

## **Quickstart (C-Control Classic)**

Die folgende Anleitung soll Ihnen die erste Inbetriebnahme und den Einstieg in die Programmierung der C-Control-Unit anhand eines einfachen Beispiels (Ansteuerung eines der beiden auf dem Starterboard befindlichen Relais) erläutern. Sie werden schrittweise mit den dazu notwendigen Handlungen vertraut gemacht. Die jeweiligen technischen Einzelheiten sollen hier noch im Hintergrund stehen. Wesentliches Ziel dieser Anleitung ist, Ihnen über die Einstiegshürden hinweg zu helfen. Besondere Vorkenntnisse, abgesehen von elektrotechnischen Grundbegriffen und ersten Erfahrungen im Umgang mit dem PC, werden hier nicht vorausgesetzt.

### **1. Voraussetzungen**

Wir gehen davon aus, daß Sie die C-Control-Unit und das C-Control-Starter-Board erworben haben. Zum Lieferumfang sollte außerdem ein Nullmodemkabel mit dazugehörigem Adapterkabel für den Anschluß an die C-Control-Unit sowie zwei Disketten gehören.

Weiterhin werden Sie benötigen:

- einen PC mit einer freien seriellen Schnittstelle und Windows 95 oder Windows 3.1x als Betriebssystem
- eine Gleichspannungsquelle (muß nicht stabilisiert sein)  $U = 8 \dots 12V$ ,  $I_{max} > 100 \text{ mA}$
- zwei Anschlußkabel zum Anschluß der Spannungsversorgung an das Starterboard

### **2. Anschluß des C-Control-Starterboards**

#### **2.1. Auspacken**

Vor dem Berühren des Starterboards sollten Sie einen gut geerdeten Gegenstand (PC-Gehäuse, Heizkörper, etc.) kurz anfassen, um die Gefahr der Beschädigung der elektronischen Bauelemente auf der Platine durch elektrostatische Ladungen zu verringern. Prüfen Sie den Inhalt der Verpackung auf Vollständigkeit. Achten Sie auf evtl. Transportschäden. Wählen Sie einen sicheren Standort für das Starterboard, der zum einen gut einsehbar sein sollte und zum anderen den versehentlichen Kontakt insbesondere mit metallischen Gegenständen und Feuchtigkeit weitestgehend ausschließt.

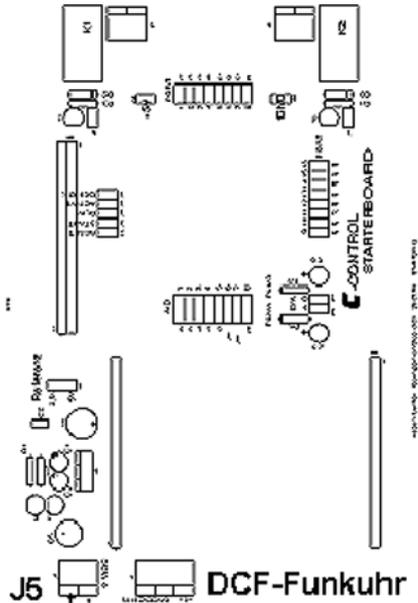
#### **2.2. Konfigurieren des Starterboards**

Auf der C-Control-Unit befinden sich drei Steckbrücken mit den Bezeichnungen AST, 232 und LED. Die Kontakte LED und 232 sollten jeweils mit kleinen Steckern (engl.: Jumper) geschlossen sein. AST muß offen bleiben. Die Karte wird mit dieser Einstellung ausgeliefert, so daß Sie normalerweise nichts verändern müssen.

Sollte die C-Control-Unit noch nicht auf das Starterboard gesteckt sein, so müssen Sie das - wie aus der Abbildung ersichtlich - tun. Aus der Lage der Stiftreihen und der dazu gehörigen Buchsen an der C-Control-Unit ist die richtige Position ebenfalls erkennbar. Wenden Sie bitte auf keinen Fall Gewalt an.

### 2.3. Anschluß der Spannungsversorgung

Vor dem eigentlichen Anschluß der Spannungsversorgung prüfen Sie bitte die Spannung (8...12 V=) und die Polarität der Spannungsquelle. Schalten Sie anschließend die Spannung aus. Verbinden Sie Spannungsquelle und die Klemme J5



auf dem Starterboard mit der richtigen Polarität (ist auf der Platine gekennzeichnet).

### 2.4. Herstellen der Verbindung zum PC

Der Steuercomputer und der PC werden über das mitgelieferte serielle Kabel verbunden. Sie benötigen also einen freien seriellen Port an Ihrem PC. Normalerweise verfügt jeder PC standardmäßig über zwei serielle Schnittstellen. Die Ausgänge dieser Ports befinden sich in der Regel auf der Rückseite des PCs und sind mit COM1, COM2, usw. beschriftet. Sie sind immer entweder als 9-polige oder 25-polige SUB-D-Verbinder (männlich, d.h. mit Stiften) ausgeführt und unterscheiden sich dadurch von allen anderen PC-Anschlüssen. (siehe Bild 2). Ein Anschluß davon wird meist durch die Maus belegt. Sollte nur noch ein 25-poliger Anschluß frei sein, müssen Sie im Fachhandel einen Adapter von 25-polig SUB-D auf 9-polig SUB-D kaufen, da das mitgelieferte Kabel nur für den letztgenannten Typ verwendbar ist. Sollte Ihr PC über keine freie serielle Schnittstelle verfügen, fragen Sie bitte einen Fachhändler nach konkreten Lösungsmöglichkeiten, da hier nicht auf alle Varianten eingegangen werden kann.

Schalten Sie bitte zunächst den PC und ggf. auch das Starterboard aus. Stecken Sie das eine Ende des Nullmodemkabels in den seriellen Port des PCs, an das andere Ende kommt das mitgelieferte Adapterkabel für den Anschluß an die C-Control-Unit. Die 10-polige, zweireihige Buchsenleiste auf der anderen Seite des Adapterkabels wird auf den Stecker J6 auf der C-Control-Unit gesteckt (siehe Bild 1). Dabei ist darauf zu achten, daß die am Adapterkabel durch eine rote Ader gekennzeichnete Seite zur mit "1" beschrifteten Seite des Steckverbinders auf der C-Control-Unit zeigt.

### 3. Installieren der Software

Legen Sie bitte die Diskette mit der Aufschrift "C-Control 1.03b" in das Disketten-Laufwerk ein. Starten Sie das Programm SETUP.EXE auf der Diskette. Die grafische Entwicklungsumgebung C-Control wird jetzt selbsttätig auf Ihrem Computer installiert. Gegebenenfalls müssen Sie noch die Systemerweiterung WIN32s installieren.

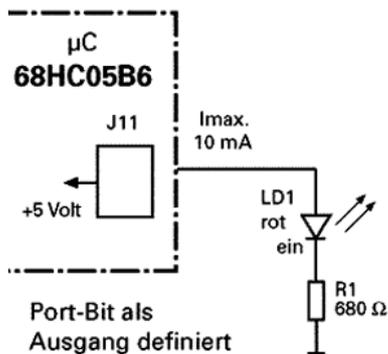
#### 4. Starten und Konfigurieren der Entwicklungsumgebung

Sie können nun die grafische Entwicklungsumgebung starten. Öffnen Sie dazu den Ordner, in dem C-Control installiert worden ist und starten Sie das Programm C-Control. Das Programm sucht automatisch alle seriellen Schnittstellen des PCs nach einer angeschlossenen C-Control-Unit ab. Im Falle einer Fehlermeldung sollten Sie die Verbindung zum PC überprüfen und sicherstellen, daß die von Ihnen verwendete Schnittstelle bzw. der betreffende Interrupt nicht gleichzeitig auch von anderen Programmen (z.B. Software für ein internes Modem) benutzt wird.

#### 5. Das erste Programm

##### 5.1. Aufgabe des Programmes

Bevor Sie jetzt Ihr erstes Programm erstellen, soll geklärt werden, was es bewerkstelligen soll. Die Aufgabe des im folgenden beschriebenen Beispiel-Programmes soll sein, das Relais 1 des Starterboards (siehe Bild 1) im Wechsel von einer Sekunde ein- und auszuschalten. Sie werden das Schalten an dem Klicken des Relais hören können. Wenn Sie das Schalten des Relais auch optisch verfolgen wollen, können Sie eine Leuchtdiode in Reihe mit einem Widerstand über die Relaiskontakte, die an die Klemme J11 herausgeführt sind an 5 Volt legen.



Hier noch eine Warnung: Mit dem Relais darf auf keinen Fall Netzspannung geschaltet werden. Isolation und Kontakte sind dafür nicht ausgelegt.

##### 5.2. Öffnen der Programmdatei

Öffnen Sie die Datei "QUICKSTC.BNT", die das Beispielprogramm enthält über den Menüpunkt *Datei/ Öffnen...*. Die Datei befindet sich auf der CD-ROM im Ordner "Ccontrol\Beispiel". Auf der Arbeitsfläche erscheint ein Blockschaltbild mit zwei verbundenen Elementen: Ein Timer-Block steuert einen Port-Block.

##### 5.3. Laden des Programmes in die Control-Unit

Rufen Sie jetzt den Menüpunkt Programm/ Laden auf. Das Programm wird jetzt von der Entwicklungsumgebung übersetzt und zur Control-Unit übertragen. Während dieses Vorgangs leuchtet die rote LED der Control-Unit. Das Laden kann einige Sekunden in Anspruch nehmen. Der erfolgreiche Abschluß wird von der Entwicklungsumgebung gemeldet. Bei der Übertragung von Programmen in die ControlUnit kann es in seltenen Fällen zu Synchronisationsproblemen kommen. Sollte Die C-Control-Software das Ende der Übertragung melden, die rote LED an der Unit jedoch nicht verlöschen, muß ein RESET an der Unit (roter Taster) von Hand ausgelöst und die Übertragung wiederholt werden. Bei weiteren Problemen lesen Sie bitte die Prüfliste (siehe weiter unten) durch um weitere Fehlerquellen auszuschließen

#### **5.4. Einstellen der Wiederholungsrate für das Programm**

Vor dem Starten des Programmes muß festgelegt werden, in welchen Zeitabständen es die Programmabarbeitung wiederholt werden soll. Rufen Sie dazu den Menüpunkt *Einstellungen/Zeitkonstante...* auf. Tragen Sie bitte in das entsprechende Eingabefeld den Wert 500 ein und beenden Sie den Dialog durch das Anklicken des Schalters *Zur Control-Unit übertragen*.

#### **5.5 Starten des Programmes**

Zum Starten des Programmes betätigen Sie bitte die gelbe Taste auf der Control-Unit. Das Relais1 muß jetzt hörbar im Sekundenrhythmus ein- und ausschalten. Die gelbe LED leuchtet ständig, die rote LED blinkt zweimal in der Sekunde (mit der vorher eingestellten Wiederholungsrate).

#### **5.6. Beenden des Programmes**

Zum Abbruch der Programmabarbeitung betätigen Sie bitte auf der Control-Unit die rote Reset- Taste. Das Relais bleibt eingeschaltet, keine LED leuchtet.

#### **6. Prüfliste zur Behebung von Fehlern**

- werden nur die Originalkabel verwendet?
- sind Nullmodemkabel und Adapterkabel verbunden?
- ist das Nullmodemkabel am PC angesteckt?
- ist das Adapterkabel polungsrichtig auf die ControlUnit gesteckt?
- sind die Jumper 232 und LED auf der ControlUnit gesetzt?
- ist die Betriebsspannung angeschlossen und eingeschaltet?
- führt eine zu hohe Belastung der Spannungsversorgung eventuell zu einem Spannungseinbruch?
- greift ein anderes Programm gerade auf die gewählte serielle Schnittstelle zu (Terminalprogramm, FAX-Programm)? Dann beenden Sie dieses, schließen Sie alle DOS-Fenster unter Windows
- gibt es Hardwarekonflikte der verwendeten seriellen Schnittstelle mit anderen Komponenten Ihres PCs? Prüfen Sie z.B. Interrupts und Adresseneinstellungen von Netzwerk-, Sound-, Modem-oder anderen Einsteckkarten

## **Quickstart (C-Control BASIC)**

Die folgende Anleitung soll Ihnen die erste Inbetriebnahme und den Einstieg in die Programmierung der C-Control-Unit anhand eines einfachen Beispiels (Ansteuerung eines der beiden auf dem Starterboard befindlichen Relais) erläutern. Sie werden schrittweise mit den dazu notwendigen Handlungen vertraut gemacht. Die jeweiligen technischen Einzelheiten sollen hier noch im Hintergrund stehen. Wesentliches Ziel dieser Anleitung ist, Ihnen über die Einstiegshürden hinweg zu helfen. Besondere Vorkenntnisse, abgesehen von elektrotechnischen Grundbegriffen und ersten Erfahrungen im Umgang mit dem PC, werden hier nicht vorausgesetzt.

### **1. Voraussetzungen**

Wir gehen davon aus, daß Sie die C-Control-Unit und das C-Control-Starter-Board erworben haben. Zum Lieferumfang sollte außerdem ein Nullmodemkabel mit dazugehörigem für den Anschluß an die C-Control-Unit sowie zwei Disketten (Sie benötigen für die folgende Anleitung nur die Diskette "C-Control/BASIC" für Windows95/NT) gehören.

Weiterhin werden Sie benötigen:

- einen PC mit einer freien seriellen Schnittstelle und Windows 95 oder Windows NT als Betriebssystem
- eine Gleichspannungsquelle (muß nicht stabilisiert sein)  $U = 8 \dots 12V$ ,  $I_{max} > 100 \text{ mA}$
- zwei Anschlußkabel zum Anschluß der Spannungsversorgung an das Starterboard

### **2. Anschluß des C-Control-Starterboards**

#### **2.1. Auspacken**

Vor dem Berühren des Starterboards sollten Sie einen gut geerdetem Gegenstand (PC-Gehäuse, Heizkörper, etc.) kurz anfassen, um die Gefahr der Beschädigung der elektronischen Bauelemente auf der Platine durch elektrostatische Ladungen zu verringern. Prüfen Sie den Inhalt der Verpackung auf Vollständigkeit. Achten Sie auf evtl. Transportschäden.

Wählen Sie einen sicheren Standort für das Starterboard, der zum einen gut einsehbar sein sollte und zum anderen den versehentlichen Kontakt insbesondere mit metallischen Gegenständen und Feuchtigkeit ausschließt.

#### **2.2. Konfigurieren des Starterboards**

Auf der C-Control-Unit befinden sich drei Steckbrücken mit den Bezeichnungen AST, 232 und LED. Die Kontakte LED und 232 sollten jeweils mit kleinen Steckern (engl.: Jumper) geschlossen sein. AST muß offen bleiben. Die Karte wird mit dieser Einstellung ausgeliefert, so daß Sie normalerweise nichts verändern müssen.

Sollte die C-Control-Unit noch nicht auf das Starterboard gesteckt sein, so müssen Sie das - wie aus der Abbildung ersichtlich - tun. Aus der Lage der Stiftreihen und der dazu gehörigen Buchsen an der C-Control-Unit ist die richtige Position ebenfalls erkennbar. Wenden Sie bitte auf keinen Fall Gewalt an.





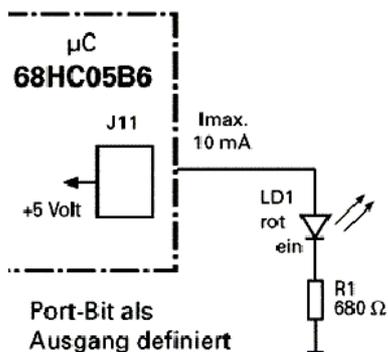
## 5. Starten und Konfigurieren der Entwicklungsumgebung

Sie können nun die integrierte Entwicklungsumgebung starten. Gehen Sie dazu in das *Startmenü/Programme* und klicken Sie CCEW32D an. Bevor Sie mit dem eigentlichen Programmieren beginnen können, muß der Entwicklungsumgebung mitgeteilt werden, an welche serielle Schnittstelle des PCs Sie das C-Control-Starterboard angeschlossen haben.

## 6. Das erste Programm

### 6.1. Aufgabe des Programmes

Bevor Sie jetzt Ihr erstes CCBASIC-Programm schreiben, muß geklärt werden, was es eigentlich bewerkstelligen soll. Die Aufgabe des im folgenden beschriebenen Beispiel-Programmes soll sein, das Relais 1 des Starterboards im Wechsel von einer Sekunde ein- und auszuschalten. Sie werden das Schalten an dem Klicken des Relais hören können. Wenn Sie das Schalten des Relais auch optisch verfolgen wollen, können Sie eine Leuchtdiode in Reihe mit einem Widerstand über die Relaiskontakte, die an die Klemme J11 heraus geführt sind, an 5 Volt legen.



Hier noch eine Warnung: Mit dem Relais darf auf keinen Fall Netzspannung geschaltet werden. Isolation und Kontakte sind dafür nicht ausgelegt.

### 6.2. Öffnen eines Dateifensters

Für Ihr Programm müssen Sie nun eine Datei anlegen in der der Programmtext (der sogenannte Quelltext) abgelegt werden kann. Ihrer Quelltext-Datei sollten Sie den Namen "relais1.bas" geben. An der Endung .bas erkennt die Entwicklungsumgebung den Text als BASIC-Quelltext; die unterschiedlichen Elemente eines BASIC-Programmes werden dann farblich voneinander abgehoben. Betätigen Sie dazu den Menüpunkt *Datei/Neu*. Nach dem Aufrufen des Menüs *Datei/Speichern unter* geben Sie bitte den Dateinamen "test1.bas" ein und bestätigen die Eingabe.

### 6.3. Schreiben des Quelltextes

Sie sollten jetzt den unten stehenden Quelltext in das eben angelegte Editor-Fenster schreiben. Der Text nach dem Hochkomma jeder Zeile stellt Kommentar dar und dient nur Ihrem Verständnis. Sie müssen ihn noch nicht im Einzelnen verstehen. Groß-/ Kleinschreibung und die Anzahl der Leerzeichen zwischen den Worten spielen keine Rolle.

```
DEFINE relais1 PORT[1]      'dem Digitalport 1 wird der Name relais1 zugewiesen
relais1 = 0                 'der Ausgang von Port 1 wird auf Null gesetzt (Relais offen)
#schleifenanfang           'Sprungmarke
TOG relais1                 'Zustand des Relais invertieren (engl.: toggle)
PAUSE 50                    '50 * 20 ms Pause
GOTO schleifenanfang       'Sprung zur Marke schleifenanfang
```

### 6.4. Programm übersetzen

Jetzt muß das Programm in eine für den Prozessor der C-Control-Unit verständliche Form übersetzt (engl.: compiliert) werden. Dazu müssen Sie den Menüpunkt *Entwicklung/BASIC-Compiler* aufrufen. Es erscheint danach das Meldungsfenster, in dem das Ergebnis des Übersetzungsvorgangs angezeigt wird. Werden Fehler gemeldet, können Sie mit der Maus doppelt auf die entsprechende Fehlermeldung klicken. Die Entwicklungsumgebung wechselt dann automatisch in Ihr Quelltextfenster und plaziert den Cursor auf die fehlerhafte Zeile. Bitte überprüfen Sie diese Zeile gründlich auf Abweichungen vom Mustertext. und wiederholen Sie den Übersetzungsvorgang.

### 6.5. Laden des Programmes in den Speicher der C-Control-Unit

Ist der Übersetzungsvorgang fehlerfrei abgeschlossen worden, kann das übersetzte Programm in die C-Control-Unit geladen werden.

*[ScreenCam 2 / Qsbas\_2.exe:: in der Entwicklungsumgebung Quelltext-Fenster wieder aktivieren, Menüpunkt Entwicklung/In C-Control-Unit übertragen aufrufen]*

Während des Ladevorgangs leuchtet die rote LED auf der C-Control-Unit. Es kann unter gewissen Umständen vorkommen, das der Ladevorgang nicht erfolgreich abgeschlossen werden kann. Sie erhalten dann eine Fehlermeldung. Bitte setzen Sie dann die Karte durch Betätigen der roten Taste zurück und wiederholen Sie den Vorgang. Bleibt auch dieser Versuch ohne Erfolg, gehen Sie bitte die Prüfliste (siehe weiter unten) durch, um nach Fehlern zu suchen.

### 6.6. Start des Programmes

Nun können Sie Ihr Programm starten. Drücken Sie auf die gelbe Taste der Control-Unit, so beginnt der Prozessor mit der zeilenweisen Abarbeitung Ihre Programmes. Sie können hören, wie das Relais im Sekundentakt schaltet. Als Zeichen der Programmabarbeitung leuchtet die gelbe LED auf der Control-Unit. Zum Anhalten des Programmes müssen Sie die rote Taste betätigen. Der Prozessor wird dann in seinen Ausgangszustand gebracht, das Programm wird angehalten.

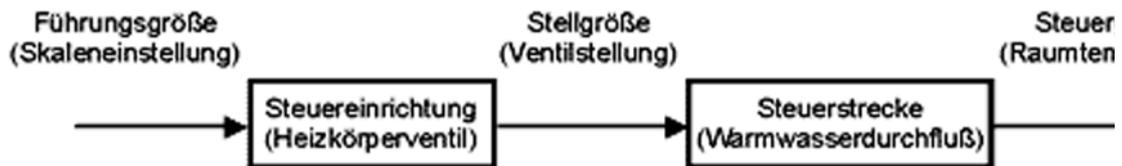
### 7. Prüfliste der möglichen Fehlerursachen

- werden nur die Originalkabel verwendet?
- sind Nullmodemkabel und Adapterkabel verbunden?
- ist das Nullmodemkabel am PC angesteckt?
- ist das Adapterkabel polungsrichtig auf die ControlUnit gesteckt?
- sind die Jumper 232 und LED auf der ControlUnit gesetzt?
- ist in der Entwicklungsumgebung die richtige COM-Schnittstelle eingestellt?

- ist die Betriebsspannung angeschlossen und eingeschaltet?
- führt eine zu hohe Belastung der Spannungsversorgung eventuell zu einem Spannungseinbruch?
- greift ein anderes Programm gerade auf die gewählte serielle Schnittstelle zu (Terminalprogramm, FAX-Programm)? Dann beenden Sie dieses und schließen Sie alle DOS-Fenster unter Windows
- gibt es Hardwarekonflikte der verwendeten seriellen Schnittstelle mit anderen Komponenten Ihres PCs? Prüfen Sie z.B. Interrupts und Adresseneinstellungen von Netzwerk-, Sound-, Modem-oder anderen Einsteckkarten

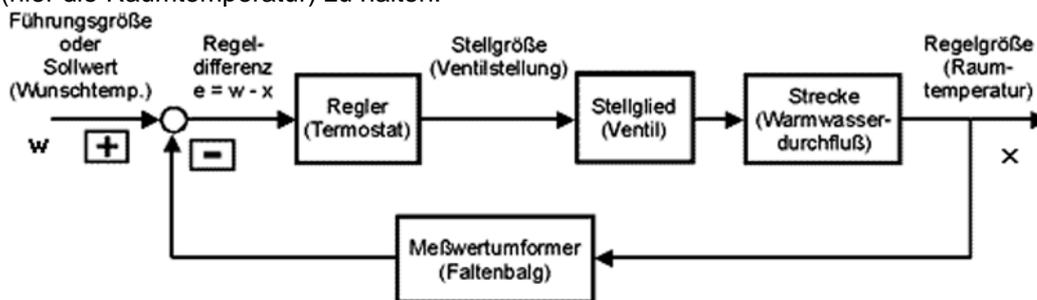
## Einführung in regelungstechnische Grundbegriffe

Die Begriffe Steuerung und Regelung bezeichnen unterschiedliche Vorgänge. Der Unterschied zwischen beiden Begriffen soll exemplarisch am Beispiel der Einstellung einer bestimmten Raumtemperatur erläutert werden. Verfügt der Heizkörper in dem betreffenden Raum über kein Thermostat (also einen Temperaturregler) wird dieser Vorgang als Steuerung bezeichnet (s.a. Bild 1). Mit einem Drehknopf oder Ähnlichem wird an einer Skala die gewünschte Temperatur (Führungsgröße) eingestellt. Infolgedessen wird dem Heizkörper über die Steuereinrichtung (Heizkörperventil) mehr oder weniger Wärmeenergie zugeführt. Die Raumtemperatur (Steuergröße) wird größer oder kleiner und nimmt, wenn die richtige Einstellung vorgenommen wurde, den gewünschten Wert an. Ändern sich aber die Umgebungsbedingungen, fällt beispielsweise die Außentemperatur, wird sich auch die Zimmertemperatur wieder verstellen. Eine neue manuelle Einstellung am Ventil muß vorgenommen werden, um den gewünschten Zustand wieder herzustellen.



**Blockschaltbild einer Steuerung  
(Wärmezufuhr an einem Heizkörper)**

Verfügt der Heizkörper jedoch über ein Thermostat, erfolgt die Einstellung der Raumtemperatur über eine Regelung. Im Unterschied zur Steuerung wird durch das Thermostat die Raumtemperatur gemessen und mit der Wunschtemperatur verglichen. Besteht zwischen beiden eine sogenannte Regeldifferenz wird die Stellung des Ventils so verändert, daß sich nach einer gewissen Zeit die gewünschte Temperatur unabhängig von den konkreten Umgebungsbedingungen einstellt. Es handelt sich, wie auch aus Bild 2 ersichtlich ist, im Gegensatz zur Steuerung um einen geschlossenen Kreis. Die Regeleinrichtung ist also in der Lage auf Störgrößen (z.B. Veränderung der Außentemperatur) zu reagieren und selbständig einen gewissen Sollwert der Regelgröße (hier die Raumtemperatur) zu halten.

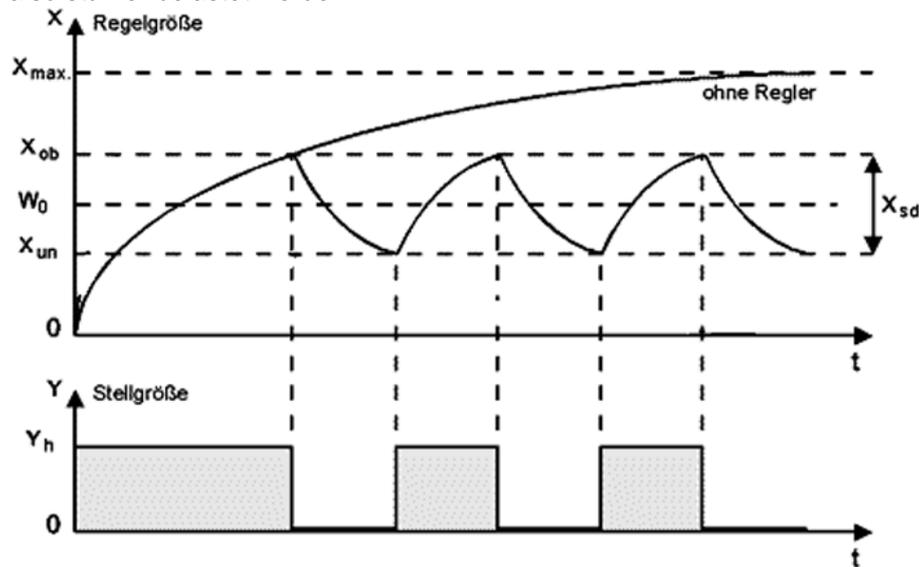


**Blockschaltbild eines Regelkreises**

Die Einrichtung, die aus der Regeldifferenz die Stellgröße erzeugt, wird als Regler bezeichnet. Für das Beispiel Regelung der Raumtemperatur ist das Thermostat, dessen Faltenbalg sich in Abhängigkeit von der Raumtemperatur ausdehnt oder zusammenzieht und damit die Stellung des Heizkörper-Ventils verändert. Der Regler hat entscheidenden Einfluß auf das Regelverhalten der Regeleinrichtung. Unter Regelverhalten versteht man, in welcher Art und Weise sich die Regelgröße verändert, wenn die Führungsgröße geändert wird (z.B. Einstellen einer anderen Raumtemperatur) oder eine Störgröße auf den Regelkreis wirkt

(Veränderung der Außentemperatur). So kann z.B. das Thermostat eines Heizkörpers mehr oder weniger schnell auf eine Änderung der Umgebungstemperatur reagieren. Zu beachten ist auch, mit welcher Genauigkeit der Regler in der Lage ist einen Sollwert einzustellen. Will man ein bestimmtes Regelverhalten festlegen, muß man das Verhalten der Regelstrecke kennen (Um welchen Betrag und wie schnell ändert sich die Raumtemperatur, wenn ich das Ventil um einen gewissen Wert verstelle ?) und einen geeigneten Regler einsetzen.

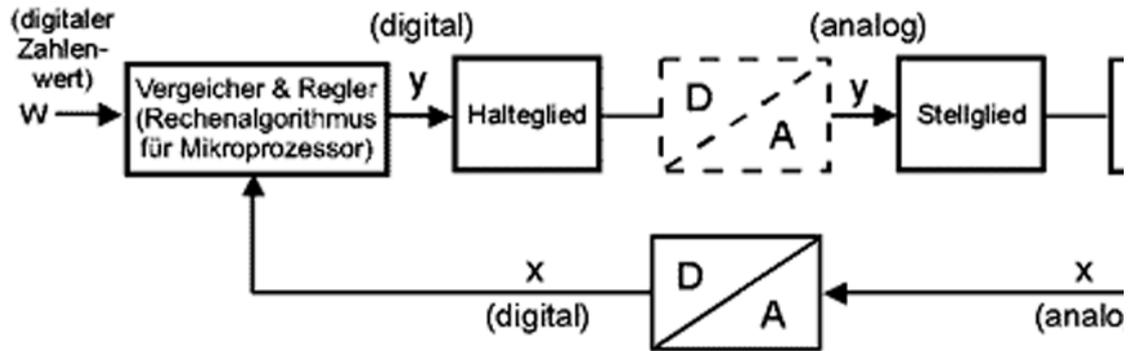
Ein einfacher Reglertyp ist der sogenannte Zweipunktregler, der häufig zur Temperaturregelung in Wärmegeräten, wie z.B. Bügeleisen, Kühlschrank, Warmwasserboiler, etc. eingesetzt wird. Das Regelverhalten eines Zweipunktreglers ist in Bild 3 dargestellt. Erreicht die Regeldifferenz (z.B. Abweichung der Wassertemperatur eines Boilers vom Sollwert) einen gewissen Wert, schaltet der Zweipunktregler (z.B. ein Bimetall-Kontaktschalter) die Wasserheizung ein, die Wassertemperatur steigt mit einer gewissen Verzögerung an. Ist der Sollwert ( $W_0$ ) um einen bestimmten Betrag überschritten ( $X_{ob}$ ), schaltet der Regler die Heizung wieder aus, die Wassertemperatur sinkt langsam, bis die untere Schaltschwelle  $X_{un}$  erreicht ist, usw. Die Regelgröße schwankt also ständig um den Sollwert. Die maximale Abweichung der Regelgröße vom Sollwert wird durch die obere und untere Schaltschwelle festgelegt. Dabei ist zu beachten, daß eine Verkleinerung dieser Grenzen zu einer erhöhten Schaltfrequenz des Reglers führt, eventuell vorhandene Kontakte also stärker belastet werden.



