kettősponttal vannak elválasztva. A régebbi Basic-től eltérően sorszámozásra nincs szükség. Ha mégis adunk sorszámot, ezt ugró utasításhoz lehet használni. A sorszámok nincsenek hatással a végrehajtás sorrendjére, az a sorok "fizikai" sorrendje szerint történik.

A megértés könnyítésére kommentet (megjegyzés) lehet a programba írni. A CCBASIC-ben ezt aposztroffal (') kell kezdeni, ami jelzi, hogy a sor további része nem tartozik a programhoz.

Azonosítók

Betűkből és számokból állhatnak, és objektumokat, úgymint változókat és konstansokat, jelölhetnek. A címke nevek és az ún. "fenntartott (rezervált) szavak" is azonosítók.

A nagy- és kisbetűk között nincsen különbség. Az azonosító betűvel vagy aláhúzással kezdődhet, és szóközt nem tartalmazhat.

Változók és konstansok

Ezek az információ feldolgozás tárgyai. A CCBASIC-ben mindkettő numerikus érték, csak a konstans a program folyamán nem változik.

A konstansokat a CCBASIC-ben decimális, hexadecimális és bináris alakban lehet megadni. Példa hexadecimális és bináris szintaxisra, 46-os szám: &H2E ill. &B101110.

A változókra azonosítójukkal lehet hivatkozni, melyet az első használat előtt DEFINE sorral meg kell adni.

Címkék

A program adott pontjainak megjelölésére valók. Algoritmusban az ugró utasítás célpontját jelölik. A CCBASIC-ben a sor elején állnak, és kettős kereszttel kezdődnek, amit szóköz nélkül követ a címke ("label") neve. Példa címkére és ugrásra:

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

Kifejezések

Ezek vagy közvetlenül adhatók meg (változóként vagy konstansként), vagy egy meghatározott érték kiszámításával. Az utasítások részét képezik, és például amikor változónak értéket adunk, az egyenlőségjeltől jobbra állnak. Operandusok és operátorok kombinációjával állíthatók elő. Példák:

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

Operandusok és operátorok

Az operandus alapvetően konstans, változó vagy függvény hívás, de lehet maga is operandusok és operátorok segítségével összeállított kifejezés.

Az operátorok számítási műveleteket jelentenek, melyeket a környező operandusokon kell végrehajtani. Meghatározott elsőbbségi sorrendjük van (precedencia), ld. később.

Függvények

Meghatározott műveletet, pl. számítás hajtanak végre, és eredményül egy számot adnak. Többnyire egy vagy több argumentumuk (függő változó) van, melyek kerek zárójelek között szerepelnek, és vesszővel vannak elválasztva. Az argumentum nélküli függvénynél nincs kerek zárójel. Példák:

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

A CCBASIC-ben minden támogatott függvény előre definiált, azonosítójuk a fenntartott szavak közé tartozik.

Értékkadás

Ez a legegyszerűbb utasításforma. Azon változó azonosítója után, melynek értéket akarunk adni, hozzárendelő egyenlőségjelet kell tenni, majd az értéket előállító kifejezés következik, vagyis egy egyszerű matematikai képlet.

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

Parancsok

Az egyszerű értékkadások mellett parancs hozzárendelések is vannak a CCBASIC-ben. Ezek fenntartott szóval kezdődnek. Egyesekhez egy vagy több paraméter is szükséges. A paramétereiket a parancs azonosító és egy szóköz után lehet megadni, és vesszővel kell elválasztani. (Kivéve a Print [nyomtatás], ld. később.) Kerek zárójel nem kell hozzájuk!

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

A programfutást vezérlő utasítások

Ezek a végrehajtás sorrendjét befolyásolják, az információ feldolgozó folyamat bemenő értékeitől függően. Nagy rugalmasságot biztosítanak az algoritmus készítésénél.

Egy vagy több fenntartott szóból állnak, és egyéb adatokat is igényelhetnek.

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

```
FOR i=0 to 10 STEP 2
```

```
...  
NEXT
```

Utasítások a fordítóprogramhoz

A CCBASIC forrásszöveg a programutasítások mellett compiler (fordítási) utasításokat is tartalmazhat, pl. adatblokkok (táblázat) készítésére, változók vagy konstansok definiálására.

Egy sorban csak egy ilyen utasítás lehet, nem lehet kettősponttal elválasztva többet megadni.

A DEFINE utasítás

Ez egy compiler utasítás.

