



C-Control I UNIT M 2.0

Best.- Nr.: 19 88 22

C-Control I UNIT M 1.2

Best.- Nr.: 19 88 09



Inhalt

Hinweise zur beschränkten Garantie und Haftung	4
Einführung	4
Bestimmungsgemäße Verwendung	4
Handhabungs- und Sicherheitshinweise.....	5
Allgemeines.....	5
Umgebungsbedingungen	5
Blitzschutz	5
Elektrostatische Entladungen	5
Einführung	5
Was ist der C-Control BASIC Steuercomputer?	5
Anschlüsse der C-Control	6
Steckleisten	6
Steckverbinder	7
Steckbrücken.....	7
Autostart:.....	7
Uref	7
Anschluss externer Baugruppen	7
Digitalports	8
Verwendung eines Digitalports als Eingang	8
Spezielle Funktionen von Digitalports	8
Analogports	8
Referenzspannung	8
Verwendung der AD-Eingänge	8
Verwendung der DA-Ausgänge	8
Aktivantenne	9
Inbetriebnahme	9
Betriebsspannung	9
Betrieb der C-Control M 2.0 am Application-Board	9
Betrieb der C-Control M1.2 u. 2.0 in Ihrer Anwenderschaltung	10
Betrieb der C-Control M 1.2 u. 2.0 am Schnittstellenadapter	10
Verbindung zum PC	10
Betrieb im Application-Board	10
Betrieb am Schnittstellenadapter oder in der Anwenderschaltung	10
Starten der Entwicklungsumgebung (IDE)	11
Einstellen der Schnittstelle	11
Laden eines Programms	11
Compilieren eines Programms	11
Download auf die C-Control	11
Start des Programms auf der C-Control	11
Die nächsten Schritte	12
Standard Functions.....	12
Extended Functions	12

Programmieren in CCBASIC	12
Speichereffizienz:.....	13
Geschwindigkeit:	13
Entwicklungsaufwand	13
Anwender	13
CCBASIC Standard Systemressourcen.....	13
Timer	13
Echtzeituhr	13
User Bytes	14
Programmspeicher	14
Ports	14
Extended Ports	14
Betriebssystem	14
CBASIC – grundlegende Elemente	14
Was ist ein Programm	14
Datentypen	15
Programmzeilen	15
Bezeichner	15
Variablen und Konstanten	15
Label	16
Terme	16
Operanden und Operatoren	16
Funktionen	16
Compileranweisungen	17
Definition symbolischer Konstanten	17
Definition von Variablen.....	17
Befehlsübersicht	20
Mathematische und logische Operatoren	20
Mathematische Funktionen und Befehle	21
Rangfolge von Operatoren und Funktionsaufrufen	21
Anweisungen zur Steuerung des Programmflusses	21
Kommunikation über die serielle Schmittstelle	24
Dateifunktionen	26
Portbefehle	27
Definition und Anwendung von Datentabellen	28
Zugriff auf die Echtzeituhr	29
Einbinden von Systemtreibern	30
CCBASIC – Extended Functions	31
Das CONFIG MODULE	31
Das LCD MODULE	32
Das IR MODULE	32
Das IIC MODULE	33
Das RF MODULE	33
Das STACK MODULE	34
Extended Ports	34
Technische Spezifikationen:	35

Hinweise zur beschränkten Garantie und Haftung

Conrad Electronic übernimmt keine Garantie dafür, dass die Leistungsmerkmale der C-Control Unit 1.2 / 2.0 individuellen Ansprüchen entsprechen.

Conrad Electronic garantiert die Funktion der mitgelieferten Applikationsbeispiele unter Einhaltung der in den technischen Daten spezifizierten Bedingungen und mit den üblichen Vorbehalten gegenüber komplexer Software. Die Gewährleistung von Conrad Electronic beschränkt sich ausschließlich auf den Austausch des Gerätes innerhalb der Garantiezeit bei offensichtlichen Defekten an der Hardware, wie mechanischer Beschädigung, fehlender oder falscher Bestückung elektronischer Bauteile, ausgenommen gesockelter integrierter Schaltkreise und Steckbrücken. Es besteht keine Haftung für Schäden, die unmittelbar durch oder in Folge der Anwendung des Gerätes entstehen.

Unberührt davon bleiben Ansprüche, die auf unabdingbaren gesetzlichen Vorschriften zur Produkthaftung beruhen. Bitte beachten Sie auch unsere Allgemeinen Geschäftsbedingungen.

Einführung

Wir danken uns für den Kauf einer C-Control M-Unit 1.2 / 2.0. Mit diesem Baustein haben Sie ein Produkt erworben, welches nach dem heutigen Stand der Technik entwickelt und gefertigt wurde. Dieses Produkt erfüllt die Anforderungen der geltenden europäischen und nationalen Richtlinien. Die Konformität wurde nachgewiesen, die entsprechenden Erklärungen und Unterlagen sind beim Hersteller hinterlegt.

Um diesen Zustand zu erhalten und einen gefahrlosen Betrieb sicherzustellen, müssen Sie als Anwender diese Bedienungsanleitung beachten.

Bei Fragen wenden Sie sich an unsere Technische Beratung

Deutschland: **Tel. 0180/5 31 21 16 oder 09604/40 88 47**
 Fax 09604/40 88 48
 e-mail: tkb@conrad.de
 Mo. - Fr. 8.00 bis 18.00 Uhr

Österreich: **Tel. 0 72 42/20 30 60 · Fax 0 72 42/20 30 66**
 e-mail: support@conrad.at
 Mo. - Do. 8.00 bis 17.00 Uhr
 Fr. 8.00 bis 14.00 Uhr

Schweiz: **Tel. 0848/80 12 88 · Fax 0848/80 12 89**
 e-mail: support@conrad.ch
 Mo. - Fr. 8.00 bis 12.00 Uhr, 13.00 bis 17.00 Uhr

Bestimmungsgemäße Verwendung

Der C-Control BASIC Steuercomputer Unit M 1.2 / 2.0 dient zur programmierbaren Ansteuerung elektrischer und elektronischer Geräte, die mit Schutzkleinspannung betrieben werden. Diese Geräte können in beliebige technische Systeme integriert werden, die nicht direkt oder indirekt medizinischen, gesundheits- oder lebens- sichernden Zwecken dienen oder durch deren Betrieb Gefahren für Personen oder Sachwerte entstehen können. Zur Programmierung der Produkte ist ausschließlich die C-Control I PC-Software zu verwenden.

Der Einsatzbereich ist auf geschlossene, trockene Räume begrenzt. Der Kontakt mit Feuchtigkeit ist unbedingt zu vermeiden. Das Produkt ist nicht für die Verwendung im industriellen Einsatz geeignet.

Für eine andere Verwendung als zuvor beschrieben, ist das Produkt nicht zugelassen. Darüber hinaus kann dies mit Gefahren, wie z.B. Kurzschluss, Brand, elektrischer Schlag, usw. verbunden sein.

Handhabungs- und Sicherheitshinweise

Allgemeines

Der C-Control BASIC Steuercomputer M-Unit 1.2 / 2.0 wurde gemäß den geltenden gesetzlichen Vorschriften einer Sicherheitsprüfung unterzogen und entsprechend zertifiziert. Bei sachgemäßem Gebrauch gehen normalerweise keine Gesundheitsgefährdungen vom diesem Gerät aus. Der Computer ist als elektrisches/elektronisches Gerät mit der dafür üblichen Vorsicht und Sorgfalt zu behandeln. Die Missachtung der aufgeführten Hinweise oder eine andere als die Bestimmungsgemäße Verwendung kann zur Beschädigung oder Zerstörung des C-Control Computers oder angeschlossener Geräte führen.

Umgebungsbedingungen

Bei Kurzschlüssen in der Versorgungsspannungszuführung besteht Brandgefahr!

Der Schnittstellenadapter ist nicht gegen Lichtbogenüberschläge geschützt und darf nicht in Starkstromindustrieanlagen verwendet werden. Die maximalen Eingangsgrößen gemäß den Spezifikationen in den Technischen Daten dürfen nicht überschritten werden. Der Schnittstellenadapter ist nicht in Räumen oder Umgebungen einzusetzen, in denen brennbare oder ätzende Gase, Dämpfe oder Stäube vorhanden sind oder vorhanden sein können. Nachdem das Gerät von einem kalten in einen wärmeren Raum gebracht wurde, darf es nicht sofort in Betrieb genommen werden. Das möglicherweise entstandene Kondenswasser könnte dabei zur Zerstörung des Gerätes führen. Vermeiden Sie starke Magnetfelder, wie sie in der Nähe von Maschinen oder Lautsprechern vorkommen.

Blitzschutz

Wird Ihre C-Control- Anwendung zusammen mit dem Schnittstellenadapter im Freien installiert, ist auf möglichen Blitzeinschlag zu achten. Montieren Sie Ihre Anwendung nie an besonders blitzgefährdeten Stellen, wie auf freien Bergkuppen, Dächern von Gebäuden, Einzelstehenden Masten oder Bäumen. Direkte und Einschläge in der Nähe elektrischer/elektronischer Geräte können zu deren Zerstörung oder Fehlfunktionen führen. Für Schäden durch Blitzeinschlag können wir keine Haftung übernehmen!

Elektrostatische Entladungen

Besonders in trockener Luft kann sich der menschliche Körper elektrostatisch aufladen. Beim Kontakt mit leitenden Gegenständen baut sich diese Ladung mit einem kleinen Funken ab. Solche Entladungen beim Berühren elektronischer Bauelemente können diese zerstören. Vermeiden Sie daher jeden unnötigen Kontakt mit dem Application-Board oder angeschlossenen Baugruppen. Vor jedem Zugriff sollten Sie einen großen, geerdeten Gegenstand berühren (z. B.: ein PC-Metallgehäuse, eine Wasserleitung oder ein Heizungsrohr), um eventuelle Aufladungen abzubauen.

Einführung

Was ist der C-Control BASIC Steuercomputer?

Der C-Control BASIC Steuercomputer ist ein kompakter Baustein für den universellen Einsatz in Mess-, Steuer- und Regelungsaufgaben und verfügt außerdem über die Fähigkeiten der seriellen Datenübertragung und der Datenspeicherung. Die Mikroprozessortechnik ist aus dem heutigen modernen Leben nicht mehr wegzudenken. In nahezu allen elektronischen Geräten führen Mikroprozessoren Regie. Ihre „Intelligenz“ erhalten diese Chips durch Programmierung.

Die Programmierung eines Mikroprozessors ist teilweise sehr kompliziert und erfordert ein umfangreiches Spezialwissen und teure Entwicklungswerkzeuge. Für Hobbyanwender und kleine Unternehmen bleibt der Zugang zur Mikroprozessortechnik somit meist versperrt. Mit dem C-Control-System eröffnen sich jedoch die Möglichkeiten dieser Technik für jeden interessierten Anwender.

Die neue Generation der C-Control/BASIC Steuercomputer baut auf der erprobten Entwicklungsumgebung der älteren Generation auf, und ist damit abwärts kompatibel. Während die Computer der älteren Generation noch graphisch programmierbar waren, macht dies bei den neueren Modellen, mit enormer Leistungsfähigkeit, keinen Sinn mehr. Die graphische Programmierung komplexer Anwendungen wird bereits nach kurzer Zeit so unübersichtlich, dass nur ein Bruchteil der vorhandenen Systemressourcen genutzt werden können.

Der C-Control/BASIC Steuercomputer der neuen Generation enthält einen weiterentwickelten Mikroprozessor der schnellere Speicherzugriffe erlaubt und damit bis zu 18000 Instruktionen pro Sekunde ausführen kann. Auch das BASIC Betriebssystem ist um wichtige Funktionen ergänzt die den Steuercomputer noch vielseitiger machen.

So wird der C-Control/BASIC Steuercomputer durch wenige Zeilen BASIC-Quelltext zur intelligenten Alarmanlage, zum komplexen Datenerfassungssystem, zur Steuerzentrale einer Heizungsanlage oder zum „Hirn“ eines kleinen Robotermodells. Das Feld der Anwendungsmöglichkeiten ist nahezu unbegrenzt.

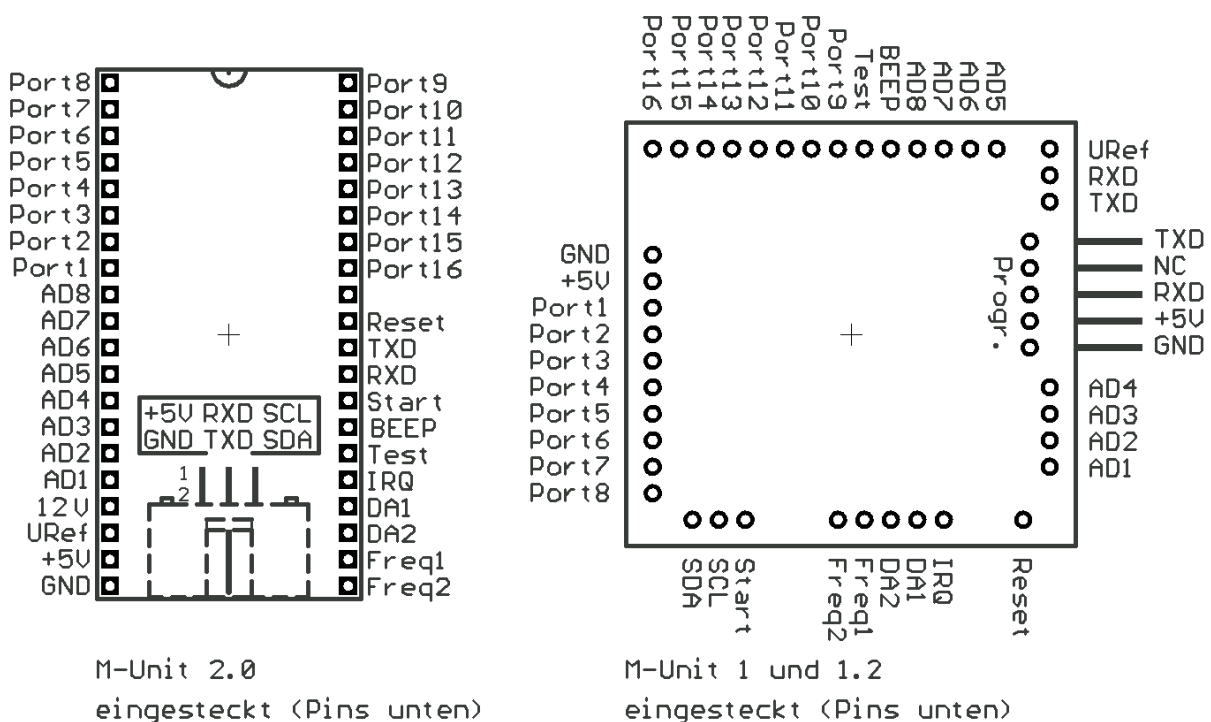
Für den Kontakt zur Außenwelt stehen acht analoge Eingänge, zwei analoge Ausgänge sowie 16 frei als Ein- oder Ausgänge programmierbare Digitalports zum Anschluss von Sensoren, Schaltern, LED's, Transistoren oder Relais zur Verfügung. Manche Ports lassen alternative Funktionen wie etwa die Ansteuerung von Servos zu.