**Regelverhalten eines Zweipunktreglers**

Regler werden heute oft durch den Einsatz von Mikroprozessoren realisiert. Das hat den Vorteil, daß das Regelverhalten jederzeit durch eine Änderung des auf dem Prozessor laufenden Programms an variable Erfordernisse angepaßt werden kann. So kann eine einmal entwickelte Hardware für verschiedenste Anwendungen eingesetzt werden. Ein weiterer Vorteil des Einsatzes eines Mikroprozessors ist, daß von ihm quasi gleichzeitig auch noch andere Aufgaben wie die Überwachung und Anzeige von Systemgrößen, oder auch die Übertragung von Daten von und zu einem übergeordnetem Computer erledigt werden können. Einen Regelkreis mit einem Mikroprozessor als digitalem Regler zeigt Bild 4. Die Regelgröße muß, bevor sie vom Mikroprozessor verarbeitet werden kann, digitalisiert, d.h. einer Analog/Digital- Wandlung unterzogen werden. Der dann digital vorliegende Wert kann durch ein entsprechendes Programm mit dem ebenfalls als digitalem Wert vorliegenden Sollwert verglichen werden. Daraus kann dann der Wert für die Stellgröße berechnet werden. Dieser muß anschließend auf einen Digital/Analog- Wandler ausgegeben werden, da das Stellglied nur mit einem analogen Wert angesteuert werden kann. So ist es z.B. denkbar eine Temperaturregelung mit einem digitalen Zweipunktregler durch den Einsatz eines

Mikroprozessors zu realisieren. Dazu muß die Temperatur elektronisch gemessen werden (z.B. Temperatursensor PT100 für die C-Control-Unit). Die analoge Meßspannung muß digitalisiert werden (z.B. Anschluß des Temperatursensors an einen analogen Eingang der C-Control-Unit.). Dieser Wert kann mit einem Programm ständig mit einem vorgegebenen (digitalem) Sollwert verglichen werden. Wird die untere Schaltschwelle unterschritten, wird vom Mikroprozessor über einen seiner digitalen Ausgänge ein Relais geschaltet, das die Heizung einschaltet. Ist die obere Schaltschwelle erreicht, wird das Relais wieder ausgeschaltet, usw.



**Regelkreis mit digitalem Regler**

## Hardware

### C-Control-Unit

#### Allgemeine Handhabungs- und Sicherheitshinweise

Vor dem Berühren elektronischer Baugruppen sollten Sie einen gut geerdeten Gegenstand (PC-Gehäuse, Heizkörper, etc.) kurz anfassen, um die Gefahr der Beschädigung der Bauelemente auf der Platine durch elektrostatische Ladungen zu verringern. Überprüfen Sie den Inhalt der Verpackung auf Vollständigkeit. Achten Sie auf evtl. Transportschäden. Haben Sie die Baugruppe vom Kalten ins Warme gebracht, warten Sie bitte mit dem Anschluß der Versorgungsspannung, bis eventuell gebildetes Kondenswasser vollständig getrocknet ist.

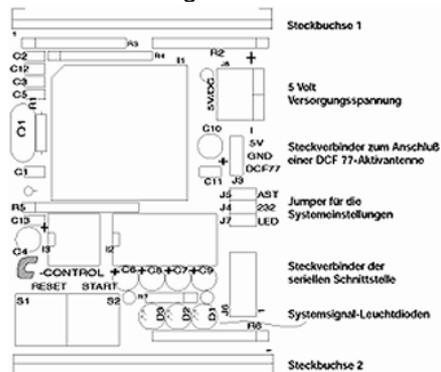
Wählen Sie einen sicheren Standort, der zum einen gut zugänglich und einsehbar sein sollte und zum anderen den versehentlichen Kontakt insbesondere mit metallischen Gegenständen und Feuchtigkeit weitestgehend ausschließt. Die Baugruppe sollte nicht in unmittelbarer Nähe starker Magnetfelder (z.B. von großen Elektromotoren, Lautsprechern oder Netzteilen) aufgestellt werden. Bei einem Einsatz im Freien sollten blitzgefährdete Standorte, wie z.B. Bergkuppen, einzelstehende Masten, Dächer, etc. vermieden, bzw. mit speziellen Schutzeinrichtungen versehen werden.

Der Einsatz darf nur entsprechend der Technischen Spezifikation erfolgen. Dabei ist insbesondere auf die Einhaltung der Grenzwerte für die Betriebsspannung, die Eingangsspannung an den digitalen und analogen Eingängen und den zulässigen Strom für die digitalen Ausgänge zu achten.

Alle Verbindungen zwischen elektronischen Baugruppen oder zu einem PC (z.B. über das serielle Kabel) sind im spannungsfreien Zustand aller beteiligten Baugruppen herzustellen. Soll eine Verbindung unterbrochen werden, ist vorher die Betriebsspannung auch der Peripherie (also auch eines eventuell angeschlossenen PC) abzuschalten. Ein gemeinsamer Netzschalter für alle verbundenen Komponenten ist zu empfehlen.

Achten Sie beim Anschluß der Betriebsspannung auf die richtige Polung und den richtigen Spannungswert. Die Spannungsquelle darf nur im spannungsfreien Zustand angeschlossen werden.

Die Mißachtung dieser Hinweise kann zur Zerstörung der Baugruppe führen.



#### Aufbau und Funktion

Die C-Control-Unit stellt einen eigenständigen Steuercomputer dar. Programme für diesen Steuercomputer werden in der leicht zu erlernenden Programmiersprache BASIC auf einem PC erstellt, getestet und anschließend in die Control-Unit geladen. Das geladene Programm bleibt dabei auch nach dem Abschalten der Betriebsspannung erhalten. Ist das Programm einmal geladen worden, kann die Baugruppe ohne Verbindung zu einem PC und damit ortsunabhängig eingesetzt werden.

Das Herzstück der Baugruppe stellt ein Mikrocontroller vom Typ MC68HC05B6 dar (Der Prozessor auf der Control-Unit trägt eine andere Beschriftung, da es sich um eine maskenprogrammierte Version handelt). Dieser Mikrocontroller führt mit Hilfe seines Betriebssystems (enthalten im internen maskenprogrammierten 6kByte ROM) das jeweils geladene Programm aus.

Die C-Control-Unit verfügt über acht A/D-Ports, zwei D/A-Ports und 16 digitale Ports, die frei als Ausgang oder Eingang programmiert werden können.

Auf der Baugruppe befindet sich weiterhin ein Speicherchip der das vom PC geladene Programm enthält und auch vom Anwenderprogramm als Datenspeicher genutzt werden kann. (serieller EEPROM mit 8 Kbyte Speicherkapazität, z.B 24C65). Der Speicherinhalt bleibt auch nach dem Ausschalten der Betriebsspannung erhalten.

Ein Pegelwandler (z.B. MAX232) dient dazu , die unterschiedlichen Logikpegel an der seriellen Schnittstelle von der Control-Unit und dem PC (CMOS <-> RS232) aneinander anzupassen.

Das Betriebssystem kann bei Anschluß einer DCF77-Aktivantenne eine intern geführte Uhr synchronisieren. Bei Empfangsausfall läuft die Uhr mit der internen Genauigkeit (maximale Gangabweichung = 0,36 Sekunden pro Stunde) weiter. Nach dem Einschalten der Betriebsspannung oder einem Reset startet die Uhr bei 01.01.97, 00:00:00 Uhr. Die Uhr kann auch per Software gestellt werden, indem die Variablen für Jahr, Monat, usw. mit den gewünschten Werten beschrieben werden. Die Systemuhr kann dazu verwendet werden, Steuerungsvorgänge zu einer bestimmten Zeit auszulösen.

Der DCF77-Anschluß kann alternativ auch zur Frequenzmessung digitaler Signale verwendet werden: Das Betriebssystem zählt die Anzahl der empfangenen Impulse pro Sekunde und liefert das Ergebnis bei Aufruf der Funktion `FREQ` zurück (s.a. `CC-BASIC-Befehl FREQ`).

Die Control-Unit kann an einem ihrer Ports einen Ton in Form einer Rechteckschwingung ausgeben (s.a. `CC-BASIC-Befehl BEEP`).

Das Betriebssystem stellt weiterhin eine 16-Bit Variable zur Verfügung, die im 20 ms - Rhythmus des internen Timers hochgezählt wird (d.h. auch, die Variable läuft alle  $65536 * 20$  ms über, beginnt also neu bei 0). Diese Variable kann innerhalb eines Anwenderprogrammes jederzeit gelesen werden (s.a. `CC-BASIC-Befehl TIMER`). Dadurch wird es möglich mit einem maximalen Fehler von +/- 20 ms Zeitabstände zu messen. Einige BASIC-Befehle beziehen sich auf diese Zeitbasis ( `BEEP`; `PAUSE`).

Zur Vereinfachung des Einsatzes dient ein zusätzlich erhältliches Starterboard, auf das die Control-Unit aufgesteckt werden kann.

### **Bedien- und Anzeigeelemente**

Auf der Baugruppe befinden sich folgende Bedien- und Anzeigeelemente:

Eine rote Taste dient dem Reset der Baugruppe und damit auch einem Programmabbruch.

Eine gelbe Taste dient dem Start des geladenen Programmes.

Eine grüne LED "DCF OK" blinkt im Sekundentakt, wenn der DCF77-Empfang ungestört funktioniert.

Eine gelbe LED "ACTIV" leuchtet, wenn sich die ControlUnit im Zustand der Programmabarbeitung befindet.

Eine rote LED "RUN" leuchtet während der Übertragung von Daten vom PC an die Control-Unit, also auch beim Laden eines Programmes.

Version CLASSIC: Die rote LED blinkt im Zustand der zyklischen Programmabarbeitung außerdem für die Zeitdauer des eigentlichen Programmablaufes kurz auf, d.h. auch sie leuchtet für die Zeitdauer der zwischen den Programmläufen liegenden Zeiträumen nicht. Geht das Blinken in dauerhaftes Leuchten über, benötigt ein Programmablauf also mehr Zeit als die eingestellte Zykluszeit erlaubt. Das Programm wird neu gestartet, bevor es bis zu Ende abgearbeitet wurde. Die Zykluszeit muß dann vergrößert werden.

Version BASIC: Die rote LED leuchtet (wie die gelbe LED), wenn sich die Control-Unit im Zustand der Programmabarbeitung befindet. Sie wird aber für die Dauer der Ausführung des PAUSE - Befehls ausgeschaltet. Befinden sich also im Programm Abschnitte, die zyklisch wiederholt werden, und die einen PAUSE-Befehl enthalten, kann man den Programmablauf am Blinken dieser LED erkennen.

Die Kathoden der drei LEDs sind an der Buchsenleiste 2 (Belegung siehe Technische Spezifikation) verfügbar. Wird der Jumper "LED" abgezogen und damit die LEDs auf der Baugruppe ausgeschaltet, können an diese Anschlüsse externe LEDs und ein Widerstand zur Strombegrenzung angeschlossen werden, die dann den Systemzustand z.B. an der Frontblende eines Gehäuses anzeigen. Ebenso sind die parallel zum Start- und Reset-Taster liegende Anschlüsse an diese Buchsenleiste herausgeführt. Es ist also möglich, jeweils einen funktionsgleichen externen Taster zu realisieren.

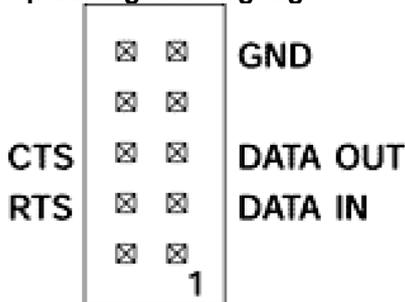
### **Installation und Anschluß**

Die Leiterplatte der Control-Unit verfügt über zwei 20polige Buchsenleisten. An ihnen sind alle verwendbaren Ports und einige Sytemsignale verfügbar. Die genaue Belegung dieser Buchsenleisten ist aus der Belegungstabelle in den Technischen Daten [s.u.] ersichtlich. Soll die Baugruppe im Zusammenhang mit dem Starterboard eingesetzt werden, ist sie auf die dafür vorgesehenen Stifteleisten des Starterboards, mit der Bestückungsseite nach oben, aufzustecken. Ein um 180° verdrehtes Aufstecken wird durch drei Stifte auf dem Starterboard verhindert, die nur in der richtigen Lage in entsprechende Bohrungen der Control-Unit passen. Soll die Control-Unit ohne Starterboard betrieben werden erscheint es ratsam, die eben beschriebene Vorrichtung auf einer eigenen Leiterplatte (Platine) nachzubauen.

Die C-Control-Unit und der PC werden über das mitgelieferte serielle Kabel verbunden. Sie benötigen also einen freien seriellen Port an Ihrem PC. Normalerweise verfügt jeder PC standardmäßig über zwei serielle Schnittstellen. Die Ausgänge dieser Ports befinden sich in der Regel auf der Rückseite des PC und sind mit COM1, COM2, usw. beschriftet. Sie sind immer entweder als 9-polige oder 25-polige SUB-D-Verbinder (männlich, d.h. mit Stiften) ausgeführt und unterscheiden sich dadurch von allen anderen PC-Anschlüssen. Ein Anschluß davon wird meist durch die Maus belegt. Sollte nur noch ein 25-poliger Anschluß frei sein, müssen Sie im Fachhandel einen Adapter von 25-polig SUB-D auf 9-polig SUB-D kaufen, da das mitgelieferte Kabel nur für den letztgenannten Typ verwendbar ist. Sollte Ihr PC über keine freie serielle Schnittstelle verfügen, fragen Sie bitte einen Fachhändler nach konkreten Lösungsmöglichkeiten, da hier nicht auf alle Varianten eingegangen werden kann.

Schalten Sie bitte zunächst den PC und ggf. auch das Starterboard aus. Stecken Sie das eine Ende des Nullmodemkabels in den seriellen Port des PC, an das andere Ende kommt das mitgelieferte Adapterkabel für den Anschluß an die C-Control-Unit. Die 10-polige, zweireihige Buchse auf der anderen Seite des Adapterkabels wird auf den Stecker J6 auf der C-Control-Unit gesteckt. Dabei ist darauf zu achten, daß die am Adapterkabel durch eine rote Ader gekennzeichnete Seite zur mit "1" beschrifteten Seite des Steckverbinders auf der C-Control-Unit zeigt.

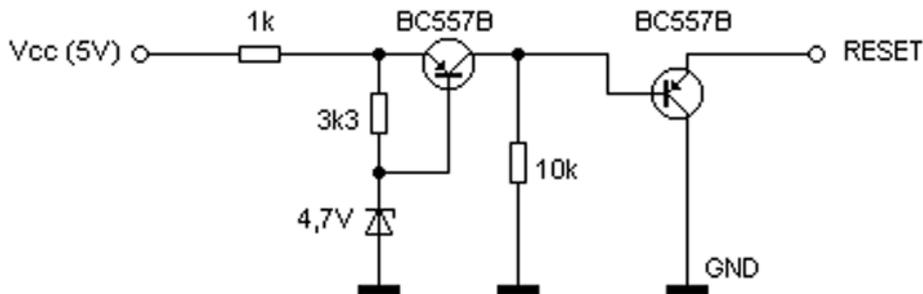
## Spannungsversorgung



Die C-Control-Unit benötigt zum Betrieb eine stabilisierte Betriebsspannung von 5V +/- 10% und einen Strom von ca. 30 mA. Werden externe Komponenten angeschlossen, ist der dadurch zusätzlich aufgenommene Strom einzukalkulieren. Die Spannung muß mit der richtigen Polung an J8 angeschlossen werden.

Die Versorgungsspannung darf bei einem laufenden Anwenderprogramm im Steuercomputer nicht ausfallen oder abgeschaltet werden! Es muß zuvor ein Reset ausgelöst werden. Anderenfalls kann es zu ungünstigen Bedingungen beim Zugriff auf den EEPROM-Speicherchip (siehe unten) führen und das gespeicherte Anwenderprogramm beschädigen. Anschließend wäre dann eine Neuprogrammierung erforderlich.

Sollte bei einem Dauerbetrieb des Steuercomputers die Gefahr eines Spannungsausfalls bestehen, ist eine externe Spannungswächterschaltung (Watchdog) mit der 5V-Betriebsspannung, Masse (GND) und dem Reset-Port zu verbinden. Optimal ist dafür das IC MAX809L des Halbleiterherstellers MAXIM geeignet, das einen Reset bei Unterschreitung des Spannungsschwellwertes von 4,65V auslöst. Alternativ dazu kann auch folgende Transistorschaltung aufgebaut werden:



## Konfiguration

Auf der Baugruppe befinden sich drei Jumper zur Systemkonfiguration:

**LED:** Dieser Jumper dient der Reduzierung der Stromaufnahme der Control-Unit. Ist er abgezogen, sind die drei LEDs von der Betriebsspannung getrennt. Dieser Jumper kann auch während des laufenden Betriebs gezogen und gesteckt werden. Während der Testphase, sollte dieser Jumper gesteckt sein, da der Zustand der LEDs wichtige Informationen über den Ablauf Ihres Programmes liefert.

232: (RS232 heißt der Standard für eine genormte serielle Schnittstelle) Dieser Jumper dient der Reduzierung der Stromaufnahme der Control-Unit. Ist er abgezogen, ist der Pegelwandler-Schaltkreis von der Betriebsspannung getrennt. Allerdings ist dann auch die serielle Schnittstelle außer Betrieb, so daß z.B. eine Verbindung zum PC nicht hergestellt werden kann. Während der Testphase muß dieser Jumper gesteckt sein, da ständig neue bzw. korrigierte Programme über die serielle Schnittstelle geladen werden müssen.

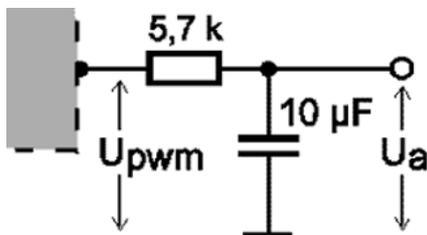
AST: (Autostart) Ist der Jumper gesetzt, beginnt die ControlUnit sofort nach dem Einschalten oder nach Betätigung des RESET-Tasters mit der Programmausführung.

### Analoge Signalverarbeitung

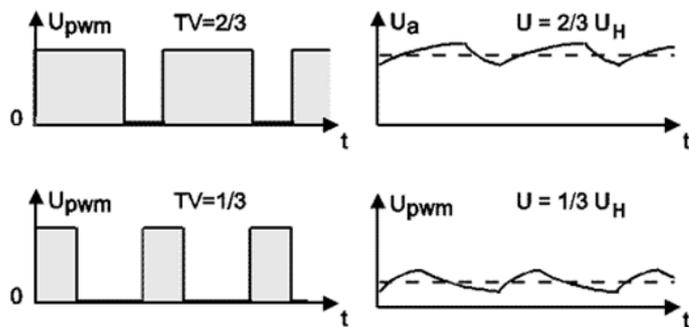
Ein Mikroprozessor kann intern nur digitale Werte verarbeiten. Soll also z.B. eine Temperatur anhand des Spannungsabfalls über einem wärmeabhängigen Widerstand gemessen und angezeigt werden, so muß dieser analoge Spannungswert mit Hilfe des internen A/D-Wandlers digitalisiert werden. Andererseits erfordert z.B. die Ansteuerung eines Motors eine analoge Spannung. Diese analoge Spannung wird mit Hilfe eines internen D/A-Wandlers aus einem, vom Mikroprozessor ermittelten digitalen Wert erzeugt.

Die C-Control-Unit verfügt über acht A/D-Ports, an denen ein analoges Signal in ein 8 Bit Digitalsignal umgewandelt wird. Der A/D-Wandler benötigt für die Wandlung eine Referenzspannung, die die Obergrenze des zu messenden Wertes darstellt (an Buchsenleiste 1.2). Wird beispielsweise die 5V Betriebsspannung als Referenzspannung verwendet, so können Eingangsspannungen im Bereich von 0...5V gewandelt werden. Die Eingangsspannung 5V erhält dann den digitalen Wert 255 (alle acht Bit gesetzt) zugewiesen. Der maximale Fehler (Quantisierungsfehler) beträgt in diesem Fall  $\frac{1}{2} * 5V/255$  plus dem Fehler, der aus der Toleranz der Referenzspannung resultiert. Die Referenzspannung darf 5V nicht überschreiten.

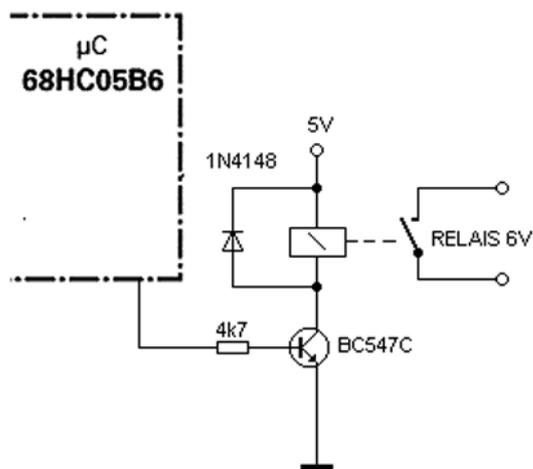
Die C-Control-Unit verfügt über zwei D/A-Ports, die nach dem Prinzip der Pulsweitenmodulation (PWM) funktionieren. Die erzeugte Gleichspannung wird dabei zunächst über das Tastverhältnis eines Rechtecksignals (Grundfrequenz = 1953 Hz) erzeugt. Ist das Tastverhältnis = 0,5, d.h. ist die Zeit, für die der Ausgang HIGH-Pegel führt, halb so groß wie die Taktperiode, beträgt der Gleichspannungsanteil dieses Signals die Hälfte der HIGH-Spannung. Wird der zeitliche Anteil des HIGH-Pegels größer, vergrößert sich auch die enthaltene Gleichspannung. Wird er kleiner, verkleinert sich proportional mit dem Tastverhältnis der Gleichspannungsanteil und kann minimal 0V werden. Die Gleichspannung kann aus dem PWM-Signal im einfachsten Fall durch eine RC-Tiefpass-Filterung gewonnen werden. Dabei kann die Spannung am Querkondensator den schnellen Änderungen des PWM-Signals nicht folgen. Es stellt sich eine mittlere, nur wenig schwankende Gleichspannung ein, die zur Ansteuerung von analogen Geräten verwendet werden kann. Werden höhere Anforderungen an die Restwelligkeit der erzeugten Gleichspannung gestellt, bzw. soll die Spannung in einen anderen Bereich transformiert werden, müssen aufwendigere Schaltungslösungen (z.B. mit Operationsverstärkern) in Betracht gezogen werden.



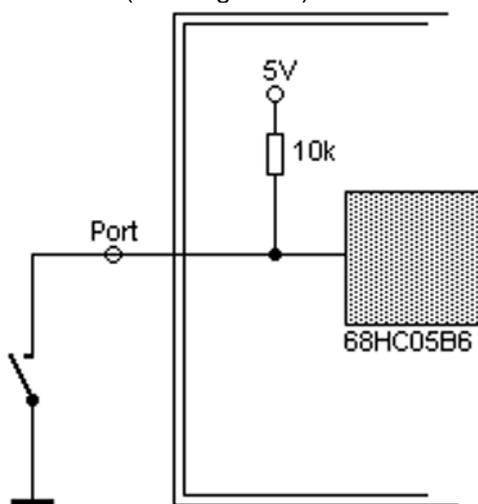
### A/D-Wandlung nach dem PWM Prinzip



## Digitale Signalverarbeitung



Die C-Control-Unit verfügt über 16 digitale Ports, die einzeln als Eingang oder Ausgang programmiert werden können. An diesen Ports werden nur zwei Zustände unterschieden. Die Spannung, die den Zustand HIGH (oder logisch 1) repräsentiert, und die Spannung, die den Zustand LOW (oder logisch 0) darstellt.



Ist ein Port als Ausgang definiert worden kann er per Software auf den Wert HIGH ( $U_a > 0,3V$ ) oder LOW ( $U_a < 0,3V$ ) gesetzt werden. Diese Ausgangspegel können von nachfolgenden digitalen Schaltungen verarbeitet werden. Es ist aber auch möglich durch einen digitalen Port direkt eine LED oder über einen Schalttransistor ein Relais anzusteuern. Achtung: Der maximal zulässige Ausgangsstrom an einem digitalen Port beträgt 10 mA. Es ist also eine entsprechende Strombegrenzung (z.B. durch einen Vorwiderstand) vorzusehen.

Ist ein Port als Eingang definiert worden kann per Software der anliegende Zustand eingelesen werden: Liegt am Eingang eine Spannung  $> 3,5V$ , so wird dieser Wert sicher als HIGH interpretiert. Ist der Wert  $< 1V$ , wird sicher auf LOW entschieden. Dazwischen liegende Werte müssen vermieden werden. Eine einfache Anwendung ist die Abfrage des Zustandes eines Schalters.

Ein als Eingang definierter Port ist hochohmig. Um einen definierten Zustand auch ohne angelegtes Signal zu gewährleisten, sind auf der Control-Unit alle digitalen Ports mit einem  $10\text{ k}\Omega$  Widerstand nach Betriebsspannung (sogenannter Pull Up-Widerstand) auf HIGH-Pegel gezogen.

### **Anschluß einer DCF77-Aktivantenne**

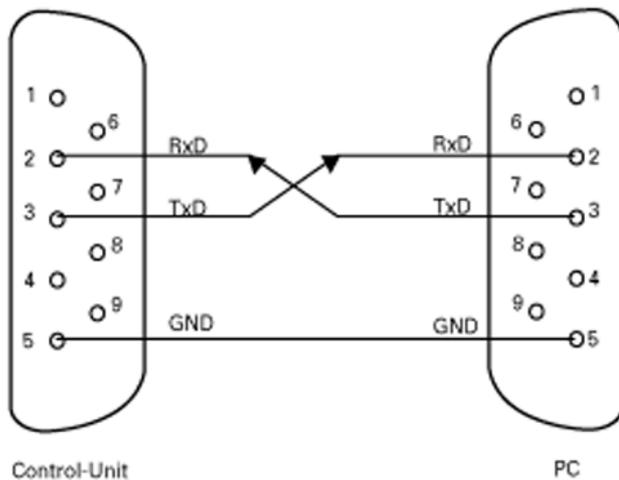
Für den Anschluß einer DCF77-Aktivantenne an die Control-Unit ist auf der Platine eine dreipolige Stiftleiste (J3) vorgesehen. Die dreipolige Stiftleiste stellt die Spannungsversorgung für die Aktivantenne zur Verfügung (5V, GND) und empfängt die Signalimpulse von der Antenne. Die Antenne muß dazu über 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. Bei korrektem Empfang des DCF77-Zeitsignals blinkt die grüne LED im Rhythmus einer Sekunde. Unregelmäßiges Blinken deutet auf Empfangsprobleme hin, die sich i.a. durch ein besseres Ausrichten der Antenne (der Pfeil soll nach Frankfurt/a.M. zeigen) bzw. durch die Wahl eines besseren Standortes beheben lassen.

### **Kommunikation über die serielle Schnittstelle**

Bei der seriellen Übertragung von digitalen Signalen, werden die einzelnen Bits nacheinander übertragen. Es wird also nur eine Verbindungsleitung in jeder Richtung und eine Masseverbindung zwischen den Baugruppen benötigt. Diesem Vorteil des geringen Schaltungsaufwandes steht der Nachteil einer geringen maximalen Übertragungsrate gegenüber. Im Gegensatz dazu steht die parallele Übertragung von Daten (z.B. vom Parallelport eines PC zu einem Drucker). Hier wird mit jeder Taktperiode ein Byte gleichzeitig übertragen. Man benötigt entsprechend mehr Verbindungsleitungen, erzielt dafür aber auch eine große maximale Übertragungsrate.

Eine weit verbreitete serielle Verbindung ist die Schnittstelle nach dem RS232 Standard. Diese Verbindung dient beispielsweise an PCs zum Anschluß der Maus. Die C-Control-Unit bedient sich ebenfalls dieser Schnittstelle um Programme vom PC zu empfangen und Daten zum PC zu senden. Die serielle Verbindung zwischen PC und Control-Unit enthält mindestens folgende Leitungen:

### Serielle Verbindung zwischen PC und Control-Unit



Die Anschlüsse der seriellen Schnittstelle haben folgende Bedeutung:

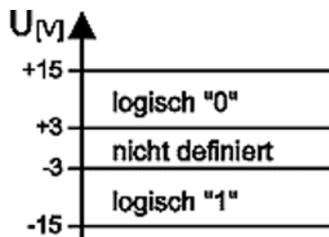
TxD Transmit Data, Ausgang, An diesem Pin werden die Daten gesendet

RxD Receive Data, Eingang, An diesem Pin werden die Daten empfangen

GND Masseverbindung

Optional sind noch eine Reihe anderer Verbindungsleitungen möglich, die den Datenaustausch zuverlässiger gestalten und teilweise speziell für den Anschluß von Modems vorgesehen sind.

Die Bitzustände logisch "1" und logisch "0" werden durch unterschiedliche Spannungspegel auf den Übertragungsleitungen dargestellt.



### Spannungspegel auf den Übertragungsleitungen

Im folgenden soll beschrieben werden, was geschieht wenn das Zeichen "A" von der Control-Unit an den PC gesendet wird (BASIC-Befehl PRINT "A"). Jedes Zeichen wird durch ein bestimmtes Byte repräsentiert. Das Zeichen "A" ist nach dem ASCII-Zeichensatz dem Byte mit dem Inhalt 65 (binär = 01000001) zugeordnet. Diese acht Bits müssen also nacheinander am Anschluß TxD der seriellen Schnittstelle der Control-Unit ausgegeben werden. Zunächst wird dem Empfänger durch die Ausgabe eines sogenannten Start-Bits (logisch "0") der Beginn der Übertragung eines Zeichens angekündigt. Er kann dadurch seinen internen Takt so synchronisieren, daß er die nachfolgenden Datenbits jeweils in Bitmitte abtasten kann. Dann werden nacheinander die acht Bits gesendet, wobei das niederwertigste (Bit 0) zuerst ausgegeben wird. Beendet wird die Übertragung durch die Ausgabe eines sogenannten Stop-Bits - der Empfänger weiß dann, daß die Übertragung beendet ist und kann das bitweise eingelesene Zeichen anzeigen bzw. in einer Datei speichern.



### Anschluß TxD der Control-Unit beim Senden des Zeichens "A"

Der Zustand für ein Bit liegt immer für die Zeitdauer T an. Ist die Übertragungsrate auf 9600 kBit/s eingestellt, beträgt diese Zeit also  $T = 1/9600 = 104 \mu\text{s}$  (Mikrosekunden =  $10^{-6}$  Sekunden). Beide Kommunikationspartner müssen selbstverständlich immer auf die gleiche Übertragungsrate eingestellt sein. Die Übertragungsrate der Control-Unit kann durch den CC-BASIC-Befehl BAUD eingestellt werden. Am PC geschieht das unter Windows95 unter *Start/Einstellungen/ Systemsteuerung/ System/ Gerätemanager/ COMx-Anschluß/ Eigenschaften/ Anschluß Einstellungen/ Bits pro Sekunde*.

Manche serielle Verbindungen übertragen vor dem Stop-Bit noch ein zusätzliches Paritätsbit das die Parität des gesendeten Zeichens kennzeichnet. Dadurch kann der Empfänger das empfangene Zeichen auf Übertragungsfehler prüfen. Stimmt die, vom Empfänger ermittelte Parität des Zeichens und das Paritätsbit nicht überein, ist ein Bit während der Übertragung verfälscht worden, d.h. es ist eine "1" anstatt einer "0" empfangen worden oder umgekehrt. Es können allerdings nur ungradzahlige Verfälschungen erkannt werden. Diese Option wird von der Control-Unit nicht unterstützt.