Szimbolikus konstansok definiálása

Áttekinthetőbbé teszi a programot, ha számokra való hivatkozások helyett jellemző neveket adunk, pl. "if x <1234" helyett a névre hivatkozunk. Még jobb a konstansokat globálisként definiálni.

Szimbolikus konstans definiálás: **DEFINE** azonosító érték
Az érték decimális, hexadecimális vagy bináris szám lehet.

A fenti példában:

```
DEFINE limit 1234
```

```
...  
if x < limit then goto alarm
```

Változók definiálása

A CControl 24 bájtot biztosít RAM-jában a felhasználói program változó számára. Bitenként vagy 16 bites egészként (szó) is használhatók.

A standard Basic-től eltérően a CCBASIC-ben a program minden változóját definiálni kell az első használatbavételük előtt. Meg kell adni az adattípust (bit, bájtt vagy szó (word)), valamint megadható a tárcella száma (bitnél kötelező megadni). A felhasználónak kell figyelni arra, hogy ne legyen átfedés a foglalások között, különben felülírások fordulhatnak elő. Például a bit[18], byte[2], word[1] egyidejű megadásával a 2-es cella egy részét közösen használhatunk.

Bit változó megadás:

```
DEFINE azonosító BIT[szám]
```

A szám 1...192 lehet (24 bájtt egyenként 8 bittel).

Bájt változó megadás cellaszámmal:

DEFINE azonosító **BYTE**[szám]

A szám 1...24 lehet (24 bájt).

Egész változó megadás cellaszámmal:

DEFINE azonosító **WORD**[szám]

A szám 1...12 lehet (1 szó = 2 bájt).

Ha a bájt- és szó definiálásnál a cella (szám) megadás elmarad, a compiler végzi el a tár felosztását. Lehetőleg ne váltogassuk a bájt és szó megadásokat, mert ez rossz tárhely kihasználáshoz vezethet. További megadásoknál ismét ügyelni kell az átfedés elkerülésére.

A compiler az 1-essel kezdi a hozzárendelést. Egy már korábban definiált változó azonosítót nem szabad másodszor is definiálni.

Digitális portok definiálása

A CCBASIC-ben a portokra változóként kell hivatkozni. A portokat is előre definiálni kell.

Egyes port definiálása:

DEFINE azonosító **PORT**[szám]

A szám 1...16 lehet.

A már ismertetett port kiosztás miatt többnyire a következő formájú definíciós blokkok adódnak:

DEFINE P1 **PORT**[1]

DEFINE P2 **PORT**[2]

...

DEFINE P6 **PORT**[6]

...

DEFINE K1 **PORT**[7]

DEFINE K2 **PORT**[8]

...

DEFINE F1 **PORT**[9]

...

DEFINE F4 **PORT**[12]

...

DEFINE LED1 **PORT**[13]

...

DEFINE LED4 **PORT**[16]

Analóg portok definiálása

DEFINE azonosító **AD**[szám]

A szám 1...8 lehet.

DEFINE A1 **AD**[1]

...

DEFINE A4 **AD**[4]

...

DEFINE T1 **AD**[5]

DEFINE T2 **AD**[6]

...

DEFINE U1 **AD**[7]

DEFINE U2 **AD**[8]

Az utasítások áttekintése

Matematikai és logikai operátorok

alapl műveletek: + - * /

modulo operátor: egész osztás maradékát adja

Pl. a = 10 **MOD** 3

eredménye 1.

összehasonlítások: >, <, >=, <=, =, <> (nem egyenlő)

Az összehasonlítás eredménye -1 (nem 1! igaz) vagy 0 (nem igaz).

Pl. a = 10 < 3

eredménye 0.

logikai operátorok: NOT, AND, NAND, OR, NOR, XOR

(negálás, és kapcsolat, és kapcs. + negálás, vagy kapcsolat, vagy kapcsolat negálással, kizáró vagy).

A logikai operátorokat feltételek megadására valamint bináris bájt- és szómanipulációra lehet használni.

eltolási operátorok: SHL, SHR

Bit-minták bitenkénti balra- vagy jobbra tolására szolgálnak, bájt- vagy szó változóknál. Az operátortól balra áll az eltolandó érték, jobboldalt a szám, ahány bittel el kell tolni. A balra tolás 2-vel való szorzás, a jobbra tolás osztás.

Pl. a = 10 **shl** 3

a = 10 * 2 * 2 * 2,

az eredmény 80.

Ügyeljen arra, hogy SHR aritmetikailag tol, vagyis előjellel! Pl.

a = &B1000000000000000 SHR 1 eredménye (értékek binárisan)

&B1100000000000000, és nem &B0100000000000000!