Der Steuercomputer besitzt einen Eingang für einen DCF77-Funkuhrempfänger. Damit ist ein sekundengenaues Ausführen von Schaltfunktionen realisierbar. Alternativ dazu ist über den DCF77-Eingang eine Frequenzmessung oder Ereigniszählung möglich.

Anschlüsse der C-Control

Steckleisten

Die Anschlüsse der C-Control M-Units 1.2 und 2.0 sind jeweils auf eine Steckerleiste geführt, so dass die Computer leicht in Ihre Applikation gesteckt werden können. Grundsätzlich gilt, dass an keinem Anschluss eine Spannung liegen darf, die größer ist als 5V (Ausnahme 10-15V Versorgung der UNIT M 2.0) Ein Port, der als Ausgang geschaltet ist, darf niemals mit einem anderen Ausgang verbunden sein, oder direkt (ohne Vorwiderstand) mit einer Spannungsquelle verbunden sein. Die Zerstörung dieser Ports wäre die unmittelbare Folge.



Steckverbinder

Die serielle Schnittstelle (und bei der Unit M 2.0 auch der I2C-Bus) sind zusätzlich an einen separaten Steckverbinder angeschlossen.



Steckbrücken

Die C-Control hat zwei Steckbrücken sog. Jumper: AUTOSTART und UREF

Autostart:

JP2 bei der Unit M2.0

JP9 bei der Unit M1.2

Ist dieser Jumper gesteckt, so wird beim Anlegen der Betriebsspannung oder nach einem Reset das geladene Programm automatisch gestartet. Ist dieser Jumper nicht gesteckt, so muss der Pin START auf der C-Control für einen kurzen Moment nach LO gezogen werden, um das Programm zu starten.

Beim Betrieb mit dem Application-Board sollte dieser Jumper grundsätzlich abgezogen sein, da eine entsprechende START-Taste vorhanden ist.

Achtung:

Für das Laden eines Programms ist es grundsätzlich erforderlich, diesen Jumper zu entfernen.

Uref

JP1 bei der Unit M2.0

JP8 bei der Unit M1.2

Dieser Jumper verbindet die Referenzspannung der AD-Wandler mit der Betriebsspannung und bestimmt damit den Messbereichsendwert. In der Regel stammt die Betriebsspannung von einem Spannungsregler und genügt den Ansprüchen hinsichtlich Stabilität und Genauigkeit. Bei gehobenen Ansprüchen kann dieser Jumper abgezogen werden und eine entsprechend genaue 5V Referenzspannung über den Anschluss Uref zugeführt werden.

Achtung:

Eine andere Spannung als 5V (+/- 10%) ist nicht zulässig.

Beim Betrieb auf dem Application-Board muss dieser Jumper grundsätzlich abgezogen sein.

Anschluss externer Baugruppen

An den Steckleisten sind alle verwendbaren Ports sowie einige Systemsignale des Steuercomputers herausgeführt. Durch die Bauform der Steckleisten kann der C-Control BASIC Steuercomputer auf andere elektronische Baugruppen aufgesteckt werden

Start und Reset

Diese Anschlüsse sind wahlweise an Taster anzuschließen, wenn ein manueller Start oder Stop von Programmen erforderlich ist. Drücken sie Start um ein geladenes BASIC Programm zu starten und Reset um es anzuhalten, oder ein neues Programm zu laden

Um die Funktion des Start-Tasters zu gewährleisten, muss der zugehörige Autostart-Jumper

JP2 bei der Unit M2.0

JP9 bei der Unit M1.2

abgezogen sein. Die Anschlüsse sind mit 10K Pullup Widerständen versehen.

Digitalports

Auf der Leiterplatte des Steuercomputers sind nur Digitalports mit spezieller Funktion einem Pullup-Widerstand von 10 K versehen. (FREQ1, FREQ2, IRQ). Sie können die anderen Digitalports aber im Konfigurationsregister mit den internen Pullup-Widerständen beschalten. Näheres finden Sie im Kapitel Extended Functions.

Verwendung eines Digitalports als Eingang

Digitaleingänge werden zur Abfrage von Schaltzuständen verwendet. Wird ein Digitalport als Eingang benutzt, führt er im unbeschalteten Zustand einen nicht definierten Pegel, der vom Anwender mittels eines Pullup Widerstandes festgelegt werden sollte. Ist beispielsweise ein Reedkontakt an diesem Port angeschlossen, wird dann bei offenem Schalter eine logische Eins („wahr“) vom Port gelesen, bei geschlossenem Schalter eine logische Null („falsch“). Achten Sie bitte unbedingt darauf, dass je nach Beschaltung des Ports und der logischen Aussage, die Ihr Programm beinhalten soll, der eingelesene Wert eventuell invertiert werden muss (NOT-Operator, siehe Befehlsbeschreibung)!

Verwendung eines Digitalports als Ausgang

Wird ein Digitalport als Ausgang verwendet, können daran nachfolgende ICs, Transistoren oder Low-Current-Leuchtdioden direkt betrieben werden. Der maximal zulässige Laststrom beträgt 10 mA. In jedem Fall ist eine ausreichende Strombegrenzung, zum Beispiel durch einen Widerstand, zu gewährleisten, da es sonst zur Zerstörung des Mikrocontrollers kommen kann! Innerhalb des Mikrocontrollers erfolgt die interne Beschaltung eines Digitalports als Ausgang oder Eingang beim ersten Ausführen des Anwenderprogramms. Nach dem Zuschalten der Betriebsspannung oder nach einem Reset verhalten sich alle Digitalports zunächst elektrisch als Eingang, sie führen also über den Pullup-Widerstand High-Pegel.

Achtung:

Die Summe der Ströme aus allen Digitalports muss auf 20 mA begrenzt werden, wenn Sie die Unit M 2.0 über den 12V Anschluss mit Betriebsspannung versorgen

Spezielle Funktionen von Digitalports

Das Betriebssystem stellt dem Anwender eine Reihe von speziellen Funktionen zur Verfügung, die wahlweise genutzt werden können z.B. den Betrieb von I2C-Bus Komponenten. Mehr dazu finden Sie im Kapitel Extended Functions

Analogports

Der C-Control/BASIC Steuercomputer verfügt über acht A/D-Ports und zwei D/A-Wandler bzw. Servo Steuersignale

Referenzspannung

Bevor die A/D-Eingänge benutzt werden können, muss eine Referenz-Spannung mit dem Referenzspannungseingang des Gerätes verbunden werden. Der angelegte Spannungswert gilt als Obergrenze des Messbereiches der A/D-Wandlung und entspricht dem Wandlungswert 255 (FF hexadezimal). Meistens kann die Betriebsspannung direkt als Referenz benutzt werden. Bei größeren Ansprüchen an Genauigkeit und Stabilität kann eine 5V Referenz extern über der Pin UREF eingespeist werden. Als Referenz für das untere Ende des Messbereiches der A/D-Wandlung dient stets das Groundpotential (Masse, „Minus“) der Betriebsspannung.

Verwendung der AD-Eingänge

An den A/D-Ports können Sensoren aller Art angeschlossen werden, die eine Ausgangsspannung von 0 bis 5 Volt liefern. In den meisten Fällen werden hier aktive Sensoren zur Anwendung kommen, um das Signal des eigentlichen Sensorelementes zu verstärken und den Ansprüchen an Auflösung, Linearität und Driftverhalten zu genügen.

Verwendung der DA-Ausgänge

Die zwei 8-Bit-D/A-Wandler arbeiten nach dem Prinzip der Pulsweitenmodulation. In einem Zeitabschnitt (Modulationsintervall), der aus 256 Teilabschnitten besteht, wird ein D/A-Ausgang für die

Dauer von so vielen Teilabschnitten high-gepulst, wie es dem 8-Bit-Wert entspricht, der zur Ausgabe bestimmt ist. Die Dauer eines Teilabschnittes beträgt $2\mu s$, die des gesamten Modulationsintervalls $512\sim s$ (1953 Hz).

Zur Demodulation, also Wandlung in ein echtes Analogsignal genügt meist ein einfaches RC-Glied. Beachten Sie dabei jedoch die Restwelligkeit und den erzielbaren Maximalwert des Ausgangssignals. Beides ist abhängig von der Last, die nach dem RC-Glied folgt.

Alternativ können die DA-Ausgänge für die Ansteuerung zweier Servos konfiguriert werden. Mehr dazu erfahren Sie im Kapitel Extended Functions.

Aktivantenne

Der Anschluss einer DCF77-Aktivantenne an den C-Control/BASIC Steuercomputer kann über den dafür vorgesehenen Eingang **FREQ1** geschehen.

Die Antenne muss dazu über einen Open-Collector-Ausgang nach Masse verfügen, der durch das empfangene Signal geschaltet wird (low-getastet). Zum Anschluss der Aktivantenne ist unbedingt abgeschirmtes Kabel zu verwenden, da sonst besonders bei größeren Kabellängen Störimpulse eingestrahlt werden können.

Inbetriebnahme

Der Betrieb der C-Control BASIC Computer erfordert eine Entwicklungsumgebung auf einem PC, welche auch den notwendigen Lader enthält um den compilierten BASIC-Code auf die C-Control zu speichern. Der Datentransfer erfolgt über die serielle Schnittstelle. Zur Anpassung an die verschiedenen Pegel von C-Control und PC ist dazu ein Schnittstellenadapter erforderlich. Wenn Ihre C-Control auf einem Application-Board steckt, brauchen Sie keinen separaten Schnittstellenadapter, da dieser auf dem Application-Board bereits enthalten ist.

Buchse 9 bis 15V DC

An dieser Buchse können Sie ein herkömmliches Steckernetzteil, das eine Gleichspannung von 9 bis 15V DC liefert anschließen.

Softwareinstallation

Legen Sie die CD in das Laufwerk Ihres Rechners und starten sie das **SETUP-** Programm. Es installiert die Entwicklungsumgebung (IDE) zusammen mit allen Beispielen und einer ausführlichen Bedienungsanleitung auf Ihrem Rechner.

Betriebsspannung

Betrieb der C-Control M 2.0 am Application-Board

Das Application-Board hat einen eigenen 5V Spannungsregler und versorgt die C-Control mit der notwendigen Betriebsspannung. Außerdem wandelt es den Schnittstellenpegel des PC auf das richtige Mass. Das Application-Board bietet Ihnen zwei Möglichkeiten zur Versorgung mit einer Gleichspannung 9-15Volt.

Anschlussklemmen 9 bis 15V DC

Stellen Sie eine Gleichspannungsversorgung mit zwei abisolierten Kabelenden zur Verfügung.

Prüfen Sie bitte, welches der beiden Kabelenden Masse („Minus“) und welches spannungsführend („Plus“) ist. Schalten Sie die Spannungsversorgung aus und klemmen Sie die Kabelenden an der zweipoligen Schraubklemme polungsrichtig an. Beachten Sie dabei bitte den Anschlussplan im Manual zum Application-Board

Buchse 9 bis 15V DC

An dieser Buchse können Sie ein herkömmliches Steckernetzteil, das eine Gleichspannung von 9 bis 15Volt liefert anschließen. Der Minuspol muss dabei außen liegen. Achten sie besonders darauf, wenn Sie ein Netzteil mit wechselbaren Steckern verwenden, da hier die Polarität frei wählbar ist.

Betrieb der C-Control M1.2 u. 2.0 in Ihrer Anwenderschaltung

Beide C-Control Ausführungen lassen sich auch in Ihrer Anwendung programmieren (incircuit programming). Ihre Schaltung muss für den Betrieb des C-Control Computers eine stabile 5V Gleichspannung zur Verfügung stellen. Bei der Ausführung M1.2 und M2.0 ist die Pin-Bezeichnung für die Betriebsspannung jeweils +5V u. GND

Bei der Ausführung M 2.0 kann auch eine Gleichspannung von 8 bis 12V an den Pins +12V u. GND zugeführt werden, da diese Version einen Festspannungsregler auf der Platine hat.

Sehen Sie hierzu bitte den Anschlussplan der Units M1.2 bzw. M 2.0

Betrieb der C-Control M 1.2 u. 2.0 am Schnittstellenadapter

Der Schnittstellenadapter hat einen eigenen 5V Spannungsregler und versorgt die C-Control mit der notwendigen Betriebsspannung. Außerdem wandelt er den Schnittstellenpegel des PC auf das richtige Mass. Der Schnittstellenadapter bietet Ihnen zwei Möglichkeiten zur Versorgung mit einer Gleichspannung 9-15Volt.