### Belegung der Steckverbinder

Kontakt-Nr.	Belegung	Eingang/Ausgang
1	Masse (GND)	
2	Referenzspann. f. d. A/D-Wandl. (UREF)	E
3	A/D-Port 1	E
4	A/D-Port 2	E
5	A/D-Port 3	E
6	A/D-Port 4	E
7	A/D-Port 5	E
8	A/D-Port 6	E
9	A/D-Port 7	E
10	A/D-Port 8	E
11	Programmzyklus-Puls; wird mit jeder Programmabarbeitung umgeschaltet	A
12	Digitalport 1	E/A
13	Digitalport 2	E/A
14	Digitalport 3	E/A
15	Digitalport 4	E/A
16	Digitalport 5	E/A
17	Digitalport 6	E/A
18	Digitalport 7	E/A
19	Digitalport 8	E/A
20	Betriebsspannung +5 V	

## Belegung von Steckbuchse 2

Kontakt-Nr.	Bezeichnung	Eingang/Ausgang
1	Betriebsspannung +5 V	
2	Digitalport1	E/A
3	Digitalport2	E/A
4	Digitalport3	E/A
5	Digitalport4	E/A
6	Digitalport5	E/A
7	Digitalport6	E/A
8	Digitalport7	E/A
9	Digitalport8	E/A
10	Start-Eing. (parall. z. Start-Taster)	E
11	Daten-Leitung des I2C-Busses (SDA) (reserviert und nicht benutzbar)	E/A
12	Takt-Leitung des I2C-Busses (SCL) (reserviert und nicht benutzbar)	E/A
13	RUN-LED-Kathode	A
14	ACTIVE-LED-Kathode	A
15	DCF-OK-LED-Kathode	A
16	D/A-Wandler 1	A
17	D/A-Wandler 2	A
18	DCF77-Eingang	E
19	Reset-Eing. (parall. z. Reset-Taster)	E
20	Masse (GND)	

## Technische Daten

Abmessungen: 80\*50\*15 mm (viertel Europaformat)  
Betriebsspannung Ub: 5 V stabilisierte Gleichspannung,  $\pm 0,5V$   
Stromaufnahme: ca. 30 mA  
< 10 mA bei abgeschalteten LEDs und RS232-Interface  
Mikrocontroller: Motorola MC68HC05B6  
4 MHz Taktfrequenz  
6 Kilobyte maskenprogrammiertes Betriebssystem

Speicherchip für Anwenderprogramm und -daten: Microchip 24C65, serielles EEPROM mit I2C-Schnittstelle, 8k x 8 Bit oder Austauschtyp

A/D-Ports: 8 x 8 Bit A/D, 0...5 Volt gegen gemeinsame Masse  
Referenzspannung Uref einstellbar (normal Ub = Uref)  
Eingangsstrom ca. 10  $\mu A$  bei Wandlung  
absoluter Fehler  $\pm 1$  Digit (= 1/256 vom Meßbereichsendwert)  
zuzüglich Fehler der Referenzspannung

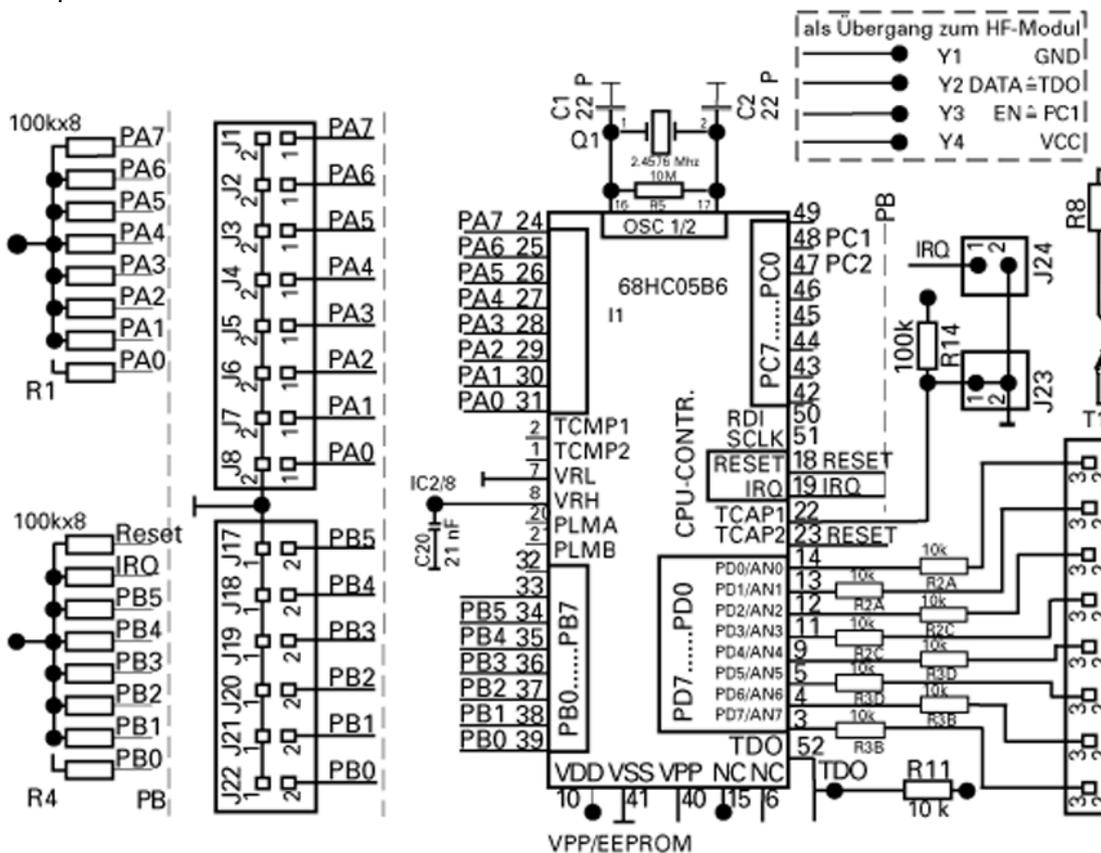
Digitalports: 16 Stück, frei als Ein- oder Ausgang programmierbar, 10 k Pull-Up  
Pegel (0,2 mA Last an Ausgängen):  
 $(U_b - 0,3 V) < U_{out,high} < (U_b - 0,1 V)$   
 $0,1 V < U_{out,low} < 0,3 V$   
 $(0,7 * U_b) < U_{in,high} < U_b$   
 $0 V < U_{in,low} < (0,2 * U_b)$   
maximal zulässiger Laststrom:  $\pm 10$  mA

Achtung!  
 Eine Überschreitung durch fehlende Strombegrenzung beim Anschluß von Spannungen an die Digitalports kann zur sofortigen Zerstörung des Mikrocontrollers führen!

D/A-Wandler : 2 pulswidenmodulierte Ausgänge, PWM-Rate 1953 Hz

DCF77-Eingang: Digitalport mit 10 k Pull-Up zum Anschluß einer DCF77-Aktivantenne mit Open-Collector Ausgang  
 serielle Schnittstelle: RS232 mit Pegelwandler MAX232 oder Austauschtyp  
 Übertragung mit 9600 Baud, 8 Bit,  
 1 Startbit, 1 Stoppbit, kein Paritätsbit, RTS/CTS- Hardware-Handshake-Verbindung mit dem PC über Nullmodemkabel

Schaltplan:



## Hardware

### C-Control-Starterboard

#### Allgemeine Handhabungs- und Sicherheitshinweise

Vor dem Berühren elektronischer Baugruppen sollten Sie einen gut geerdeten Gegenstand (PC-Gehäuse, Heizkörper, etc.) kurz anfassen, um die Gefahr der Beschädigung der Bauelemente auf der Platine durch elektrostatische Ladungen zu verringern. Überprüfen Sie den Inhalt der Verpackung auf Vollständigkeit. Achten Sie auf evtl. Transportschäden. Haben Sie die Baugruppe vom Kalten ins Warme gebracht, warten Sie bitte mit dem Anschluß der Versorgungsspannung, bis eventuell gebildetes Kondenswasser vollständig getrocknet ist.

Wählen Sie einen sicheren Standort, der zum einen gut zugänglich und einsehbar sein sollte und zum anderen den versehentlichen Kontakt insbesondere mit metallischen Gegenständen und Feuchtigkeit weitestgehend ausschließt. Die Baugruppe sollte nicht in unmittelbarer Nähe starker Magnetfelder (z.B. von großen Elektromotoren, Lautsprechern oder Netzteilen) aufgestellt werden. Bei einem Einsatz im Freien sollten blitzgefährdete Standorte, wie z.B. Bergkuppen, einzelstehende Masten, Dächer, etc. vermieden, bzw. mit speziellen Schutzeinrichtungen versehen werden.

Der Einsatz darf nur entsprechend der Technischen Spezifikation erfolgen. Dabei ist insbesondere auf die Einhaltung der Grenzwerte für die Betriebsspannung, die Eingangsspannung an den digitalen und analogen Eingängen und den zulässigen Strom für die digitalen Ausgänge zu achten.

Alle Verbindungen zwischen elektronischen Baugruppen oder zu einem PC (z.B. über das serielle Kabel) sind im spannungsfreien Zustand aller beteiligten Baugruppen herzustellen. Soll eine Verbindung unterbrochen werden, ist vorher die Betriebsspannung auch der Peripherie (also auch eines eventuell angeschlossenen PC) abzuschalten. Ein gemeinsamer Netzschalter für alle verbundenen Komponenten ist zu empfehlen.

Achten Sie beim Anschluß der Betriebsspannung auf die richtige Polung und den richtigen Spannungswert. Die Spannungsquelle darf nur im spannungsfreien Zustand angeschlossen werden.

Die Mißachtung dieser Hinweise kann zur Zerstörung der Baugruppe führen  
Aufbau und Funktion

Das C-Control-Starterboard dient ausschließlich der Unterstützung der C-Control-Unit. Auf dem Starterboard befinden sich im einzelnen folgende Funktionseinheiten:

- die 5V Spannungsstabilisierung für die Control-Unit
- eine 6,5V- Spannungsstabilisierung zur Versorgung externer Sensormodule
- zwei Referenzspannungen für die A/D-Wandlung:
- Auf dem Starterboard stehen die zwei Referenzspannungen 5 V und 2,5 V zur Verfügung, die über die Steckbrücke J7 mit dem Referenzspannungseingang der Control-Unit verbunden werden können.
- zwei Relais mit je einem Schließer und zugehöriger Ansteuerung:
- Die Relais sind per Steckbrücke (J8 und J10) mit den Digitalports 1 und 2 der Control-Unit verbunden. Werden die Relais nicht benötigt, sind die Steckbrücken abzuziehen. Die Digitalports stehen dann für eine andere Verwendung zur Verfügung. Die Relais verfügen über je einen Schließer im Arbeitskreis. Der Arbeitskontakt ist geschlossen, wenn der entsprechende Digitalport der Control-Unit High-Pegel führt (logisch "1"). Zu beachten ist dabei, daß alle Digitalports unmittelbar nach dem Einschalten der Betriebsspannung bis

- zum Start des Anwenderprogramms High-Pegel führen! Über die Relais können Kleinspannungsgeräte direkt geschaltet werden, deren Stromaufnahme den in den Technischen Daten spezifizierten Wert nicht überschreiten.
- Keinesfalls dürfen 230-V-Netzspannungsgeräte über die Relais betrieben werden!
- zwei RC-Glieder (Kombination von Widerständen und Kondensatoren) zur Demodulation der pulswidenmodulierten D/A-Ausgänge der Control-Unit
- ein Experimentierfeld von 14 x 14 Lötäugen (Pads)
- Hier können kleine Schaltungen direkt auf dem Starterboard aufgebaut werden.
- 2 x 20polige Stiftleisten (J1 und J2)
- Die Control-Unit wird zum Betrieb auf diese beiden Stiftleisten gesteckt. Dabei bieten drei Führungsstifte mechanisch Schutz vor Verpolung.
- 2 x 20polige Stiftleisten (J3 und J4), Signale der Control-Unit
- Auf diese Stiftleisten sind alle Signale der Control-Unit parallel herausgeführt. J3 entspricht in der Belegung der Steckbuchse 1 der Control-Unit, J4 der Steckbuchse 2. Kontakt 1 ist jeweils auf der Platine beschriftet
- 2 x 2 x 8polige Stiftleiste (J17 bis J32), Digitalports
- 3 x 8polige Stiftleiste (J33 bis J40), A/D-Eingänge der Control-Unit
- An diese Stiftleisten können Sensormodule angeschlossen werden. Sie werden mit einer 6,5 V Betriebsspannung versorgt, die vom Starterboard zur Verfügung gestellt wird
- 2 x 2polige Stiftleisten (J41 und J42), D/A-Ausgänge
- An J41 und J42 stehen die per RC-Glied demodulierten D/A-Ausgänge zur Verfügung. An den Widerständen R5 und R6 sind die Pins beschriftet, die noch die pulswidenmodulierten D/A-Signale (PLMA und PLMB) vor den RC-Gliedern führen und sind bei Bedarf dort abgreifbar

#### **Installation und Anschluß/ Aufstecken der C-Control-Unit**

Die beiden Buchsen der Control-Unit werden auf J1 und J2 des Starterboards aufgesteckt. Dabei ist die Lage der Führungsstifte des Starterboards, die vollständig in die dafür vorgesehenen Bohrungen der Control-Unit passen, zu beachten. Gewaltanwendung ist in jedem Fall zu vermeiden.

#### **Spannungsversorgung**

Zur Versorgung des Starterboards ist eine Gleichspannung von 8 V bis 12 V, nominal 9 V, an die zweipolige Schraubklemme (J5) des Starterboards anzuschließen. Dabei ist auf polungsrichtigen Anschluß gemäß der Beschriftung auf dem Board zu achten. Das Starterboard versorgt die aufgesteckte Control-Unit mit der erforderlichen 5 V Betriebsspannung. Eine getrennte Versorgung der Control-Unit mit einer externen 5 V Spannung ist nicht notwendig und nicht zulässig!

#### **Anschluß einer DCF77- Aktivantenne**

Zum Anschluß der DCF77-Aktivantenne an das Starterboard darf nur abgeschirmtes, dreiadriges Kabel verwendet werden. Es sollte nicht länger als drei Meter sein.

Der Schirm und die Masse-Ader des Kabels werden an die Klemme "GND" von J6 angeschlossen. Die Ader für die Betriebsspannung kommt an die Klemme "+ 5V" von J6. Die Signalader kommt an die Klemme "DCF IN" von J6.

Die nun angeschlossene DCF77-Aktivantenne muß so ausgerichtet werden, daß die grüne LED auf der aufgesteckten Control-Unit regelmäßig blinkt. Nach einigen Minuten sollte die Zeit-Synchronisation erfolgen und die grüne LED dauerhaft leuchten.

## Anschluß von Sensormodulen

Die getrennt erhältlichen Sensormodule werden zum Betrieb mit dem Starterboard auf einen der dreipoligen A/D-Steckverbinder (J33 bis J40) so aufgesteckt, daß die weiße Ader (Sensor-signal) zum Experimentierfeld weist.

Werden Sensormodule am Starterboard betrieben, muß die Steckbrücke J7 so gesetzt sein, daß die Control-Unit mit der 2,5V- Referenzspannung des Starterboards verbunden ist. Die dreipoligen Sensormodulkabel dürfen bei Bedarf um maximal zwei Meter verlängert werden.

Belegung der Steckverbinder

Pin- Nr	J1 und J3	J2 und J4
1	Masse (GND)	+5 V
2	Referenzspannung	Digitalport 1
3	A/D-Port 1	Digitalport 2
4	A/D-Port 2	Digitalport 3
5	A/D-Port 3	Digitalport 4
6	A/D-Port 4	Digitalport 5
7	A/D-Port 5	Digitalport 6
8	A/D-Port 6	Digitalport 7
9	A/D-Port 7	Digitalport 8
10	A/D-Port 8	Start-Eingang (parallel zum Start-Taster)
11	Programmzyklus-Puls	Daten-Leitung des I2C-Busses (SDA) *
12	Digitalport 9	Takt-Leitung des I2C-Busses (SCL) *
13	Digitalport 10	RUN-LED-Kathode
14	Digitalport 11	ACTIVE-LED-Kathode
15	Digitalport 12	DCF-OK-LED-Kathode
16	Digitalport 13	D/A-Wandler 1
17	Digitalport 14	D/A-Wandler 2
18	Digitalport 15	DCF77-Eingang
19	Digitalport 16	Reset-Eingang (parallel zum Reset-Taster)
20	+ 5 V	Masse (GND)

J17...32 (Digitalports): Alle Stifte dieser zweireihigen Steckverbinder, die zum Experimentierfeld weisen, sind signalführend, die Stifte, die zum Platinenrand weisen, führen Massepotential. Auf J17 liegt Digitalport 1, auf J32 liegt Digitalport 16.

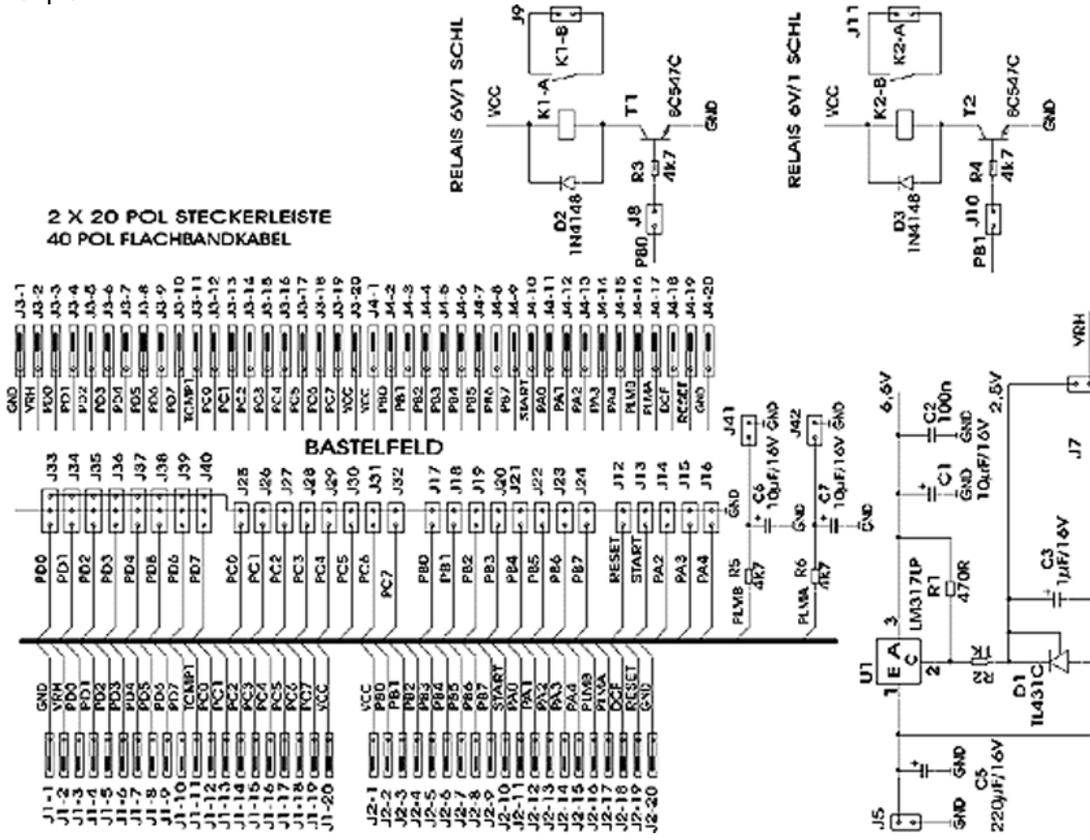
J33...40 (Digitalports): Alle Stifte dieser dreireihigen Steckverbinder, die zum Experimentierfeld weisen, sind signalführend, der jeweils mittlere Kontakt führt 6,5V zur Versorgung der Sensormodule, die Stifte, die zum Platinenrand weisen, führen Massepotential. Auf J33 liegt A/D-Port 1, auf J40 liegt A/D-Port 8.

J41...42 (Digitalports): Alle Stifte dieser zweireihigen Steckverbinder, die zum Experimentierfeld weisen, sind signalführend, die Stifte, die zum Platinenrand weisen, führen Massepotential. J41 liegt an D/A-Wandler 1, J42 liegt an D/A-Wandler 2.

## Technische Daten

Abmessungen:	160 mm x 100 mm (Euroformat)
Betriebsspannung Ub:	8...12 V Gleichspannung
max. zulässige Stromaufnahme der Experimentierschaltung	
aus der 5 V Spannung:	250 mA
Leistungsdaten der Relais:	max. 24 V, max. 3 A

Schaltplan:



## C-Control-Classic (V. 1.03b)

### Das Erstellen eines Programms mit der grafischen Entwicklungsumgebung

Die Programmierung erfolgt, indem mit dem Grafikeditor ein Blockschaltbild erstellt wird. Stellvertretend für jede Funktion, die C-Control ausführen soll, wird ein entsprechendes Symbol auf der Arbeitsfläche plaziert. Diese Symbole werden auch als Blöcke bezeichnet. Am linken Rand des Editorfensters befindet sich eine Tastenleiste, aus der die gewünschten Blöcke abgerufen werden können.

Die Tabelle listet die verfügbaren Blöcke und ihre grundsätzlichen Eigenschaften auf. Die Angaben zu Ein- und Ausgängen verstehen sich aus der Sichtweise des Programms und sind im ersten Moment vielleicht etwas verwirrend. Ein Block, der z.B. einen Eingang (Port) des Mikrocontrollers abfragt, liefert ja ein Signal "in das Programm hinein". Programmtechnisch betrachtet muß er aber einen Ausgang haben, an dem das betreffende Signal abgegriffen werden kann. Die Spalte Anzahl gibt an, wieviel Blöcke je Programm maximal eingesetzt werden können.

Kategorie	vorh. Blöcke	Anzahl	Funktion	Eingänge	Ausgänge
Ports	Eingang	16	Liest Digitalport aus	-	1 Bit
	Ausgang	16	Schreibt in Digitalport	1 Bit	-
	A/D-Wandler	8	Liest A/D-Port aus	-	1 Byte
	D/A-Wandler	2	Gibt Analogwert aus	1 Integer	-
Logik	Und	bel.	"Und"-Verknüpfung	2 ... 8 Bit	1 Bit
	Oder	bel.	"Oder"-Verknüpfung	2 ... 8 Bit	1 Bit
	Nicht	bel.	Negiert Signal	1 Bit	1 Bit
	Flag (1-Bit-Sp.)	8	Speichert Signal	-	1 Bit
	Timer	bel.	Zeitgest. Signal	-	1 Bit
Numerik	Summe	bel.	Addiert Zahlen	2 Integer	1 Integer
	Negator	bel.	Wechselt Vorzeich.	1 Integer	1 Integer
	Produkt	bel.	Multipliziert Zahlen	2 Integer	1 Integer
	Betragsbildner	bel.	Bildet Betrag	1 Integer	1 Integer
	Tabellenkonv.	bel.	Formt 8-Bit-Zahl um	1 Byte	1 Integer
	Konstante	bel.	Gibt Zahl aus	-	1 Integer
	Umschalter	bel.	Schaltet Signalw. um	2 Int. 1 Bit	1 Integer
	Speicher	8	Speichert Zahl	1 Integer	1 Integer
Vergleicher	Größer	bel.	Vergleicht Zahlen	2 Integer	1 Bit
	Kleiner	bel.	Vergleicht Zahlen	2 Integer	1 Bit
	Gleich 0	bel.	Prüft, ob Zahl 0 ist	1 Integer	1 Bit
	Gleich	bel.	Vergleicht Zahlen	2 Integer	1 Bit
	Maximum	bel.	Gibt größere Zahl weiter	2 Integer	1 Integer
	Minimum	bel.	Gibt kleinere Zahl weiter	2 Integer	1 Integer
Transfer	Text ausgeb.	bel.	Gibt Text seriell aus	1 Bit	-
	Ergeb. ausgeb.	bel.	Gibt Zahl seriell aus	1 Int. 1 Bit	-
	Zeit ausgeben	bel.	Gibt Uhrzeit seriell aus	1 Bit	-
Aufzeichnung	Logbuch lösch.	bel.	Löscht EEPROM	1 Bit	-
	Ergeb. aufz.	bel.	Schr. Zahl in EEPROM	1 Int. 1 Bit	-
	Zeitpunkt ausg.	bel.	Schreibt Uhrzeit in EEPROM	1 Bit	-

Ein Programm wird erstellt, indem die benötigten Blöcke auf der Arbeitsfläche des Editors platziert und in geeigneter Weise durch Signalflußlinien verbunden werden. Der Signalfluß durch diese Linien verläuft immer unidirektional, weshalb diese als Pfeile dargestellt werden. Die Art des übertragenen Signals richtet sich nach dem Ausgang des Blocks, von dem es abgegriffen wird.

Es kommen logische und numerische Signale vor. Logische Signale sind Bitsignale, d.h. deren Zustand kann entweder 0 oder 1 sein. Die Signalflußlinien können in diesem Fall als einpolige Leitung gedeutet werden. Alle Logikblöcke arbeiten mit solchen Signalen. Numerische Signale können entweder Byte- oder Integerstruktur haben. Bei einem Bytesignal können die Werte 0 ... 255, bei einem Integersignal die Werte von -32768 ... 32767 übertragen werden. Hier kann man sich die Signalflußlinie als Bussystem von 8 bzw. 16 parallel geführten Einzelleitungen vorstellen. Für alle Signalformen haben die Linien das gleiche Aussehen. Bei der Programmerstellung ist also mit einer gewissen Sorgfalt vorzugehen.

Man sollte sich bereits im Vorfeld genau darüber klar werden, wie das zu lösende Problem anzugehen ist. Es ist ratsam sich zumindest eine Skizze über die geplante Umsetzung anzufertigen. Für umfangreichere Projekte bieten sich mehrere Verfahren an, bei denen die Lösung einer gegebenen Aufgabe schrittweise erfolgt. Beispielsweise können mit Hilfe von Funktionstabellen und Karnaughdiagrammen auch sehr komplexe Zusammenhänge systematisch erfaßt, vereinfacht und in ein Blockschaltbild überführt werden. Einzelheiten hierzu finden Sie in der Fachliteratur zum Thema Steuerungstechnik. Programmablaufpläne sind in diesem Zusammenhang weniger geeignet, da sie von ihrer Struktur her auf zeilenorientierte Programme zugeschnitten sind.

Die Bedienung der grafischen Entwicklungsumgebung wird durch das enthaltene Hilfesystem unterstützt. Darin enthalten ist eine Referenzliste der zur Verfügung stehenden Funktionsblöcke (*Blockvorrat*). Zu jedem Block kann eine kurze Erläuterung aufgerufen werden. Das zeichnerische Erstellen eines Programms wird im Hilfefunkt "Bearbeiten von Blockschaltbildern" unterstützt. Einige zusätzliche Hinweise befinden sich in der Datei "readme.txt" auf der Programmdiskette.

*TIP:* Die grafische Entwicklungsumgebung verfügt nicht über die Möglichkeit, das erstellte Blockschaltbild auszudrucken. Durch gleichzeitiges Drücken der Tasten [Alt][Druck] wird ein Screenshot des aktuellen Fensters in die Zwischenablage gelegt. Diese kann dann in Paintbrush (oder einem anderen Pixelgrafikprogramm) sichtbar gemacht werden. Hierzu führen Sie die Menüfolge "Bearbeiten / Einfügen" aus. Danach können Sie das Blockschaltbild nach Belieben weiterverarbeiten.

Ein fertiges Blockschaltbild wird in einer Datei mit der Endung ".bnt" gespeichert. Vor dem Übertragen des Programms in den Mikrocontroller muß es kompiliert werden, d.h. aus der graphischen Lösung wird ein zeilenorientiertes Programm erzeugt. Es besteht aus einem Code, der vom Mikrocontroller verstanden werden kann. Dieser Vorgang erfolgt automatisch, wenn der sog. Lader aufgerufen wird.

Der Start und das Beenden des Programms kann wahlweise vom PC oder direkt auf der Control-Unit ausgelöst werden. Ist das Programm von der Entwicklungsumgebung aus gestartet worden, muß es auch von dieser beendet werden. Nach der Programmentwicklung sollte eine ausführliche Testphase erfolgen, um eventuelle Fehler aufzuspüren

Die folgende Demonstration zeigt exemplarisch den Vorgang der Programmerstellung mit C-Control anhand des im Abschnitt Quickstart gezeigten Programms.

## Die Grundregeln der Programmerstellung mit C-Control

1. Falls ein Block über einen Ausgang verfügt, kann dieser beliebig oft abgegriffen werden. (z.B. AND-, Timer, Additionsblock). Einige Blöcke haben allerdings keinen Ausgang (z.B. Eingangsport-Block). Da der Signalflußgraph dort endet, werden derartige Blöcke auch als Wurzelblöcke bezeichnet
2. Ein Block kann entweder keinen (z.B. Eingangsport-Block), einen oder mehrere Eingänge haben. Im letztgenannten Fall verarbeiten die Eingänge zumeist gleichartige Signale (z.B. AND, SUMME). Es existieren jedoch einige Blöcke, die zwar numerische Signale verarbeiten aber zusätzlich einen Steuereingang aufweisen, der mit einem logischen Signal angesteuert werden muß (z.B. Zeitaufzeichnungs-Block).
3. Ähnlich wie Eingänge, können auch Ausgänge entweder logischer oder numerischer Natur sein. Bei der Erstellung des Blockschaltbilds müssen Sie sicherstellen, daß jeder Eingang das ihm zugewiesene Signal auch verarbeiten kann. Logische Eingänge können nur von logischen Ausgängen, numerische Eingänge nur von numerischen Ausgängen angesteuert werden.
4. Ein Blockschaltbild enthält einen oder mehrere Signalflußgraphen, die die Control-Unit voneinander unabhängig bearbeitet. Jeder Signalflußgraph endet in genau einem Ausgangsport-, Transfer- oder Aufzeichnungsblock und fächert sich baumartig bis zu einem oder mehreren Eingangsport- oder Konstantenblöcken auf. Dabei können mehrere Signalflußgraphen auf dieselben Eingangsport-Blöcke und Konstantenblöcke zugreifen.
5. Normalerweise werden die Signalflußgraphen beim Programmdurchlauf in der selben Reihenfolge abgearbeitet, in der die zugehörigen Wurzelblöcke im Blockschaltbild platziert worden sind. Wenn eine andere Abfolge gewünscht wird, kann diese jedoch frei programmiert werden.
6. Für die Verwendung von Blöcken, die zwei numerische Eingänge und einem numerischen Ausgang besitzen (z.B. Summe), gilt folgende Einschränkung: Von den beiden Eingangszweigen dieser Blöcke, darf nur einer wieder einen Block der beschriebenen Art enthalten.
7. Für die Verwendung von AND- und OR-Blöcken gilt folgende Einschränkung: Von allen Eingangszweigen eines AND- oder OR-Blocks darf nur ein Zweig wieder einen AND- oder OR-Block enthalten.

### Programmierbeispiel 1:

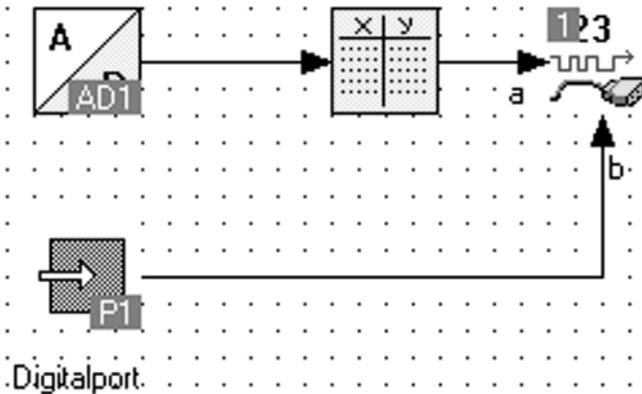
Ausgabe eines Temperatur-Meßwertes über die serielle Schnittstelle

Das Programm soll über den A/D-Port AD1 eine Spannung messen, die von einem KTY-Sensormodul geliefert wird. Mit Hilfe der Tabelle KTY.TAB, welche in Form einer gleichnamigen Datei hinterlegt ist, soll das eingelesene Sensorsignal in °C umgerechnet. Immer dann, wenn ein Taster (Öffner) gedrückt wird, der an den Digitalport P1 angeschlossen ist, soll der ermittelte Temperaturwert im über die serielle Schnittstelle ausgegeben werden.