Matematikai függvények és utasítások

Az x és y argumentumok mindig kifejezések (ld. fentebb).

SQR(x): x négyzetgyökének közelítése; a tizedesek le lesznek vágva.

SGN(x): 1, ha x pozitív, -1, ha x negatív, 0, ha x=0.

MAX(x,y): =x, ha x nagyobb, egyébként y.

MIN(x,y): =x, ha x kisebb, egyébként y.

RANDOMIZE x: x-szel kezdve belső pszeudo véletlen generátort indít. Ugyanazon számtól indítva ugyanazt a számsort kapjuk. A **RANDOMIZE TIMER** speciális forma a szabadonfutó időzítő (timer) aktuális értékét tölti be indításul a generátorba.

A **RAND** véletlen függvény a pszeudo véletlen generátor következő egész értékét szolgáltatja. A véletlen számok a multiplikatív eljárás és az azt követő modulo osztás szerint képződnek a mindenkorai megelőző értékből (ld. egy jobb matematikai szakkönyvben).

Operátorok és függvények precedenciája (rangsorrendje)

A kifejezések a hagyományos műveleti rangsorrendben lesznek számítva, egyenrangú esetben pedig balról jobbra haladva.

A sorrendet zárójellezéssel lehet módosítani. A CCBASIC-ben a zárójellezési mélység legfeljebb 6.

Az áttekinthetőség érdekében javasoljuk a sok zárójel helyett inkább több programsor használatát.

Precedencia a CCBASIC -ben

Rang	Operátor
9	()
8	függvényhívás
7	negatív előjel
6	* / MOD SHL SHR
5	+ -
4	> >= < <= = <>
3	NOT
2	AND NAND
1	OR NOR XOR

A program futását vezérlő utasítások

Ciklus

FOR változó = kezdet **TO** vég **STEP** lépésmagyság

...

NEXT

A FOR-ciklus addig hajtja végre a NEXT-ig szereplő utasításokat, míg a változó el nem éri a vége kifejezés értékét. Az első futtatás előtt a kezdet kifejezés értéke ki lesz számítva, és hozzárendelve a ciklusváltozóhoz. A ciklusváltozó minden lefutáskor a lépés értékével nő.

A

FOR változó = kezdet **TO** vég

...

NEXT

formánál a lépésmagasság 1.

A zárókifejezés és a lépésmagasság kifejezés minden lefutáskor újra számíttódik.

A FOR ciklusok egymásba ágyazhatók, ennek mélységét csak a ciklusváltó számára rendelkezésre álló memória korlátozza.

A programfutás során minden FOR ciklus csak saját NEXT-utasításával futhat (pl. nem lehet ugrással más NEXT-hez küldeni). Egy ilyen program ugyan lefordul és betöltődik, de az eredmény kétséges.

Ügyelni kell a ciklusváltónál és a zárókifejezésnél megengedett értéktartományra is (255), mivel ennek túllépésekor 0-ról indulhat újra, és végtelen ciklus állhat elő.

·Feltételes végrehajtás

IF feltétel **THEN** utasítás 1

vagy

IF feltétel **THEN** utasítás 1 **ELSE** utasítás 2

Az IF-THEN-ELSE szerkezet lehetővé teszi a programvégrehajtás illesztését a futás során előálló feltételekhez. Az első esetben a feltétel egy tetszőleges kifejezés; ha ennek értéke nem 0, a feltétel teljesültnek van tekintve, és az 1-es utasítás lesz végrehajtva. Ha ELSE-el egy második utasítást is hozzáfűzünk, akkor a feltétel kifejezés 0 értéke esetén a második utasítás lesz végrehajtva.

Az IF-THEN-ELSE szerkezet egyetlen forrásorban kell álljon. THEN és ELSE után nem következhet utasításblokk (több utasítás).

·Ugró utasítás

GOTO címke

A program tetszőleges, címkével jelölt helyétől lehet folytatni a futást. A cél a GOTO előtt és után egyaránt lehet.

·Szubrutin hívás és visszatérés

GOSUB címke

Itt a címke a szubrutin kezdőpontja.

Az ún. szubrutinokban programrészek vannak összefogva, melyeket a programfutás során többször ugyanúgy kell végrehajtani. Címkével kezdődik, majd az utasítások sorozata következik, végül a lezárás (RETURN).

A RETURN után a program a GOSUB után következő utasítással folytatódik. Minden RETURN-höz egy megelőző GOSUB-nak kell tartozni.

Szubrutin hívásnál az egymásba ágyazás megengedett legnagyobb mélysége 4.