Anschlussklemmen 9 bis 15V DC

Stellen Sie eine Gleichspannungsversorgung mit zwei abisolierten Kabelenden zur Verfügung.

Prüfen Sie bitte, welches der beiden Kabelenden Masse („Minus“) und welches spannungsführend („Plus“) ist. Schalten Sie die Spannungsversorgung aus und klemmen Sie die Kabelenden an der zweipoligen Schraubklemme polungsrichtig an. Beachten Sie dabei bitte den Anschlussplan im Manual zum Schnittstellenadapter

15Volt liefert anschließen. Der Minuspol muss dabei außen liegen. Achten sie besonders darauf, wenn Sie ein Netzteil mit wechselbaren Steckern verwenden, da hier die Polarität frei wählbar ist.

Achtung:

Wenn sie eine C-Control M1.2 oder M2.0 an einen Programmieradapter anschließen muss die Betriebsspannung entweder vom Programmieradapter oder von der Anwenderschaltung bzw. dem Application-Board zur Verfügung gestellt werden. Auf keinen Fall darf die C-Control von beiden Einheiten gleichzeitig versorgt werden.

Verwenden sie niemals den Schnittstellenadapter zur Versorgung der C-Control Unit, wenn diese in einer Anwenderschaltung steckt deren Strombedarf insgesamt mehr als 100 mA beträgt. In diesem Fall muss die C-Control und der Programmieradapter von der Anwenderschaltung mit Spannung versorgt werden.

Verbindung zum PC

Betrieb im Application-Board

Der C-Control Computer braucht für das Laden des compilierten Programmes eine Verbindung zur seriellen Schnittstelle des PC. Mit dem Application-Board oder dem Schnittstellenadapter wurde Ihnen ein Schnittstellenkabel ein 9-poliges Nullmodemkabel (ca. 1,5 Meter) ausgeliefert. Stecken Sie nun das Nullmodemkabel an eine freie serielle Schnittstelle Ihres Computers. Viele Computer verfügen über eine 9-poligen und eine 25-polige serielle Schnittstelle. Sollte bei Ihrem Computer nur noch eine 25-polige Schnittstelle frei sein, benötigen Sie einen zusätzlichen Adapter.

Betrieb am Schnittstellenadapter oder in der Anwenderschaltung

Verbindung zur Unit M 1.2

Stecken Sie die Unit M1.2 oder M 1.1 in die dafür vorgesehen Buchse am Schnittstellenadapter. Beide Bestückungsseiten der Leiterplatten zeigen dabei nach oben. Sollten Sie unsicher sein, ob die Verbindung polungsrichtig hergestellt wurde, sehen Sie sich hierzu bitte die Anschlusszeichnung im Manual des Schnittstellenadapters an.

Verbindung zur Unit M 2.0

Die Version 2.0 hat an der Stirnseite einen 6 poligen Stecker der mittels einem passenden Anschluss-Kabel mit dem Schnittstellenadapter verbunden wird. Der Stecker hat einen Verpolungsschutz.

Starten der Entwicklungsumgebung (IDE)

Schalten Sie jetzt die Spannungsversorgung ein. Starten Sie die CCBASIC Entwicklungsoberfläche über das Windows Startmenü

Start -> CCBASIC -> CCEW32d.exe

Oder starten sie die EXE vom Explorer aus mit einem Doppelclick

Programme->CCBASIC -> CCEW32d.exe

Dieses Programm ist mit einer Standard-Oberfläche mit Menü und Mausbedienung ausgestattet und ermöglicht die Eingabe von BASIC-Quellcode (Editor), das übersetzen (Compiler) und das Laden (Lader) des C-Control Codes in den Steuercomputer.

Einstellen der Schnittstelle

Stellen Sie zunächst über das „Optionen“ Menü ein, über welchen seriellen Port (COM1, COM2 ..) das C-Control System mit dem PC verbunden ist.

Laden eines Programms

Mit der Entwicklungsumgebung wurden zahlreiche Beispiele in Ihr Zielverzeichnis kopiert. Sie zeigen Ihnen Schritt für Schritt den Gebrauch von CCBASIC. Laden Sie jetzt das erste Beispiel indem sie in der Entwicklungsumgebung Auf Das Menü „Datei öffnen“ klicken und das erste Beispiel laden. Der Pfad ist:

Programme -> CCBASIC_2.0 -> Beispiele -> APPLICATION_BOARD ->Standard functions -> START.BAS

Compilieren eines Programms

Wählen sie in der Menüleiste die Compilerfunktion der Entwicklungsumgebung.

Menü Entwicklung -> compilieren

Das von Ihnen erstellte oder geladene Programm BASIC-Programm wird von einem Compiler auf seine syntaktische Korrektheit überprüft und danach in eine Folge von Befehlsbytes umgesetzt.

Die Befehle und die zugehörigen Parameterbytes werden über die serielle Schnittstelle zum Steuercomputer übertragen, wo sie von dessen Betriebssystem im FLASH-Speicher abgelegt werden. Durch das C-Control Konzept können Ihre Anwendungsprogramme in sehr kompakter Form gespeichert werden.

Download auf die C-Control

Wenn keine syntaktischen Fehler im Programm sind, verläuft die Compilierung erfolgreich und Sie können das übersetzte Programm in die C-Control übertragen. Andernfalls gibt Ihnen der Compiler Hinweise darauf, welcher Art der Fehler ist, und in welcher Zeile er gefunden wurde. Die C-Control muss zum Download im RESET-Zustand sein (dies ist nach Anlegen der Betriebsspannung oder nach Drücken der entsprechenden Taste RESET am Application-Board der Fall). Achten Sie auch darauf, dass der AUTOSTART Jumper nicht gesetzt ist. Starten Sie jetzt die Übertragung.

Menü Entwicklung -> in C-Control übertragen

Die serielle Schnittstelle zwischen PC und dem Steuercomputer arbeitet bidirektional. über sie werden neben den Befehlscodes zur Programmierung eines Anwenderprogramms auch einige Konfigurations- und Steuerbefehle sowie Statusinformationen übertragen. Prinzipiell ist eine Verbindung von PC und Steuercomputer nur zur Programmierung erforderlich. Anschließend kann der Steuercomputer in die Zielapplikation eingesetzt werden und dort eigenständig arbeiten. Die Verbindung zum PC kann allerdings auch während der Arbeit in der Zielapplikation bestehen bleiben und z.B. zur Übertragung von Messdaten benutzt werden. Der Übertragungsvorgang wird mit einer Leuchtdiode angezeigt und dauert bei diesem kleinen Beispiel kaum eine Sekunde.

Start des Programms auf der C-Control

Wenn alle Verbindungen richtig hergestellt wurden ist die Übertragung auf den BASIC-Steuercomputer erfolgreich verlaufen

Das Anwenderprogramm ist jetzt dauerhaft auf die C-Control geladen und bleibt auch nach dem Ausschalten der Betriebsspannung gespeichert. Zum Start des Programms muss grundsätzlich der

PIN START auf LO gezogen werden. Sie erreichen dass indem der AUTOSTART Jumper gesteckt wird, oder eine entsprechende Taste in Ihrer Schaltung drücken, die diese Leitung bedient. Auf dem Application-Board ist diese Taste mit START beschriftet.

Das geladene Programm wird gestartet, im Falle des Beispiels START.BAS geht es nur darum die prinzipielle Funktionstüchtigkeit aller Einstellungen und Verbindungen zu prüfen. Das Beispiel lässt die Beleuchtung des LCD auf dem Application-Board blinken.

Die nächsten Schritte

Die Installation der Entwicklungsumgebung beinhaltet eine Reihe von Beispielen die Ihnen die Programmierung von CC-BASIC erklären. Damit sich auch erfahrene Anwender nicht mühsam durch längst bekannte Fakten wühlen müssen, sind die *Standard Functions* und *Extended Functions* in getrennten Ordnern angelegt.

Standard Functions

Im Ordner

Programme->CCBASIC_2.0 -> Beispiele -> TERMINAL_AUSGABE

finden Sie Beispiel die ohne Applikation-Board bzw. LCD auskommen und Ihre Ausgaben auf ein Terminalprogramm leiten. Diese Beispiele zeigen Ihnen die Standardfunktionen von CCBASIC, berücksichtigen allerdings nicht die neuen, erweiterten Funktionen, da hierfür ein Application-Board mit dem entsprechenden Zubehör erforderlich ist um die Beispiele sinnvoll auszuprobieren.

Eine umfassendere Sammlung an Beispielen zu CCBASIC und dem Betrieb mit dem Application-Board finden Sie im Ordner

Programme->CCBASIC_2.0 -> Beispiele -> APPLICATIONBOARD -> STANDARD_FUNCTIONS

Hier finden Sie eine Einführung in die Standard-Funktionen von CCBASIC, basierend auf einem Application-Board mit LCD und Tastatur. Die Beispiele sind nummeriert und tragen die Thematik im Filenamen. Wenn Sie noch keine Erfahrung mit CCBASIC haben, ist es sinnvoll die Beispiele der Reihe nach zu laden und auch nach eigenen Vorstellungen zu modifizieren. Achten Sie bitte auf die Erklärungen im Kopf der Programme, sie enthalten oft wichtige Hinweise zur Funktion oder möglichen Problemen.

Extended Functions

Hier finden Sie die Beispiele zu den neuen Funktionen des Betriebssystems, basierend auf einem Application-Board mit LCD und Tastatur.

Programme->CCBASIC_2.0 -> Beispiele -> APPLICATIONBOARD -> EXTENDED_FUNCTIONS

Sie zeigen Ihnen wie vorhandene Ressourcen alternativ genutzt werden können bzw. wie neue Möglichkeiten und neues Zubehör einfach zu benutzen sind. Details dazu finden Sie hier im Manual, im Kapitel EXTENDED FUNCTIONS.

Programmieren in CCBASIC

Kleine Microcontroller haben die Welt erobert und sind überall zu finden. Microcontroller werden vorwiegend in Assembler programmiert werden, was Erfahrung in dieser Programmiersprache und genaue Kenntnis der Prozessorarchitektur erfordert. Nicht selten ist allein das Manual zum Prozessor mehrere hundert Seiten lang und von Laien praktisch nicht zu verstehen. Auch wer Assembler programmieren möchte, ohne diese Kenntnisse bereits zu haben ist auf Bücher angewiesen deren Seitenumfang ganz sicher noch größer ist.

CCBASIC ist der BASIC-Dialekt, der zur Programmierung des C-Control BASIC Steuercomputers verwendet wird und keine der genannten Spezialkenntnisse erfordert. Die Syntax entspricht in etwa der des Standard-BASIC. Bei einigen Befehlen gibt es Abweichungen oder Erweiterungen, die speziell auf die Hardware des Steuercomputers zugeschnitten sind.

Die neuere Generation der C-Control Computer bietet einige Erweiterungen die den Umgang mit CCBASIC noch komfortabler machen. Sie sind werden Kapitel Extended Functions erläutert.

Speichereffizienz:

Die Computer der C-Control Serie haben ein komplettes Betriebssystem auf dem Chip und lassen sich in einem BASIC-Dialekt programmieren. BASIC ist nicht nur leicht zu erlernen, sondern auch sehr Platz sparend. Ein BASIC- Programm braucht nur etwa ein Fünftel des Speichers, das ein Assemblerprogramm gleicher Funktion benötigen würde. Grundsätzlich gilt, je länger das BASIC - Programm ist, desto größer ist der Speicher-Vorteil gegenüber einer Assembler Programmierung.

Geschwindigkeit:

Der Ausspruch „BASIC ist langsam“ stammt aus der Zeit, als BASIC auf dem Computer als Quelltext gespeichert war und der gesamte Text bei der Ausführung erst gelesen und interpretiert werden musste. Bei CCBASIC wird der Quelltext von einem Compiler gelesen und in die von Betriebssystem lesbaren sog. Token verwandelt, welche als eigentliches Programm in die C-Control geladen wird. Das Betriebssystem muss also nicht die vielen Buchstaben einer Programmzeile lesen sondern nur das dafür repräsentative Token. Und das geht sehr schnell. Die C-Control Unit M 2.0 benötigt nur etwa 50us für die Ausführung eines BASIC-Befehls.

Entwicklungsaufwand

Wenn man bedenkt, dass ein einziger BASIC-Befehl Routinen des Betriebssystems aufruft, die meistens mehrere hundert Bytes lang sind, ist leicht einzusehen, dass es eine enorme Zeitersparnis darstellt ein Programm in BASIC zu programmieren. Ein Programm in BASIC zu programmieren dürfte nur etwa ein Zehntel der Zeit beanspruchen, welches ein Assemblerprogramm gleicher Funktion zum Entwurf benötigt.

Anwender

CCBASIC und C-Control ist also auch für Entwickler und Programmierer, welche die Kenntnisse über Prozessorarchitektur und Assemblerprogrammierung haben, ein leistungsfähiges Tool komplexe Programme in kürzester Zeit zu entwickeln und eine flexibles Controller-System für die Steuerung in Ihrer Anwendung.

CCBASIC Standard Systemressourcen

Dieses Kapitel beschreibt die bei C-Control verfügbaren Standard-Ressourcen welche Ihnen bereits von der älteren C-Control Generation bekannt sein wird, wenn Sie bereits Erfahrung mit unseren C-Control Computern haben.

Unter dem Begriff „Systemressourcen“ sind hier alle internen Funktionseinheiten zusammengefaßt, die sich nicht nur aus den Eigenschaften des Mikrocontrollers ableiten, sondern durch das auf dem Chip integrierte Betriebssystem zur Verfügung gestellt werden. Wie diese Systemressourcen im BASIC-Programm angesprochen werden, wird weiter unten in der Befehlsübersicht beschrieben.