Programmaufbau:

.Ausgabe eines Temperaturwertes auf die serielle Schnittstelle

.KTY-Sensormodul. .Tabelle.1 (KTY.TAB)



[Das Programm temp.bnt]

Vom A/D-Wandler-Block wird ein numerisches Signal (Byte-Format) abgegriffen. Der Ausgang des entsprechenden Blocks ist mit dem Eingang eines Tabellen-Blocks verbunden. Diesem ist die Tabelle KTY.TAB zugewiesen. Der Ausgang des Tabellenblocks gibt ein numerisches Signal im Integer-Format ab, welches auf den Eingang eines Ergebnisausgabe-Blocks wirkt. Dieser benötigt noch ein logisches Steuersignal. Es wird vom Digitalport P1 entnommen. Im Programm wird zu diesem Zweck ein Eingangsport-Block eingesetzt, dessen Signal dann den Steuereingang versorgt.

#### Programmablauf:

Am Ausgang des A/D-Wandler-Blocks steht das Ergebnis der A/D-Wandlung an. Es handelt sich dabei um eine Zahl zwischen 0 und 255 (Byte-Format). Sie wird an den Eingang des Tabellen-Blocks weitergegeben. Hier wird in der Tabelle KTY.TAB die entsprechende Zeile aufgesucht. Der dort aufgefundene Wert steht nun am Ausgang des Tabellen-Blocks zur Verfügung und wird dem Eingang a des Ergebnisaufzeichnungsblocks übergeben. Dieser überprüft nun den Zustand seines Steuereingangs b. Bei nicht gedrücktem Taster erhält dieser vom Ausgang des Eingangsport-Blocks ein 0-Signal. In diesem Fall verhält sich der Ergebnisausgabe-Block passiv, d.h. es findet keine Datenübertragung über die serielle Schnittstelle statt.

Bei gedrücktem Taster erhält der Steuereingang ein 1-Signal und der vom Tabellen-Block gelieferte Wert wird über die serielle Schnittstelle ausgegeben. Die Ausgabe erfolgt in Form einer nullterminierten Zeichenkette (C-String). Es handelt sich dabei um folgendes Format: (-)#####\0\n. "\0" steht für das NULL-Zeichen, "\n" für eine Zeilenvorschub. "(-)" bedeutet, daß optional ein "-"-Zeichen vorangestellt wird (bei negativen Meßwerten).

Im angeschlossenen PC wird die Datenübertragung von der C-Control-Software registriert und kann dort angezeigt oder weiterverarbeitet werden. Der gesamte Vorgang läuft einmal je Programmzyklus des Mikrocontrollers ab. Bei fortlaufend gedrücktem Taster hängt die Häufigkeit, mit der Daten übertragen werden, von der eingestellten Wiederholrate ab. Sie kann 50 Millisekunden bis 1 Minute betragen.

## Arbeiten mit dem Tabellenkonverter-Block

Das Programm veranschaulicht, wie Tabellen zur bequemen Umrechnung von Meßwerten herangezogen werden können. Jeder an der Tabelle anliegende Wert (0 ... 255) adressiert eine Tabellenzeile, deren Inhalt (Integerwert) dann auf den Ausgang des Tabellenblocks gelegt wird. Zusammen mit dem Programm können maximal 2 Tabellen an den Mikrocontroller übergeben werden. Es können jedoch beliebig viele Tabellenkonverter eingesetzt werden, die dann jeweils eine dieser beiden Tabellen zur Grundlage haben.

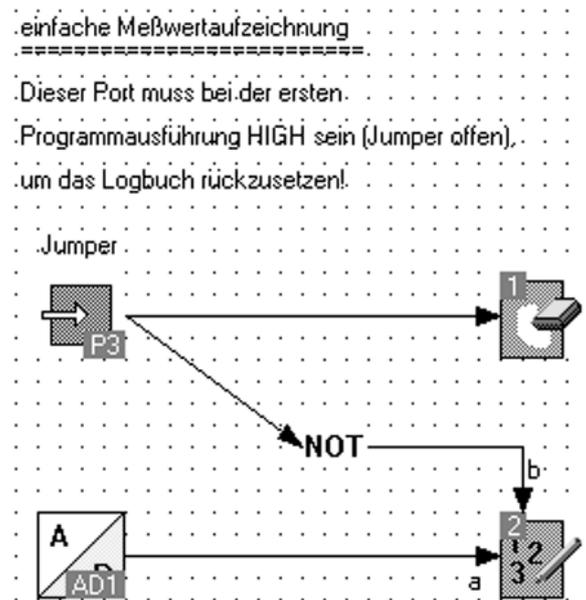
Die Zuweisung der Tabellen erfolgt über die Menüfolge "*Einstellungen / Tabelle...*". Dort kann eine Text-Datei mit der Endung "TAB" bestimmt werden, deren Inhalt übertragen wird. Die Datei muß 256 Zeilen enthalten, wobei in jeder Zeile eine Integerzahl stehen soll. Die erste Zeile enthält die Zahl, die dem Eingangswert 0 zugeordnet werden soll usw. Eine derartige Datei kann leicht vom Anwender selbst erstellt werden. Sämtliche Tabellendateien müssen im Verzeichnis /TAB abgelegt werden.

## Programmierbeispiel 2:

Aufzeichnung eines Meßwertes im Speicher der Control-Unit

Das Programm soll einmal je Zyklus den Analogport AD1 einlesen. Der so erhaltene Wert soll auf dem EEPROM der Mikrocontroller-Platine gespeichert werden. Dies soll solange erfolgen, wie der Jumper auf Digitalport P3 gesteckt ist. Wird er gezogen, soll der Inhalt des EEPROMs gelöscht werden. Sobald der Jumper wieder aufgesteckt wird, soll die Aufzeichnung von Neuem beginnen.

Programmaufbau:



[Das Programm aufzeich.bnt]

Der Eingangsport-Block stellt an seinem Ausgang das logische Signal des Digitalports P3 zur Verfügung. Dieses verzweigt sich zu den Eingängen eines Logbuch-löschen- und eines Nicht-Blocks. Der Ausgang des Nicht-Blocks stellt ein negiertes Signal zur Verfügung, mit welchem der Steuereingang b eines Ergebnis-aufzeichnen-Blocks angesprochen wird. Der Signaleingang a dieses Blocks erhält vom Ausgang eines A/D-Wandler-Blocks ein numerisches Signal im Byte-Format. Dieses stammt vom Analogport A/D1.

### **Programmablauf**

Bei gezogenem Jumper steht am Ausgang des Eingangsport-Blocks ein 1-Signal an. Dieses veranlaßt den Logbuch-löschen-Block dazu, den Datenspeicher des EEPROMs zu löschen. Der Steuereingang erhält wegen des vorgeschalteten Nicht-Blocks ein 0-Signal und verhält sich deshalb passiv.

Bei gestecktem Jumper liegen genau umgekehrte Verhältnisse vor. Der Logbuch-löschen-Block verhält sich passiv, da an seinem Eingang ein 0-Signal ansteht. Der Ergebnis-aufzeichnen-Block erkennt an seinem Steuereingang b ein 1-Signal. Er schreibt deshalb das numerische Signal, welches ihm vom A/D-Wandler-Block an seinen Signaleingang a geliefert wird, ins EEPROM.

Bleibt der Jumper andauernd gesteckt, erfolgt bei jedem Zyklus des Mikrocontrollers eine Aufzeichnung. Die Häufigkeit, mit der Daten aufgezeichnet werden richtet sich dann also von der eingestellten Wiederholrate ab. Sie kann 50 Millisekunden bis 1 Minute betragen. Das EEPROM verfügt über Kbyte Speicherplatz. Jede Aufzeichnung belegt davon 2 Byte. Theoretisch finden als 4096 Aufzeichnungen darin Platz.

## C-Control-Basic

### Die Erstellung eines Programmes mit der Integrierten Entwicklungsumgebung

Vor dem eigentlichen Schreiben eines noch so kleinen Programmes sollte man sich immer so genau wie möglich über das zu lösende Problem im klaren werden. Anschließend ist es ratsam sich ein schrittweises Konzept für die Lösung der Aufgabenstellung zu erarbeiten. Für die schriftliche Fixierung eignet sich z.B. ein sogenannter Programmablaufplan. Dann muß dieses Konzept in eine Folge von CC-BASIC-Befehlen umgesetzt werden. Dieser sogenannte Quelltext wird dazu mit Hilfe des Editors der Integrierten Entwicklungsumgebung in eine Datei mit der Endung ".bas" geschrieben. Ist der Quelltext vollständig, muß er durch Aufruf des Compilers der Integrierten Entwicklungsumgebung in eine für den Prozessor verständliche Form übersetzt werden. In der Regel ist der ursprüngliche Quelltext nie fehlerfrei. Die vom Compiler entdeckten Fehler werden in einem separaten Fenster unter Angabe der fehlerhaften Zeile und der Fehlerursache angezeigt. Nach der Korrektur des Quelltextes ist dann ein neuer Compiler-Lauf notwendig. Ist das Programm fehlerfrei kompiliert worden und somit frei von Formfehlern, muß es auf logische Fehler überprüft werden (d.h. löst es die Aufgabenstellung wie geplant?). Dazu ist in der Integrierten Entwicklungsumgebung ein Simulator enthalten, der den Programmablauf auf dem PC nachbildet. Man ist dadurch in der Lage, das Programm an einem Haltepunkt anzuhalten, das Programm zeilenweise abzuarbeiten und sich den Zustand einzelner Variablen oder Ports anzeigen zu lassen. Werden Fehler entdeckt, müssen diese im Quelltext oder sogar im Konzept korrigiert werden. Ist man der Meinung, daß das Programm das bewerkstelligt was man will, kann man es in die C-Control-Unit laden. Auch jetzt können sich noch Fehler (z.B. auch mit elektronischen Ursachen) bemerkbar machen. Es kann u.U. Fehler geben, die sich erst nach Monaten und Jahren äußern (auch bei professionellen Anwendungen).

### Erläuterung der Programmiersprache C-Control-BASIC

#### Was ist C-Control-Basic?

Jeder Mikroprozessor verfügt über einen spezifischen Befehlssatz, mit dem einzelne Bytes gelesen, manipuliert (z.B. negiert) oder miteinander verknüpft (z.B. addiert, subtrahiert) und ausgegeben werden können. Die Programmierung mit Hilfe dieser Befehle (Assemblerprogrammierung) ist sehr mühselig und fehleranfällig. Der Programmierer muß sozusagen jedes Byte seines Programmes kennen und bearbeiten. Hinzu kommt, daß derartige Programme sehr unübersichtlich und schwer verstehbar sind. Sie lassen sich nachträglich nur schwer erweitern und modifizieren. Deshalb wird diese Programmierung nur angewendet, wenn es um Programmteile geht, die sehr schnell sein müssen und dennoch nicht allzuviel Speicher belegen dürfen. Für den Großteil der Programmieraufgaben werden sogenannte höhere Programmiersprachen eingesetzt, von denen es eine große Vielzahl gibt (z.B. PASCAL für technisch-mathematische Aufgabenstellungen, C für Systemprogrammierung, BASIC für einfache Mehrzweckanwendungen, usw.). Höhere Programmiersprachen verfügen über Befehle mit denen der Programmierer auf einfache Weise die gängigen Aufgabenstellungen lösen kann. So ermöglicht z.B. der CC-BASIC Befehl **SQR(x)** das einfache Ziehen der Quadratwurzel (engl.: Square Root) aus der Zahl x. Die Programmierung dieser Aufgabe würde auf der Assemblerebene (also in der eigentlichen Programmiersprache des jeweiligen Prozessors) ungleich mehr Kenntnisse und Arbeitsaufwand erfordern. Programme, die in einer höheren Programmiersprache geschrieben sind, sind außerdem besser lesbar (die Befehle orientieren sich eher an der menschlichen Sprache als an dem Befehlssatz des jeweiligen Prozessors) und können auch leicht auf Prozessoren eines anderen Typs übertragen werden (Portabilität). CC-BASIC ist ein Dialekt der populären und einfach zu erlernenden Programmiersprache BASIC (Abk. für Beginner's All Purpose Instruction Code = Allzwecksprache für Anfänger).

## **Aufbau und Bestandteile eines CC-BASIC-Programmes**

(Beispiel BSP01.BAS)

Ein CC-BASIC-Programm besteht aus zeilenweise angeordneten Programmanweisungen. Die einzelnen Elemente einer Zeile müssen mindestens durch ein Leerzeichen voneinander getrennt werden. Zeilen müssen nicht, wie bei anderen BASIC-Dialekten üblich, mit Zeilennummern versehen werden. Werden Zeilennummern angegeben, können sie als Marken dienen, haben aber im übrigen keinen Einfluß auf den Programmablauf, d.h. im Programm BSP01.BAS wird trotz der höheren Zeilennummer zuerst der Befehl PRINT "HALLO" abgearbeitet und dann der Sprungbefehl GOTO 20, der wieder zur ersten Zeile zurückspringt. Auch die Fehlermeldungen des Compilers beziehen sich auf die Zeilennummern des Editors (in der Entwicklungsumgebung unten links angegeben). Zeilen, die Steueranweisungen für den Compiler (z.B. DEFINE) enthalten, zählen nicht zum eigentlichen Programm und dürfen nicht mit Zeilennummern versehen werden.

(Beispiel BSP02.BAS)

Innerhalb von CC-BASIC-Programmen wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Dieses Programm führt die gleichen Schritte aus, wie BSP01.BAS. Viele Programmierer verwenden allerdings Großschreibung für vordefinierte BASIC-Sprachelemente (reservierte Worte), um sie besser von selbst zugewiesenen Bezeichnern abzuheben. Reservierte Worte sind alle im Sprachumfang enthaltenen Bezeichner. Der Programmierer darf diese Worte nicht für seine selbst festgelegten Namen verwenden.

(Beispiel BSP03.BAS)

Steht in einer Zeile (auch als erstes Zeichen) das Hochkomma ' , so wird dadurch der Rest der Zeile zum Kommentar erklärt. Ausführliche Kommentierung verbessert die spätere Lesbarkeit des Programmes.

(Beispiel BSP04.BAS)

In einer Programmzeile können auch mehrere Anweisungen enthalten sein, die dann durch einen Doppelpunkt : voneinander getrennt sein müssen. Steueranweisungen für den Compiler müssen immer einzeln in einer Zeile stehen.

Elemente eines CC-BASIC-Programmes sind Compileranweisungen, Variablen und Konstanten, Marken, Terme, Operanden und Operatoren, Funktionen und Befehle.

### **Steueranweisungen für den Compiler**

Compileranweisungen steuern den Übersetzungsvorgang des Compilers, d.h. sie stellen keine Handlungsanweisungen an den Steuercomputer dar. Sie haben zur Laufzeit des Programms keine Funktion. Compileranweisungen sind die Definition von Variablen, symbolischen Konstanten, Portbezeichnern (alle mit der DEFINE-Anweisung), Anweisungen zum Anlegen von Tabellen (TABLE ... TABEND) oder die Einbindung von Assemblerprogrammen (SYSCODE).

- Definition von symbolischen Konstanten (s.a. Variablen und Konstanten)

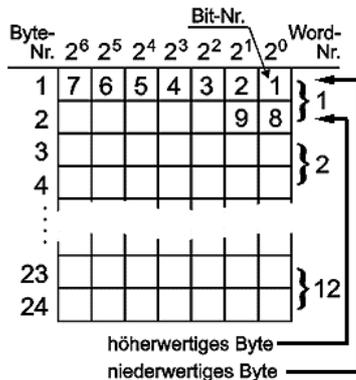
DEFINE konstante wert

"konstante" ist ein symbolischer Name für den numerischen Wert "wert". Wird die symbolische Konstante im Quelltext verwendet, ersetzt der Compiler sie beim Übersetzungsvorgang durch den entsprechenden numerischen Wert. Konstanten müssen im Quelltext immer vor ihrer ersten Verwendung definiert sein.

- Definition von Variablen (s.a. Variablen und Konstanten)

Der Name einer Variablen repräsentiert immer einen Speicherplatz im internen RAM des Mikrocontrollers. Von den insgesamt 240 Byte, über die der Mikrocontroller verfügt, stehen für benutzerdefinierte Variablen 24 Byte zur Verfügung. Diese Bytes können bitweise, als Bytes oder auch als 16-Bit-Integerzahl angesprochen werden (siehe Abb.).

### Speicheraufteilung



Der Typ wird bei der Definition festgelegt. Die Speicherzelle, die die Variable verwenden soll, kann (BYTE, WORD) oder muß (BIT) bei der Definition festgelegt werden. Der Programmierer muß selbst darauf achten, daß es durch Überlappungen im Speicherbereich nicht zu Programmfehlern kommt. (BIT[18], BYTE[2] und WORD[1] belegen gemeinsam einen Teil der Speicherzelle 2). Variablen müssen im Quelltext immer vor ihrer ersten Verwendung definiert worden sein.

- Definition einer Bit-Variablen

```
DEFINE bitvar BIT[bitnr]
```

"bitvar" ist ein symbolischer Name für die definierte Bitvariable. BIT[bitnr] legt den Typ und die verwendete Speicherzelle fest (bitnr muß einen Wert im Bereich von 1...192 haben).

- Definition einer Byte-Variablen

```
DEFINE bytevar BYTE[bytenr]
```

"bytevar" ist ein symbolischer Name für die definierte Variable. BYTE[bytenr] legt den Typ und die verwendete Speicherzelle fest ("bytenr" muß einen Wert im Bereich von 1...24 haben).

- Definition einer Word-Variablen

```
DEFINE wordvar WORD[wordnr]
```

"wordvar" ist ein symbolischer Name für die definierte Variable. WORD[wordnr] legt den Typ und die verwendete Speicherzelle fest ("wordnr" muß einen Wert im Bereich von 1...12 haben).

Läßt man für Variablen vom Typ WORD und BYTE die eckigen Klammern und die Angabe des Speicherplatzes weg, weist der Compiler den Variablen automatisch einen Platz zu (Er beginnt dabei mit der Nummer 1). Da Variablen vom Typ WORD nur auf Speicherplätze mit

ungerader Nummer (1,3,5,...) gelegt werden können, führen folgende Anweisungen dazu, daß jedes 2. Byte unbelegt bleibt und somit nicht nutzbar ist.

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

Besser wäre deshalb:

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

Bei der gemischten Definition von automatisch und manuell zugewiesenen Variablen muß wiederum auf möglicherweise entstehende Überlappungen im Speicherbereich geachtet werden.

- Definition von Digitalports

Die C-Control-Unit verfügt über 16 digitale Ports, die wahlweise als Eingang oder Ausgang definiert werden können. Diese Ports werden in CC-BASIC wie Variablen behandelt. Ports können bitweise, als Byteport (8 Bit breit) oder auch als ein 16 Bit breiter Port (Wordport) angesprochen werden.

- Definition von Bit-Ports

```
DEFINE portname PORT[portnr]
```

"portname" ist ein symbolischer Name für den definierte Port. "portnr" legt den verwendeten Port fest ("portnr" muß einen Wert im Bereich von 1...16 haben).

- Definition von Byte-Ports

```
DEFINE portname BYTEPORT[portnr]
```

"portname" ist ein symbolischer Name für den definierte Port. "portnr" legt den verwendeten Port fest ("portnr" muß einen Wert im Bereich von 1...2 haben).

- Definition eines WORD-Ports

```
DEFINE portname WORDPORT[portnr]
```

"portname" ist ein symbolischer Name für den definierten Port. "portnr" legt den verwendeten Port fest ("portnr" kann nur den Wert 1 haben).

- Definition von analogen Ports

Die C-Control-Unit verfügt über acht A/D-Ports, an denen analoge Eingangssignale als digitale Werte eingelesen werden können und zwei D/A-Ports, die digitale Werte als PWM-Signale (pulsweitenmoduliert) ausgeben können. Diese Ports werden in CC-BASIC wie Variablen behandelt. Die Definition erfolgt ähnlich der von Variablen:

- Definition eines A/D-Ports

DEFINE portname AD[portnr]  
Für "portnr" sind Werte von 1 bis 8 zulässig

- Definition eines D/A-Ports

DEFINE portname DA[portnr]  
Für "portnr" sind die Werte von 1 und 2 zulässig

## Variablen und Konstanten, Datentypen

Variablen und Konstanten speichern einen numerischen Wert (eine Zahl), also keine Zeichenketten (z.B. "HALLO"), wie in anderen Programmiersprachen üblich. Konstanten repräsentieren dabei einen unveränderlichen Wert, der Inhalt einer Variablen kann während des Programmablaufes beliebig oft verändert werden. Variablen und auch Konstanten haben in CC-BASIC immer die Größe von 1- (BIT), 8- (BYTE) oder 16 Bit (WORD). Es werden ausschließlich ganzzahlige Werte (Integerzahlen) verarbeitet.

(Beispiel BSP05.BAS)

Konstanten können direkt durch Angabe ihres Zahlenwertes auf dezimaler, hexadezimaler oder binärer Zahlenbasis verwendet werden. Übersichtlicher ist es, am Programmanfang einen symbolischen Bezeichner, also einen Namen, für die Konstante zu vereinbaren (mit Hilfe der DEFINE-Anweisung), der dann im Programm beliebig oft anstelle des Zahlenwertes hingeschrieben werden kann.

(Beispiel BSP06.BAS)

Einer Variablen wird zunächst durch die DEFINE-Anweisung ein symbolischer Name verliehen. Bei dieser Definition muß angegeben werden, von welchem Datentyp die Variable sein soll. Im Programm kann ihr dann unter diesem Namen beliebig oft ein Wert zugewiesen bzw. aus ihr gelesen werden.

(Beispiel BSP07.BAS)

Variablen vom Typ BIT können die Werte 0 oder 1, vom Typ BYTE 0...255 und vom Typ WORD -32768...32767 annehmen (s.a. Definition von Variablen. Dieser Wertebereich muß vom Programmierer immer berücksichtigt werden. Wird er überschritten kommt es zu falschen Ergebnissen. Addiert man beispielsweise zu einer 8 Bit-Zahl, die den Inhalt 255 hat (als Binärzahl geschrieben: &B11111111), die Zahl 1 (&B00000001), so kommt es zu einem Überlauf. Das Ergebnis 256 (&B10000000) erfordert zu seiner Darstellung 9 Bit und paßt somit nicht mehr in eine 8 Bit-Zahl. Die führende 1 wird abgeschnitten und in der Variable stehen nur die letzten 8 Bit. Die Variable enthält also nach der Addition den Wert 0 (&B00000000). Diese Effekt ähnelt dem Überlauf eines Kilometerzählers im Auto nach Überschreiten des größtmöglichen Kilometerstandes (Der Kilometerzähler zeigt dann den Wert 0 an).

Zum Zugriff auf die Echtzeituhr sind die folgenden Variablen vordefiniert und werden vom Betriebssystem ständig aktualisiert. Sie können vom Benutzer nicht geschrieben werden.

YEAR	Jahr (0...99)
MONTH	Monat (1...12)
DAY	Tag des Monats (1...31)
DOW	Wochentag (0=Sonntag ... 6= Sonnabend)

HOUR	Stunde (0...23)
MINUTE	Minute (0...59)
SECOND	Sekunde (0...59)

Der Timer des Systems kann über den vordefinierten Bezeichner TIMER ausgelesen werden. Ein Rücksetzen oder Schreiben des Timers ist nicht möglich.

## Marken

(Beispiel BSP08.BAS)

Eine Marke (engl.: Label) gibt einer Zeile einen symbolischen Namen (im Beispiel "marke1"), der dann als Ziel für Programmsprünge (GOTO) oder Unterprogrammaufrufe (GOSUB) dienen kann. Sie müssen bei ihrer Definition am Anfang einer Zeile stehen und mit einem führenden Doppelkreuz # versehen werden. Nach der Marke können noch Programmanweisungen folgen, meist wird jedoch aus Gründen der Übersichtlichkeit eine extra Zeile verwandt. Beim Aufruf als Sprungziel muß das Doppelkreuz weggelassen werden.

## Operanden und Operatoren

(Beispiel BSP09.BAS)

Operanden können Konstanten, Variablen (s.a. Variablen und Konstanten), ein Funktionsaufruf, aber auch ein wiederum aus Operanden und Operatoren zusammengesetzter Term sein.

Operatoren beeinflussen einen Operanden oder verknüpfen mehrere Operanden. Das Ergebnis kann einer Variablen zugewiesen oder direkt in einem Term verwendet werden. In CC-BASIC werden folgende Operatoren bereitgestellt:

- Mathematische Operatoren: (Beispiel BSP10.BAS)

+	Addition
-	Subtraktion
*	Multiplikation
/	Integer Division
MOD	Rest einer Division

- Vergleichsoperatoren: (Beispiel BSP11.BAS)

>	größer als
<	kleiner als
>=	größer oder gleich
<=	kleiner oder gleich
=	gleich
<>	ungleich

Vergleichsoperationen haben als Ergebnis den Wert "wahr" (repräsentiert durch den numerischen Wert 255 oder -1) oder "unwahr" (repräsentiert durch den Wert 0).

- Logische Operatoren: (Beispiel BSP12.BAS und BSP13.BAS)

NOT	Negation
AND	UND-Verknüpfung
NAND	UND-Verknüpfung mit anschließender Negation

OR	ODER-Verknüpfung
NOR	ODER-Verknüpfung mit anschließender Negation
XOR	EXKLUSIV-ODER-Verknüpfung

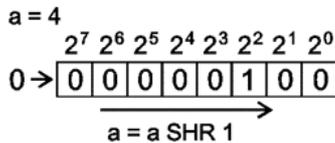
Die logischen Operatoren können einerseits zur Verknüpfung logischer Aussagen verwendet werden BSP12.BAS. Sie werden dann für sogenannte bedingte Programmausführungen verwendet. Andererseits können diese Operatoren auch Bit-, Byte- und Word-Variablen bitweise manipulieren BSP13.BAS. Bitweise heißt, das die gleichwertigen Bits zweier Werte miteinander logisch verknüpft werden. Werden beispielsweise die beiden Zahlen &B00000011 und &B01000001 durch den AND-Operator UND-verknüpft, ist das Ergebnis die Zahl &B00000001.

- Schiebeoperatoren (Beispiel BSP14.BAS)

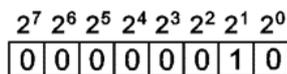
Die Schiebeoperatoren SHL (engl.: SHift Left = Linksschieben) und SHR (engl.: SHift Right = Rechtsschieben) dienen dem bitweisen Verschieben von Bitmustern in BYTE- oder WORD-Variablen. Links vom Operator steht der Name der Variablen, deren Inhalt verschoben werden soll, rechts steht die Anzahl der Stellen, um die der Inhalt verschoben werden soll. Die freigewordenen Stellen werden mit Nullen aufgefüllt, die hinaus geschobenen Bits gehen verloren. Die Rechtsverschiebung um eine Position ist dabei gleichwertig mit einer Division durch 2, da alle Bits auf die nächst niedrigere Position (die ja um den Faktor 0.5 weniger zählt) verschoben werden. Eine Linksverschiebung um eine Position entspricht einer Multiplikation mit 2. (siehe Abb.)

**Verschiebung des Wertes 4  
um eine Stelle nach rechts**

Inhalt von a vor der  
Verschiebung:



Inhalt von a nach der  
Verschiebung (a = 2):



- Der Klammeroperator und die Rangfolge der Operatoren (Beispiel BSP15.BAS)

Enthält ein Ausdruck (z.B. 2 \* 2 + 4) mehrere Operanden, stellt sich die Frage, in welcher Reihenfolge diese Operationen durchgeführt werden. Die Operatoren in CC-BASIC werden mit einer gewissen Rangfolge behandelt (ähnlich der mathematischen Regel "Punktrechnung geht vor Strichrechnung"). Für CC-BASIC ist die in Tabelle 1 aufgeführte Rangfolge von Operatoren und Funktionsaufrufen festgelegt worden. Operatoren gleicher Rangfolge werden innerhalb einer Zeile von links nach rechts abgearbeitet.

Rang	Operatoren
9 (höchster)	()
8	Funktionsaufrufe
7	Negatives Vorzeichen

6	* / MOD SHL SHR
5	+ -
4	> >= < <= = <>
3	NOT
2	AND NAND
1	OR NOR XOR

Ungeachtet dessen kann die Reihenfolge der Bearbeitung durch das Setzen von Klammern beeinflusst werden. CC-BASIC unterstützt maximal drei Klammerebenen.

## Funktionen und Befehle

(Beispiel BSP16.BAS)

CC-BASIC enthält einige einfache mathematische Funktionen. Den meisten Funktionen wird in Klammern, die sich unmittelbar an den Funktionsnamen anschließen, ein oder mehrere Argumente übergeben (z.B. steht x in **SQR(x)** für einen Term, der das Argument dieser Funktion darstellt). Einige Funktionen benötigen aufgrund ihrer Aufgabe keinen Übergabewert (z.B.: RAND, Erzeugen einer Zufallszahl). Funktionen führen genau definierte Operationen durch (z.B. Ziehen der Quadratwurzel aus dem übergebenen Argument). Das Ergebnis eines Funktionsaufrufes kann einer Variablen zugewiesen oder weiter verwendet werden. Die Definition eigener Funktionen wie in einigen anderen Programmiersprachen üblich, ist in CC-BASIC nicht vorgesehen.

Befehle unterscheiden sich von Funktionen dadurch, daß sie als Ergebnis keinen Wert liefern. Sie führen ebenso wie Funktionen eine bestimmte Operation durch, deren Inhalt aber nicht die Berechnung eines Wertes ist (z.B. hält der Befehl PAUSE 50 die Programmausführung für die Zeit von 50 \* 20 ms an). Einige Befehle erwarten, ähnlich wie Funktionen, einen oder mehrere Parameter. Diese Parameter werden im Anschluß an den Befehlsbezeichner, von diesem durch mindestens ein Leerzeichen getrennt, und untereinander durch ein Komma getrennt, angegeben. Eine Ausnahme bildet der PRINT-Befehl.

## Anweisungen zur Steuerung des Programmflusses

Anweisungen zur Steuerung des Programmflusses erzeugen Schleifen, bedingte Programmausführungen, Programmsprünge, Aufrufe eines Unterprogramms, Rückkehr ins Hauptprogramm, wertgesteuerte Programmverzweigungen, das Programmende und eine Verzögerung der Programmausführung.

- Programm-Schleifen (Beispiel BSP17.BAS)

```
FOR variable = anfang TO ende STEP schrittweite
  anweisung1
  anweisung2
...
NEXT
```

Programm-Schleifen sind eine Anzahl nacheinander wiederholter Ausführungen ein und desselben Programmabschnittes. Beim ersten Durchlauf wird dem Schleifenindex "variable" ein Startwert zugewiesen (im ersten Beispiel: a = 1). Zu Beginn jeder Schleifenabarbeitung wird geprüft, ob der Schleifenindex gleich dem Endwert (10) ist. Ist das der Fall wird der Schleifendurchlauf abgebrochen und das Programm hinter NEXT fortgesetzt. Andernfalls

werden die Anweisungen bis NEXT ausgeführt. Dann wird zum Schleifenindex der mit STEP festgelegte Wert addiert und zur FOR-Anweisung gesprungen. Die Schleifenabarbeitung beginnt von vorn. Wird die Angabe der Schrittweite mit STEP weggelassen, beträgt die verwendete Schrittweite 1. Aus Übersichtlichkeitsgründen ist es ratsam die Programmanweisungen innerhalb einer FOR ... NEXT-Konstruktion etwas einzurücken.