RETURN segítségével számértéket is vissza lehet adni a behívónak, így visszatérő vagy kiterjedtebb számításokat vagy státusz meghatározásokat be lehet skatulyázni egy szubfunkcióba. (példa ld. eredeti útm. 47. o. közepe)

·Értékvezérelt programelágazás

ON változó **GOTO** címke0, címke1, ...címken

ON változó **GOSUB** címke0, címke1, ...címken

A szelektáló változó értékétől függően programelágazás vagy szubrutin hívás következhet 0 értéknél a 0-ás címkéhez stb. ugorva. Ha a változó értéke negatív, vagy nagyobb a felsorolt ugrási célpontoknál, nem történik programelágazás.

·Program vége

END

Ha a komputer a végrehajtás során egy END utasításhoz ér, a program végrehajtás befejeződik, a rendszer inaktív állapotba kerül. Ekkor újabb felhasználói program vihető át, vagy a Start gombbal újra lehet indítani a programot.

·Várakozási utasítás

WAIT feltételkifejezés

Megszakítja a futást, amíg a feltétel kiszámítása nem adja ki a 0-tól különböző eredményt.

Pl. define F1 port[9]

...

WAIT key

Itt addig tart a várakozás, míg a 9-es digitális portról (F1 gomb az állomáson) logikai 1-est nem olvas.

A PAUSE parancs meghatározott időre állítja le a programot. A paraméter kifejezéssel 20ms-ot megszorozva kapjuk a várakozási időt, pl. a PAUSE 50 egy másodpercet jelent. A legnagyobb eltérés a valóságos szünetidő és a megadott érték között eszerint +/-20ms lehet.

Kommunikáció a soros interfészen keresztül

·Adat kivitel

Az adatok a CControl vezérlő soros interfészén át szöveges formában mennek ki. Ha ide interfész kábelen keresztül PC-t kötünk, amelynek terminál programja van, a kiadott adatok megjeleníthetők. A

PRINT kifejezés

hatására átvitelre kerül a kifejezés kiszámított értéke;

PRINT "szöveg"

hatására megjelenik az idézőjelek közötti szöveg.

Mindkét esetben soremelés jel adódik az átvitelhez, melynek hatására a terminál program új sorba írja a képernyőn a következő kimenő adatot. A soremelés úgy iktatható ki, hogy a Print parancsot követő kifejezés vagy szöveg végére pontosvesszőt teszünk, pl.: **PRINT** "szöveg";

A CCBASIC több adat kivitelét is támogatja egy Print parancs alatt. Az egyes paramétereket vesszővel vagy pontosvesszővel kell elválasztani. A vessző hatására tabulátor kerül a kimenő adatba, ami a képernyőn hosszabb szóközöként jelenik meg. Ha két kimenő adatnak köz nélkül kell egymást követni, akkor a Print parancsban pontosvesszővel kell őket elválasztani:

PRINT "a= ", a

PRINT "a= "; a

Egyetlen Print parancs paraméter nélkül soremelést eredményez.

PRINT

·Adat bevitel

Az **INPUT** változó

paranccsal egész értéket tud a soros interfész beolvasni, és további feldolgozásra eltárolni.

Az értéket terminál programmal lehet PC-be vinni, és innen az enter megnyomása után interfész kábelen át kerül a CControl-ba.

Az Input parancs addig vár, amíg a terminál nem vett egy komplett adatátvitelt. Ha az Input parancsot rákövetkező adatátvitel nélkül hívjuk, a program végtelen időre ebben az állapotban marad, és csak resettel valamint újraindítással lehet ismét használni!

·Bájtsoros kommunikáció a soros interfészen

A Print és Input rövid karaktorsorozatokat továbbít ill. fogad. Kívánatos lehet egyes bájtokat sorosan átvinni. Erre szolgál a CCBASIC PUT és GET parancsa. A

PUT kifejezés

a kifejezés kiszámított értékét küldi el. Szükség esetén a 0...255 bájttartományra redukálja. A

GET változó

sorosan fogadott bájtra vár, és az értéket a megadott változóba tárolja.