Timer

Im Hintergrund des Betriebssystems läuft ein mit 20 Millisekunden getakteter 16-Bit-Timer, dessen Wert jederzeit ausgelesen und zum Herstellen von Zeitbezügen im BASIC-Programm benutzt werden kann.

Echtzeituhr

Die Zeit- und Datumsinformation wird vom Betriebssystem in sieben interne Speicherzellen (Jahr, Monat, Tag, Wochentag, Stunde, Minute, Sekunde) übertragen und bis zur nächsten Synchronisation in Portionen von 20 Millisekunden erhöht. Die Ganggenauigkeit der Echtzeituhr ist bestimmt durch die Abweichung des Quarzes von seiner Normalfrequenz von bis zu 0,1 Promille, abhängig von

Streuungen in der Serienproduktion und von der Temperatur. Das entspricht einer Abweichung von bis zu 0,36 Sekunden pro Stunde.

Nach dem Zuschalten der Betriebsspannung und nach einem Reset startet die Uhr mit dem 01.01.04, 00:00:00 Uhr. Die internen Speicherzellen für Datum und Uhrzeit können vom BASIC-Programm aus gelesen und beschrieben werden. Durch das Beschreiben der Zeitspeicherzellen kann die Uhr gestellt werden. Bei größeren Ansprüchen an die Genauigkeit der Uhr, kann diese mittels eines DCF77 Empfängermoduls synchronisiert werden.

User Bytes

Während die Computer der alten Generation nur 24 User Bytes für Variablen zur Verfügung stellten, können Sie bei der C-Control Unit M 2.0 u. 1.1 64 Bytes für sich beanspruchen. Die Verwendung dieser Userbytes ist im Abschnitt zum DEFINE-Befehl weiter unten beschrieben.

Programmspeicher

Die neue Generation der C-Control Computer stellt nicht nur mehr Programmspeicher zur Verfügung (10 kByte statt bisher 8 kByte) sondern kann den Programmspeicher mit voller Busgeschwindigkeit lesen, da dieser jetzt Bestandteil des Prozessors ist und nicht mehr auf ein externes EEPROM ausgelagert ist, wie es bisher der Fall war.

Ports

Die bisher verfügbaren 16 Digitalports, 2 DA-Ports, 2 FREQ-Ports und 8 AD-Ports sind weiter verfügbar, haben aber teilweise eine alternative Funktionen, die C-Control noch flexibler in der Anwendung machen. Mehr darüber finden Sie im Kapitel Extended Functions.

Extended Ports

Ports sind mit die wichtigsten Ressourcen, eines C-Control Computers. Um auch den höchsten Ansprüchen zu genügen, können über den I2C-Bus Portbausteine angeschlossen werden. Das Betriebssystem verwaltet bis zu 16 PCF8574(A) als Bitport oder Byteport über den I2C-Bus.

Diese Extended Ports werden in BASIC genauso angesprochen, wie auch die zwei Digitalports der C-Control. Mehr finden Sie auch hier im Kapitel zu den Extended Functions

Betriebssystem

Erweitert wurde auch das Betriebssystem der neuen C-Control Computer. Obwohl es abwärts kompatibel ist, damit auch Programme der alten C-Control Computer laufen, ist es um verschiedene Funktionen ergänzt, welche die Anwendung noch komfortabler machen. So ist z.B. ein direkter Zugriff auf verschiedenen Hardwarekomponenten, welche als Zubehör erhältlich sind, möglich

CBASIC – grundlegende Elemente

Was ist ein Programm

Ein Programm ist die Beschreibung eines Informationsverarbeitungsprozesses.

Im Laufe eines solchen Prozesses wird aus einer Menge von variablen oder konstanten Eingangswerten eine Menge von Ausgangswerten berechnet. Die Ausgangswerte sind entweder selbst Ziel der Informationsgewinnung oder dienen mittelbar zur Reaktion auf die Eingangswerte. Neben den eigentlichen Berechnungen kann ein Programm Anweisungen zum Zugriff auf die Hardware des Computers oder zur Steuerung des Programmflusses enthalten.

Ein BASIC-Programm besteht aus mehreren Zeilen sogenannten Quelltextes. Dabei enthält jede Zeile eine oder mehrere Rechen- oder Steueranweisung. Außer diesen Anweisungen selbst bestimmt ihre Reihenfolge ganz wesentlich die eingangs beschriebene Informationsverarbeitung. Die Ausführung der den Anweisungen entsprechenden Operationen durch den Steuercomputer erfolgt sequentiell,

also nacheinander. Eine Folge von Programmanweisungen mit einem bestimmten Ziel nennt man auch Algorithmus.

Datentypen

Daten sind die Objekte des Informationsverarbeitungsprozesses, sie repräsentieren die gespeicherten Informationen. Der C-Control BASIC Steuercomputer verarbeitet und speichert ausschließlich ganzzahlige numerische Daten - sogenannte „Integerzahlen“ von 1, 8 oder 16 Bit. Eine Variable von 8 Bit (Byte) kann nur nichtnegative Werte von 0 bis 255 annehmen. Der Wertebereich einer Integervariablen von 16 Bit (Word) reicht von -32768 bis +32767. Achten Sie bei allen Berechnungen darauf, daß die Ergebnisse diese Grenzwerte nicht über- oder unterschreiten, da es sonst zu sogenannten „Überläufen“ kommt.

$a = 255 + 1$

ergibt beispielsweise für a den Wert 0 und nicht 256, wenn a nur ein Byte repräsentiert!

$a = -32768 - 1$

ergibt 32767 und nicht -32769, wenn a ein Word repräsentiert!

Programmzeilen

Jede Programmzeile enthält eine oder mehrere Anweisung, die durch Doppelpunkte : getrennt sind. Zeilennummern, wie in älteren BASIC-Dialekten üblich, sind nicht notwendig. Werden dennoch Zeilennummern angegeben, so können diese als Sprungziel verwendet werden.

10...

GOTO 10

Einen Einfluß auf die Reihenfolge der Programmoperationen haben die Nummern darüber hinaus nicht. Wenn beispielsweise im Quelltext auf eine mit 200 numerierte Zeile eine Zeile 100 folgt, wird trotzdem die Zeile 200 vor der 100 abgearbeitet.

Kommentare können zur Erläuterung des geschriebenen Programms mit in den Quelltext aufgenommen werden und steigern dessen Lesbarkeit und Wartungsfreundlichkeit. Ein Kommentar in CCBASIC beginnt stets mit einem Hochkomma ' und erklärt den Rest der Zeile zum nicht zum Programm gehörigen Text.

$a = b + c$ '... *Kommentar* ...

Bezeichner

Bezeichner sind Programmelemente aus alphanumerischen Zeichen (Abis Z, 0 bis 9) die in vom Programmierer festgelegter Weise Objekte, wie Variablen und Konstanten, bezeichnen. Label-Namen und die sogenannten „reservierten Worte“ sind ebenfalls Bezeichner.

Es erfolgt keine Unterscheidung von Groß- und Kleinbuchstaben. Ein Bezeichner beginnt stets mit einem Buchstaben oder mit einem Unterstrich. Leerzeichen innerhalb eines Bezeichners sind nicht erlaubt.

Variablen und Konstanten

Variablen und Konstanten sind Objekte des Informationsverarbeitungsprozesses. In CCBASIC speichern beide einen numerischen Wert. Während der Wert einer Konstante einmal angegeben wird und dann unverändert bleibt, kann sich der Wert einer Variablen im Lauf des Programms beliebig oft ändern. Konstanten können in CCBASIC in dezimaler, hexadezimaler und binärer Form angegeben werden.

Die Syntax für Hexadezimal- und Binärzahlen sei hier am Beispiel der Zahl 46 (dezimal) gezeigt:

&H2E

&B10111000

Außerdem können per DEFINE-Zeilen (siehe unten) symbolische Konstanten vereinbart werden.

Auf Variablen wird stets über ihren Bezeichner zugegriffen. Dieser Bezeichner muß vor der ersten Verwendung der Variable im Programm in einer DEFINE-Zeile definiert werden.

Label

Label markieren bestimmte Punkte in der Folge der Programmoperationen. Label sind Ziele von Sprungoperation innerhalb eines Algorithmus. In CCBASIC stehen Label am Anfang einer Zeile und beginnen stets mit einem Doppelkreuz, dann folgt - ohne Leerzeichen - der Bezeichner des Labels. Das Beispiel zeigt die Definition des Labels „Label1“ und die Verwendung in einem Sprungbefehl:

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

Terme

Ein Term ergibt sofort (als Variable oder Konstante) oder durch Berechnung einen bestimmten Wert. Terme sind Teile von Anweisungen und stehen beispielsweise bei der Zuweisung eines Wertes an eine Variable rechts des Zuweisungszeichens „=“.

Terme werden durch Kombinationen von Operanden und Operatoren gebildet.

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

Operanden und Operatoren

Ein Operand ist in der Grundform entweder eine Konstante, eine Variable oder ein Funktionsaufruf, kann aber auch selbst wieder ein aus Operanden und Operatoren zusammengesetzter Term sein.

Operatoren bezeichnen Rechenoperationen, die mit den umstehenden Operanden auszuführen sind. Dabei gibt es eine definierte Rangfolge der Operatoren (siehe Befehlsbeschreibung), die die Reihenfolge der Berechnungen bestimmt.

Funktionen

Eine Funktion führt eine definierte Operation - zum Beispiel eine Berechnung - durch und liefert durch ihren Aufruf einen Ergebniswert. Die meisten Funktionen erwarten ein oder mehrere Argumente, die in runden Klammern „()“ nach dem Funktionsbezeichner übergeben werden und durch Kommas getrennt sind. Einige Funktionen werden ohne Argument aufgerufen. In diesem Fall werden keine runden Klammern geschrieben.

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

In CCBASIC sind alle unterstützten Funktionen vordefiniert. Deren Bezeichner gehören zu den reservierten Worten. Die Formulierung anwenderdefinierter Funktionen ist in CCBASIC nicht vorgesehen. Die Zuweisung ist die einfachste Form einer Programmanweisung. Nach dem Bezeichner einer Variablen, der ein Wert zugewiesen werden soll, folgt das Zuweisungszeichen „=“ und dann ein Term, der den zu zuweisenden Wert bestimmt. Eine Zuweisung entspricht damit einer einfachen mathematischen Formel.

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

Neben den einfachen Zuweisungen sind Befehle Anweisungen zur Ausführung von Programmoperationen durch den C-Control/BASIC Steuercomputer. Befehle beginnen stets mit einem reservierten Wort. Einige Befehle erwarten einen oder mehrere Parameter zur genauen Spezifikation der auszuführenden Programmoperation. Diese Parameter werden nach dem Befehlsbezeichner und einem Leerzeichen aufgeführt und dabei durch Kommas getrennt (Ausnahme PRINT, siehe Befehlsübersicht). Im Gegensatz zu den Argumenten beim Aufruf einer Funktion stehen die Befehlsparameter nicht innerhalb runder Klammern!

RANDOMIZE
PAUSE 100
BEEP 40,50,50

Diese Anweisungen erlauben, die Reihenfolge der an sich streng sequentiell abgearbeiteten Programmoperationen zu steuern und an Eingangswerte des Informationsverarbeitungsprozesses anzupassen. Sie bieten eine hohe Flexibilität bei der Algorithmenformulierung und sind für die Lösung mancher anwendungs-technischer Probleme sogar Grundvoraussetzung. Anweisungen zur Steuerung des Programmflusses bestehen aus einem oder mehreren reservierten Worten und erfordern in jeweils spezieller Weise eventuell weitere Angaben.

GOTO label1
IF a > b THEN GOSUB label2
FOR i = 0 TO 10 STEP 2
..
NEXT

Compileranweisungen

Zusätzlich zu den Programmanweisungen enthält ein CCBASIC-Quelltext Compileranweisungen, die zum Beispiel zum Anlegen von Datenblocks (Tabellen) oder zur Definition von Variablen - und Konstanten dienen. Für Compileranweisungen gilt die Doppelpunktregel zum Trennen mehrerer Anweisungen in einer Zeile nicht. Es darf jeweils nur eine Compileranweisung in einer Zeile stehen. Die DEFINE-Anweisung ist eine Compileranweisung.

Definition symbolischer Konstanten

Es ist guter Programmierstil, statt „magischer“ Zahlen im Programm

IF x > 1234 THEN GOTO alarm

besser symbolische Konstanten zu verwenden. Durch Vergabe signifikanter Bezeichner für Konstanten erhöht sich die Lesbarkeit des Quelltextes. Wenn alle Konstanten global definiert werden, ist ein Programm auch leichter zu warten. Das gilt besonders, wenn ein und dieselbe Konstante mehrmals im Programm benötigt wird. Die Definition einer symbolischen Konstanten erfolgt wie folgt:

DEFINE bezeichner wert

Dabei ist wert entweder eine dezimale, hexadezimale oder binäre Zahl. So sollte das Beispiel zuvor besser so lauten:

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

Definition von Variablen

Der C-Control/BASIC Steuercomputer stellt 64 Byte-Speicherzellen seines internen Speichers (RAM) dem Anwender zur Verwendung in seinen Programmen zur Verfügung. In diesem Speicherbereich werden alle Variablen eines BASIC-Programms gespeichert. Die 64 Bytes können je nach Bedarf auch bitweise oder als 16bit Integer (Word) verwendet werden. Im Gegensatz zum Standard-BASIC müssen in CCBASIC alle vom Programm benutzten Variablen vor ihrer ersten Verwendung definiert werden. Dabei ist der Datentyp zu spezifizieren (Bit, Byte oder Word) und kann (für Bits muß!) eine Speicherzellennummer angegeben werden. Der Anwender muß selbst darauf achten, daß keine unerwünschten Überlappungen bei der Vergabe der Speicherplätze entstehen, da es sonst zum gegenseitigen überschreiben der Variablen kommen kann. Beispielsweise belegen bit[18], byte[2] und word[1] jeweils einen Teil der Zelle 2 des Speicherbereiches.