(Beispiel BSP18.BAS)

Schleifen können ineinander verschachtelt werden. Die Anzahl der Verschachtelungen ist nur durch den zur Verfügung stehenden Speicherplatz begrenzt.

(Beispiel BSP19.BAS)

Jede FOR-Schleife darf nur über ihre eigene NEXT-Anweisung laufen, d.h. man darf nicht aus einer Schleife mit der GOTO-Anweisung in eine andere Schleife springen. Es ist außerdem wichtig, darauf zu achten, daß die Abbruchbedingung auch erreicht werden kann. Ist das nicht der Fall, wird die Schleife endlos ausgeführt und das Programmende nie erreicht.

- bedingte Programmausführung (Beispiel BSP20.BAS)

IF bedingung THEN anweisung1 ELSE anweisung2

Zur bedingten Programmausführung dient die IF ... THEN ... ELSE ... - Konstruktion. Ist der Term "bedingung" wahr, hat er also den Wert ungleich Null, wird "anweisung1" ausgeführt, andernfalls "anweisung2". Der ELSE-Zweig kann auch weggelassen werden. Dann wird für "bedingung" = falsch zur nächsten Zeile gesprungen, "anweisung1" also einfach übersprungen. Die gesamte Konstruktion muß in einer Quelltextzeile stehen. Mehrere Anweisungen nach THEN oder ELSE, wie in einigen anderen Programmiersprachen üblich, sind nicht zulässig.

- Sprunganweisung (Beispiel BSP20.BAS)

GOTO marke

Im Sprachumfang von CC-BASIC ist die Sprunganweisung GOTO marke enthalten. "marke" ist dabei der Name des Sprungziels. Mit der GOTO-Anweisung kann zu einer Marke gesprungen werden, die vor oder nach der Anweisung im Quelltext steht.

- Aufruf und Rückkehr aus einem Unterprogramm (Beispiel BSP21.BAS)

Ein Unterprogramm stellt einen Programmabschnitt dar, der im Quelltext nur einmal hingeschrieben werden muß und dann unter seinem Namen im Hauptprogramm beliebig oft aufgerufen und ausgeführt werden kann. Durch den Einsatz von Unterprogrammen wird der Umfang des Programmes kleiner. Ein Unterprogramm beginnt immer mit einer Marke und endet mit der Anweisung RETURN. Der Aufruf eines Unterprogrammes erfolgt mit der Anweisung GOSUB marke. (GOSUB von engl.: Go Subroutine)  
"marke" ist dabei die Marke, die in der ersten Zeile des Unterprogrammes steht. Mit dieser Anweisung merkt sich das Programm zunächst die Zeile vor der GOSUB-Anweisung, dann wird zur ersten Zeile des Unterprogrammes gesprungen. Der dort stehende Quelltext wird abgearbeitet, bis die RETURN-Anweisung erreicht ist. Dann wird zu der Zeile zurückgesprungen, die im Hauptprogramm auf die GOSUB-Anweisung folgt. Es darf nicht vorkommen, das eine RETURN-Anweisung abgearbeitet wird bevor ein Unterprogrammaufruf mit GOSUB erfolgt ist. Deshalb sollten alle Unterprogramme in einem Block am Ende des Programmes angeordnet werden. Innerhalb eines Unterprogrammes kann wieder in ein anderes Unterprogramm gesprungen werden. Die maximale Verschachtelungstiefe ist vier.

- wertgesteuerte Programmverzweigung (Beispiel BSP22.BAS)

ON variable1 GOTO marke0, marke1, ..., markex  
ON variable1 GOSUB marke0, marke1, ..., markex

Ist der Wert von "variable1" gleich Null wird zu "marke0" gesprungen, ist er gleich eins, wird zu "marke1" verzweigt, usw. Ist der Wert der Variablen negativ oder größer als die Anzahl der angegebenen Marken, wird die Programmabarbeitung in der nächsten Zeile fortgesetzt. Die gleiche Konstruktion ist auch mit Unterprogrammaufrufen möglich:

- Programmende (ohne Beispiel)

END

Wird während der Programmarbeitung die END-Anweisung erreicht, wird der Steuercomputer in einen inaktiven Zustand versetzt. Es kann dann ein neues Programm geladen werden.

- bedingte Programmunterbrechung (Beispiel BSP23.BAS)

WAIT bedingung

Diese Anweisung bewirkt eine Unterbrechung der Programmabarbeitung bis "bedingung" einen Wert ungleich Null hat. (Hinweis zum Beispiel: Das Drücken der Taste also das Nullsetzen des Ports 9 kann im Simulator nachgebildet werden, indem mit F4 das Programm bis WAIT ... ausgeführt wird, dann die Variable "taste" mit STRG+F7 überwacht wird. Im dann aufgehenden Fenster kann "taste" durch doppeltes Anklicken verändert werden.)

- zeitgesteuerte Programmunterbrechung (Beispiel BSP24.BAS)

PAUSE term

PAUSE unterbricht die Programmausführung für die Zeitdauer von 20 ms mal dem Wert den "term" repräsentiert. Die Genauigkeit der Wartezeit beträgt dabei +/- 20ms.

### **Befehle zur Kommunikation über die serielle Schnittstelle**

Über die serielle Schnittstelle der C-Control-Unit können Daten seriell (d.h. Bit für Bit nacheinander) ausgegeben oder eingelesen werden. Ist die Control-Unit über ein Kabel mit einem PC verbunden, können so Daten ausgetauscht werden. (s.a. Kommunikation über die serielle Schnittstelle)

- Ausgabe (Beispiel BSP26.bas)

PRINT term

gibt den Wert von "term" als Text aus, d.h. hat "term" den Wert 1 wird das Byte &H31 (das nach dem ASCII- Zeichensatz die 1 darstellt) auf die serielle Schnittstelle ausgegeben. Außerdem wird an jede Ausgabe ein Zeichen für den Zeilenvorschub angehängt (PRINT

ohne Parameter gibt nur einen Zeilenvorschub aus). Durch ein Abschließen des Befehls mit einem Semikolon kann diese Funktion unterdrückt werden.

PRINT term ;

PRINT "Text"

gibt den in Anführungszeichen stehenden Text aus.

Soll ein PRINT-Befehl mehrere Ausgaben erledigen, sind die Parameter durch ein Komma (Zeilenvorschub nach jeder Ausgabe) oder Semikolon (alle in eine Zeile ) zu trennen.

PUT term

gibt den Wert von "term" als Byte, also ohne Übersetzung in den ASCII-Zeichensatz aus.

- Eingabe (Beispiel BSP27.bas)

INPUT variable

liest einen Wert von der seriellen Schnittstelle und speichert ihn in der Variable mit dem Namen "variable". Das Programm wartet an dieser Stelle solange, bis ein Zeichen empfangen wurde. D.h. ggf. kann dieser Zustand nur durch einen Reset der Baugruppe beseitigt werden.

GET variable

liest ein Byte und speichert es in der angegebenen Variablen.

- Steuerbefehle

RXD

ermittelt, ob an der Schnittstelle ein Zeichen empfangen wurde. Ist das der Fall, liefert RXD den Wert -1, andernfalls 0. Dadurch kann verhindert werden, daß INPUT oder GET endlos auf eine Eingabe warten (s.a. INPUT und GET).

BAUD rate

stellt die Übertragungsgeschwindigkeit der seriellen Schnittstelle ein (Standard: 9600 Baud). Für "rate" existieren in CC-BASIC die vordefinierten Konstanten R1200, R2400, R4800 und R9600. Die anderen Schnittstellenparameter können nicht geändert werden (8 Datenbits, keine Parität, 1 Stoppbit).

### **Befehle zur Dateiverwaltung**

Eine Datei im Sinne von CC-BASIC ist eine Folge von Integerwerten, die im seriellen EEPROM gespeichert werden, und dort auch nach einer Unterbrechung der Spannungsversorgung erhalten bleiben. Die Dateiposition, auf die zugegriffen werden soll ist durch einen im Betriebssystem geführten Zeiger festgelegt, der bei jedem Schreiben bzw. Lesen um den Wert eins erhöht wird. Es kann nur eine Datei verwaltet werden, die aus diesem Grund auch keinen Namen trägt. Bevor die Datei bearbeitet werden kann, muß sie geöffnet werden. Dazu dient die OPEN Anweisung.

OPEN# FOR WRITE

öffnet die Datei ausschließlich zum Schreiben. (Beispiel BSP28.bas) Der Dateizeiger wird auf die erste Dateiposition gesetzt, d.h. evtl. vorhandene alte Inhalte werden überschrieben.

OPEN# FOR APPEND

öffnet die Datei ebenfalls ausschließlich zum Schreiben. Neu geschriebene Werte werden aber an evtl. vorhandene alte Inhalte angehängt, da der Zeiger jetzt auf den ersten freien Speicherplatz gesetzt wird.

OPEN# FOR READ

öffnet die Datei ausschließlich zum Lesen. (Beispiel BSP29.bas) Der Zeiger wird auf die erste freie Position gesetzt.

Vor dem Schreiben ist zu prüfen, ob noch genügend Speicherplatz frei ist. Die Funktion

FILEFREE

liefert die Anzahl freier Speicherplätze á 2 Byte zurück.

PRINT# term

schreibt den Wert, den "term" ergibt, in die Datei und erhöht den internen Dateizeiger. Vor jedem Lesen ist zu prüfen, ob das Ende der Datei schon erreicht worden ist. Dazu dient die Funktion

EOF

(engl.: End Of File) Diese Funktion liefert als Ergebnis den Wert -1, wenn das Ende der Datei erreicht wurde, andernfalls den Wert 0.

INPUT# variable

liest von der Adresse des EEPROMs, auf die der Dateizeiger zeigt.

Nach Abschluß ist die Datei mit dem Befehl

CLOSE#

wieder zu schließen. Nur dann sind die Daten vor einem Spannungsausfall oder Reset geschützt.

### **Portbefehle**

Der Zugriff auf Ports erfolgt prinzipiell wie auf Variablen. Soll ein Digitalport auf HIGH-Pegel geschaltet werden, muß ihm eine Eins zugewiesen werden. Die Zuweisung einer Null setzt ihn auf LOW-Pegel. Es gibt jedoch einige speziell für Ports vorgesehene Befehle:

TOG portname

Dieser Befehl invertiert den Zustand des Ports und ist somit identisch zum Befehl port = NOT port. (engl.: toggle - kippen)

DEACT portname

Dieser Befehl schaltet den Port mit dem Bezeichner "portname" in den hochohmigen Zustand (Deaktivierung) und arbeitet dann als Eingang. Aufgrund des Pull Up-Widerstandes, mit dem jeder Digitalport auf der C-Control-Unit versehen ist, führt ein unbeschalteter Eingang HIGH-Pegel.

PULSE bitportname

Dieser Befehl erzeugt am Ausgang des Portes mit dem Namen "bitportname" einen Impuls von einigen Millisekunden Länge. Die Polarität des Impulses ist dabei immer der des Ausgangszustandes entgegengesetzt. Dieser Befehl kann zweckmäßig für die Ansteuerung externer digitaler Baugruppen (z.B. flankengetriggerte Zähler) verwandt werden.

### Definition und Anwendung von Datentabellen

(Beispiel BSP30.bas)

Eine Datentabelle speichert Integer-Konstanten, die jeweils zwei Byte belegen. Die Konstanten können direkt im Quelltext definiert werden ("konst1", "konst2", ... sind dabei Integer-Konstanten):

```
TABLE tabellenname konst1 konst2 ... konstn  
TABEND
```

oder aus einer Datei eingelesen werden:

```
TABLE tabellenname "dateiname"
```

Tabellendefinitionen müssen stets am Ende eines Programmes, hinter dem END-Befehl stehen, da die Tabellenwerte ansonsten bei der Programmausführung als Programmcode interpretiert würden. Der Zugriff auf die Daten der Tabelle erfolgt mit dem Befehl

```
LOOKTAB tabellenname, index, variable
```

"tabellenname" ist der Bezeichner der Tabelle, "index" kann ein beliebiger Term sein, dessen Wert nicht negativ und nicht größer als N-1 (N - Anzahl der Einträge) sein darf. "variable" ist der Name der Variablen, in der der Tabellenwert abgespeichert werden soll.

### Weitere Befehle

- Ausgabe von Tönen mit BEEP

Der C-Control-BASIC-Steuercomputer kann an einem seiner Pins (BEEP-Pin, entspricht Prozessorausgang TCMP1) 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 nach der Formel

$$\text{ton} = 250000 / \text{freq [Hz]}$$

"tTon" bestimmt die Dauer des Tons und "tPause" die Pause nach dem Ton. Die Einheit für die Zeitangaben beträgt 20 Millisekunden. Der Befehl

BEEP 568, 10, 3

gibt also für  $10 \cdot 20 = 200$  Millisekunden einen Ton von etwa 440 Hz (Kammerton A) 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. Ist für die Tonlänge 0 angegeben, wird ein Dauerton erzeugt. Der Tongenerator schaltet den Ton ein und fährt mit der Abarbeitung des BASIC-Programms fort. Mit dem Wert 0 für "ton" kann der Tongenerator wieder abgeschaltet werden.

- Frequenzmessung mit der Funktion FREQ

Ist am DCF77-Eingang 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.

- Stromsparmodus mit SLOWMODE

Anwendungen, die keine hohe Rechenleistung benötigen, können durch Aufruf des Befehls

SLOWMODE ON

den internen Takt des Mikroprozessors verlangsamen (1/16). In Kombination mit den Jumpern "LED" und "232" läßt sich so der Leistungsbedarf des Steuercomputers nochmals senken. Sollte im Verlauf des Programms wieder eine höhere Geschwindigkeit erforderlich sein, so läßt sich mit

SLOWMODE OFF

wieder der Ausgangszustand herstellen.

Achtung: Programme, die serielle Datenübertragungen verwenden, sollten den SLOWMODE nicht aktivieren, da die eingestellten Übertragungsraten mit dem Prozessortakt herabgesetzt werden.

### **Einbinden von Assemblerroutinen**

Die folgenden Informationen richten sich an professionelle Anwender des C-Control/BASIC Steuercomputers und sind für die eigentliche BASIC-Programmierung nicht erforderlich! Vorausgesetzt werden die Kenntnis des internen Aufbaus des Mikrocontrollers MC68HC05B6 und Kenntnisse in der Assemblerprogrammierung dieses Controllers. Außerdem wird ein Assembler für 68HC05-Prozessoren benötigt. An dieser Stelle sei das Buch "Motorola 68HC05" von Zekeriya Zengin aus dem Heise-Verlag (ISBN 3-88229-034-X, Conrad Electronic Best.-Nr.: 91 91 79) empfohlen, das den Mikrocontroller vollständig beschreibt und

auf einer beiliegenden Diskette u.a. einen Assembler und zahlreiche Beispiele liefert. 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 stehen neben dem externen EEPROM-Speicherchip im Mikroprozessor selbst noch einmal 255 EEPROM Bytes zur Aufnahme von Assembler-routinen zur Verfügung. Diese Routinen können aus dem BASIC-Programm heraus aufgerufen werden. Der Befehl dazu lautet

`SYS adr`

wobei "adr" eine Konstante ist und die Adresse bestimmt, zu der gesprungen werden soll, beispielsweise &H101, da der interne EEPROM-Bereich an dieser Adresse beginnt. Der Assemblercode muß also per ORG-Befehl an die Adresse &H101 gelegt werden. Die Rückkehr aus einer Assembler-routine zum BASIC erfolgt per RTS-Befehl. Der Datenaustausch zwischen BASIC und Assembler kann über die in der Datei SYSADR.INC aufgelisteten RAM-Adressen erfolgen.

Wie kommt der Assemblercode in den C-Control/*BASIC* Steuercomputer?

Die in Assembler geschriebenen Zusatzroutinen werden in einer separaten Quelltextdatei (z.B. ADDONS.ASM) gespeichert. Anschließend wird Assembler aufgerufen, daraus eine Objektcode-datei im S19-Format zu erstellen (z.B. ADDONS.S19). Lesen Sie dazu die Dokumentation zu dem Ihnen zur Verfügung stehenden Assembler. Ihr BASIC-Programm, das den SYS-Befehl enthält, muß mit dem Befehl

`SYSCODE "ADDONS.S19"`

den generierten Code einbinden. Der SYSCODE Befehl darf nur einmal in einem CCBASIC-Programm erscheinen und sollte am Ende, noch hinter eventuellen Tabellendefinitionen, stehen.

## Allgemeine Handhabungs- und Sicherheitshinweise

Der externe Temperatur-Sensor ist ein Zubehör zum modularen Telemetrie-System. Er kann aber ohne weiteres auch an die C-Control-Unit angeschlossen werden. Mit ihm können Temperaturen im Bereich von - 27,5...+ 100 °C gemessen werden. Die Aufzeichnung und Anzeige der Meßdaten sind vom Anwender durch ein geeignetes C-Control-Programm zu realisieren.

### Aufbau und Funktion

Der externe Temperatur-Sensor dient zur Messung der Umgebungstemperatur. Der entsprechende **Sensor** sowie die dazugehörige Auswerteelektronik sind in, bzw. an dem Modulgehäuse untergebracht. Die Meßwerte werden in Form eines **normierten Ausgangssignals** geliefert. Es handelt sich dabei um eine **Spannung** im Bereich von 0 ... 2,55 Volt.

Die Meßsignale werden über die Anschlußleitung zu einem der acht A/D-Ports der Control-Unit übertragen und dort **digitalisiert**. Jeder Meßwert wird als Zahl im Bereich von 0... 255 gespeichert. Die Zahlenwerte können umgerechnet und dann in der gewohnten Einheit °C ausgegeben oder weiterverarbeitet werden.

Dieses Konzept hat den Vorteil, daß die Signale unterschiedlichster Sensoren im **Byte-Format** erfaßt werden können. Die Berücksichtigung der speziellen Sensoreigenschaften erfolgt erst bei der Auswertung durch eine geeignete Software.

### Temperaturmessung

Als **Sensor** kommt ein **PTC-Widerstand** des Typs KTY 87-250 zum Einsatz. Grundlage des Meßverfahrens ist die Tatsache, daß sich bei bestimmten Halbleitermaterialien der elektrische **Widerstand** mit steigender Temperatur deutlich erhöht. Der **Sensor** besteht aus zwei in Reihe geschalteten Silizium-Elementen, welche in ein transparentes Kunststoff-Gehäuse eingegossen sind.

Der **Sensor** liefert als "Meßsignal" einen veränderlichen Widerstandswert. Um ein **normiertes Signal** in Form einer **Spannung** von 0 ... 2,55 Volt zu erhalten, wird eine Signalaufbereitung durchgeführt.

Eine **Konstantstromquelle** sorgt dafür, daß der Temperatursensor stets von einem gleichbleibenden **Strom** durchflossen wird. Als Ergebnis steht eine veränderliche **Spannung** zur Verfügung, die proportional zum Widerstand des Sensors ist. Diese **Spannung** wird verstärkt und kann nun von der C-Control-Unit verarbeitet werden.

Das Meßergebnis ist zunächst eine Zahl im Bereich von 0 ... 255. Der dazugehörige Meßwert in °C muß also erst noch berechnet werden. Da der **Sensor** keine lineare **Kennlinie** aufweist, ist zusätzlich eine Korrektur erforderlich. Im Lieferumfang von C-Control ist dazu die Tabelle kty.tab als Datei enthalten.

### Installation und Anschluß

Das Sensormodul wird betriebsfertig geliefert. Ein Abgleich ist nicht erforderlich.

Der Sensor muß an einen der acht A/D-Ports der Control-Unit angeschlossen werden.

Überzeugen Sie sich daß die Spannungsversorgung abgeschaltet ist und trennen sie eine eventuelle Kabelverbindung zum PC. Wählen Sie einen freien A/D-Port (J33-J40 auf dem Starterboard) und stecken Sie die dreipolige Buchse so auf den gewählten Portstecker, daß das weiße Kabel (Signalader) zum Experimentierfeld des Starterboards zeigt. Das Anschlußkabel darf um maximal zwei Meter verlängert werden. Der Jumper zur Wahl der Referenzspannung der A/D-Wandler muß auf 2,5V gesetzt sein.

Nach dem Zuschalten der Spannungsversorgung benötigt das Modul etwa eine Minute zur internen Stabilisierung. Beachten Sie bitte, daß vorher keine gültigen Meßwerte gewonnen werden können

## Programmtechnische Realisierung einer Temperaturmessung

Nach dem Einlesen des aktuellen Abtastwertes muß dieser noch in einen Wert für die Temperatur umgesetzt werden. Im Lieferumfang von C-Control ist dafür die Tabelle kty.tab enthalten. Der aus der Tabelle ermittelte Wert kann dann über die serielle Schnittstelle zur Anzeige gebracht werden (s.a. Datenauswertung/-anzeige).

### Kalibrieren

Das Sensormodul ist bereits werksseitig abgeglichen. Ein Abgleich von Ihrer Seite ist daher nicht erforderlich. Sie können jedoch auch selbst eine Kalibrierung vornehmen. Gehen Sie dazu nach folgender Anweisung vor:

#### Abgleich des Temperatursensors KTY 87

Stellen Sie sicher, daß keine Betriebsspannung am Sensormodul anliegt.

Öffnen Sie das Modulgehäuse durch Abziehen des Klebebands.

Löten Sie den KTY 87 aus der Platine aus. Sie erkennen ihn am durchsichtigen Gehäuse. Beachten Sie, daß Sie diesen anschließend mit der richtigen Polarität wieder einsetzen müssen.

Löten Sie anstelle des KTY 87 einen **Meßwiderstand** von 2000 Ohm mit (+/- 0,1% Toleranz) ein.

Legen Sie eine Betriebsspannung von 6,5 Volt an und warten Sie eine Minute.

Messen Sie mit einem **Voltmeter** die Referenzspannung (Uref). Verbinden Sie dazu die "+"-Leitung des Voltmeters mit Pin 1 und die "-"-Leitung mit Pin 4 des **Operationsverstärkers** LM 10. Dieser ist auf der Platine mit "I1" gekennzeichnet.

Falls Sie ein **Multimeter** verwenden, muß das Gerät bei allen Messungen auf **Gleichspannungsmessung** geschaltet sein. Überzeugen Sie sich, daß das Gerät auf keinen Fall auf **Strommessung** eingestellt ist! Dies könnte eine Zerstörung des Sensormoduls zur Folge haben.

Berechnen Sie nun den einzustellenden Korrekturwert für den Referenzstrom (Usense) in mV. Runden Sie das Ergebnis auf 2 Stellen nach dem Komma.

$$\begin{array}{ll} \text{Uref} < 200 \text{ mV:} & x = 200 \text{ mV} - \text{Uref} \\ & \text{Usense} = x * 0,66 + 200 \\ \text{Uref} > 200 \text{ mV:} & x = \text{Uref} - 200 \text{ mV} \\ & \text{Usense} = 200 - (x * 0,66) \end{array}$$

Schließen Sie ihr **Voltmeter** parallel zum Meßwiderstand an, den Sie vorher eingelötet haben. Der Innenwiderstand des verwendeten Meßgeräts muß größer als 10 MOhm sein.

Stellen Sie den Spindeltrimmer R1 (500 Ohm) so ein, daß Sie das vorher berechnete Ergebnis von Usense mit einer Genauigkeit von +/- 0,1 mV angezeigt bekommen.

Klemmen Sie das **Voltmeter** vom **Meßwiderstand** ab.

Verbinden Sie die "-"-Leitung des Voltmeters mit dem Anschlußstift "Y21" (GND) und die "+"-Leitung mit Pin 6 des **Operationsverstärkers** LTC 1049. Diese ist auf der Platine mit "I2" gekennzeichnet.

Stellen Sie mit dem **Spindeltrimmer** R2 (2 kOhm) so ein, daß sie eine **Spannung** von +865 mV mit einer Genauigkeit von +/- 1 mV angezeigt bekommen.

Klemmen Sie das Meßgerät und die Betriebsspannung vom Sensormodul ab

Entfernen Sie den **Meßwiderstand** und löten Sie den KTY 87 wieder ein. Beachten Sie dabei die Polarität.

Verschließen Sie das Gehäuse. Benutzen Sie dazu ein neues Klebeband.

## Technische Daten

### Modul

Betriebstemperatur des Moduls (für die angegebene Toleranz)

- 25 ... + 85 °C

Betriebsspannung (von der Wetterstation)

6,5 Volt

Stromaufnahme

ca. 0,65 mA

### Temperatursensor

Betriebstemperatur

- 40 ... + 125 °C

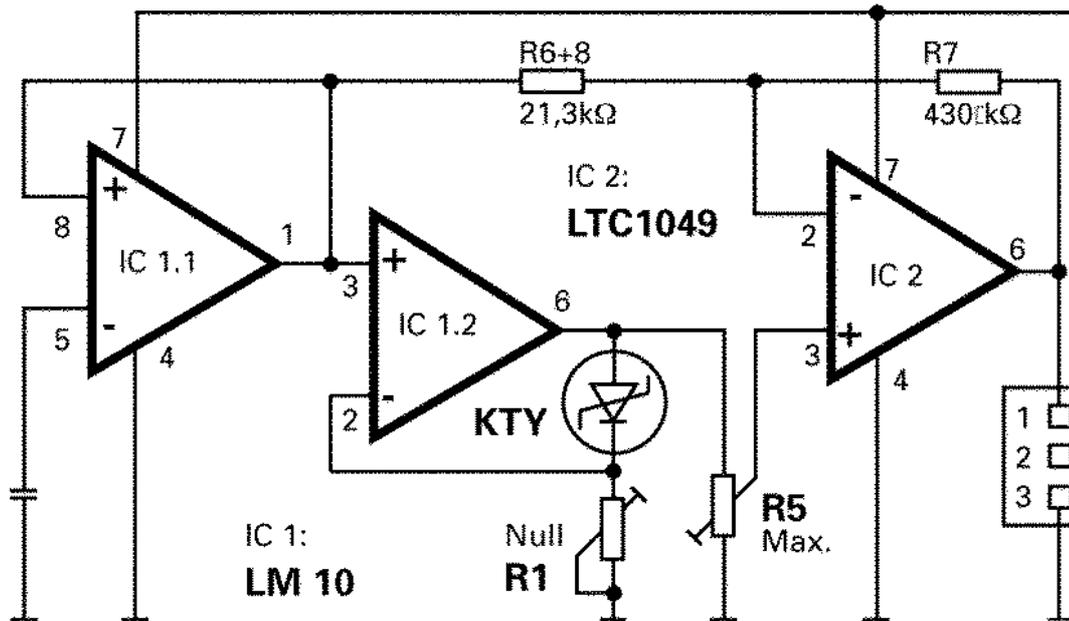
Meßbereich

- 27,5 ..+ 100 °C

Toleranz

+/- 1 %

Schaltplan:



## Allgemeine Handhabungs- und Sicherheitshinweise

Der Temperatur-Sensor ist ein Zubehör zum modularen Telemetrie-System. Er kann aber ohne weiteres auch an die C-Control-Unit angeschlossen werden. Mit ihm können Temperaturen im Bereich von - 50...+ 600 °C gemessen werden. Die Aufzeichnung und Anzeige der Meßdaten sind vom Anwender durch ein geeignetes C-Control-Programm zu realisieren.

### Aufbau und Funktion

Der Temperatursensor dient zur Messung der Umgebungstemperatur. Der entsprechende **Sensor** sowie die dazugehörige Auswerteelektronik sind in, bzw. an dem Modulgehäuse untergebracht. Die Meßwerte werden in Form eines **normierten Ausgangssignals** geliefert. Es handelt sich dabei um eine **Spannung** im Bereich von 0 ... 2,55 Volt. Die Meßsignale werden über die Anschlußleitung zu einem der acht A/D-Ports der Control-Unit übertragen und dort **digitalisiert**. Jeder Meßwert wird als Zahl im Bereich von 0 ... 255 gespeichert. Die Zahlenwerte können umgerechnet und dann in der gewohnten Einheit °C ausgegeben oder weiterverarbeitet werden. Dieses Konzept hat den Vorteil, daß die Signale unterschiedlichster Sensoren im **Byte-Format** erfaßt werden können. Die Berücksichtigung der speziellen Sensoreigenschaften erfolgt erst bei der Auswertung durch eine geeignete Software.

### Temperaturmessung

Es kommt ein Platinensensor zum Einsatz. Grundlage des Meßverfahrens ist die Tatsache, daß sich bei Metallen der elektrische **Widerstand** mit steigender Temperatur erhöht. Der **Sensor** besteht aus einem Keramiksubstrat, auf dem eine sehr dünne Platinschicht aufgebracht ist. Diese ist zum Schutz gegen äußere Einflüsse mit einer Glasschicht überzogen.

Der **Sensor** liefert als "Meßsignal" einen veränderlichen Widerstandswert. Um ein **normiertes Signal** in Form einer **Spannung** von 0 ... 2,55 Volt zu erhalten, wird eine Signalaufbereitung durchgeführt.

Eine **Konstantstromquelle** sorgt dafür, daß der Temperatursensor stets von einem gleichbleibenden **Strom** durchflossen wird. Als Ergebnis steht eine veränderliche **Spannung** zur Verfügung, die proportional zum Widerstand des Sensors ist. Diese **Spannung** wird verstärkt und kann nun von der C-Control-Unit verarbeitet werden.

Das Meßergebnis ist zunächst eine Zahl im Bereich von 0 ... 255. Der dazugehörige Meßwert in °C muß also erst noch berechnet werden. Da der **Sensor** keine lineare **Kennlinie** aufweist, ist zusätzlich eine Korrektur erforderlich. Im Lieferumfang von C-Control ist dazu die Tabelle pt100.tab als Datei enthalten.

### Installation und Anschluß

Das Sensormodul wird betriebsfertig geliefert. Ein Abgleich ist nicht erforderlich.

Der Sensor muß an einen der acht A/D-Ports der Control-Unit angeschlossen werden. Überzeugen Sie sich, daß die Spannungsversorgung abgeschaltet ist und trennen Sie eine eventuelle Kabelverbindung zum **PC**. Wählen Sie einen freien A/D-Port (J33-J40 auf dem Starterboard) und stecken Sie die dreipolige Buchse so auf den gewählten Portstecker, daß das weiße Kabel (Signalader) zum Experimentierfeld des Starterboards zeigt. Das Anschlußkabel darf um maximal zwei Meter verlängert werden. Der Jumper zur Wahl der Referenzspannung der A/D-Wandler muß auf 2,5V gesetzt sein.

Nach dem Zuschalten der Spannungsversorgung benötigt das Modul etwa eine Minute zur internen Stabilisation. Beachten Sie bitte, daß vorher keine gültigen Meßwerte gewonnen werden können.

## Programmtechnische Realisierung einer Temperaturmessung

Nach dem Einlesen des aktuellen Abtastwertes muß dieser noch in einen Wert für die Temperatur umgesetzt werden. Im Lieferumfang von C-Control ist dafür die Tabelle pt100.tab enthalten. Der aus der Tabelle ermittelte Wert kann dann über die serielle Schnittstelle zur Anzeige gebracht werden (s.a. Datenauswertung/-anzeige).