További interfész parancsok és funkciók

Az Input és Get parancsok adott esetben vég nélkül várnak a soros adatra. Ha el akarjuk kerülni a program "kiakadását", minden Input és Get előtt meg lehet vizsgálni az RXD státusz függvény behívásával, hogy van-e továbbítandó adat. Ha van, a függvény értéke 1. Ha az interfész puffer üres, az eredmény 0.

```
...  
If RXD then GET thebyte  
...
```

A soros interfész beállított adó- és vevő átviteli sebessége 9600 baud. Ezt a BAUD paranccsal meg lehet változtatni, a program által rögzített (1200, 2400, 4800, 9600) vagy egyéb értékre: BAUD R2400

hatására az adó és vevő 2400 bit/s-ra áll. Az adó és vevő sebessége nem kell azonos legyen. A soros interfész átviteli sebessége a CControl mikroprocesszor belső ütemadójából van származtatva. A Baud parancsnak átadandó bájt érték tartalmazza a szükséges Nxx osztási értéket.

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

A 7-es és 6-os bit az adó és vevő számára közös NP előosztó. Az NP értéke 1, 3, 4 és 13 lehet. NP1 és NP0 ehhez szükséges beállításai:

Előosztó	NP1	NP0
1	0	0
3	0	1
4	1	0
13	1	1

A 3 és 5 NT bitek valamint a 0...2 NR bitek további osztó értékeket határoznak meg, külön az adóra (NT) ill. a vevőre (NR), a következő kódolás szerint:

Osztó	NT2 ill. NR2	NT1 ill. NR1	NT2 ill. 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

Az adó átviteli sebességének számítása: $125000/(NP * NT)$
A vevőé: $125000/(NP * NR)$
[T=transmitter, R=receiver]

A további interfész paraméterek - 8 adatbit; paritásbit nincs; 1 stop bit - rögzítettek, nem változtathatók.

Fájl használat

Mérési eredményeket vagy egyéb adatokat rögzíthetünk, melyeket tápfeszültség kimaradás után vissza kell tölteni mint programváltozót. Erre a célra az EEPROM chipnek a felhasználói program után következő - többnyire nagyobbik - része áll rendelkezésre. A memória tartomány fájlként lesz kezelve, mely a megfelelő attribútummal való megnyitással írás és olvasás céljára hozzáférhető.

```
OPEN# FOR WRITE  
OPEN# FOR APPEND  
OPEN# FOR READ
```

(megnyitás (felül)íráshoz, hozzáíráshoz, olvasáshoz).
Csak egész értékeket lehet beírni vagy kiolvasni. Mindegyik érték két bájtot foglal az EEPROM-ban.
Írás ill. olvasás parancsok:

```
PRINT# kifejezés  
(kifejezés eredményének tárolása)
```

ill.
INPUT# változó
(a változó a program definiált egész változója)

A fájl beírása és kiolvasása szigorúan sorban történik. Ehhez egy belső fájlmutató értéke minden hozzáfordulás után 1-gyel meg lesz növelve.

Beírás előtt mindig ellenőrizni kell, van-e elég hely a memóriában, a FILEFREE függvénnyel. Ez a szabad tár mennyiségét adja eredményül (szavakban). Példát ld. eredeti útm. 53. o.

Olvásás előtt ellenőrizni kell, hogy van-e még rögzített adat, az EOF (end of file) függvénnyel. Ennek eredménye 1, ha nincs a fájlban több hozzáférhető adat, egyébként 0. A lekérdezésnek ugyanúgy keretbe kell foglalni az adatblokk kiolvasást, mint az előbbi példánál (53.o.)

A fájlkezelés befejezése után a fájlt az adatok védelme miatt azonnal be kell zárni (CLOSE# parancs, paraméter nélküli).

Portkezelő parancsok

Átkapcsolás parancs (TOG)

Elvileg a komputer portjait változóként lehet hozzáférni. Digitális port be- ill. kikapcsolása:

```
P = 1  
ill.  
P = 0
```

A be- ill. kikapcsolás váltogatása:

```
P = NOT P  
vagy  
TOG P
```

A TOG (angol toggle-ból (váltogat)) kevesebb memóriát igényel, és gyorsabb.

TOG-nál a P port változó csak egyes digitális portot jelenthet, nem bájt- vagy szó portot.

Port deaktiválás (DEACT)

Amint egy port változóhoz első ízben értéket rendelünk, a komputer a processzorban a hozzátartozó hardver struktúrát (tranzisztorok) kimenet állapotba kapcsolja, tehát a csatlakoztatott áramkörnek megfelelően folyik az áram ki vagy be a processzornál (max. 10mA!). A

DEACT portvar parancs deaktiválja az adott portot, vagyis nagyimpedanciás állapotba teszi, így az bemenetként működhet.

A PULSE parancs

PULSE portvar parancs hatására a kijelölt porton néhány ms-os pulzus megy ki. Ez hasznos lehet pl. külső élvezérelt logikai áramkörökhöz. Ha a parancsot megelőzően a port alacsony (LOW, 0) szinten van, High-pulzus (0-1-0) jön létre, egyébként low pulzus (1-0-1).