Definition einer Bitvariablen:

DEFINE bezeichner BIT [nr]

Dabei sind für nr Werte von 1 bis 255 (32 Bytes mit je 8 Bit) zulässig. Es können also nicht alle 64 Variablen als Bitvariable definiert werden da der Compiler selbst nur ein Byte für die Verwaltung der Bitvariablen verwendet.

Definition einer Bytevariablen mit Zellennummer:

DEFINE bezeichner BYTE[nr]

Dabei sind für nr Werte von 1 bis 64 (64 Bytes) zulässig.

Definition einer Integervariablen mit Zellennummer:

DEFINE bezeichner WORD[nr]

Dabei sind für nr Werte von 1 bis 32 (ein Word belegt 2 Bytes) zulässig. Wenn bei Byte- und Worddefinitionen die Zellenangabe [nr] weggelassen wird, übernimmt der Compiler die Aufteilung auf den Speicherbereich. Achten Sie dann darauf, daß nicht abwechselnd Bytes und Words definiert werden. Die folgenden Anweisungen

```
DEFINEa BYTE  
DEFINE b WORD  
DEFINE c BYTE  
DEFINE d WORD
```

führen zu zwei ungenutzten (verschenkten kostbaren!) Bytes, zwischen a und b sowie zwischen c und d, da Words prinzipiell an den Bytes 1,3,5,7,... usw. der 24 Bytes ausgerichtet werden. Man schreibt deshalb besser:

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

Die automatische Aufteilung der Variablen auf den Speicher durch den Compiler beginnt bei Zellennummer 1. Das obige (bessere) Beispiel belegt 6 Bytes. Bei Definition weiterer Bits, Bytes und Words mit Angabe der Zellennummer ist wieder auf unerwünschte Überlappung zu achten. Ein bereits definierter Variablenbezeichner darf nicht ein zweites Mal definiert werden.

Definition von Digitalports

In CCBASIC wird auf Ports wie auf Variablen zugegriffen. Auch hier muß jeder verwendete Port zuvor definiert sein.

Definition eines der 16 Digitalports:

DEFINE bezeichner PORT[nr]

Dabei sind für nr Werte von 1 bis 16 zulässig.

Definition eines der 64 Extended Digitalports:

Über den I2C-Bus des BASIC-Computers können bis zu acht PCF8574 und weitere acht PCF8574A adressiert werden. Die Extended Ports werden genauso angesprochen wie einer der 16 Digitalports der CPU, jedoch müssen Sie natürlich wissen, welcher PCF8574 Adresse welche Ports gehören: Mehr darüber finden Sie im Kapitel Extended Functions.

PCF 8574	ADR 0	Ports 17 – 24	BYTEPORT 3
PCF 8574	ADR 1	Ports 25 – 32	BYTEPORT 4
PCF 8574	ADR 2	Ports 33 – 40	BYTEPORT 5
PCF 8574	ADR 3	Ports 41 – 48	BYTEPORT 6
PCF 8574	ADR 4	Ports 49 – 56	BYTEPORT 7

PCF 8574	ADR 5	Ports 57 - 64	BYTEPORT 8
PCF 8574	ADR 6	Ports 65 - 72	BYTEPORT 9
PCF 8574	ADR 7	Ports 73 - 80	BYTEPORT 10

PCF 8574A	ADR 0	Ports 81 - 88	BYTEPORT 11
PCF 8574A	ADR 1	Ports 89 - 96	BYTEPORT 12
PCF 8574A	ADR 2	Ports 97 - 104	BYTEPORT 13
PCF 8574A	ADR 3	Ports 105 - 112	BYTEPORT 14
PCF 8574A	ADR 4	Ports 113 - 120	BYTEPORT 15
PCF 8574A	ADR 5	Ports 121 - 128	BYTEPORT 16
PCF 8574A	ADR 6	Ports 129 - 136	BYTEPORT 17
PCF 8574A	ADR 7	Ports 137 - 144	BYTEPORT 18

DEFINE bezeichner PORT[nr]

Dabei sind für nr Werte von 17 bis 144 zulässig, aber der PCF8574 muss entsprechend adressiert sein.

Definition eines 8 Bit breiten Ports:

DEFINE bezeichner BYTEPORT[nr]

Dabei sind für nr nur die Werte 1 (Ports 1 bis 8 als Byteport) und 2 (Ports 9 bis 16) zulässig.

Definition eines 8 Bit breiten Extended Digitalports:

Der PCF8574 am I2C-Bus kann natürlich auch als BYTEPORT angesprochen werden. In der Aufstellung oben finden Sie die Zuordnung zur Adresse. Mehr darüber finden Sie im Kapitel Extended Functions.

DEFINE bezeichner BYTEPORT[nr]

Dabei sind für nr nur Werte 3 (Ports 17 bis 24 als Byteport) bis 18 (Ports 137 bis 144) zulässig.

Definition eines Bezeichners für den gemeinsamen Zugriff auf alle 16 Digitalports als ein 16 Bit Port:

DEFINE bezeichner WORDPORT[nr]

Für nr ist nur der Wert 1 zulässig. Die Extended Ports sind nicht als WORDPORT verfügbar.

Definition von Analogports

Definition eines der 8 A/D-Potts:

DEFINE bezeichner AD[nr]

Dabei sind für nr Werte von 1 bis 8 zulässig.

Definition eines der 2 D/A-Ports:

DEFINE bezeichner DA[nr]

Dabei sind für nr nur die Werte 1 und 2 zulässig.

Die beiden DA-Ports können eine alternative Funktion zur Ansteuerung von Servos annehmen. Erläuterungen dazu finden Sie bei den Extended Functions

Definition einer Interruptroutine

Wird während der Abarbeitung eines Programmes ein Interruptsignal (Low-Flanke) am IRQ-Pin detektiert, wird die momentan bearbeitete BASIC-Anweisung zum nächstmöglichen Zeitpunkt unterbrochen und die Programmabarbeitung an der zuvor mit dem Befehl

INTERRUPT label

festgelegten Stelle fortgesetzt. Der Rücksprung zur Ausgangsposition erfolgt durch den Befehl

RETURN INTERRUPT

Gehen weitere IRQ-Signale ein, während sich der Steuercomputer gerade in der Bearbeitung eines Interrupts befindet, wird maximal ein IRQ-Ereignis gespeichert und im Anschluß an RETURN INTERRUPT ausgeführt. Alle anderen Signale gehen verloren. Die Interrupt-Routine kann durch mehrfachen Aufruf des INTERRUPT-Befehls beliebig oft umdefiniert werden, auch während der Ausführung einer Interruptroutine.

Im folgenden Beispiel wird durch jeden Interrupt eine Leuchtdiode ein/ausgeschaltet:

```
DEFINE led PORT[8]  
led = OFF  
' Festlegen der Interruptroutine  
INTERRUPT switch_it  
' Endlosschleife  
#loop  
GOTO loop  
' Interruptroutine  
#switch_it  
tog led  
RETURN INTERRUPT
```

Befehlsübersicht

Dieses Kapitel gibt einen kompletten Überblick über die CCBASIC Operatoren, Funktionen und Anweisungen.

Mathematische und logische Operatoren

Grundrechenarten: + - * /

Der Modulooperator MOD liefert den Rest einer Integerdivision,

a = 10 MOD 3

ergibt beispielsweise für a den Wert 1.

Vergleichsoperatoren: > (größer als), < (kleiner als), >= (größer oder gleich), <= (kleiner oder gleich), = (gleich), <> (ungleich)

Das Ergebnis einer Vergleichsoperation ist entweder -1 bzw. 255 (Vergleich wahr) oder 0 (Vergleich falsch).

a=10 <3

ergibt beispielsweise für a den Wert 0.

Logische Operatoren:

NOT (Negation), AND (Und-Verknüpfung), NAND (Und-Verknüpfung mit anschließender Negation), OR (Oder-Verknüpfung), NOR (Oder-Verknüpfung mit anschließender Negation), XOR (Exklusiv-Oder-Verknüpfung).

Die logischen Operatoren können außer zur Formulierung von Bedingungen (meist in Verbindung mit Vergleichsoperationen) auch für binäre Byte- oder Wordmanipulationen benutzt werden.

Schiebeoperatoren: SHL (nach links schieben), SHR (nach rechts schieben)

werden zum bitweisen Verschieben von Bitmustern in Byte- oder Wordvariablen benutzt. Links des Operators steht der zu schiebende Wert, rechts die Zahl, um wieviel Bits verschoben werden soll. Beim Linksschieben entspricht jede einzelne Verschiebung einer Multiplikation mit 2, beim Rechtsschieben einer Division durch 2.

a = 10 SHL 3

entspricht also $a=10*2*2*2$ und ergibt beispielsweise für a den Wert 80.

Mathematische Funktionen und Befehle

Die Argumente x und y, je nach Funktion oder Befehl, sind stets Terme (Definition siehe oben).

Die Wurzelfunktion SQR(x) liefert eine Näherung für die Quadratwurzel aus dem Argument x. Dabei werden die Nachkommastellen abgeschnitten.

Die Signumfunktion SGN(x) ergibt 1, wenn der Wert des Argumentes x größer 0, und ergibt -1, wenn der Wert kleiner 0 ist. Für x = 0 ist auch das Ergebnis der SGN-Funktion gleich 0.

Die Maximumfunktion MAX(x,y) ergibt x, wenn x > y ist, sonst y.

Die Minimumfunktion MIN(x,y) ergibt x, wenn x < y ist, sonst y.

Der Befehl RANDOMIZE x initialisiert den internen Pseudo-Zufallsgenerator des Steuercomputers mit dem Wert von x. Ein und derselbe Initialisierungswert führt stets zu einer identischen Folge von Zahlen.

Die Spezialform RANDOMIZE TIMER lädt den Wert des freilaufenden Timers in den Generator.

Die Zufallsfunktion RAND liefert den nächsten Integer-Zufallswert des Pseudo-Zufallsgenerators. Die Zufallszahlen werden nach dem multiplikativen Verfahren mit anschließender Modulodivision (siehe ein gutes Mathematikbuch) aus dem jeweils vorangehenden Wert erzeugt.

Rangfolge von Operatoren und Funktionsaufrufen

Bei der Berechnung von Termen mit Operatoren und Funktionen ist deren Rangfolge von entscheidender Bedeutung. Teilausdrücke mit Operatoren von hohem Rang werden vor denen mit einem niedrigerem Rang berechnet (vergleiche Rechenregel: „Punktrechnung vor Strichrechnung“). Bei gleichrangigen Operatoren erfolgt die Berechnung von links nach rechts. Wie in der Mathematik kann jedoch durch Klammersetzung zusätzlich Einfluß auf die Berechnungsreihenfolge genommen werden. CCBASIC unterstützt maximal 3 Klammerebenen. Im Sinne der Übersichtlichkeit eines Programmes sollten jedoch „wilde“ Klammerausdrücke vermieden und komplexe Berechnungen auf mehrere BASIC-Zeilen aufgeteilt werden.

Die folgende Liste zeigt die CCBASIC Operatorenrangfolge :

Rang Operatoren

9	()
8	Funktionsaufrufe
7	negatives Vorzeichen
6	* / MOD SHL SHR
5	+ -
4	> >= < <= = <>
3	NOT
2	AND
1	OR XOR

Achtung:

Die C-Control hat die NAND und NOR Operatoren nicht. Diese können aber durch eine anschließende Negation gebildet werden:

Anweisungen zur Steuerung des Programmflusses

Schleife

FOR variable = anfang TO ende STEP schrittweite

...

NEXT

Die FOR-Schleife führt die Anweisungen bis zum NEXT solange aus, bis der Wert der Variable gleich dem Wert des Terms ende ist. Vor dem ersten Durchlauf wird der Wert des Terms anfang berechnet und der Schleifenvariablen zugewiesen. In jedem Durchgang wird der Wert des schrittweite-Terms zur Schleifenvariablen addiert. In der Form

FOR variable = anfang TO ende

...

NEXT

beträgt die Schrittweite konstant 1. Die Werte des ende-Terms und des schrittweite-Terms werden mit jedem Schleifendurchlauf neu berechnet. Das gestattet eine erweiterte Kontrolle des Programm-Verlaufes. FOR-Schleifen können ineinander verschachtelt werden. Die Verschachtelungstiefe ist nur durch den für die Schleifenvariablen erforderlichen Speicherplatz beschränkt.

FOR v1 = anfang1 TO ende1

FOR v2 = anfang 2 TO ende2

FOR v3 = anfang3 TO ende3

...

NEXT

NEXT

NEXT

Jede FOR-Schleife darf im Verlauf des Programms nur über ihre eigene NEXT-Anweisung laufen. Folgender Quelltext kann zwar compiliert und in den Steuercomputer geladen werden, wird jedoch nicht wie vielleicht erwartet funktionieren:

FOR v1 = anfang1 TO ende1

...

GOTO anothernext

...

NEXT

FOR v2 = anfang 2 TO ende2

...

#anothernext

NEXT

Achten Sie außerdem auf den Wertebereich von Schleifenvariable und ende-Term!

DEFINE v BYTE

FOR v = 1 TO 1000

...

NEXT

wird zu einer Endlosschleife, da v als Bytevariable nie den Wert 1000 erreichen kann, sondern bereits nach 255 wieder auf 0 überrollt.