### Kalibrieren

Das Sensormodul ist bereits werksseitig abgeglichen. Ein Abgleich von Ihrer Seite ist daher nicht erforderlich. Sie können jedoch auch selbst eine Kalibrierung vornehmen. Gehen Sie dazu nach folgender Anweisung vor:

#### Abgleich des PT100 Temperatursensors

Stellen Sie sicher, daß keine Betriebsspannung am Sensormodul anliegt.

Öffnen Sie das Modulgehäuse durch Abziehen des Klebebands

Löten Sie den Sensor aus der Platine aus.

Löten Sie anstelle des PT100 einen **Meßwiderstand** von 100 Ohm mit (+/- 0,1% Toleranz) ein.

Legen Sie eine Betriebsspannung von 6,5 Volt an und warten Sie etwa eine Minute.

Schließen Sie ihr **Voltmeter** parallel zum Meßwiderstand an, den Sie vorher eingelötet haben. Der Innenwiderstand des verwendeten Meßgeräts muß größer als 10 MOhm sein.

Falls Sie ein **Multimeter** verwenden, muß das Gerät bei allen Messungen auf **Gleichspannungsmessung** geschaltet sein. Überzeugen Sie sich, daß das Gerät auf keinen Fall auf **Strommessung** eingestellt ist! Dies könnte eine Zerstörung des Sensormoduls zur Folge haben.

Stellen Sie den **Spindeltrimmer** R8 (50 Ohm) so ein, daß Sie eine Spannung von 100 mV mit einer Genauigkeit von +/- 0,1 mV angezeigt bekommen.

Klemmen Sie das **Voltmeter** vom **Meßwiderstand** ab.

Verbinden Sie die "-"-Leitung des Voltmeters mit Pin 3 und die "+"-Leitung mit Pin 1 des **Operationsverstärkers** LM10. Dieser ist auf der Platine mit "I1" gekennzeichnet.

Stellen Sie mit dem **Spindeltrimmer** R9 (25 kOhm) so ein, daß sie eine **Spannung** von +211 mV mit einer Genauigkeit von +/- 1 mV angezeigt bekommen.

Klemmen Sie das Meßgerät und die Betriebsspannung vom Sensormodul ab

Entfernen Sie den **Meßwiderstand** und löten Sie den Platinensensor wieder ein.

Verschließen Sie das Gehäuse. Benutzen Sie dazu ein neues Klebeband.

### Technische Daten

#### Modul

Betriebstemperatur des Moduls (für die angegebene Toleranz)

- 25 ... + 85°C

Betriebsspannung (von der Wetterstation)

6,5 Volt

Stromaufnahme

ca. 1,25 mA

#### Temperatursensor

Betriebstemperatur

- 50 ... + 600 °C

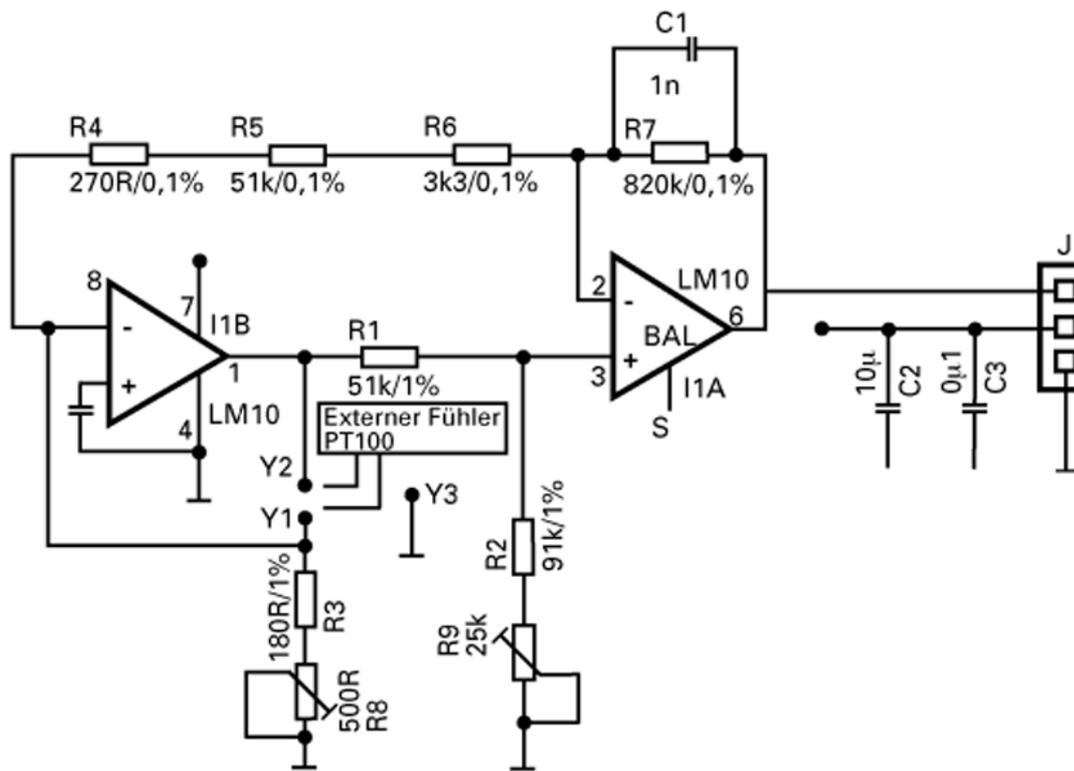
Meßbereich

- 50 ... + 600 °C

Toleranz

+/- 3 %

Schaltplan:



## Allgemeine Handhabungs- und Sicherheitshinweise

Der externe Luftfeuchte-Sensor ist ein Zubehör zum modularen Telemetrie-System. Er kann aber ohne weiteres auch an die C-Control-Unit angeschlossen werden. Mit ihm kann die relative Luftfeuchte im Bereich von 10 bis 90 Prozent gemessen werden. Die Aufzeichnung und Anzeige der Meßdaten sind vom Anwender durch ein geeignetes C-Control-Programm zu realisieren.

### Aufbau und Funktion

Der externe Luftfeuchtesensor dient zur Messung der relativen Luftfeuchtigkeit. Der entsprechende **Sensor** sowie die dazugehörige Auswerteelektronik sind in einem Modulgehäuse untergebracht. Die Meßwerte werden in Form eines **normierten Ausgangssignals** geliefert. Es handelt sich dabei um eine **Spannung** im Bereich von 0 ... 2,55 Volt.

Die Meßsignale werden über die Anschlußleitung zu einem der acht A/D-Ports der Control-Unit übertragen und dort **digitalisiert**. Jeder Meßwert wird als Zahl im Bereich von 0 ... 255 gespeichert. Die Zahlenwerte können umgerechnet und dann in der Einheit Prozent relative Luftfeuchte ausgegeben oder weiterverarbeitet werden.

Dieses Konzept hat den Vorteil, daß die Signale unterschiedlichster Sensoren im **Byte-Format** erfaßt werden können. Die Berücksichtigung der speziellen Sensoreigenschaften erfolgt erst bei der Auswertung durch eine geeignete Software.

### Luftfeuchtemessung

Es kommt ein **kapazitiver Sensor** zum Einsatz. Grundlage des Meßverfahrens ist die Tatsache, daß bestimmte Kunststoffolien ihre **Dielektrizitätskonstante** in Abhängigkeit von der **relativen Luftfeuchte** ändern. Der Sensor ist ein **Kondensator**, der eine solche Kunststoffolie als **Dielektrikum** besitzt. Die **Elektroden** bestehen aus Goldfolie, wodurch eine Änderung der Eigenschaften durch **Korrosion** ausgeschlossen werden kann.

Der **Sensor** liefert als "Meßsignal" eine veränderliche **Kapazität**. Die C-Control-Unit benötigt jedoch ein normiertes Signal in Form einer **Spannung** von 0 ... 2,55 Volt. Deshalb ist eine Signalaufbereitung erforderlich.

Der Feuchtesensor dient als frequenzbestimmendes Bauteil in einem **Oszillator**. Dieser liefert eine **Rechteckspannung**, dessen **Frequenz** sich mit der Kapazität des Feuchtesensors ändert. Daneben existiert noch ein zweiter Oszillator, der eine **Rechteckspannung** mit konstanter Frequenz abgibt. Beide Signale werden durch ein **XOR-Gatter** verknüpft. Als Ergebnis steht eine **pulsbreitenmodulierte Rechtspannung** zur Verfügung. Durch **Glätten** erhält man eine veränderliche **Spannung**, die dem **arithmetischen Mittelwert** des Signals entspricht und proportional zur **Kapazität** des Feuchtesensors ist. Diese **Spannung** kann nun von der Control-Unit verarbeitet werden. Das Meßergebnis ist zunächst eine Zahl im Bereich von 0 ... 255. Der dazugehörige Meßwert in **% Frel** muß also erst noch berechnet werden. Da der **Sensor** eine nicht lineare **Kennlinie** aufweist, die zudem noch temperaturabhängig ist, ist zusätzlich eine Korrektur erforderlich. Im Lieferumfang von C-Control ist dazu die Tabelle hyg.tab als Datei enthalten.

### Installation und Anschluß

Das Sensormodul wird betriebsfertig geliefert. Ein Abgleich ist nicht erforderlich.

Der Sensor muß an einen der acht A/D-Ports der Control-Unit angeschlossen werden.

Überzeugen Sie sich daß die Spannungsversorgung abgeschaltet ist und trennen sie eine eventuelle Kabelverbindung zum **PC**. Wählen Sie einen freien A/D-Port (J33-J40 auf dem Starterboard) und stecken Sie die dreipolige Buchse so auf den gewählten Portstecker, daß das weiße Kabel (Signalader) zum Experimentierfeld des Starterboards zeigt. Das Anschlußkabel darf um maximal zwei Meter verlängert werden. Der Jumper zur Wahl der Referenzspannung der A/D-Wandler muß auf 2,5V gesetzt sein.

Nach dem Zuschalten der Spannungsversorgung benötigt das Modul etwa eine Minute zur internen Stabilisierung. Beachten Sie bitte, daß vorher keine gültigen Meßwerte gewonnen werden können.

### Programmtechnische Realisierung einer Luftfeuchtemessung

Nach dem Einlesen des aktuellen Abtastwertes muß dieser noch in einen Wert für die Luftfeuchte umgesetzt werden. Im Lieferumfang von C-Control ist dafür die Tabelle hyg.tab enthalten. Der aus der Tabelle ermittelte Wert kann dann über die serielle Schnittstelle zur Anzeige gebracht werden (s.a. Datenauswertung/-anzeige).

### Kalibrieren

Das Sensormodul ist bereits werksseitig abgeglichen. Ein Abgleich von Ihrer Seite ist daher nicht erforderlich.

### Technische Daten

#### Modul

Betriebstemperatur des Moduls (für die angegebene Toleranz)	0 ... + 85°C
Betriebsspannung (von der Wetterstation)	6,5 Volt
Stromaufnahme	ca. 4 mA

#### Sensor

Relative Feuchte über 90 % führt leicht zu Meßfehlern, da es dabei zu einer Betauung des Sensors kommen kann. Relative Feuchte unter 10% kann zwar gemessen werden; für diesen Bereich ist die Empfindlichkeit des Sensors jedoch nicht spezifiziert.

Betriebstemperatur	0 ... 85 °C
--------------------	-------------

Ansprechzeit auf 85 % des Endwertes in bewegter Luft (25 °C)

im Bereich Frel = 10 ... 43 %

< 3 Min

im Bereich Frel = 43 ... 90 %

< 5 Min

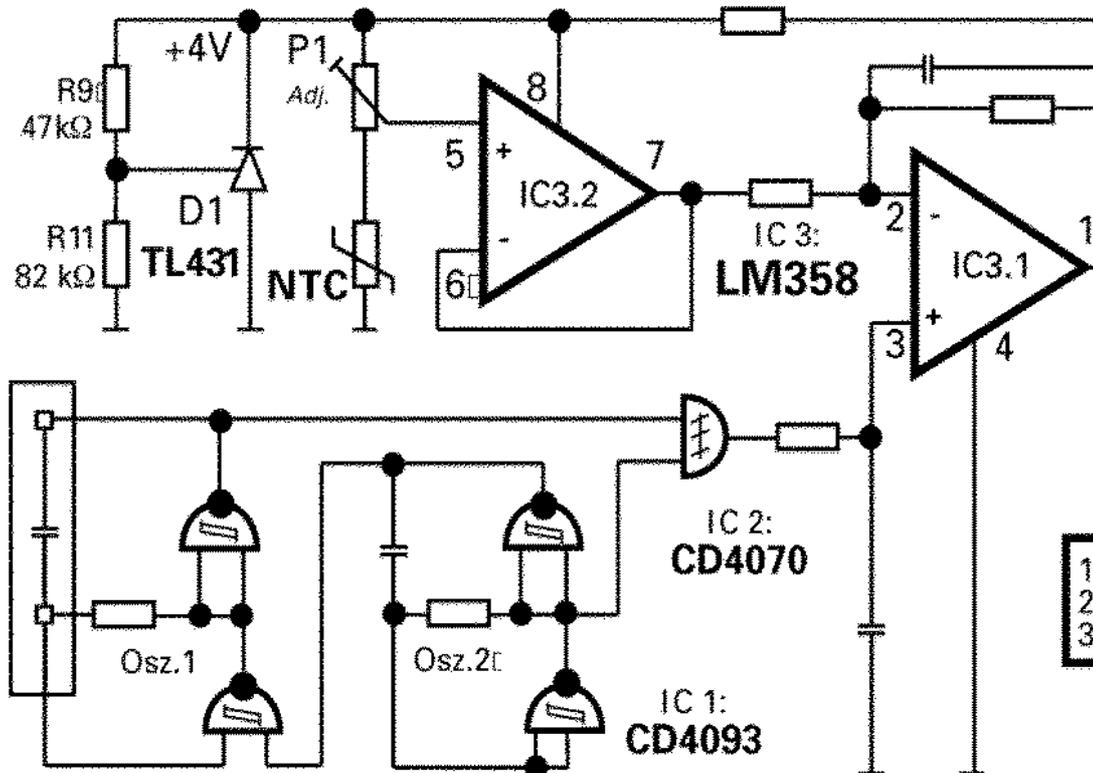
Meßbereich

10 ... 90 %

Toleranz bei Umgebungstemperatur + 10 ... + 50 °C

+/- 3 %

Schaltplan:



## Allgemeine Handhabungs- und Sicherheitshinweise

Der externe Höhenmesser-/Barometer-Sensor ist ein Zubehör zum modularen Telemetrie-System. Er kann aber ohne weiteres auch an die C-Control-Unit angeschlossen werden. Mit ihm kann der absolute Luftdruck im Bereich von 815...1070 hPa gemessen werden. Damit ist er als Höhenmesser oder als Barometer einsetzbar. Die Aufzeichnung und Anzeige der Meßdaten sind vom Anwender durch ein geeignetes C-Control-Programm zu realisieren.

## Aufbau und Funktion

Der externe Höhenmesser-/Barometer-Sensor dient wahlweise zur Messung des Luftdrucks oder der Höhe. Der entsprechende **Sensor** sowie die dazugehörige Auswerteelektronik sind in einem Modulgehäuse untergebracht. Die Meßwerte werden in Form eines **normierten Ausgangssignals** geliefert. Es handelt sich dabei um eine **Spannung** im Bereich von 0 ... 2,55 Volt.

Die Meßsignale werden über die Anschlußleitung zu einem der acht A/D-Ports der Control-Unit übertragen und dort **digitalisiert**. Jeder Meßwert wird als Zahl im Bereich von 0 ... 255 gespeichert. Die Zahlenwerte können umgerechnet und dann in den gewohnten Einheiten hPa (Hekto-Pascal) bzw. m abgelesen und weiterverarbeitet werden.

Dieses Konzept hat den Vorteil, daß die Signale unterschiedlichster Sensoren im **Byte-Format** erfaßt werden können. Die Berücksichtigung der speziellen Sensoreigenschaften erfolgt erst bei der Auswertung durch eine geeignete Software.

## Druckmessung

Es kommt ein **piezoresistiver Absolutdrucksensor** der Firma Bosch zum Einsatz. Der **Luftdruck** wirkt auf die eine Seite einer Siliziummembran. Auf der anderen Seite der Membran befindet sich ein abgeschlossener Raum, in welchem ein Vakuum herrscht. Je nach der Höhe des anliegenden Drucks biegt sich die Membran unterschiedlich stark durch. Druckänderungen werden also zunächst auf mechanischem Wege in Dehnungsänderungen der Membranoberfläche überführt. Diese Dehnungsänderungen können meßtechnisch erfaßt werden.

Auf der Vakuumseite der Membran, die gegen Umwelteinflüsse geschützt ist, sind Dehnungsmeßstellen in Form von **Widerständen** aufgebracht. Grundlage des Meßverfahrens ist die Tatsache, daß bei der Dehnung eines Materials eine Querschnittsverringering stattfindet. Davon sind auch die Meßwiderstände betroffen. Ähnlich wie bei einem Draht erhöht sich bei einer Dehnung die stromdurchflossene Länge, während gleichzeitig der leitfähige Querschnitt abnimmt. Dadurch ergibt sich eine Erhöhung des Widerstandswertes.

Die durch die Druckschwankungen hervorgerufenen Widerstandsänderungen sind sehr gering. Gleichzeitig sind die Widerstandswerte stark temperaturabhängig. Um den Schaltungsaufwand für die Auswertung zu verringern findet bereits im **Sensor** selbst eine Signalaufbereitung statt. Zu diesem Zweck ist im Sensorgehäuse eine Schaltung integriert. Diese sorgt für eine **Temperaturkompensation**. Der **Sensor** liefert als Meßsignal eine veränderliche Spannung. Diese Spannung wird verstärkt und kann nun von der C-Control-Unit verarbeitet werden.

Das Meßergebnis ist zunächst eine Zahl im Bereich von 0 ... 255. Der dazugehörige Meßwert in hPa bzw. m muß also erst noch berechnet werden. Da der **Sensor** eine nicht ganz lineare **Kennlinie** aufweist, ist zusätzlich eine Korrektur erforderlich. Im Lieferumfang von C-Control sind dazu die Tabellen baro.tab (Barometer), h510.tab (Höhenmesser bis 510 m) und high2041.tab (Höhenmesser bis 2041 m) als Dateien enthalten.

## Installation und Anschluß

Zur Einstellung der Betriebsart müssen Sie das Modulgehäuse öffnen. Stellen Sie sicher, daß die Versorgungsspannung abgeschaltet ist und entfernen Sie das Klebeband. Wenn Sie den Sensor als Barometer betreiben möchten, muß der **Jumper** J3 auf die Stellung "Baro" gesteckt werden. In der Stellung "Alt" arbeitet das Modul als Höhenmesser. In dieser

Betriebsart können Sie mit dem **Jumper** J2 zwei Meßbereiche definieren. Ist J2 gesteckt, arbeitet der Sensor mit einem Meßbereich von 0 ... 2041 m. Andernfalls erstreckt sich der Meßbereich von 0 ... 512 m, wodurch eine entsprechend feinere Auflösung möglich wird. Jetzt muß der Sensor an einen der acht A/D-Ports der Control-Unit angeschlossen werden. Überzeugen Sie sich daß die Spannungsversorgung abgeschaltet ist und trennen sie eine eventuelle Kabelverbindung zum **PC**. Wählen Sie einen freien A/D-Port (J33-J40 auf dem Starterboard) und stecken Sie die dreipolige Buchse so auf den gewählten Portstecker, daß das weiße Kabel (Signalader) zum Experimentierfeld des Starterboards zeigt. Das Anschlußkabel darf um maximal zwei Meter verlängert werden. Der Jumper zur Wahl der Referenzspannung der A/D-Wandler muß auf 2,5 V gesetzt sein.

Nach dem Zuschalten der Spannungsversorgung benötigt das Modul etwa eine Minute zur internen Stabilisierung. Beachten Sie bitte, daß vorher keine gültigen Meßwerte gewonnen werden können.

#### **Programmtechnische Realisierung einer Luftdruckmessung**

Nach dem Einlesen des aktuellen Abtastwertes muß dieser noch in einen Wert für den Luftdruck umgesetzt werden. Im Lieferumfang von C-Control ist dafür die Tabelle baro.tab enthalten. Der aus der Tabelle ermittelte Wert kann dann über die serielle Schnittstelle zur Anzeige gebracht werden (s.a. Datenauswertung/-anzeige).

#### **Programmtechnische Realisierung einer Höhenmessung**

Nach dem Einlesen des aktuellen Abtastwertes muß dieser noch in einen Wert für die Höhe umgesetzt werden. Im Lieferumfang von C-Control sind dafür die Tabellen h510.tab (Höhenmesser bis 510 m) und high2041.tab (Höhenmesser bis 2041 m) als Dateien enthalten. Der aus der betreffenden Tabelle ermittelte Wert kann dann über die serielle Schnittstelle zur Anzeige gebracht werden.

#### **Kalibrieren**

##### **Betriebsart Barometer**

Das Sensormodul ist bereits werksseitig abgeglichen. Ein Abgleich von Ihrer Seite ist daher nicht erforderlich.

##### **Betriebsart Höhenmesser**

Da der Luftdruck je nach Wetterlage schwankt, ist in dieser Betriebsart regelmäßig ein Nullabgleich erforderlich. Sie führen diesen folgendermaßen durch.

Öffnen Sie das Modulgehäuse

Verbinden Sie den Sensor mit der Control-Unit. Verbinden Sie diese mit dem PC und

schalten Sie die Spannungsversorgung ein. Starten Sie ein entsprechendes

Aufzeichnungsprogramm in der Control-Unit, daß die ermittelten Luftdruckwerte an die serielle Schnittstelle ausgibt. Zeigen Sie diese Werte auf ihrem PC (z.B. mit Hyperterminal) an.

Warten Sie mindestens eine Minute, damit sich das Sensormodul intern stabilisieren kann.

Stellen Sie den **Spindeltrimmer** R19 so ein, daß die Anzeige gerade Null wird.

Kleben Sie das Modulgehäuse wieder zu. Verwenden Sie dazu am Besten neues Klebeband.

## Technische Daten

### Modul

Betriebstemperatur des Moduls (für die angegebene Toleranz)

0 ... + 70 °C

Betriebsspannung (von der Wetterstation)

6,5 Volt

Stromaufnahme

ca. 9,5 mA

### Sensor

Betriebstemperatur

- 40 ... + 110 °C

Meßbereich Barometer

815 ... 1070 hPa

Höhenmesser, Bereich 1

0 ... 510 m

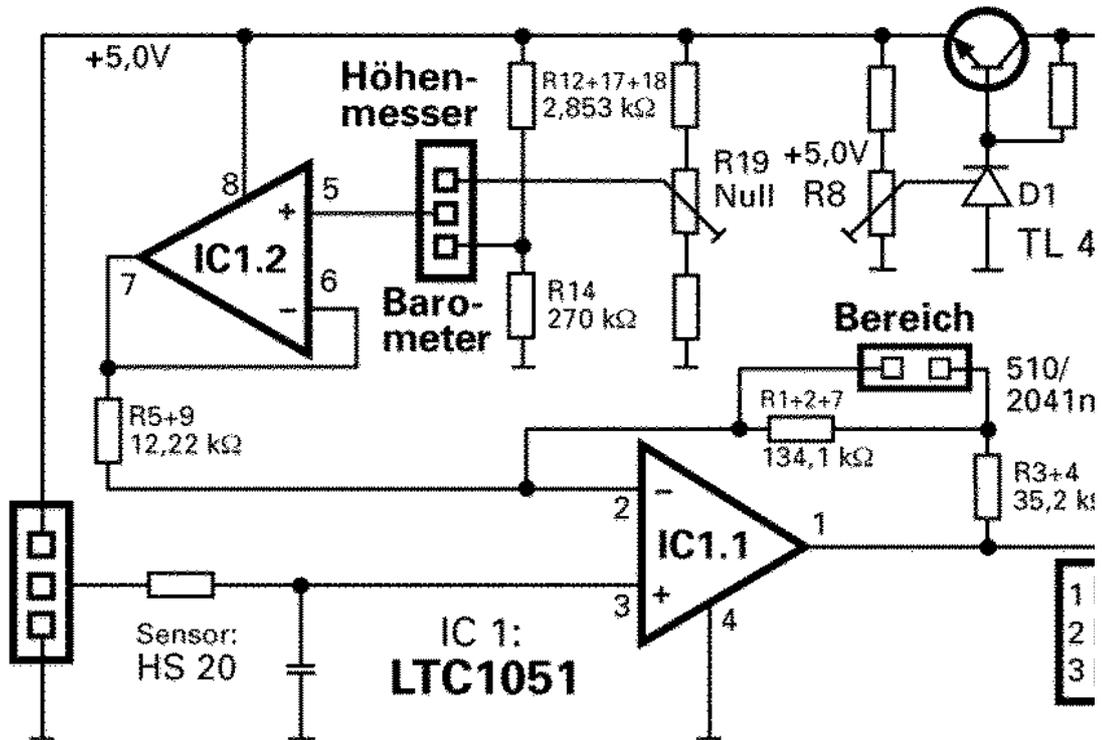
Höhenmesser, Bereich 2

0 ... 2041 m

Toleranz bei + 10 ... 85 °C und konstantem Luftdruck

+/- 2 %

Schaltplan:



## Allgemeine Handhabungs- und Sicherheitshinweise

Der Windgeschwindigkeitsmesser ist ein Zubehör zum modularen Telemetrie-System. Er kann aber ohne weiteres auch an die C-Control-Unit angeschlossen werden. Dieser Sensor dient der Messung der zum Meßzeitpunkt herrschenden Windgeschwindigkeit. Die Aufzeichnung der Meßdaten und die Übertragung zum PC muß vom Anwender durch ein entsprechendes C-Control-Programm realisiert werden.

### Aufbau und Funktion

Der Windgeschwindigkeitsmesser dient der Erfassung der momentanen Windgeschwindigkeit. Dieser **Sensor** ist zur Außenmontage vorgesehen.

Der Windgeschwindigkeitsmesser besteht aus einem **Schalenanemometer**. Dieses wird durch seine Schalen vom Wind in eine Dehbewegung versetzt. Die Drehzahl des Rotors ändert sich proportional mit der Windgeschwindigkeit. Ein Magnet auf der Achse des Rotors schließt bei jeder Umdrehung kurzzeitig einen **Reed-Kontakt**, der im feststehenden Gehäuseteil des Anemometers eingebaut ist. Die Meßwerte werden also in Form von Impulsen geliefert, deren **Frequenz** ein Maß für die Windgeschwindigkeit ist. Diese Impulsfrequenz wird in der Control-Unit gemessen und kann in die gewohnten Einheiten **km/h** oder **m/s** umgerechnet und angezeigt oder weiterverarbeitet werden.

### Installation und Anschluß

Der Windgeschwindigkeitsmesser wird einbaufertig geliefert. Zur Installation muß er lediglich an einem geeigneten Standort befestigt und mit der Control-Unit verbunden werden. Der Sensor muß so montiert werden, daß die Drehachse senkrecht zur zu messenden Windrichtung steht. Für den Anschluß an die C-Control-Unit gibt es zwei Möglichkeiten:

1. Anschluß an den DCF-77-Eingang
2. Anschluß an einen der Digital-Ports

Im ersten Fall ist ein gleichzeitiger Betrieb der Funkuhr nicht möglich, dafür wird das Programm zur Messung der Windgeschwindigkeit einfacher.

### Programmtechnische Realisierung der Messung

Wird der Sensor an den DCF-77-Eingang angeschlossen, erfolgt die Messung über die BASIC- Funktion **FREQ**. Das Betriebssystem zählt im Hintergrund, d.h. ständig und ohne Zutun des Anwenders, die während einer Sekunde am DCF-77- Eingang registrierten Impulse. Der Aufruf der Funktion **FREQ** in einem BASIC-Programm gibt dann jeweils das letzte vollständige Zählergebnis zurück. Da eine Sekunde als Meßperiode hier nur sehr kleine Zählwerte und damit ungenaue Meßergebnisse ergeben würde, ist es zweckmäßig, die Ergebnisse der Zählung über einen längeren Zeitraum (z.B. eine Minute) auf zu summieren und dann entsprechend umzurechnen. Die Umrechnung des Zählergebnisses für eine Zähldauer von einer Minute in einen Wert für die Windgeschwindigkeit erfolgt durch Multiplikation mit dem Faktor 0,0179 für die Einheit km/h bzw. 0,0645 für m/s. Werden andere Meßperioden gewählt, ist auch der Umrechnungsfaktor proportional zu vergrößern oder zu verkleinern.

Wird der Sensor an einen der Digital-Ports angeschlossen, ist der Vorgang der Impulzzählung vom Anwender selbst zu programmieren. Ansonsten gilt das oben gesagte.

### Kalibrieren

Das Sensormodul ist bereits werksseitig abgeglichen. Ein Abgleich von Ihrer Seite ist daher nicht erforderlich.

### Technische Daten

Betriebstemperatur für die angegebenen Toleranzen:

0...+ 70 °C

Meßbereich :

2 bis 40 m/s

Toleranz :

+/- 1 %

## Allgemeine Handhabungs- und Sicherheitshinweise

Der Windrichtungsmesser ist ein Zubehör zum modularen Telemetrie-System. Er kann aber ohne weiteres auch an die C-Control-Unit angeschlossen werden. Er dient zur Messung der zum Meßzeitpunkt herrschenden Windrichtung.

### Aufbau und Funktion

Der Windrichtungsmesser dient zur Erfassung der momentanen Windrichtung. Der entsprechende **Sensor** ist zur Außenmontage vorgesehen. Die Meßwerte werden in Form eines **normierten Ausgangssignals** geliefert. Es handelt sich dabei um eine Spannung im Bereich von 0 ... 2,55 Volt.

Die Meßsignale werden an einen frei wählbaren A/D-Port der Control-Unit übertragen. Dort werden die Meßsignale **digitalisiert**. Jeder Meßwert wird als Zahl im Bereich von 0... 255 gespeichert. In einem **PC** werden die Zahlenwerte umgerechnet und können dann in der gewohnten Einheit **Grad** abgelesen und weiterverarbeitet werden.

Dieses Konzept hat den Vorteil, daß die Signale unterschiedlichster Sensoren im **Byte-Format** erfaßt werden können. Die Berücksichtigung der speziellen Sensoreigenschaften erfolgt erst bei der Auswertung durch eine geeignete Software.

### Windrichtungsmessung

Der Windrichtungsmesser besteht aus einer drehbar gelagerten Fahne. Durch die spezielle Formgebung richtet sie sich selbsttätig nach der jeweiligen Windrichtung aus. Die Achse der Windfahne ist mit dem Schleifer eines **Potentiometers** verbunden. Es handelt sich dabei um eine Sonderbauform ohne Anschlag, d.h. die Fahne läßt sich "durchdrehen".

Herstellungsbedingt hat das **Potentiometer** einen toten Bereich von etwa 20 °. Dies hat zur Folge, daß Windrichtungen zwischen NNW bis N nicht aufgelöst werden.