Adattáblázatok definiálása és használata

A standard Basic-ben a DATA sorok szolgálnak konstans adatblokkok készítésére, ahol az adatokhoz sorosan hozzá lehet férni. A CCBASIC nem támogatja a DATA sorokat, hanem más rugalmas lehetőséget kínál az adatblokk hozzáféréshez. A konstans adatokat táblázat formájában lehet eltenni. A táblázatok nevet kapnak, és annyi adat tehető be, amennyit csak enged a memória. A bevitel (Cx) egészként lesznek letéve, és két bájtot foglalnak. Az adatok közvetlenül megjeleníthetők a forrásszövegben

```
TABLE tablename C0 C1...
```

```
...Cn
```

TABEND

vagy pedig a CCBASIC compilerrel importálható külső szövegfájlból

```
TABLE tablename "tabfilename"
```

Az adattáblázat definícióknak a program végén kell lenni, az END parancs után, mivel az adatok közvetlenül az EEPROM-ban levő kódbájtok után következnek, és különben esetleg parancsoknak is lehetne őket értelmezni.

Adattáblázathoz való hozzáférés: LOOKTAB parancs.

LOOKTAB táblázatnév, index, változó

A név egy érvényes táblázat neve, az index tetszőleges kifejezés lehet, a változó pedig a memória cella, ahova az eredménynek kerülnie kell. Az index értéke nem lehet negatív, és max. N-1-ig terjedhet, ha a táblázatnak N eleme van. Ha az index 0, C0 kerül a változóba, stb. (példa ld. 56. o.; Bildschirm=képernyő)

A táblázatok különösen jól használhatók az A/D értékeknek fizikai mennyiségekké való alakításában. Az átváltó táblázatnak rendszeren 256 eleme van. A mért A/D érték táblázat indexként szerepel a fizikai mennyiség meghatározásában.

Hozzáférés a valósidejű órához

A belső óra olvasásához és állításához szolgáló parancsok:

YEAR	év (0...99)
MONTH	hó (1...12)
DAY	a hó napja (1...31)
DOW	a hét napja (0=vas., ...6=szo)
HOURL	óra (0...23)
MINUTE	perc (0...59)
SECOND	másodperc (0...59)

Figyelni kell arra, hogy a hozzáférés során az óra tovább fut, ezért először a másodpercet kell kiolvasni. Ha 59-en áll, a teljes kiolvasás (mikor már az év is megvan) után újra ellenőrizni kell, nem áll-e 0-n. Ebben az esetben a kiolvasást meg kell ismétlni, mivel éppen egy perc váltásnál voltunk. (Ez tovább is gyűrűzhet, szélsőséges eset a Szilveszter-nap.)

A CControl az évet csak két jeggyel tárolja, az 1999-2000 átmenet 99-0 átmenetet jelent.

Timer

A belső 20ms-os időzítőt az előre definiált **TIMER** azonosítóval lehet kiolvasni. Szabadonfutó, módosítani, visszaállítani nem lehet.

Egyéb parancsok

·Hangjelzés utasítás (BEEP)

A készülék piezo jeladójának megszólaltatása:

BEEP hang, thang, tszünet

A paraméterek konstansok vagy kifejezések lehetnek. A hang magasságát a

hang=250000/frekv.(Hz)

képlet adja, thang a hangjelzés időtartama, tszünet a jelzés után következő szünet. Ha a jelzés hosszúra adunk meg 0-t, folyamatos hangjelzést kapunk. A hanggenerátor bekapcsol, és folytatódik a Basic program feldolgozása. Ha a hang érték 0, a generátor kikapcsol.

·Frekvenciamérés (FREQ)

A mérés 1s kapuidővel történő pulzusszámláláson alapul, és a háttérben zajlik, a Basic program futásával párhuzamosan. Lekérdezés: FREQ2 parancssal.

x = **FREQ2**

A mérési tartomány kb. 30kHz-ig terjed, 1% alatti hibával.

Ha a DCF-77 bemenetre nincsen aktív antenna kötve, alternatívaként frekvencia mérésére használhatjuk. Az eredmény lekérdezésére szolgál a FREQ parancs.

A mérési tartomány kb. 5kHz-ig terjed, 1% alatti hibával. Ez után az eredmény egyre pontatlanabb lesz.

x = **FREQ**

·Áramtakarékos üzemmód (SLOWMODE)

Kisebb számítási igényű alkalmazásoknál a SLOWMODE ON

parancssal a mikroprocesszor belső órája lelassítható (1/16). Ha a program során valahol mégis nagyobb sebesség kell, a SLOWMODE OFF

parancssal visszatérhetünk a kiindulási helyzethez.