Bedingte Ausführung

IF bedingung THEN anweisung

Oder

IF bedingung THEN anweisung1 ELSE anweisung

Die IF...THEN...ELSE-Konstruktion ermöglicht die Anpassung des Programmflusses an Bedingungen zur Laufzeit des Programms. Als Bedingung ist ein beliebiger Term einzusetzen. Ergibt dessen Berechnung einen Wert ungleich 0, dann gilt die Bedingung als erfüllt, und die an-Weisung1 wird ausgeführt. Werden zusätzlich ein ELSE und eine zweite Anweisung angegeben, so wird diese Anweisung alternativ ausgeführt, wenn der berechnete Term einen Wert gleich 0 ergibt. Die gesamte

IF...THEN...ELSE-Konstruktion muß in einer Quelltextzeile stehen. Anweisungsblöcke (mehrere Anweisungen) nach THEN und ELSE sind nicht zulässig.

Sprunganweisung

GOTO label

Mit der GOTO-Anweisung kann der Steuercomputer veranlaßt werden, die Programmabarbeitung an einer bestimmten Stelle fortzusetzen. Als Ziel des Sprungs wird ein Label-Bezeichner angegeben. Das Sprungziel kann sich vor oder nach der GOTO-Anweisung im Quelltext befinden.

Aufruf und Rückkehr aus einer Unterroutine

Der Aufruf einer Unterroutine erfolgt mit der Anweisung

GOSUB label

Dabei ist label der Anfangspunkt der Unterroutine.

In den sogenannten UnterROUTINEN sind Programmabschnitte zusammen gefaßt, die mehrfach im Verlauf der Programmabarbeitung benötigt werden. Eine Unterroutine beginnt stets mit einem Label, enthält dann eine oder mehrere Anweisungen und abschließend ein RETURN. Nach dem RETURN wird die Programmabarbeitung mit der Anweisung nach dem GOSUB fortgesetzt. Die Programmabarbeitung darf ohne ein vorheriges GOSUB niemals an eine RETURN-Anweisung gelangen. Die maximal zulässige Verschachtelungstiefe bei Aufrufen von UnterROUTINEN aus UnterROUTINEN ist zwölf, hier ist ein Beispiel von vier Verschachtelungen gezeigt:

```
.  
#hauptprogramm  
GOSUB sub1  
...  
#sub1  
GOSUB sub2  
...  
RETURN  
#sub2  
GOSUB sub3  
...  
RETURN  
#sub3  
GOSUB sub4  
...  
RETURN  
#sub4  
...  
RETURN
```

Wertgesteuerte Programmverzweigung

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

In Abhängigkeit des Wertes des Selektors variable erfolgt eine Programmverzweigung oder ein UnterROUTINENaufruf zu den aufgelisteten Einsprungpunkten. Ist der Wert 0, dann wird zu label0 verzweigt, bei Wert gleich 1 zu label1 usw. Ist der Variablenwert negativ oder größer als die Anzahl der aufgeführten Sprungziele, dann wird die Programmabarbeitung ohne Verzweigung fortgesetzt.

Programmende

Gelangt der Steuercomputer im Verlauf der Programmabarbeitung zur END-Anweisung, wird die Programmabarbeitung beendet. Das System verharrt dann in einem inaktiven Zustand dass durch einen RESET beendet werden muss. Danach kann ein neues Anwenderprogramm übertragen oder die Ausführung per Start-Taster wieder gestartet werden.

Verzögerung des Programmflusses

WAIT conditionterm

unterbricht die Programmausführung solange, bis die Berechnung des conditionterm einen Wert ungleich 0 ergibt.

define key port[9]

...

WAIT key

In diesem Beispiel wird solange gewartet, bis vom Digitalport 9 ein HIGH-Pegel (= logisch 1) gelesen wird.

PAUSE

Befehl unterbricht die Programmausführung für eine gewisse Zeit. Der berechnete Wert des Parameterterms geht als Multiplikationsfaktor mit der Grundeinheit von 20 Millisekunden in die Festlegung der Pausenzeit ein.

PAUSE term

Beispielsweise wird durch den Befehl **PAUSE 50** die Programmausführung für ca. 50*20 Millisekunden = 1 Sekunde unterbrochen. Die maximale Zeitabweichung der tatsächlichen Pause vom angegebenen Wert beträgt dabei Prinzip bedingt + 20 Millisekunden. Der maximale Wert ist

PAUSE 32767

und dauert etwa 655 Sekunden, also etwa 10 Minuten

Kommunikation über die serielle Schnittstelle

Datenausgabe

Die Datenausgabe erfolgt als Text über die serielle Schnittstelle des C-Control BASIC Steuer-Computers. Ist über ein Schnittstellenkabel zum Beispiel ein PC mit einem Terminalprogramm angeschlossen, können die ausgegebenen Daten dort angezeigt werden.

PRINT term

gibt das Ergebnis der Berechnung von term aus.

PRINT "text"

überträgt den in Anführungszeichen stehenden Text. In beiden Fällen wird an die Übertragung ein Zeilenvorschubzeichen angehängt, welches das Terminalprogramm veranlaßt, die nächste Ausgabe in der nächsten Bildschirmzeile vorzunehmen. Der Zeilenvorschub kann unterdrückt werden, wenn dem PRINT-Befehl nach dem Parameter (term oder „text“) ein Semikolon hinzugefügt wird.

PRINT term;

PRINT "text";

CCBASIC unterstützt außerdem mehrere Ausgaben mit einem PRINT-Befehl, wobei die einzelnen Parameter durch Komma oder Semikolon getrennt werden. Ein Komma fügt in die Ausgabe ein Tabulatorzeichen ein, das entsprechend den Einstellungen im Terminalprogramm als eine Anzahl von Leerzeichen am Bildschirm erscheint. Sollen zwei Ausgaben ohne Zwischenraum aufeinander folgen, so sind diese im PRINT-Befehl durch ein Semikolon zu trennen.

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

Ein einzelner PRINT-Befehl ohne Parameter gibt nur einen Zeilenvorschub aus.

Der PRINT-Befehl wird bei den neuen Versionen der C-Control benutzt Schnittstellenausgaben auf verschiedene interne System-Module umzuleiten z.B. um mit **PRINT"text"** direkt auf ein LCD zu schreiben. Diese Umleitung wird mit dem Befehl

print"#ON_MODULE#"

aktiviert. Deshalb darf in einem PRINT-Befehl welcher der seriellen Schnittstelle gilt niemals das Doppelkreuz # als Zeichen stehen.

Print"Ausgabe #1" ist also verboten

Einzelheiten über die Umleitung des PRINT-Befehl finden sie im Kapitel Extended Functions

Dateneingabe

Mit dem Befehl

INPUT variable

kann ein Integerwert von der seriellen Schnittstelle gelesen und für die anschließende Weiterbearbeitung in einer Variablen gespeichert werden. Der Wert wird in einem Terminalprogramm an einem PC eingegeben und nach dem Drücken der ENTER-Taste per Schnittstellenkabel an den C-Control/BASIC Steuercomputer übertragen. Der INPUT-Befehl wartet solange, bis eine komplette Datenübertragung vom Terminal empfangen wurde. Wird der INPUT-Befehl aufgerufen, ohne daß eine Datenübertragung vom Terminal erfolgt, wird das Programm endlos an dieser Stelle stehen bleiben! Hier hilft dann nur noch der Reset-Taster und der anschließende Neustart des C-Control Gerätes.

Byteweise Kommunikation über die serielle Schnittstelle

Während PRINT und INPUT kurze Zeichenketten zur Darstellung eines numerischen Wertes senden beziehungsweise erwarten, kann es wünschenswert sein, einzelne Bytes seriell zu übertragen. Dafür bietet CCBASIC die Befehle PUT und GET.

PUT term

sendet den berechneten Wert eines Terms. Falls erforderlich, wird das Ergebnis zuvor auf den Byte-Wertebereich (0...255) reduziert.

GET variable

wartet auf ein seriell empfangenes Byte und speichert den Wert dann in der angegebenen Variablen.

Weitere Schnittstellenbefehle und -funktionen

Wie beschrieben warten INPUT und GET unter Umständen endlos auf den Empfang serieller Daten. Soll ein „Aufhängen“ des Programms in dieser Art verhindert werden, kann vor jedem von INPUT oder GET durch Aufruf der Statusfunktion RXD ermittelt werden, ob empfangene Daten zur Verfügung stehen. Die Funktion liefert in diesem Fall den Wert -1. Ist der Schnittstellenpuffer leer, so ist das Funktionsergebnis gleich 0.

if RXD then GET thebyte

Die voreingestellte Übertragungsrate der seriellen Schnittstelle beträgt für Sender und Empfänger 9600 Bit pro Sekunde (baud). Mit dem BAUD Befehl können jedoch auch andere Raten eingestellt

werden. CCBASIC enthält dafür einige vordefinierte Konstanten: R1200, R2400, R4800, R9600 für die Raten 1200 bis 9600 Bit pro Sekunde.

BAUD R2400

schaltet beispielsweise Sender und Empfänger auf die Rate von 2400 Bit pro Sekunde um.

Die weiteren Schnittstellenparameter - 8 Datenbits, kein Paritätsbit, 1 Stopbit - sind fest und können nicht geändert werden. Die Handshake-Signale RTS und CTS werden in der aktuellen Version nicht verwendet und sind in CCBASIC-Programmen nicht ansprechbar.

Dateifunktionen

Die Dateifunktionen erlauben das Aufzeichnen von Messwerten oder anderen Daten oder können zum Abspeichern von Information benutzt werden, die nach Ausfall der Betriebsspannung wieder in die Programm-variablen geladen werden sollen. Der Speicherbereich im FLASH Speicher nach dem Anwenderprogramm –meist der größte Teil - steht für diesen Zweck zur Verfügung. Der Speicher-Bereich wird als eine Datei verwaltet, auf die lesend oder schreibend zugegriffen werden kann, nachdem sie mit dem entsprechenden Attribut geöffnet wurde. Der Befehl zum Öffnen der Datei lautet wie folgt:

OPEN# FOR WRITE
OPEN# F OR APPEND
OPEN# FOR READ

Dabei bedeutet WRITE das Öffnen zum Schreiben mit Überschreiben eventueller alter Aufzeichnungen,

APPEND das Öffnen zum Schreiben mit Anhängen der neuen an die alten Aufzeichnungen und READ das Öffnen zum Auslesen der Aufzeichnungen.

Es können nur Integerwerte gespeichert und gelesen werden. Jeder Wert belegt also 2 Bytes im Speicher. Das Schreiben und Lesen erfolgt mit den Befehlen

PRINT# term

wobei das berechnete Ergebnis des Terms gespeichert wird, beziehungsweise

INPUT# variable

,wobei variable eine definierte Integervariable des Programms bezeichnet. Schreiben in und Lesen aus der Datei erfolgt streng sequentiell. Dafür wird intern ein Dateizeiger verwaltet, der nach jedem Zugriff um 1 erhöht wird. Vor jedem Schreiben sollte geprüft werden, ob noch genügend Platz im Speicher zur Aufnahme der Daten vorhanden ist. Dafür kann die Funktion FILEFREE abgefragt werden, die als Ergebnis die Größe des noch freien Speichers liefert (in Words). Folgendes Beispiel zeigt die Anwendung der Funktion

```
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
```

Vor jedem Lesen sollte geprüft werden, ob noch weitere Daten aufgezeichnet sind. Die Funktion dafür lautet EOF („end of file“). Ihr Ergebnis ist -1, wenn in der Datei keine weiteren Daten verfügbar sind,

sonst 0. Die Abfrage der EOF Funktion sollte das Auslesen von Datenblocks in gleicher Weise rahmen, wie beim Schreiben der Daten.

IF NOT EOF THEN GOSUB readblock

```
...  
#readblock  
INPUT# a  
INPUT# b  
INPUT# c  
RETURN
```

Nach Beenden eines Dateizugriffs sollte die Datei sofort wieder geschlossen werden. Erst dann sind die Daten vor einem Spannungsausfall oder Reset des Systems sicher. Der Befehl dafür lautet

CLOSE#

und hat keinen Parameter.