Am **Potentiometer** kann eine veränderliche **Spannung** abgegriffen werden, der ein bestimmter Stellwinkel, d.h. die Windrichtung zugeordnet werden kann. Dem **Potentiometer** ist ein **Spannungsteiler** nachgeschaltet. Dieser begrenzt die Ausgangsspannung auf einen Bereich von 0 ... 2,55 V. Somit kann das Meßsignal unmittelbar von der Control-Unit eingelesen werden. Das Meßergebnis ist zunächst eine Zahl im Bereich von 0 ... 255. Der dazu gehörende Meßwert in **Grad** muß also erst noch berechnet werden. Dazu muß der Abtastwert mit dem Faktor 1,35 multipliziert werden:

### Installation und Anschluß

Befestigen Sie zunächst den Windrichtungsmesser an einer Stelle, an der ihn der Wind ungehindert erreichen kann. Er ist vom Hersteller in 0°-Richtung fixiert. Die Windfahne muß also in dieser Stellung nach Norden ausgerichtet werden. Ein Kompaß erleichtert Ihnen das Justieren der Fahne. Entfernen Sie dann die Fixierung. Verlegen Sie nun die Anschlußleitung zur Control-Unit. Diese sollte bereits an ihrem endgültigen Aufstellungsort montiert sein. Verbinden Sie die Adern der Anschlußleitung mit dem beiliegenden Abschlußstück in folgender Weise:

Grün -	Schwarz (GND)
Braun -	Rot (Betriebsspannung 6,5V)
Weiß -	Weiß (Signalader)

Jetzt muß der Sensor an einen der acht A/D-Ports der Control-Unit angeschlossen werden. Überzeugen Sie sich daß die Spannungsversorgung abgeschaltet ist und trennen sie eine eventuelle Kabelverbindung zum **PC**. Wählen Sie einen freien A/D-Port (J33-J40 auf dem Starterboard) und stecken Sie die dreipolige Buchse so auf den gewählten Portstecker, daß das weiße Kabel (Signalader) zum Experimentierfeld des Starterboards zeigt. Das Anschlußkabel darf um maximal zwei Meter verlängert werden. Der Jumper zur Wahl der Referenzspannung der A/D-Wandler muß auf 2,5V gesetzt sein.

### **Programmtechnische Realisierung einer Windrichtungsmessung**

Das Einlesen des aktuellen Spannungswertes des Windrichtungsmessers sollte in einer Schleife erfolgen. Um den digitalen Meßwert in Grad umzurechnen muß er mit dem Wert 1,35 multipliziert werden. Diese Umrechnung kann entweder durch ein Programm in der Control-Unit oder auch nach der Übertragung zum PC vor der Anzeige erfolgen.

[*TIP*: Soll die Umrechnung innerhalb eines CC-BASIC-Programmes erfolgen, ist zu berücksichtigen, daß mit CC-BASIC nur ganze Zahlen verarbeitet werden können. Es muß also zuerst mit dem Wert 135 multipliziert und dann wieder durch 100 geteilt werden. Man erhält dann den ganzzahligen Anteil der Multiplikation als Ergebnis]

### **Kalibrieren**

Das Sensormodul ist bereits werksseitig abgeglichen. Ein Abgleich von Ihrer Seite ist daher nicht erforderlich.

### **Technische Daten**

Betriebstemperatur für die angegebenen Toleranzen:	0...+ 70 °C
Betriebsspannung (von der Wetterstation) :	6,5 Volt
Stromaufnahme :	kleiner 1 mA
Meßbereich :	0 bis 340°
Toleranz :	+/- 5°

## Allgemeine Handhabungs- und Sicherheitshinweise

Der Schallpegel-Sensor ist ein Zubehör zum modularen Telemetrie-System. Er kann aber ohne weiteres auch an die C-Control-Unit angeschlossen werden. Mit ihm können Schallpegel im Bereich von 35 dB(A) bis 120 dB(A) gemessen werden. Die Aufzeichnung und Anzeige der Meßdaten sind vom Anwender durch ein geeignetes C-Control-Programm zu realisieren.

### Aufbau und Funktion

Der Schallpegelsensor dient zur Messung der Lautstärke. Der entsprechende **Sensor**, sowie die dazugehörige Auswerteelektronik sind in bzw. an einem Modulgehäuse angebracht. Die Meßwerte werden in Form eines **normierten Ausgangssignals** geliefert. Es handelt sich dabei um eine **Spannung** im Bereich von 0 ... 2,55 Volt.

Die Meßsignale werden über die Anschlußleitung zu einem der **Analogeingänge** der C-Control-Unit übertragen und dort **digitalisiert**. Jeder Meßwert wird als Zahl im Bereich von 0... 255 gespeichert. Im **Datendisplay** oder einem **PC** werden die Zahlenwerte umgerechnet und können dann in der gewohnten Einheit **dB(A)** abgelesen und weiterverarbeitet werden. Dieses Konzept hat den Vorteil, daß die Signale unterschiedlichster Sensoren im **Byte-Format** erfaßt werden können. Die Berücksichtigung der speziellen Sensoreigenschaften erfolgt erst bei der Auswertung durch eine geeignete Software.

### Schallmessung

Als Sensor kommt ein **Kondensatormikrofon** zum Einsatz. Dieses liefert als Meßsignal eine veränderliche **Kapazität**. Die Control-Unit benötigt jedoch ein **normiertes Signal** von 0... 2,55 Volt. Deshalb ist eine Signalaufbereitung erforderlich.

Das Mikrofon ist über einen Vorwiderstand mit der Betriebsspannung verbunden und wirkt so als Spannungsteiler. Es läßt sich, entsprechend den Kapazitätsänderungen des Mikrofons, eine schwache Wechselspannung abgreifen und diese wird anschließend **logarithmisch** verstärkt. Darüber hinaus findet eine dem menschlichen Ohr nachempfundene Gewichtung der unterschiedlichen Frequenzanteile statt (siehe Abbildung).

Das so gewonnene Signal wird gleichgerichtet und kann nun von der Control-Unit eingelesen werden. Das Meßergebnis ist zunächst eine Zahl im Bereich von 0... 255. Der dazugehörige Meßwert in **dB(A)** muß also erst noch berechnet werden. Im Lieferumfang von C-Control ist dazu die Tabelle spl.tab als Datei enthalten.

### Installation und Anschluß

Das Sensormodul wird betriebsfertig geliefert. Ein Abgleich ist nicht erforderlich.

Der Sensor muß an einen der acht A/D-Ports der Control-Unit angeschlossen werden. Überzeugen Sie sich daß die Spannungsversorgung abgeschaltet ist und trennen sie eine eventuelle Kabelverbindung zum **PC**. Wählen Sie einen freien A/D-Port (J33-J40 auf dem Starterboard) und stecken Sie die dreipolige Buchse so auf den gewählten Portstecker, daß das weiße Kabel (Signalader) zum Experimentierfeld des Starterboards zeigt. Das Anschlußkabel darf um maximal zwei Meter verlängert werden. Der Jumper zur Wahl der Referenzspannung der A/D-Wandler muß auf 2,5V gesetzt sein.

### Programmtechnische Realisierung einer Schallpegelmessung

Nach dem Einlesen des aktuellen Abtastwertes muß dieser noch in einen Wert für den Schallpegel umgesetzt werden. Im Lieferumfang von C-Control ist dafür die Tabelle spl.tab enthalten. Der aus der Tabelle ermittelte Wert kann dann über die serielle Schnittstelle zur Anzeige gebracht werden (s.a. Datenauswertung/-anzeige).

## Kalibrieren

Das Sensormodul ist bereits werksseitig abgeglichen. Ein Abgleich von Ihrer Seite ist daher nicht erforderlich.

## Technische Daten

### Modul

Betriebstemperatur des Moduls (für die angegebene Toleranz)

0 ... + 50 °C

Betriebsspannung (von der Wetterstation)

6,5 Volt

Stromaufnahme

ca. 5 mA

### Schallpegelensensor

Meßbereich

35 ... 120 dB(A)

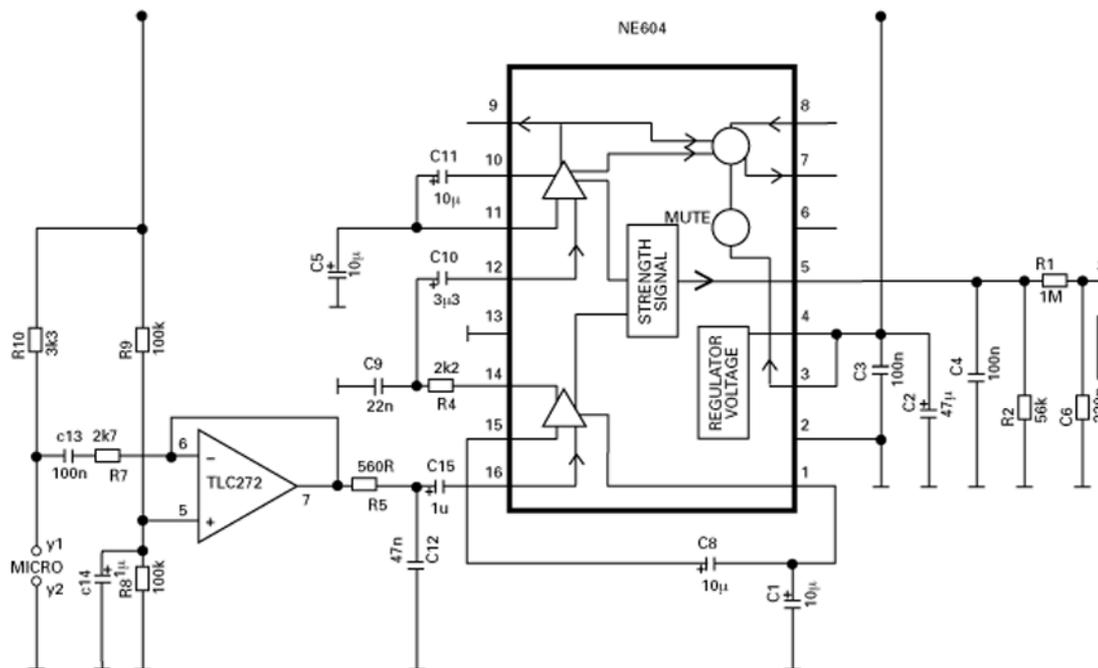
Toleranz

+/- 5 %

Richtcharakteristik

Kugel

Schaltplan:



## Allgemeine Handhabungs- und Sicherheitshinweise

Der Geschwindigkeitssensor ist ein Zubehör zum modularen Telemetrie-System. Er kann aber ohne weiteres auch an die C-Control-Unit angeschlossen werden. Mit ihm können Geschwindigkeiten nach dem Staudruckverfahren im Bereich von 14 bis 230 km/h gemessen werden. Die Aufzeichnung und Anzeige der Meßdaten sind vom Anwender durch ein geeignetes C-Control-Programm zu realisieren.

### Aufbau und Funktion

Der Geschwindigkeitssensor dient zur Messung der Geschwindigkeit über den Staudruck der Luft. Der entsprechende **Sensor** sowie die dazugehörige Auswerteelektronik sind in einem Modulgehäuse untergebracht. Die Meßwerte werden in Form eines **normierten Ausgangssignals** geliefert. Es handelt sich dabei um eine **Spannung** im Bereich von 0... 2,55 Volt.

Die Meßsignale werden über die Anschlußleitung zu einem der acht A/D-Ports der Control-Unit übertragen und dort **digitalisiert**. Jeder Meßwert wird als Zahl im Bereich von 0 ... 255 gespeichert. Die Zahlenwerte können umgerechnet und dann in der gewohnten Einheit km/h ausgegeben oder weiterverarbeitet werden.

Dieses Konzept hat den Vorteil, daß die Signale unterschiedlichster Sensoren im **Byte-Format** erfaßt werden können. Die Berücksichtigung der speziellen Sensoreigenschaften erfolgt erst bei der Auswertung durch eine geeignete Software.

### Staudruckmessung

Es kommt ein **piezoresistiver Differenzdrucksensor** zum Einsatz. Der **Staudruck** wird über einen Schlauchanschluß auf die eine Seite einer Siliziummembran geleitet. Die andere Seite der Membran ist durch eine Bohrung mit der Umgebungsluft verbunden. Je nach der Höhe des anliegenden Drucks biegt sich die Membran unterschiedlich stark durch. Druckänderungen werden also zunächst auf mechanischem Wege in Dehnungsänderungen der Membranfläche überführt. Diese Dehnungsänderungen können meßtechnisch erfaßt werden.

Auf der Innenseite der Membran, die weitgehend gegen Umwelteinflüsse geschützt ist, sind Dehnungsmeßstellen in Form von **Widerständen** aufgebracht. Grundlage des Meßverfahrens ist die Tatsache, daß bei der Dehnung eines Materials eine Querschnittsverringeringung stattfindet. Davon sind auch die Meßwiderstände betroffen. Ähnlich wie bei einem Draht erhöht sich bei einer Dehnung die stromdurchflossene Länge, während gleichzeitig der leitfähige Querschnitt abnimmt. Dadurch ergibt sich eine Erhöhung des Widerstandswertes.

Die durch die Druckschwankungen hervorgerufenen Widerstandsänderungen sind sehr gering. Gleichzeitig sind die Widerstandswerte stark temperaturabhängig. Um den Schaltungsaufwand für die Auswertung zu verringern findet bereits im **Sensor** selbst eine Signalaufbereitung statt. Zu diesem Zweck ist im Sensorgehäuse eine Schaltung integriert. Diese sorgt für eine **Temperaturkompensation**. Der **Sensor** liefert als Meßsignal eine veränderliche Spannung. Diese Spannung wird verstärkt und kann nun von der C-Control-Unit verarbeitet werden.

Das Meßergebnis ist zunächst eine Zahl im Bereich von 0 ... 255. Der dazugehörige Meßwert in **km/h** muß also erst noch berechnet werden. Da der Zusammenhang zwischen **Staudruck** und Geschwindigkeit nicht linear ist, ist zusätzlich eine Korrektur erforderlich. Im Lieferumfang von C-Control ist dazu die Tabelle speed230.tab als Datei enthalten.

### Installation und Anschluß

Das Sensormodul wird betriebsfertig geliefert. Ein Abgleich ist nicht erforderlich. Der Sensor muß an einen der acht A/D-Ports der Control-Unit angeschlossen werden. Überzeugen Sie sich daß die Spannungsversorgung abgeschaltet ist und trennen sie eine eventuelle Kabelverbindung zum **PC**. Wählen Sie einen freien A/D-Port (J33-J40 auf dem

Starterboard) und stecken Sie die dreipolige Buchse so auf den gewählten Portstecker, daß das weiße Kabel (Signalader) zum Experimentierfeld des Starterboards zeigt. Das Anschlußkabel darf um maximal zwei Meter verlängert werden. Der Jumper zur Wahl der Referenzspannung der A/D-Wandler muß auf 2,5 V gesetzt sein. Nach dem Zuschalten der Spannungsversorgung benötigt das Modul etwa eine Minute zur internen Stabilisation. Beachten Sie bitte, daß vorher keine gültigen Meßwerte gewonnen werden können.

### **Programmtechnische Realisierung einer Geschwindigkeitsmessung**

Nach dem Einlesen des aktuellen Abtastwertes muß dieser noch in einen Wert für die Geschwindigkeit umgesetzt werden. Im Lieferumfang von C-Control ist dafür die Tabelle speed230.tab enthalten. Der aus der Tabelle ermittelte Wert kann dann über die serielle Schnittstelle zur Anzeige gebracht werden (s.a. Datenauswertung/-anzeige).

### **Kalibrieren**

Das Sensormodul ist bereits werksseitig abgeglichen. Ein Abgleich von Ihrer Seite ist daher nicht erforderlich. Sollte jedoch im Stillstand, d.h. ohne Luftströmung am Staurohr, eine Geschwindigkeit von 14 km/h oder mehr angezeigt werden, ist eine Nullpunktkorrektur erforderlich. Gehen Sie dazu wie folgt vor:

#### **Null-km/h-Abgleich**

Öffnen Sie das Modulgehäuse

Verbinden Sie den Sensor mit der **Telemetrie-Haupteinheit**. Verbinden Sie diese entweder mit dem PC oder dem **Datendisplay** und schalten Sie die Spannungsversorgung ein.

Warten Sie mindestens eine Minute, damit sich das Sensormodul intern stabilisieren kann.

Stellen Sie den **Spindeltrimmer** P2 so ein, das die Anzeige gerade Null wird.

Kleben Sie das Modulgehäuse wieder zu. Verwenden Sie dazu am Besten neues Klebeband.

### **Technische Daten**

#### **Modul**

Betriebstemperatur des Moduls (für die angegebene Toleranz)

0 ... + 70 °C

Betriebsspannung (von der Haupteinheit)

6,5 Volt

Stromaufnahme

ca. 7,5 mA

#### **Sensor**

Betriebstemperatur

+ 10 ... + 85 °C

Meßbereich

14 ... 230 km/h

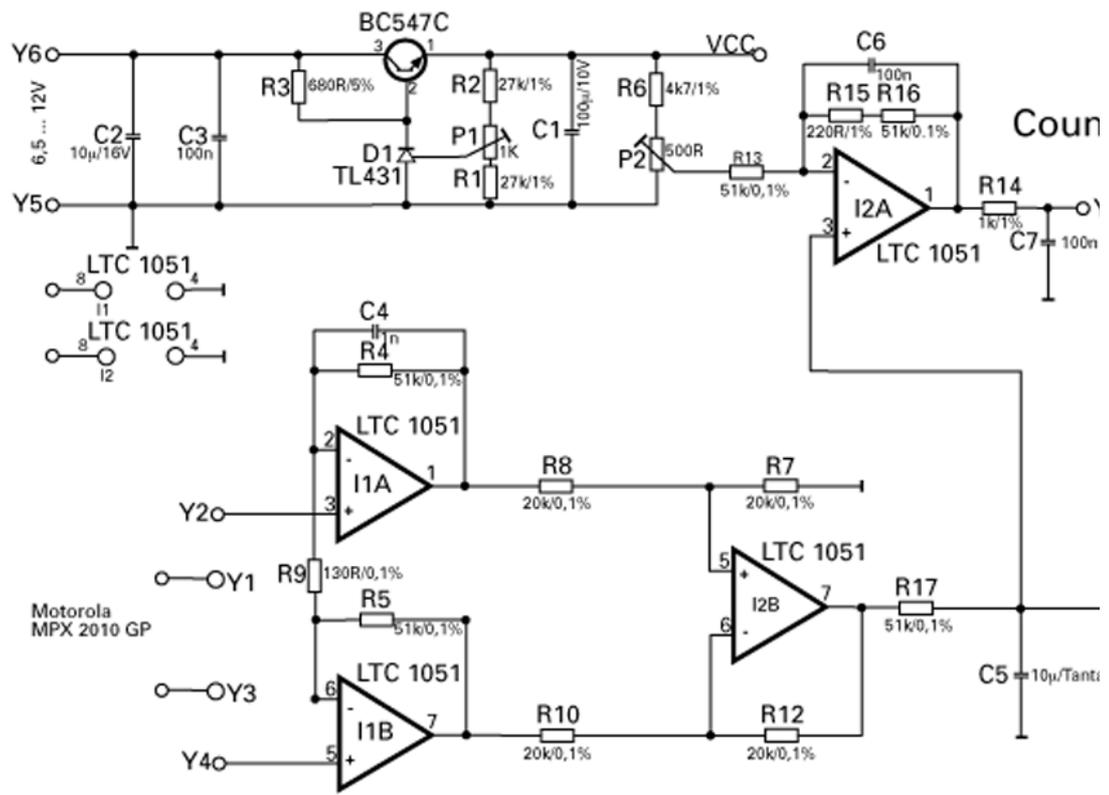
Druckmeßbereich

0 ... 10 kPa

Toleranz bei + 10 ... 85 °C und konstantem Luftdruck

+/- 4 %

Schaltplan:



## Allgemeine Handhabungs- und Sicherheitshinweise

Der Drehzahl-Sensor ist ein Zubehör zum modularen Telemetrie-System. Er kann aber ohne weiteres auch an die C-Control-Unit angeschlossen werden. Er dient zum berührungslosen Messen von Drehzahlen durch optische Reflektions- oder optische Abschattungs-Marken im Bereich 10 bis 60000 U/min. Die Aufzeichnung und Anzeige der Meßdaten sind vom Anwender durch ein geeignetes C-Control-Programm zu realisieren.

### Aufbau und Funktion

Der Drehzahlsensor dient zur berührungslosen Messung von Drehzahlen. Der entsprechende **Sensor** sowie die dazugehörige Auswerteelektronik sind in einem Modulgehäuse untergebracht. Die Meßwerte werden in Form von Impulsen geliefert, deren **Frequenz** ein Maß für die Drehzahl ist. Diese Impulsfrequenz wird in der Control-Unit gemessen und kann in der gewohnten Einheit U/min ausgegeben werden.

### Drehzahlmessung

Als Sensor kommt ein **Fototransistor** zum Einsatz. Dieser reagiert sehr empfindlich auf Hell-Dunkel-Unterschiede und ändert dabei jeweils seinen Verstärkungsfaktor. Über ein Differenzglied werden die Helligkeitsänderungen herausgefiltert und als Impulse für eine Zählstufe aufbereitet. Die Frequenz dieser Impulse kann durch ein frei wählbares Verhältnis von 2... 10 geteilt werden. Das so gewonnene Signal kann nun von der C-Control-Unit eingelesen werden.

### Installation und Anschluß

Das Sensormodul wird einbaufertig geliefert. Zur Installation muß es lediglich auf geeignete Weise am Meßobjekt (z.B. eine Drehachse) angebracht werden.

Für den Anschluß an die C-Control-Unit gibt es zwei Möglichkeiten:

1. Anschluß an den DCF-77-Eingang
2. Anschluß an einen der Digital-Ports

Im ersten Fall ist ein gleichzeitiger Betrieb der Funkuhr nicht möglich, dafür wird das Programm zur Messung der Windgeschwindigkeit einfacher.

Zum Einstellen des Teilverhältnisses müssen Sie das Modulgehäuse öffnen. Schalten Sie dazu die Spannungsversorgung ab.

Wenn Sie eine unbeeinflusste Weitergabe der Meßergebnisse wünschen, muß der **Jumper** J11 auf die Stellung "Dir" gesteckt sein. Andernfalls stecken Sie ihn in die Stellung "Teil".

Dann können Sie über die **Jumper** J1 ... J9 ein Teilverhältnis von 2 bis 10 wählen. Das zum jeweiligen Steckplatz gehörende Teilverhältnis ist auf der Platine angegeben.

### Programmtechnische Realisierung der Messung

Wird der Sensor an den DCF-77-Eingang angeschlossen, erfolgt die Messung über die BASIC-Funktion **FREQ**. Das Betriebssystem zählt im Hintergrund, d.h. ständig und ohne Zutun des Anwenders, die während einer Sekunde am DCF-77-Eingang registrierten Impulse. Der Aufruf der Funktion **FREQ** in einem BASIC-Programm gibt dann jeweils das letzte vollständige Zählergebnis zurück.

Wird der Sensor an einen der Digital-Ports angeschlossen, ist der Vorgang der Impulszählung vom Anwender selbst zu programmieren.

### Kalibrieren

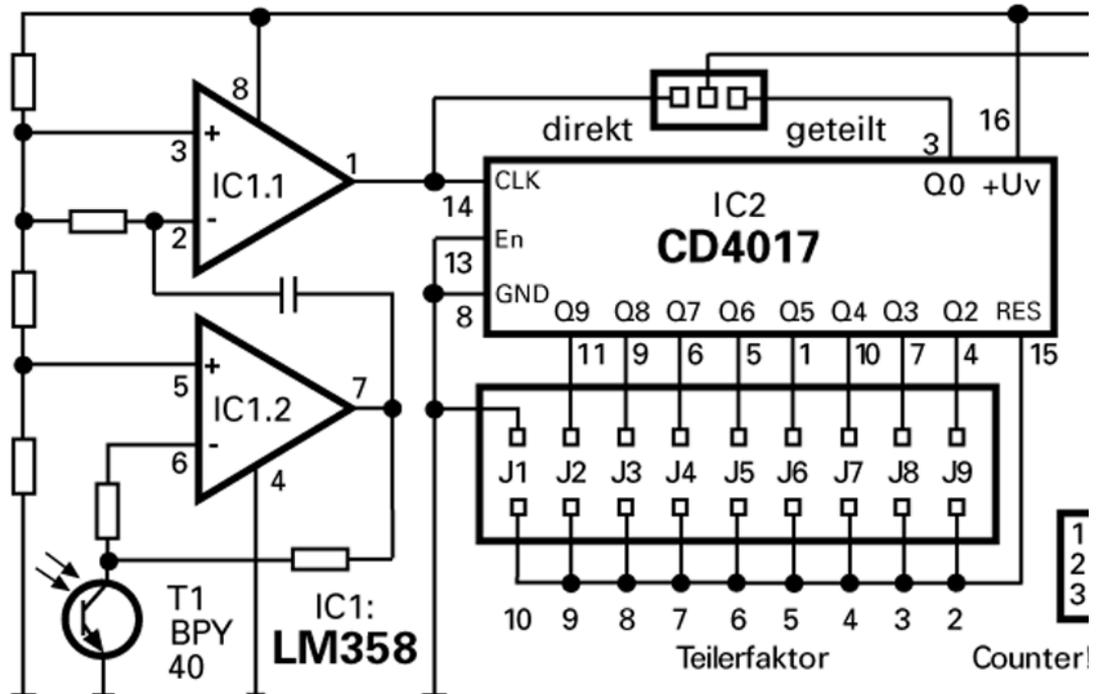
Das Sensormodul ist bereits werksseitig abgeglichen. Ein Abgleich von Ihrer Seite ist daher nicht erforderlich.

### Technische Daten

Betriebstemperatur des Moduls (für die angegebene Toleranz)  
Betriebsspannung (von der Wetterstation)  
Stromaufnahme  
Meßbereich

0 ... + 70 °C  
6,5 Volt  
ca. 1,6 mA  
10 .60000 U/min

Schaltplan:



## Allgemeine Handhabungs- und Sicherheitshinweise

Der Gleichstrom-Sensor ist ein Zubehör zum modularen Telemetrie-System. Er kann aber ohne weiteres auch an die C-Control-Unit angeschlossen werden. Mit ihm kann Gleichstrom in zwei Meßbereichen von 0...2,55 A bzw. 0...25,5 A gemessen werden. Die Aufzeichnung und Anzeige der Meßdaten sind vom Anwender durch ein geeignetes C-Control-Programm zu realisieren.

### Aufbau und Funktion

Der Gleichstrom-Sensor dient zur Messung der Stromstärke. Der entsprechende **Sensor** sowie die dazugehörige Auswerteelektronik sind in einem Modulgehäuse untergebracht. Die Meßwerte werden in Form eines **normierten Ausgangssignals** geliefert. Es handelt sich dabei um eine **Spannung** im Bereich von 0 ... 2,55 Volt.

Die Meßsignale werden über die Anschlußleitung zu einem der acht A/D-Ports der Control-Unit übertragen und dort **digitalisiert**. Jeder Meßwert wird als Zahl im Bereich von 0 ... 255 gespeichert. Die Zahlenwerte können umgerechnet und dann in der gewohnten Einheit Ampere [A] ausgegeben oder weiterverarbeitet werden.

Dieses Konzept hat den Vorteil, daß die Signale unterschiedlichster Sensoren im **Byte-Format** erfaßt werden können. Die Berücksichtigung der speziellen Sensoreigenschaften erfolgt erst bei der Auswertung durch eine geeignete Software.

### Gleichstrommessung

Als **Sensor** kommt ein **Präzisions-Shuntwiderstand** zum Einsatz. Der zu messende Strom wird an die herausgeführten Leitungen angelegt und bewirkt einen Spannungsabfall. Um diesen und somit die **Verlustleistung** klein zu halten hat der **Shuntwiderstand** den sehr geringen Wert von 0,01 Ohm.

Um zu vermeiden, daß der Spannungsabfall an **Kontaktwiderständen**, Lötstellen, Meßklemmen, etc. das Ergebnis verfälschen, ist der Meßwiderstand in **Vierleitertechnik** ausgeführt. Die abzugreifende Meßspannung wird verstärkt und kann nun von der C-Control-Unit verarbeitet werden. Das Meßergebnis ist zunächst eine Zahl im Bereich von 0 ... 255. Der dazugehörige Meßwert für die beiden einstellbaren Meßbereiche in Ampere muß also erst noch berechnet werden.

### Installation und Anschluß

Das Sensormodul wird betriebsfertig geliefert. Zur Installation muß es lediglich mit einem A/D-Port der Control-Unit verbunden werden.

Sie können zwischen zwei Meßbereichen wählen. Um eine Einstellung vorzunehmen, müssen Sie das Modulgehäuse öffnen. Der **Jumper J3** kann wahlweise auf 2,55 oder auf 25,5 A Meßbereich gesteckt werden. Kleben Sie nach erfolgter Einstellung das Gehäuse wieder zu. Sie verwenden am besten neues Klebeband.

Beim Anschluß des Sensormoduls an das Meßobjekt muß das Meßobjekt unbedingt spannungsfrei sein! Außerdem sind die folgenden Anschlußregeln zu beachten. Ansonsten können Sensormodul und Telemetrie-Haupteinheit zerstört werden.

Grundsätzlich sind zwei Anwendungsfälle zu unterscheiden: Messen mit gemeinsamer oder getrennter Gleichspannungsquelle von Meßobjekt und Telemetrie-Haupteinheit. Gemeinsame Spannungsquelle heißt, daß ein gemeinsames Netzteil, eine Batterie oder ein Akku verwendet wird. In diesem Fall muß der Stromsensor in den Minuszweig des vermessenen Stromkreises eingeschaltet werden. Bei getrennter Spannungsversorgung kann die Strommessung auch im Pluszweig erfolgen.

Bei beiden Meßmöglichkeiten sind die Polaritätsangaben zu beachten. Werden die Meßleitungen am Stromsensor vertauscht, ist eine Strommessung nicht möglich, auch wenn Strom fließt.

Bei geschlossenem Sensor-Modulgehäuse ist ein hoher Stromfluß zeitlich unbegrenzt nicht möglich, da es dann zu einer starken Erwärmung des Shunts kommt, wodurch das Modul zerstört werden kann. Es gelten folgende maximale Betriebszeiten bei gegebener Stromstärke und 20 °C Umgebungstemperatur:

Strom [A]	max. Meßzeit [min]
5	unbegrenzt
10	10
15	2
25	1

Sind hohe Ströme über einen längeren Zeitraum zu messen, so ist der Stromsensor ohne Modulgehäuse zu betreiben und der Meßwiderstand R7 (Shunt) so aufzurichten, daß er senkrecht zur Platine steht. Desweiteren besteht die Möglichkeit am Shunt einen Kühlkörper zu montieren.

Wurde das Modulgehäuse entfernt, so muß darauf geachtet werden, daß die Platine nicht mit Metallteilen in Berührung kommt. Es besteht Kurzschlußgefahr!

Dann muß der Sensor an einen der acht A/D-Ports der Control-Unit angeschlossen werden. Überzeugen Sie sich daß die Spannungsversorgung abgeschaltet ist und trennen sie eine eventuelle Kabelverbindung zum **PC**. Wählen Sie einen freien A/D-Port (J33-J40 auf dem Starterboard) und stecken Sie die dreipolige Buchse so auf den gewählten Portstecker, daß das weiße Kabel (Signalader) zum Experimentierfeld des Starterboards zeigt. Das Anschlußkabel darf um maximal zwei Meter verlängert werden. Der Jumper zur Wahl der Referenzspannung der A/D-Wandler muß auf 2,5 V gesetzt sein.

Nach dem Zuschalten der Spannungsversorgung benötigt das Modul etwa eine Minute zur internen Stabilisierung. Beachten Sie bitte, daß vorher keine gültigen Meßwerte gewonnen werden können.

#### **Programmtechnische Realisierung einer Gleichstrommessung**

Nach dem Einlesen des aktuellen Abtastwertes muß dieser noch in einen Wert für den Gleichstrom umgerechnet werden. Für den Meßbereich 0..2,55 A ist der Abtastwert mit dem Faktor 0.01 zu multiplizieren. Für den Meßbereich 0..25,5 A ist der Faktor 0.1 zu verwenden. Der so ermittelte Wert kann dann über die serielle Schnittstelle zur Anzeige gebracht werden (s.a. Datenauswertung).