Figyelem: soros adatátvitelt alkalmazó programoknál nem szabad a SLOWMODE-t aktiválni, mivel az adatátviteli sebesség a processzor órával együtt változik.

A lassítás megnöveli az áramkimaradásakor akkuval áthidalható üzemidőt, mindazonáltal ennek akkor van értelme, ha a LED-ek és relék nincsenek bekapcsolva.

Példa CCBASIC-ben való programozásra

A CD-n további példák is találhatóak a CCBASIC megismerésére.

Tegyük fel, hogy a modul F1 gombjával kívánja a K1 relét be/ki kapcsolgatni, és a gombnyomásokat hangjelzéssel nyugtázni, valamint ezzel párhuzamosan percenként be/ki kapcsolni kell a K2 relét.

Indítsa el Windows alatt a CCBASIC-ot (CCEW32D.EXE). nyisson egy új fájlt (Datei/neu), és mentse el pl. "TEST1.BAS" néven.

A forrásszöveg

A szöveg megírásához először definiálni kell az alkalmazott periféria egységeket:

```
define K1 port[7]
```

```
define K2 port[8]
```

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

Definiálunk még egy további változót, melynek használata később lesz megmagyarázva.

```
define oldsec byte
```

Egy program többnyire a következő részekből áll: inicializálás, főhurok és alfunkciók. Itt pl. inicializálásul kapcsolni kell a K1 és K2 relét:

```
K1=ON
```

```
K2=OFF
```

Példánkban legyen a főhurok a "Loop" címkével ellátva. A hurok tartalma: F1 gomb lekérdezése, K2 relé percenkénti működtetése. Új perc kezdetekor a másodperc érték 1 másodpercig nulla. Hogy a relé ezalatt ne kapcsolgasson, a perc kezdet megállapítás a másodperc váltáshoz is lesz kötve.

```
#loop
```

```
if not F1 then gosub toggle K1
```

```
if (0=second) and (second <> oldsec) then tog K2
```

```
oldsec = second
```

```
goto loop
```

Ügyeljen a gomb értékek negálására. A gombok hardver oldalról 5V-ra felhúzó ellenállással vannak ellátva, vagyis megnyomott állapotban Low (alacsony), kiengedett állapotban High szintet adnak. (A grafikus programozásnál a gomb lekérdezés ezt a negációt már magában foglalja.)

F1 megnyomott állapotának észlelésekor a program a toggleK1 alfunkcióra ágazik, amely átkapcsolja a K1 relét. Pergementesítésre és a gomb reteszelésére 100 (5x20) ms-es várakozás van, majd az elengedést kell figyelni. Ezután visszatérünk a főprogramba.

```
#toggleK1
```

```
tog K1
```

```
pause 5
```

```
wait F1
```

```
return
```

Kompilálás, szimulálás

A forrásszöveg bevitele után le kell fordítani a programot (Entwicklung/compilieren). Egy üzenő ablakban (Meldungen) követhető a fordítás folyamata, esetleg megjelenik a hibák listája. (fehlerfrei=hibátlan). (20. ábra.)

A program szimuláláshoz vissza kell kapcsolni a forrásszöveg ablakhoz, az üzenő ablak bezárásával vagy "Ctrl+Tab"-al.

Nyissa meg a "C-Control Digitalports" szimulátor ablakot a nézet (Ansicht) menüből. A grafikus módtól eltérően a BASIC program nem ad közvetlen képet a hardverről. A "C-Control Digitalports" ablakban ábrázolódik a 16 digitális port. A P1...6 kapcsokhoz kötődő 1...6 portok és az F1...4 fólia billentyűkhöz kapcsolódó 9...12 portok felhúzó ellenállásokkal vannak ellátva. Kapcsoljon a szimulátor ablakban először ezeket a portokra, miközben a shift gombot nyomva tartva az 1...6 és 9...12 LED-ekre kattint, úgy hogy ezek világos zöldek lesznek. (21. ábra.)

Kattintson a jobb egérgombbal az ablakra, és a megjelenő menüből válassza a "Als Anfangsstellung nach Reset Verwenden" (Reset utáni kezdeti beállításként alkalmazni) pontot. A szimulátor ablak most a C-Control hardver feltételeinek felel meg. Ezt a beállítást csak egyszer kell végigcsinálni, a beállítások a program végéig megőrződnek.