Portbefehle

der Umschaltbefehl TOG

Prinzipiell erfolgt der Zugriff auf die Ports des Steuercomputers wie auf Variablen. Um einen Digitalport P einzuschalten, schreibt man

P=I oder **p=on**

P=O oder **p=off**

, um ihn auszuschalten. Um den Port umzuschalten (EIN nach AUS; AUS nach EIN), kann man schreiben

P = NOT P

oder den Befehl

TOG P

benutzen. TOG steht für englisch „toggle“. Der TOG Befehl benötigt weniger Platz im Speicher und wird schneller als die klassische NOT-P-Konstruktion ausgeführt. Die Portvariable P darf beim TOG Befehl nur für einen einzelnen Digitalport stehen, nicht für einen Byte- oder Wordport und nicht für einen Extended Port (sehen Sie dazu auch die Ausführungen im Kapitel Extended Functions)

Deaktivieren eines Ports mit DEACT

Sobald einer Portvariablen erstmalig ein Wert zugewiesen wird, schaltet der Steuercomputer die zugehörigen Hardwarestrukturen im Prozessorchip (Transistoren) auf Ausgangsbetrieb. Es fließt also entsprechend der angeschlossenen Schaltung Strom aus bzw. in den Prozessor (max. 10 mA zulässig!). Der Befehl

DEACT portvar

deaktiviert den angegebenen Port. Das heißt, der Port wird in einen hochohmigen Zustand geschaltet und arbeitet im Eingangsbetrieb. Der DEACT Befehl darf auf einzelne Digitalports oder Byteports angewendet werden, nicht jedoch auf die Extended Ports, da diese durch einen PCF 8574 gebildet werden und sich seine Ports nicht deaktivieren lassen (sehen Sie dazu auch die Ausführungen im Kapitel Extended Functions)

der PULSE Befehl

Mit dem Befehl

PULSE portvar

wird ein Puls von einigen Mikrosekunden Breite am mit portvar bezeichneten Port ausgegeben. Das ist beispielsweise nützlich zum Schalten extern angeschlossener flankengetriggelter Logikschaltkreise. Steht der Port vor Ausführung des PULSE Befehls auf low (=0), wird ein High-Puls (O-I-O), ansonsten ein Low-Puls (I-O-1) ausgegeben. Die Portvariable darf beim PULSE Befehl nur für einen einzelnen Digitalport stehen, nicht für einen Byte- oder Wordport und nicht für einen Extended Port

Definition und Anwendung von Datentabellen

Im Standard-BASIC dienen DATA-Zeilen zum Ablegen von konstanten Datenblöcken, auf die dann sequentiell zugegriffen werden kann. CCBASIC unterstützt keine DATA-Zeilen, bietet jedoch ein weitaus flexibleres Werkzeug zur Definition und zum Zugriff auf Datenblocks. Konstante Daten können in Form von Tabellen abgelegt werden. Jede Tabelle bekommt einen Bezeichner (tablename) zugewiesen und kann beliebig viele Einträge enthalten, soweit der Programmspeicher Platz bietet. Jeder Dateneintrag (Cx) wird als Integerwert abgelegt und belegt somit zwei Bytes. Dabei können die Daten direkt im Quelltext aufgeführt werden

```
TABLE tablename CO C1 C2 C3 ...  
..c4 c5 . . . . Cn  
TABEND
```

oder vom CCBASIC-Compiler aus einer externen Textdatei importiert werden

```
TABLE tablename "tabfilename"
```

Die Tabellendefinitionen müssen stets am Ende eines Programms, hinter dem END Befehl stehen, da die Daten nahtlos hinter den vorangehenden Codebytes im EEPROM- Speicherchip abgelegt werden. Die Programmabarbeitung darf nie über Tabellendaten laufen, da die Daten sonst als BASIC Befehle interpretiert werden würden, was sicher zum Absturz des Systems führt. Der Zugriff auf die Tabellendaten erfolgt mit dem Befehl

```
LOOKTAB tablename,index,variable
```

tablename bezeichnet eine gültige Tabelle, für index kann ein beliebiger Term stehen und die variable bezeichnet die Speicherzelle, in der das Ergebnis abgespeichert werden soll. Der berechnete Wert des index-Terms darf nicht negativ sein und maximal N-1 betragen, wenn die indizierte Tabelle N Einträge hat. Ergibt index den Wert 0, so wird CO in der angegebenen Variablen gespeichert, für index gleich 1 C1 und so weiter. Folgendes Beispiel gibt den Inhalt einer Tabelle seriell aus

```
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
```

Am Bildschirm des Terminalprogramms sollte erscheinen

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

Besonders nützlich erweisen sich Tabellen beim Umsetzen von A/D-Werten in echte physikalische Größen. Eine Umsetzungstabelle hat dann in der Regel 256 Einträge. Der gemessene A/D-Wert geht dann als Tabellen-Index in die Bestimmung der physikalischen Größe ein.

Zugriff auf die Echtzeituhr

Um den Stand der internen Echtzeituhr auszulesen und zu setzen, sind folgende Systemvariablen definiert:

YEAR Jahr (0...99)
MONTH Monat (1...12)
DAY Tag des Monats (1...31)
DOW Wochentag (7=Sonntag...1=Montag)
HOURL Stunde (0...23)
MINUTE Minute (0...59)
SECOND Sekunde (0...59)

Beachten Sie bitte, daß während des Zugriffs die interne Uhr weiterläuft. Der Sekundenwert sollte daher stets zuerst ausgelesen werden. Steht er auf 59, so muß nach dem Lesen der letzten interessierenden Zeitinformaton (z.B. YEAR) der Sekundewert nochmals gelesen und auf =0 getestet werden. In diesem Fall ist das Auslesen der Echtzeituhr zu wiederholen, da eine neue Minute angebrochen ist (Extremfall Silvester mit Weiterschalten aller Stellen in Uhr und Datum). Die Jahreszahl wird im C-Control System nur zweistellig abgespeichert..

Timer

Der interne 20-Millisekunden-Timer kann über die Systemvariable **TIMER** ausgelesen werden. Der Timer ist freilaufend und kann gestellt und rückgesetzt werden. Nach Programmstart steht der Timer auf Null und wird alle 20 ms um eins erhöht.

Der Timer ist eine Wortvariable und zählt nur bis 32767 danach wird der **TIMER** angehalten. Das findet etwa nach 10 Minuten statt. Sie können dem Timer zu beliebigen Zeiten andere Werte zuweisen, also auch löschen. Dazu werden der Variablen die entsprechenden Werte zugewiesen.

TIMER=0 Löscht den Timer

Ausgabe von Tönen mit BEEP

Der C-Control BASIC Steuercomputer kann an einem seiner Pins Töne als Rechteckschwingungen ausgeben. Der Befehl dazu lautet

BEEP ton, tTon, tPause

Für die drei Parameter können Konstanten oder Terme eingesetzt werden. Dabei bestimmt ton die Tonhöhe, tTon bestimmt die Dauer des Tons und tPause die Pause nach dem Ton. Die Einheit für die Zeitangaben beträgt 20 Millisekunden. Der Befehl

BEEP 6, 10, 3

gibt also für $10 \cdot 20 = 200$ Millisekunden einen Ton von etwa 2500 Hz aus und macht danach eine Pause von $3 \cdot 20 = 60$ Millisekunden. Wenn nach einem BEEP kein weiterer BEEP folgt, kann die Pause auch auf 0 gesetzt werden.

Der Frequenzbereich der mit BEEP erzeugt wird erstreckt sich von ca. 7kHz bis 100 Hz
Werte für ton kleiner als 2 sollten sie nicht verwenden, da wegen der häufigen durch BEEP ausgelösten Interrupts das System beginnt instabil zu werden und sich "aufhängen" könnte, vor allem wenn gleichzeitig andere Interrupts (wie in FREQ auslöst) aktiv sind.

ton=2	7500 Hz
ton=3	5000 Hz
ton=4	3700 Hz
ton=5	3000 Hz
ton=6	2500 Hz

Sie sehen dass die Abstufungen mit zunehmendem Wert für Ton immer feiner werden und ab einem Wert für Ton von etwa 60 ist die Abstufung auf wenige Herz gesunken. Beachten Sie bitte, dass einige Funktionen des Erweiterten Betriebssystems (z.B. zur IR-Kommunikation) höhere Interrupt-Prioritäten haben und einen BEEP vorzeitig abschalten können

Frequenzmessung mit der Funktion FREQ

Ist am DCF77-Eingang (FREQ1) keine Aktivantenne angeschlossen, so kann mit diesem Eingang alternativ eine Frequenzmessung erfolgen, deren Ergebnis mit der Funktion FREQ jederzeit abgefragt werden kann.

x = FREQ

Die Frequenzmessung basiert auf dem Pulszählprinzip bei einer Torzeit von 1 Sekunde. Die Messung erfolgt ständig im Hintergrund, parallel zur Abarbeitung des BASIC-Programms. Der Messbereich reicht bis etwa 5 Kilohertz mit einem Messfehler unter einem Prozent. Danach wird das Ergebnis zunehmend ungenauer.

Für Messungen kleinere Frequenzen oder zur Zählung von Ereignissen kann FREQ alternativ konfiguriert werden. Sie finden darüber mehr im Kapitel Extended Functions

SLOWMODE

Eine wesentliche Voraussetzung für Kompatibilität zu Programmen für die C-Control der älteren Generation ist auch die Geschwindigkeit. Dies ist vor allem dann wichtig, wenn Laufzeiten für Verzögerungen benutzt wurden, oder z.B. Portfunktionen auch ohne Bremse für die Peripherie langsam genug waren. Ein typisches Beispiel ist der BASIC-LCD Treiber für die ältere Generation der C-Control. Er läuft auf den neuen Versionen der C-Control bereits zu schnell für das LCD. Um diese Programme auch ohne große Änderungen an die neuen C-Control Versionen anzupassen gibt es den zweistufigen SLOWMODE Befehl.

SLOWMODE ON

Die erste Stufe bremst die Geschwindigkeit auf etwa das Niveau der älteren C-Control M 1.1 und ist konstant etwa 600 Instruktionen/ sek für die UNIT M 1.2 Die UNIT M 2.0 läuft hier mit etwa 2500 I/s

SLOWMODE ON:SLOWMODE ON

Die zweite Stufe ist (wie ursprünglich auch) der sehr langsame Stromsparmodus. Hier hängt die Geschwindigkeit außerdem von der Häufigkeit interner/externer Interrupts ab, da pro Interrupt jeweils nur ein BASIC-Befehl abgearbeitet wird.

Läuft die C-Control M-UNIT 1.2 im default PLM-Mode der DACs ist die Geschwindigkeit etwa 700 Instruktionen/ sek (ohne weitere externe Interrupts) Die UNIT M 2.0 erreicht hier etwa 2800 I/s

Im Servo-Mode werden nur mehr etwa 40 Instruktionen/sek von der UNIT 1.2 erreicht, weil hier die Interruptrate sehr gering ist. Die UNIT 2.0 liegt bei etwa 100 I/s

Bedingt durch die hohe Stromaufnahme des Oszillators (etwa 30 mA) ist die Stromersparnis bei der Unit M 2.0 nicht bedeutend.

SLOWMODE OFF

Schaltet den BASIC-Computer wieder auf max. Geschwindigkeit.

Einbinden von Systemtreibern

Die meisten anwendungstechnischen Probleme lassen sich sicher ausschließlich durch ein BASIC-Programm lösen. Dennoch kann es vorkommen, daß für eine spezielle Aufgabe eine höhere Verarbeitungsgeschwindigkeit oder besondere Hardwarezugriffe erforderlich sind. Für diesen Fall sind die letzten beiden Seiten im FLASH (zus. 512 Bytes)des Controllers nutzbar um spezielle Treiber zu laden. Der dafür benötigte Speicherplatz steht dann aber nicht mehr für BASIC Programme zur Verfügung. Der Anwender muss, wenn solche Treiber geladen sind, selbst Sorge tragen, dass sein BASIC Programm nicht so lang ist, dass es diese Treiber überschreibt.

Während bei den älteren C-Control Varianten das Betriebssystem unlöschar im Prozessor verankert war, ist es bei der neuen Generation der C-Control Computer löschar (FLASH). Aus diesem Grund können nur Treiber geladen werden die von Conrad Elektronik zur Verfügung gestellt werden. Damit wird verhindert dass bei eigenen Experimenten mit Assemblerprogrammen versehentlich das Betriebssystem gelöscht wird.

Beschreibungen zum Laden und Ansprechen von Systemtreibern finden sie in den Beispielen zu diesen Treibern.

CCBASIC – Extended Functions

Die Extended Functions sind spezielle Module des Betriebssystems die den Funktionsumfang Ihrer C-Control gegenüber älteren Versionen erweitern.

Es sind folgende Module im Betriebssystem implementiert.

- CONFIG MODULE	zur Konfiguration von alternativen Portfunktionen und anderen Optionen
- LCD MODULE	zum direkten Schreiben auf das LCD mit Formatierungshilfen
- IR MODULE	zur IR-Kommunikation
- IIC MODULE	zur Kommunikation mit I2C-Bus Bausteinen und Geräten
- RF MODULE	zur Kommunikation über 433 MHz Sender/Empfänger
- STACK MODULE	zur Nutzung eines Variablen-Stacks
- EXTENDED PORTS	zur Porterweiterung auf PCF 8574 I2C-Bus Basis

Wie Sie bereits bei den Standard Functions beim PRINT-Befehl gesehen haben, ist das Doppelkreuz # innerhalb einer PRINT"string" Anweisung für die Benutzung der Extended Functions reserviert.

PRINT"#ON_MODULNAME#";

Leitet alle Schnittstellen-Befehle temporär auf das jeweilige Modul um.

!!! Wichtig ist die Schreibweise in Grossbuchstaben und das Semikolon danach !!!

Ist eine Umleitung aktiviert können an das Modul Kommandos gesendet werden sofern es das jeweilige Modul erfordert.

PRINT"#COMMAND#";

Daten an das aktive Modul werden in der herkömmlichen Art mit

PRINT"abcd";

PRINT a;

PUT a

GET a

übergeben erfordern jedoch bei PRINT immer ein Semikolon danach.

Die Umleitung wird aufgehoben, d.h. auf die serielle Schnittstelle zurückgestellt mit

PRINT"#OFF#";

Eine Umleitung muss immer aufgehoben werden, bevor eine andere Umleitung aktiviert wird.

Das CONFIG MODULE

Das Konfigurations-Register bietet folgende Optionen.

Bit 0	Schaltet beide PWM-DACs in den SERVO-Mode
Bit 1	Schaltet den Frequenzzähler 1 in den EREIGNISZÄHLER Mode
Bit 2	Schaltet den Frequenzzähler 2 in den EREIGNISZÄHLER Mode
Bit 3	Aktiviert die PULLUP-Widerstände an PORT 1 bis 8
Bit 4	Aktiviert die PULLUP-Widerstände an PORT 9 bis 15
Bit 5	Zeigt an wenn die interne Uhr mit DCF77 synchronisiert wurde
Bit 6	Signalisiert einen Fehler bei der I2C-Kommunikation
Bit 7	Zeigt den Zustand der Start-Taste

Die jeweilige Funktion wird ausgeführt wenn das zugehörige Bit gesetzt ist

Demos zu den einzelnen Konfigurationen finden sie bei den Beispielen zu den EXTENDED FUNCTIONS

PRINT"#ON_CONFIG#";

Aktiviert die Umleitung auf das CONFIG MODULE.

Daten werden an das Modul mit dem gewöhnlichen PUT-Befehl übergeben.

PUT &B00000001

Das CONFIG MODULE kann natürlich auch gelesen werden, z.B. um nur ein bestimmtes Bit zu manipulieren und alle anderen unverändert zu lassen. Das Beispiel liest das Register, setzt das Bit 0 und schreibt den neuen Wert auf das Register zurück.

GET a

PUT (a or 1)

Das LCD MODULE

Es ermöglicht Ihnen eine komfortable Ausgabe auf das LCD in dem die PRINT-Funktion auf das LCD umgeleitet wird. Zusätzlich stehen Ihnen einige Kommandos zur einfachen Formatierung der Ausgabe zur Verfügung. Die Umleitung auf das LCD wird mit

PRINT"#ON_LCD#";

aktiviert. Danach können Sie mit PRINT oder PUT auf das LCD schreiben bzw. Kommandos zur Formatierung übermitteln.

PRINT"#INIT#";	-> Initialisierung des LCD
PRINT"#CLR#";	-> LCD Löschen
PRINT"#L1nn#";	-> Schreiben in Zeile 1 Position nn (immer zweistellig)
PRINT"#L2nn#";	-> Schreiben in Zeile 2 Position nn (immer zweistellig)
PRINT"#SR#";	-> Scroll right
PRINT"#SL#";	-> Scroll left
PRINT Var;	-> Schreibt den Wert der Variable ins LCD
PRINT "xyz";	-> Schreibt einen String „xyz“ ins LCD

Das IR MODULE

Das IR-MODULE unterstützt das Format RC5. Das Format wird von den meisten Geräten der Unterhaltungsindustrie verwendet und ermöglicht es Ihnen diese Geräte mit dem BASIC-Computer zu steuern oder Ihre Anwendung mit einer üblichen IR-Fernsteuerung zu bedienen. Das RC5 Format besteht aus einer Geräteadresse und einem Kommando.

RC5 FORMAT:

13-12-11-10-09-08-07-06-05-04-03-02-01-00 DATA BIT
S S T a4 a3 a2 a1 a0 c5 c4 c3 c2 c1 c0 RC5

Das IR-Module erwartet an Port 2 den Ausgang eines IR-Empfängers (z.B. TSOP 4836) und an Port 3 eine IR-LED als Datensender. Wollen Sie nur Daten empfangen, so ist Port 3 zur freien Verfügung. Ein fertiges Sender/Empfänger-Modul gibt es als Zubehör zum Application-Board 2.0 (für die M-UNIT 2.0)

Ist die Umleitung auf das IR MODULE mit

PRINT"#ON_IR#";

aktiviert, so können sie mit PUT u. GET Daten senden bzw. empfangen..

PUT adr

PUT cmd

Übergibt zunächst die Adresse des angesprochenen Gerätes an das IR MODULE, danach das Kommando an dieses Gerät. Haben sie beide Werte übergeben, wird der RC5 Datenrahmen gesendet.

GET adr
GET cmd

GET holt hier zunächst die Adresse eines empfangenen Datenrahmens und danach das Kommando, das an das Gerät an dieser Adresse gerichtet war, aus dem Datenpuffer. Wurde nichts empfangen, oder wurde der Puffer geleert, ist der Inhalt des Puffers jeweils 255. Der Empfang selbst geschieht interruptgesteuert im Hintergrund

Das IR MODULE ist weiterhin empfangsbereit, auch wenn sie die Umleitung zeitweise abschalten (um z.B. eine Ausgabe auf das LCD zu machen).

Für das Lesen oder Schreiben von Daten muss die Umleitung aber auf jeden Fall wieder aktiviert werden.

Werte für Adresse oder Kommando werden auf die max. zulässige Bitlänge reduziert.

Achtung:

Das IR MODULE ist nicht mit dem RF-MODULE kombinierbar, da sie gleiche Ports nutzen

Das IIC MODULE

Das IIC MODULE erleichtert es an den IIC-BUS angeschlossene Komponenten anzusprechen bzw. Daten über den IIC-BUS zu lesen oder zu schreiben. Kenntnisse über die angeschlossenen Komponenten muss der Anwender jedoch haben, da die Details im Protokoll gerätespezifisch sind.

Der I2C-Bus liegt an PORT 9 (SDA) und PORT 10 (SCL)

PRINT"#ON_IIC#";

Aktiviert die Umleitung und das Protokoll für das angeschlossene Gerät wird aus folgenden Befehlen gebildet:

PRINT"#START#";	-> Sendet die I2C-Bus START Bedingung
PUT Wert	-> Sendet ein Byte über den Bus
GET Wert	-> Empfängt ein Byte über den Bus
PRINT"xyz";	-> Sendet einen String über den Bus z.B. wenn ein LCD angeschlossen ist
PRINT"#STOP#";	-> Sendet die I2C-Bus STOP Bedingung

Das RF MODULE

Das verwendete Datenformat ist du den gebräuchlichsten Geräten der Haushaltstechnik kompatibel, obwohl es hier keinen Standard gibt. Es wird eine 8Bit lange Geräteadresse verwendet (die beim Empfänger entsprechend eingestellt sein muss) und ein 4Bit langes Kommando um z.B. ein Relais am Empfänger zu schalten. Jeder Datenrahmen wird 1 mal gesendet und benötigt dafür rund 12ms.

Eine angehängte Pause von 10ms macht es einfach, vier dieser Rahmen im erforderlichen Zeitlichen Abstand zu senden um dem gebräuchlichen Protokoll zu entsprechen.

Das ermöglicht es Ihnen diese Geräte mit dem BASIC-Computer zu steuern oder Ihre Anwendung mit einer üblichen 433 MHz Funk-Fernsteuerung zu bedienen.

FORMAT:

11-10-09-08-07-06-05-04-03-02-01-00 DATA BIT
c3 c2 c1 c0 a7 a6 a5 a4 a3 a2 a1 a0 ADRESSE/KOMMANDO

Das RF-Module erwartet an Port 2 den Ausgang eines 433MHz-Empfängers (z.B. das als Zubehör erhältliche Modul) und an Port 3 einen 433MHz Sender, der ebenfalls als Zubehör zum Application-Board 2.0 erhältlich ist.

PRINT"#ON_RF#";

aktiviert, so können sie mit PUT u. GET Daten senden bzw. empfangen.

PUT adr
PUT cmd

Übergibt zunächst die Adresse des angesprochenen Gerätes an das IR MODULE, danach das Kommando an dieses Gerät. Haben sie beide Werte übergeben, wird der Datenrahmen gesendet.

GET adr
GET cmd

GET holt hier zunächst die Adresse eines empfangenen Datenrahmens und danach das Kommando, das an das Gerät an dieser Adresse gerichtet war, aus dem Datenpuffer. Wurde nichts empfangen, oder wurde der Puffer geleert, ist der Inhalt des Puffers jeweils 255. Der Empfang selbst geschieht interruptgesteuert im Hintergrund

Das RF MODULE ist weiterhin empfangsbereit, auch wenn sie die Umleitung zeitweise abschalten (um z.B. eine Ausgabe auf das LCD zu machen).

Für das Lesen oder Schreiben von Daten muss die Umleitung aber auf jeden Fall wieder aktiviert werden.

Werte für Adresse oder Kommando werden auf die max. zulässige Bitlänge reduziert.

Achtung:

Das RF-MODULE steht nur bei UNIT M 2.0 zur Verfügung

Das RF MODULE ist nicht mit dem IR-MODULE kombinierbar, da sie gleiche Ports nutzen

Das STACK MODULE

Das Stack Module stellt dem Benutzer 10 weitere Bytes zur zeitweiligen Speicherung von Byte-Variablen zur Verfügung z.B. zur Variablenübergabe an Unterprogramme

PRINT"#ON_STACK#";

Aktiviert die Umleitung von PUT und GET auf das Stack

PUT Wert -> Legt ein Byte auf das Stack und erhöht den Stackpointer, so dass er
Auf den nächsten Speicherplatz zeigt.

GET Wert -> Holt ein Byte vom Stack und dekrementiert den Stackpointer, so dass

Die Ablage/Wegnahme auf das Stack erfolgt streng sequenziell nach dem Prinzip

FIRST IN – LAST OUT

Beispiel:

PUT a

PUT b

GET a

GET b

Tauscht die Inhalte der Variablen a und b

Extended Ports

Über den I2C-Bus des BASIC-Computers können bis zu acht Porterweiterungen PCF8574 und weitere acht PCF8574A adressiert werden. Jeder dieser Bausteine stellt dem Benutzer acht digitale Ein/Ausgabeports zur Verfügung. Es können auf diese Weise also bis zu 16 Byteports mit insgesamt 128 Bitports gebildet werden. Jeder dieser Bausteine hat eine vom Benutzer festlegbare 3 Bit Adresse. Jeder dieser 8 möglichen Adressen ist im Betriebssystem ein BYTEPORT zugeordnet. Diese Extended Ports werden genauso angesprochen wie die Digitalports der CPU, jedoch müssen Sie natürlich wissen, welcher PCF8574 Adresse welche Ports gehören:

PCF 8574	ADR 0	Ports 17 – 24	BYTEPORT 3
PCF 8574	ADR 1	Ports 25 – 32	BYTEPORT 4
PCF 8574	ADR 2	Ports 33 – 40	BYTEPORT 5
PCF 8574	ADR 3	Ports 41 – 48	BYTEPORT 6
PCF 8574	ADR 4	Ports 49 – 56	BYTEPORT 7

PCF 8574	ADR 5	Ports 57 - 64	BYTEPORT 8
PCF 8574	ADR 6	Ports 65 - 72	BYTEPORT 9
PCF 8574	ADR 7	Ports 73 - 80	BYTEPORT 10

PCF 8574A	ADR 0	Ports 81 - 88	BYTEPORT 11
PCF 8574A	ADR 1	Ports 89 - 96	BYTEPORT 12
PCF 8574A	ADR 2	Ports 97 - 104	BYTEPORT 13
PCF 8574A	ADR 3	Ports 105 - 112	BYTEPORT 14
PCF 8574A	ADR 4	Ports 113 - 120	BYTEPORT 15
PCF 8574A	ADR 5	Ports 121 - 128	BYTEPORT 16
PCF 8574A	ADR 6	Ports 129 - 136	BYTEPORT 17
PCF 8574A	ADR 7	Ports 137 - 144	BYTEPORT 18

Definition eines der Extended Digital Bitports:

DEFINE bezeichner PORT[nr]

Dabei sind für nr Werte von 17 bis 144 zulässig, aber der PCF8574 muss entsprechend adressiert sein.

Definition eines 8 Bit breiten Extended Digital Byteports:

Der PCF8574 am I2C-Bus kann natürlich auch als BYTEPORT angesprochen werden. In der Aufstellung oben finden Sie die Zuordnung zur Adresse.

DEFINE bezeichner BYTEPORT[nr]

Dabei sind für nr Werte von 3 bis 18 zulässig, aber der PCF8574 muss entsprechend adressiert sein.

Wordports werden nicht unterstützt!

Technische Spezifikationen:

Betriebsspannung	5V stabilisierte Gleichspannung, +0,5V
Erweiterter Spannungsbereich	9-15V DC (nur Unit M 2.0)
Stromaufnahme	M 2.0 ca. 50 mA / M 1.2 ca. 8 mA
Abmessungen	M 2.0 ca. 50mm x 25mm / M1.2 ca 42mm x 42mm
Variablenspeicher	64 Byte RAM
Programmspeicher	10 kByte FLASH
Betriebssystem	ca. 6 kB FLASH
Bustakt	M 2.0 8 MHz / M 1.2 2MHz
A/D-Ports	8 x 8 Bit A/D, 0...5 Volt gegen gemeinsame Masse
Referenzspannung Uref	5V intern oder extern
Digitalports	16, frei als Ein- oder Ausgang programmierbar
Digitalport Pegel 0.2mA Last	$0,1\text{ V} < U_{\text{out,low}} < 0,3\text{V}/(U_{\text{B}} - 0,3\text{V}) < U_{\text{out,high}} < (U_{\text{B}} - 0,1\text{V})$
Digitalport Pegel Eingang	$0\text{V} < U_{\text{in low C}} (0,2 \cdot U_{\text{B}}) / (0,7 \cdot U_{\text{B}}) < U_{\text{in,high}} < U_{\text{B}}$
maximal zulässiger Laststrom	+/- 10mA
D/A-Ports	2 x 8 Bit 1930 Hz PWM
Serielle Schnittstelle	8N1, 9600 Baud, kein Handshake
Zul. Temperaturbereich	0.....60 C

Achtung:

Die Summe der Ströme aus allen Digitalports muss auf 20 mA begrenzt werden, wenn Sie die Unit M 2.0 über den 12V Anschluss mit Betriebsspannung versorgen



Impressum

Diese Bedienungsanleitung ist eine Publikation der Conrad Electronic GmbH,
Klaus-Conrad-Str. 1, 92240 Hirschau/ Germany.

Alle Rechte einschließlich Übersetzung vorbehalten. Reproduktionen jeder Art, z. B. Fotokopie, Mikroverfilmung, oder die Erfassung in elektronischen Datenverarbeitungsanlagen, bedürfen der schriftlichen Genehmigung des Herausgebers. Nachdruck, auch auszugsweise, verboten. Diese Bedienungsanleitung entspricht dem technischen Stand bei Drucklegung. Änderung in Technik und Ausstattung vorbehalten.

© Copyright 2003 by Conrad Electronic GmbH. Printed in Germany.



Imprint

These operating instructions are published by Conrad Electronic GmbH,
Klaus-Conrad-Str. 1, 92240 Hirschau/ Germany.

No reproduction (including translation) is permitted in whole or part e. g. photocopy, microfilming or storage in electronic data processing equipment, without the express written consent of the publisher. The operating instructions reflect the current technical specifications at time of print. We reserve the right to change the technical or physical specifications.

© Copyright 2003 by Conrad Electronic GmbH. Printed in Germany.