(ld. 22. ábra):

alle eingange einschalten=minden bemenetet bekapcsolni
 alle eingange ausschalten=minden bemenetet kikapcsolni
 Indítsa el a szimulátort. Aktiválja az editor ablakot. Válassza a "Simulator" menü "Ausführen" (futtatás) parancsát.

A 7 és 8 LED-nek piros kerete lesz, a 7. LED világospiros. Ez azt jelenti, hogy mindkét port kimenetként lesz használva, a 7-es port be van kapcsolva. A 7 és 8-as portok a K1 ill. K2 relével vannak kapcsolatban. Kattintson a 9-es LED-re, amely az F1-es főbillentyűnek felel meg. A tervnek megfelelően a K1 relé állapota váltogatható.

Nyissa meg az "Ansicht" menüből az "Überwachte Variablen" (figyelt változók) ablakot. Nyomja meg az Insert ("Einf", beillesztés) gombot, és írja be az ablakba: second (másodperc megadás).

(23. ábra):

Geben sie den Bezeichner ein = megnevezés beadás

"OK" után a következő jelenik meg (ld. 24. ábra):

változó: másodperc, érték (wert) 11.

Az "Überwachte Variablen" ablakban a rendszer- és saját változókat lehet követni, ebben az esetben a valósidejű órát. Hagyja egy ideig futni a szimulátort, és figyelje a másodperceknek 59-ről 0-ra történő váltását, vagyis a perc kezdetét, és a 8-as LED átváltását, ami a K2 relé állapotát mutatja. Ezzel a demo program második részét is megtekintette. Állítsa le a szimulátort, majd állítsa vissza. A szükséges parancsok a szimulátor menüben vannak.

Program átvitele és indítása

- Győződjön meg, hogy az állomás megfelelő módon össze van kötve a tápfeszültséggel és soros interfészen keresztül a PC-vel.
- Nyomja meg az állomáson a Reset gombot.
- Állítsa be az alkalmazott interfészt az Optionen / Umgebung / Simulator und Lader (Opciók, környezet, szimulátor és töltő) pontokon keresztül.
- Aktiválja a "test1.bas" editor ablakot.
- Válassza ki az "Entwicklung / In C-Control Unit übertragen" menü parancsot.

Az állomás RS232 kapcsain ég a TX világítódióda. A képernyő kiírja az átvitel sikeres befejezését vagy az esetleges hibaüzeneteket. Hiba esetén vizsgálja meg az összekötő vezetékét, a soros interfész beállítást, valamint a tápfeszültség meglétét. Újabb átvitel megkísérlése előtt ismét aktiválni kell az editor ablakot.

Sikeres átvitel után az állomás be van programozva.

- A Start gombbal indítsa el a kis tesztprogramot.

Láthatja, hogy a modul F1 gombjának minden megnyomására a K1 relé be/ki kapcsol. A kapcsolási állapotot a relé kapcsokon levő jelződióda mutatja. A K2 relé pedig percenként átkapcsol.

A tesztprogram valamint a CD-n a SAMPLES alkönyvtárban levő grafikus és BASIC mintaprogramok alapján áttérhet az önálló munkákra.

Műszaki adatok

Tápfeszültség	230V/50Hz hálózati feszültség a 16 és 17 kapcsokra, vagy stabilizált 12V-os hálózati egység a 7 és 8 kapcsokra
Max. teljesítményfelvétel	2,5W
Max. megengedett áram a 6-os kapcsos (5V)	16,max=50mA
Max. megengedett áram a 9-es kapcsos (12...24V)	19,max=50mA
Max. port áram a 28...33 kapcsokon (P1...P6)	128,max...133,max=egyenként 10mA
Max. kivehető áram összesen	16,max + 19,max + 128,max + ... + 133,max = 100mA
Akku puffer áram a 7-es kapcsról	kb. 5mA

Relé kapcsolási teljesítmény (K1, K2)	max. 230VAC, rövid ideig 8A bekapcsolási áram, max. 6A tartós áram
Hőmérséklet mérési tartomány (T1, T2)	-25...102,5 °C felbontás 0,5K, max. hiba +/-2,5%
A/D portok (A1...A4) mérési tartománya	0...2,55V felbontás 0,01V, max. hiba +/-1 digit
Frekvencia mérési tartomány (35.kapocs)	1Hz...30000Hz
Környezeti feltételek	0...40 °C, 20...60% rel. légnedvesség

Ábrák

Felirat a kapcsolási rajzon:

"Wird nicht bestückt" - nincs beültetve