#### **Kalibrieren**

Das Sensormodul ist bereits werksseitig abgeglichen. Ein Abgleich von Ihrer Seite ist daher nicht erforderlich.

## Technische Daten

### Modul

Betriebsspannung (von der Haupteinheit)  
Stromaufnahme  
Betriebstemperatur für die angegebenen Toleranzen

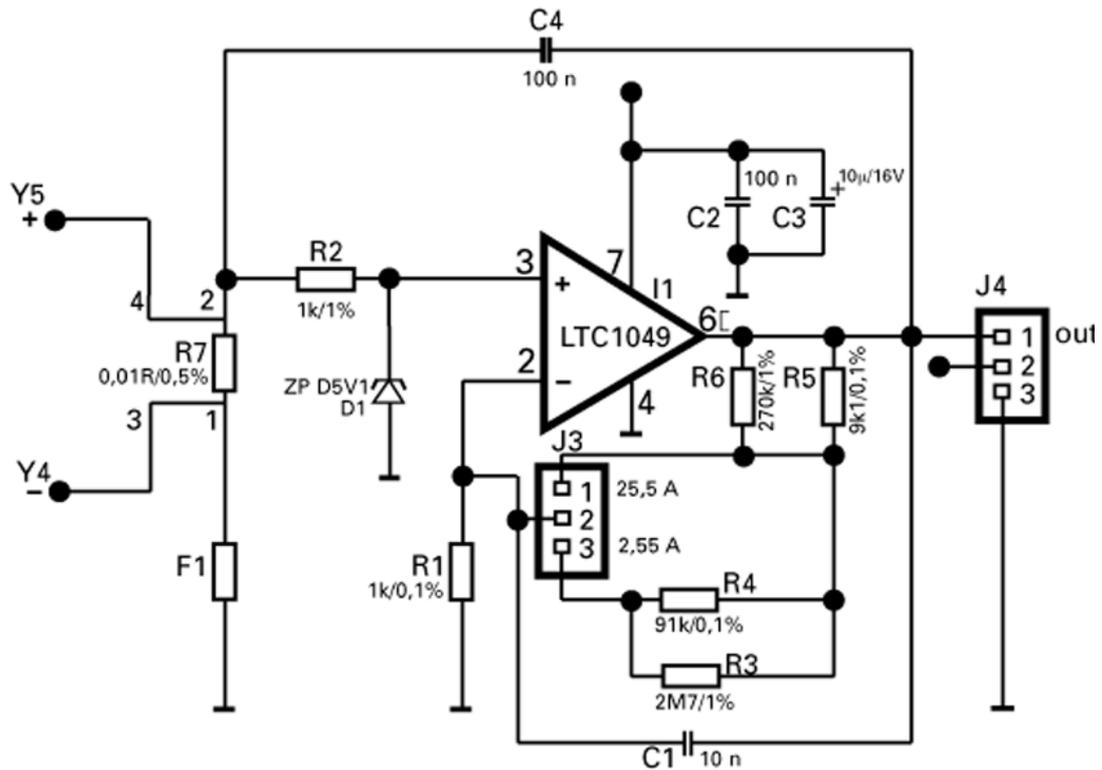
6,5 Volt  
ca. 0,15 mA  
- 40 ... + 85 °C

### Sensor

Toleranz  
Meßbereich 1  
Auflösung Meßbereich 1  
Meßbereich 2  
Auflösung Meßbereich 2

+/- 1 %  
0 ... 2,55 A  
10 mA  
0 ... 25,5 A  
100 mA

Schaltplan:



## Allgemeine Handhabungs- und Sicherheitshinweise

### Aufbau und Funktion

Der Spannungssensor dient zur Messung von Gleichspannung. Es handelt sich dabei um einen einfachen **Spannungsteiler**, der für eine Bereichsanpassung sorgt. Dieser ist in Isolierschlauch eingeschrumpft. Die Meßwerte werden in Form eines **normierten Ausgangssignals** geliefert. Es handelt sich dabei um eine **Spannung** im Bereich von 0 ... 2,55 Volt.

Der Sensor wird an einen der acht A/D-Ports der Control-Unit übertragen.

Jeder Meßwert wird als Zahl im Bereich von 0 ... 255 gespeichert. Die Zahlenwerte können umgerechnet und dann in der gewohnten Einheit Volt [V] abgelesen und weiterverarbeitet werden.

Dieses Konzept hat den Vorteil, daß die Signale unterschiedlichster Sensoren im **Byte-Format** erfaßt werden können. Die Berücksichtigung der speziellen Sensoreigenschaften erfolgt erst bei der Auswertung durch eine geeignete Software.

### Spannungsmessung

Der **Spannungsteiler** hat die Aufgabe, die Meßspannung so zu teilen, daß der gewünschte Meßbereich auf das **normierte Ausgangssignal** von 0 ... 2,55 V "abgebildet" wird. Es sind zwei Spannungssensoren verfügbar.

Für den Meßbereich 0 ... 10 V wird eine Teilung des Eingangssignal im Verhältnis von etwa 1 : 4 vorgenommen. Hierzu sind zwei Widerstände mit den Werten 30 kOhm und 10 kOhm in Reihe geschaltet. Der Bereich 0 ... 25 V erfordert ein Teilungsverhältnis von etwa 1/10. Die hierzu erforderlichen Widerstandswerte haben Werte von 53 kOhm und 6 kOhm.

Das zwischen den beiden Widerständen abzugreifende Meßsignal kann von der Control-Unit direkt verarbeitet werden. Das Meßergebnis ist zunächst eine Zahl im Bereich von 0 ... 255. Der dazugehörige Meßwert in Volt muß also erst noch berechnet werden.

### Installation und Anschluß

Das Sensormodul wird betriebsfertig geliefert und muß lediglich mit einem A/D-Port der Control-Unit verbunden werden.

Jetzt muß der Sensor an einen der acht A/D-Ports der Control-Unit angeschlossen werden. Überzeugen Sie sich daß die Spannungsversorgung abgeschaltet ist und trennen sie eine eventuelle Kabelverbindung zum **PC**. Wählen Sie einen freien A/D-Port (J33-J40 auf dem Starterboard) und stecken Sie die dreipolige Buchse so auf den gewählten Portstecker, daß das weiße Kabel (Signalader) zum Experimentierfeld des Starterboards zeigt. Das Anschlußkabel darf um maximal zwei Meter verlängert werden. Der Jumper zur Wahl der Referenzspannung der A/D-Wandler muß auf 2,5V gesetzt sein.

### Programmetechnische Realisierung einer Spannungsmessung

Nach dem Einlesen des aktuellen Abtastwertes muß dieser noch in einen Wert für die Spannung umgesetzt werden. Dazu ist der Abtastwert für den 10 V-Sensor durch 25,5 zu teilen, für den 25 V-Sensor durch 10. Das Ergebnis kann dann über die serielle Schnittstelle zur Anzeige gebracht werden (s.a. Datenauswertung).

### Kalibrieren

Das Sensormodul ist bereits werksseitig abgeglichen. Ein Abgleich von Ihrer Seite ist daher nicht erforderlich.

### Technische Daten

Meßbereich

0...10 Volt  
bzw. 0...25 Volt

Meßtoleranz

2%

Eingangswiderstand beim 10-Volt-Sensor

ca. 40 kOhm

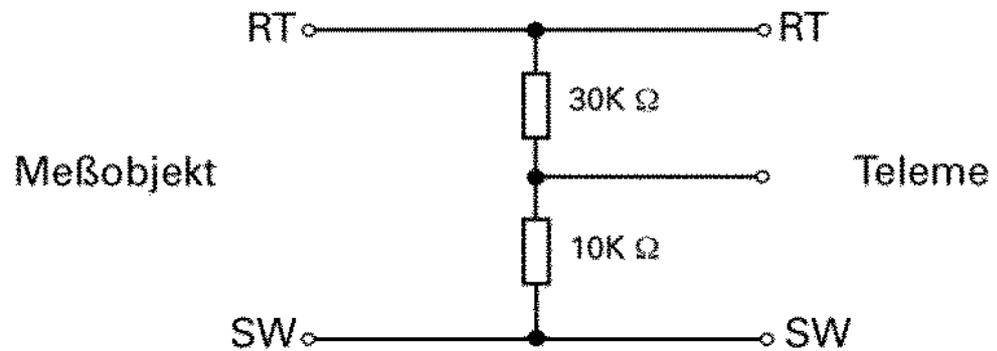
Eingangswiderstand beim 25-Volt-Sensor

60 kOhm

Schaltplan:

## Spannungs-Sensor

### Spannungsteiler 10 V



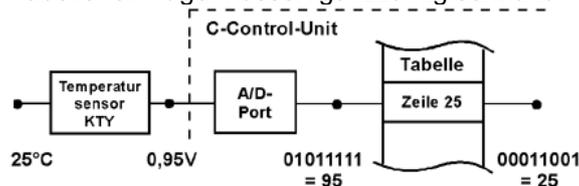
## Anwendung

### Datenerfassung

Sollen analoge Signale erfaßt werden (z.B. Temperatur, Drehzahl), müssen diese Signale zunächst so umgewandelt werden, daß sie von der Control-Unit verarbeitet werden können. Dazu dient in der Regel ein Sensor, der die zu messende physikalische Größe in ein elektrisches Signal umsetzt. Die Meßgröße äußert sich dann durch eine Spannung (s.a. Temperatursensor KTY), oder eine Impulsfrequenz (s.a. Drehzahlmesser). Eine Spannung stellt ein analoges elektrisches Signal, eine Impulsfrequenz ein digitales Signal dar. Analoge Signale können durch die Control-Unit an den acht prozessoreigenen A/D-Ports verarbeitet werden (s.a. Analoge Signalverarbeitung). Diese Ports wandeln (ohne Zutun des Anwenders) analoge Spannungen im Bereich von 0... Referenzspannung in digitale Werte zwischen 0 und 255 um, die dann die jeweilige Meßgröße repräsentieren. Digitale Signale (z.B. Impulsfrequenz) werden an den digitalen Ports der Control-Unit eingelesen (s.a. Digitale Signalverarbeitung).

Liegt das Meßsignal als digitaler Wert vor, muß dieser so umgerechnet werden, daß er unmittelbar den richtigen Zahlenwert der gemessenen physikalischen Größe darstellt. Wurde z.B. eine Temperatur von 25 °C mit dem Temperatursensor KTY gemessen, so liegt an dessen Ausgang eine Spannung von ca. 0,95V. Der am Sensor angeschlossene A/D-Port erzeugt daraus den digitalen Wert 95. Um daraus den richtigen Zahlenwert für die Temperatur zu machen müßte der digitale Wert also noch mit dem Faktor  $25/95 = 0,26$  multipliziert werden. Da einerseits Multiplikationen für einen Prozessor eine rechenintensive Aufgabe darstellen und andererseits für viele Sensoren ein derartiger Faktor von der Größe des Meßwertes abhängt (nichtlineare Kennlinie) stellt diese Umrechnungsmöglichkeit keine ideale Lösung dar.

Wesentlich günstiger für die schnelle Umrechnung von Werten ist die Verwendung von Datentabellen. Soll beispielsweise die Kennlinie eines Temperatursensors korrigiert werden, wird der Meßbereich einmal durchfahren, und für jeden digitalen Meßwert der Control-Unit die zugehörige exakte Temperatur mit einem genauen Thermometer gemessen. Der jeweilige exakte Temperaturwert wird in einer Tabelle so abgelegt, das der zugehörige digitale Abtastwert die Zeilennummer des Tabelleneintrages angibt, in der der exakte Wert gespeichert ist. Zum Lieferumfang der Control-Unit gehören einige fertige Tabellen, mit deren Hilfe die Kennlinien der als Zubehör erhältlichen Sensoren (s.a. Sensoren) linearisiert werden können (z.B. kty.tab für den Temperatursensor). Tabellen können mit einem einfachen Texteditor aber ohne weiteres auch selbst erstellt werden. Dazu sind die entsprechenden Werte als Text zeilenweise in eine Datei mit der Endung .tab einzutragen. Die Tabelleneinträge müssen ganzzahlig sein und im Bereich von -32768 ... 32767 liegen.



Im folgenden soll die Anwendung von Datentabellen für CC-BASIC anhand einer Temperaturmessung demonstriert werden (siehe Beispiel kty.bas).

Am A/D-Port 1 der Control-Unit ist der Temperatursensor KTY (Best-Nr. 108812) angeschlossen.

Durch den CC-BASIC-Befehl

```
TABLE temptab "kty.tab"
```

am Ende des Quelltextes, wird die mitgelieferte Tabelle kty.tab unter dem Namen "temptab" an den Quelltext angehängt. Angenommen die Control-Unit liest jetzt an ihrem A/D-Port den digitalen Wert 36 ein und legt ihn in der Variable "sample" ab.  
Durch den CC- BASIC- Befehl

```
LOOKTAB temptab, sample, temp
```

wird dann der Wert, der an 36. Stelle in der Tabelle mit dem Namen temptab steht (hier 32) in die Variable "temp" kopiert. Dieser Wert entspricht dann der exakten Temperatur und kann z.B. ausgegeben werden.

Auch unter dem grafischen Entwicklungssystem C-Control (Classic) ist der Einsatz von Datentabellen vorgesehen. Im folgenden soll die Anwendung anhand einer Temperaturmessung demonstriert werden (siehe Beispiel kty.bnt). Am A/D-Port 1 der Control-Unit ist der Temperatursensor KTY angeschlossen. Das Ergebnis wird auf Tabelle1 gegeben, der unter dem Menü *Einstellungen/ Tabelle1...* die Datei "kty.tab" zugeordnet wurde. Der Ausgang des Tabellenblocks gibt dann den jeweils korrigierten Wert (also die exakte Temperatur) aus.

### Datenübertragung/ -speicherung

Gemessene und aufgezeichnete Daten können über die serielle Schnittstelle zu einem PC übertragen werden (s.a. Kommunikation über die serielle Schnittstelle).

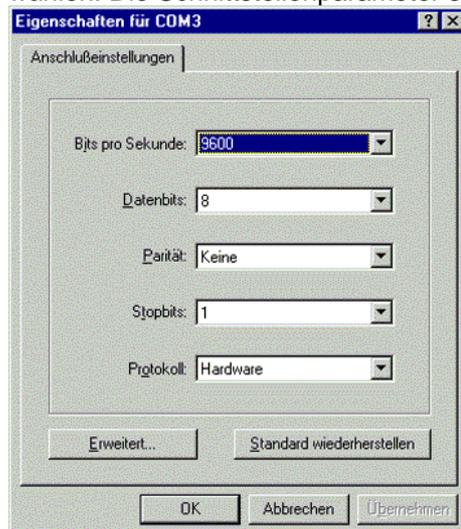
In CC-BASIC dienen die Befehle PRINT und PUT zum Senden von Zeichen und numerischen Werten (s.a. Befehle zur Kommunikation über die serielle Schnittstelle).

Im Beispiel-Programm kty.bas gibt die Zeile

```
PRINT temp
```

den Inhalt der Variablen "temp" als Text über die serielle Schnittstelle aus. Zum Empfang auf dem PC kann das Windows95-Programm Hyperterminal verwendet werden. Sollte es nicht installiert sein, können Sie das noch nachträglich mittels *Systemsteuerung/Software/Windows Setup*.

Unter *Start/Programme/Zubehör/Hyperterminal* öffnet sich der Ordner Hyperterminal. Durch doppeltes Anklicken des Icons "Hypertrm" öffnet sich das Dialogfenster "Neue Verbindung". Geben Sie einen Namen (z.B. "C-Control") und ein Symbol für die Verbindung an. Nach dem Bestätigen Ihrer Auswahl werden Sie zum Konfigurieren der Verbindung aufgefordert. Hier ist zunächst im Optionsfeld "Verbinden über" die Einstellung "Direktverbindung über COMx" zu wählen. Die Schnittstellenparameter sind standardmäßig wie folgt einzustellen:



Für die Übertragungsrate "Bit pro Sekunde" kann auch ein anderer Wert gewählt werden. Dieser ist dann natürlich auch für die Schnittstelle der Control-Unit einzustellen (siehe CC-BASIC-Befehl BAUD). Danach ist die Konfiguration der Verbindung abgeschlossen und Sie können sie jederzeit unter dem gewählten Icon erneut laden. Nach dem Starten des BASIC-Programmes "kty.bas" werden jetzt die gemessenen Temperaturen im Sekundenrhythmus angezeigt. Sollen die Werte gespeichert werden, ist im Hyperterminal-Menü *Übertragung/Text aufzeichnen ...* der Name einer Textdatei anzugeben, in die die Werte geschrieben werden sollen. Beendet wird die Aufzeichnung durch den Hyperterminal-Menüpunkt *Übertragung/Text aufzeichnen .../Beenden*. Die aufgezeichneten Werte können jetzt am PC weiterverarbeitet werden.

Bei der grafischen Programmierumgebung C-Control ist der eben beschrittene Weg der Datenaufzeichnung am PC mit Hilfe eines Terminalprogrammes (z.B. Hyperterminal) ebenfalls möglich (siehe Beispiel kty.bnt). Dazu ist ein Ergebnisausgabeblock einzubinden (s.a. Aufzeichnung eines Meßwertes im Speicher der Control-Unit).

[*Tip: Ist Hyperterminal gestartet worden, kann ein CControl-Programm nur von der Control-Unit aus und nicht von der Entwicklungsumgebung gestartet und angehalten werden.*]

Eine andere Möglichkeit besteht darin, die gewonnenen Daten zunächst durch das Einbinden eines Aufzeichnungsblocks im Logbuch abzulegen (siehe Beispiel kty.bnt). Nach dem Anhalten des Programmes durch den Menüpunkt *Programm/Stop* können die aufgezeichneten Daten mit *Datei/Daten auslesen* zum PC übertragen werden. Zum Abschluß dieses Vorgangs ist es möglich, sich die aufgezeichneten Daten als grafische Darstellung anzeigen zu lassen. Auf jeden Fall jedoch wird der entsprechende Datensatz in der Datei "logbook.txt" gespeichert. Diese Datei wird immer im aktuellen Programm-Verzeichnis angelegt. Die einzelnen Daten werden zeilenweise als Text gespeichert. Jede Zeile beginnt mit einem "V", dem dann der jeweilige immer ganzzahlige Wert folgt.

[*Tip: Automatisches Entfernen des führenden "V " bzw.. "T" in "logbook.txt"*

Für eine Weiterverarbeitung der Meßwerte in "logbook.txt" (siehe Datenauswertung) kann das führende "V" hinderlich sein. Durch die Funktion "Ersetzen" eines einfachen Textverarbeitungsprogrammes (z.B. mit dem Windows95-Programm WordPad) lassen sich die führenden Zeichen einfach aus der Datei entfernen.]

Bei der grafischen Entwicklungsumgebung kann es aufgrund von Timing-Problemen des Prozessors dazu kommen, daß einzelne Meßwerte doppelt im Logbuch abgelegt werden. Bei der Aufzeichnung von Daten von mehreren Signalquellen oder Ereigniszeitpunkten wird dadurch die Reihenfolge der Daten gestört, eine spätere Zuordnung der Daten zu der entsprechenden Signalquelle wird schwierig. Diese Programmversion von C-Control ist daher nur bedingt für diesen Zweck einsetzbar. Es wird empfohlen, jeweils nur eine Datenquelle aufzuzeichnen.

[*Tip: Sollten Sie Schwierigkeiten mit Ihren WINCOM-Treibern haben, können Sie aus dem Verzeichnis "Treiber" auf der CD-ROM einen anderen Treiber installieren. Lesen Sie davor unbedingt die Datei "readme.txt". CONRAD Electronic übernimmt keine Garantie oder Haftung für Fremdsoftware.*]

### **Datenauswertung/-anzeige**

Die Auswertung der gewonnenen Daten kann prinzipiell durch ein C-Control-Programm oder nach einer Übertragung am PC erfolgen. Ein C-Control-Programm kann die betreffenden Daten dazu verwenden, bestimmte Schaltvorgänge (Schalten eines Relais) oder eine entsprechende Signalisierung (LED, Signalton) auszulösen.

Sollen Datenwerte in übersichtlicher Form angezeigt und archiviert bzw. zusammengefaßt werden, bietet sich die Übertragung zum PC und dortige Weiterverarbeitung an.

Prinzipiell besteht natürlich die Möglichkeit eigene PC-Programme zu erstellen (z.B. Erweiterung des zur graphischen Entwicklungsumgebung mitgelieferten BASIC- Programmes "terminal.bas"), die diese Aufgaben bewerkstelligen. Einfacher zu handhaben sind aber Standardanwendungen, wie z.B. das Programm Microsoft® EXCEL. Dieses Programm zur Tabellenkalkulation bietet die Möglichkeit numerische Daten aus Textdateien einzulesen und in eigenen Tabellen zu speichern. Verwenden Sie dazu den Menüpunkt *Datei/Daten importieren*. Sind die Daten geladen, können Sie unter dem Menüpunkt *Einfügen/Diagramm* sehr einfach als Diagramm in fast beliebiger Darstellungsform angezeigt werden. Lesen Sie gegebenenfalls die entsprechenden Hilfe-Texte.

## Tips und Tricks

### Hardware / C-Control-Unit

#### **Tip: Das Buch zum Prozessor**

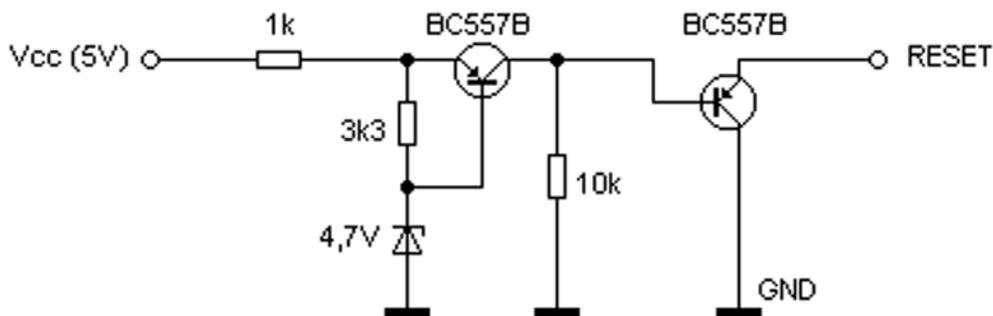
Genauere Auskunft über den Prozessor und seine Programmierung gibt das Buch "Motorola 68HC05" von Zekeriya Zengin aus dem Heise-Verlag (ISBN 3-88229-034-X, Conrad Electronic Best.-Nr.: 91 91 79)]

#### **Tip: Anschluß von externen LEDs**

Die Kathoden der drei LEDs der Control-Unit sind an der Buchsenleiste 2 (Belegung siehe Technische Spezifikation) verfügbar. Wird der Jumper "LED" abgezogen und damit die LEDs auf der Baugruppe ausgeschaltet, können an diese Anschlüsse externe LEDs und ein Widerstand zur Strombegrenzung angeschlossen werden, die dann den Systemzustand z.B. an der Frontblende eines Gehäuses anzeigen. Ebenso sind die parallel zum Start- und Reset-Taster liegende Anschlüsse an diese Buchsenleiste herausgeführt. Es ist also möglich, jeweils einen funktionsgleichen externen Taster zu realisieren.

#### **Tip: Überwachung der Betriebsspannung der Control-Unit**

Sollte bei einem Dauerbetrieb des Steuercomputers die Gefahr eines Spannungsausfalls bestehen, ist eine externe Spannungswächterschaltung (Watchdog) mit der 5V-Betriebsspannung, Masse (GND) und dem Reset-Port zu verbinden. Optimal ist dafür das IC MAX809L des Halbleiterherstellers MAXIM geeignet, das einen Reset bei Unterschreitung des Spannungsschwellwertes von 4,65V auslöst. Alternativ dazu kann auch folgende Transistorschaltung aufgebaut werden:



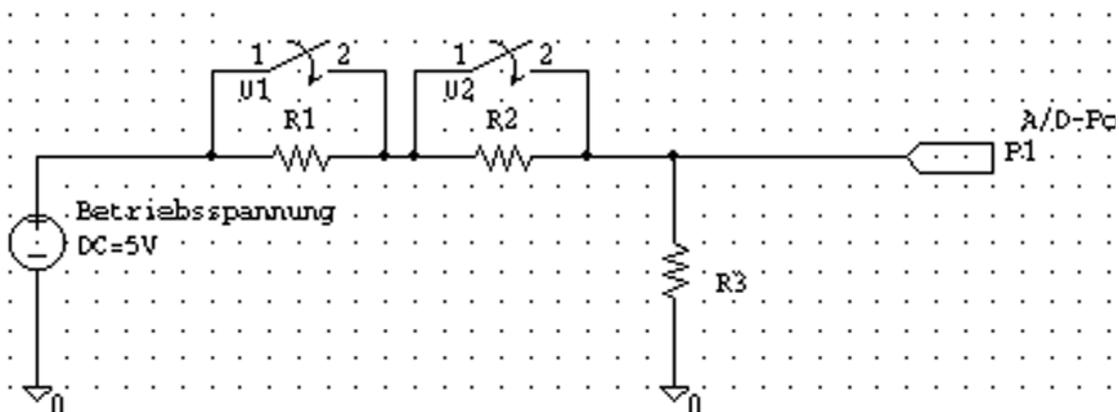
#### **Tip: Anpassung einer Meßspannung an den Spannungsbereich der A/D-Ports**

Der Maximalwert der Meßspannung sollte nach Möglichkeit nur unwesentlich geringer sein als die Referenzspannung, da man ansonsten Meßgenauigkeit verschenkt. Der Bereich der zu wandelnden Eingangsspannung läßt sich auf den Eingangsbereich des A/D-Wandlers anpassen, indem man einen nichtinvertierenden Operationsverstärker mit einer entsprechenden Verstärkung vorschaltet. Will man z.B. eine Spannung im Bereich von 0...10V an einen 5V-Wandelbereich des A/D-Umsetzers anpassen, benötigt man einen Operationsverstärker mit der Verstärkung 0,5. Dieser Operationsverstärker schützt gleichzeitig die Eingänge des A/D-Wandlers vor Überspannungen. Die Vorverstärkung ist bei der weiteren Auswertung der gemessenen Werte per Software zu berücksichtigen. Eine andere Möglichkeit besteht darin, die Referenzspannung an den Eingangsbereich anzupassen. Für einfachste Anwendungen kann ein nicht zu hochohmiger Eingangsspannungsteiler die Spannungsanpassung realisieren.

### **Tip: Tastatur an einen A/D-Port anschließen**

Eine Methode, an die Control-Unit eine kleine Tastatur anzuschließen ohne viele Eingänge zu belegen ist im folgenden Bild für zwei Tasten dargestellt. Das Betätigen einer Taste schließt einen Widerstand kurz, der Bestandteil eines Spannungsteilers ist. Dementsprechend verändert sich die Ausgangsspannung des Spannungsteilers. Diese Spannung liegt an einem A/D-Port der Control-Unit und wird durch ein Tastaturabfrage-Programm ausgewertet. Folgende Gesichtspunkte sind bei der Realisierung zu beachten:

- Der Widerstand, der parallel zum A/D-Port liegt, sollte 1 kOhm nicht überschreiten.
- Die Werte der den Schaltern zugeordneten Widerstände sollten maximal weit auseinander liegen, um am Ausgang möglichst unterschiedliche Spannungen zu erhalten.
- Es ist darauf zu achten, daß es nicht aufgrund von Toleranzen der einzelnen Widerstände zu Überlappungen der den Schaltern zugeordneten Spannungsbereiche kommen kann.
- Auch das gleichzeitige Drücken von mehreren Tasten muß zweifelsfrei als solches erkannt werden können.
- Im Tastatur-Programm ist zu berücksichtigen, daß mechanische Tasten bei ihrer Betätigung prellen, d.h. beim Übergang von "Zu" nach "Auf" und umgekehrt mehrere Male ihren Zustand wechseln, bevor sie stabil geschaltet haben. Im Programm ist deshalb ein Mehrheitsentscheid anhand von drei, im Abstand von ca. 20ms gemessenen Spannungswerten, vorzusehen.



### **Tip: Erweiterung der Anzahl der A/D-Ports**

Sollten die acht zur Verfügung stehenden A/D-Ports nicht ausreichen, besteht die Möglichkeit durch eine Zusatzschaltung weitere Ports zur Verfügung zu stellen. Dazu wird an einen A/D-Port der Control-Unit ein Umschalter angeschlossen, der es erlaubt, aus mehrere Signalquellen (z.B. vier Temperatursensoren) eine auf den A/D-Port durchzuschalten. Im einfachsten Fall kann dafür ein geeigneter mechanischer Umschalter eingesetzt werden. Eine bessere und einfache Möglichkeit stellt die Verwendung eines elektronischen Umschalters (Analogschalter/ Analogmultiplexer) dar. Ein derartiger IC ist z.B. der Standardschaltkreis 4051. Mit ihm kann durch das Anlegen von einer aus acht möglichen Kombinationen an die drei digitalen Steuerleitungen, eins von acht analogen Signalen auf den Schaltkreisausgang durchgeschaltet werden. Schließt man an die drei Steuerleitungen dieses Analogmultiplexers drei Digitalports der Control-Unit an, kann innerhalb eines C-Control-Programmes durch geeignetes Ein- und Ausschalten der Ports eine Signalquelle ausgewählt werden.

### **Tip: Installation der DCF77-Aktivantenne außerhalb von Stahlbeton-Gebäuden**

Die Wände vieler moderner Bauten sind zur Stabilisierung mit einem Stahlgeflecht versehen. Dieses Geflecht kann abschirmende Wirkung haben, so daß ein Empfang des DCF77-Signals nur außerhalb solcher Gebäude möglich ist. Die Aktivantenne sollte dann in einem wasserdichten Gehäuse untergebracht werden.

***Tip: Störung des DCF77-Empfangs durch elektromagnetische Störfelder***

Die elektromagnetische Strahlung, die von einem PC und einem Monitor ausgeht, kann den DCF77-Empfang beeinträchtigen. Es wird empfohlen, einen möglichst großen Abstand zwischen Antenne und PC einzuhalten.

## **Software C-Control Classic**

***Tip: Drucken eines grafischen Programms***

Die grafische Entwicklungsumgebung verfügt nicht über eine eigene Funktion zum Drucken. Durch das gleichzeitige Drücken der Tasten [ALT][Druck] wird unter Windows 95 ein Screenshot des aktuellen Fensters in die Zwischenablage gelegt. Diese kann man sich dann durch das Aufrufen von *Start/Zubehör/Zwischenablage* ansehen und den Inhalt als \*.clp Datei abspeichern. Diese lässt sich unter einem entsprechenden Grafik-Programm ausdrucken.

## **Software C-Control BASIC**

***Tip: Verwendung des integrierten Hilfesystems***

Die Bedienung der Integrierten Entwicklungsumgebung wird durch das enthaltene Hilfesystem unterstützt. Sehr hilfreich ist die Referenzliste der reservierten Worte [*Lexikon*] von CC-BASIC, die teilweise auch mit Beispielen versehen ist.

***Tip: Auskommentieren von Programmzeilen***

Soll eine Programmanweisung zu Testzwecken aus dem Programm entfernt werden, genügt es, durch das Vorsetzen eines Hochkommata die betreffende Zeile "auszukommentieren".

***Tip: Reihenfolge beim Auslesen der Variablen der Echtzeituhr***

Die Uhr läuft während des Auslesevorgangs weiter. Daher kann es vorkommen, dass während des Auslesens ein Stellenüberlauf auftritt (z.B. von 23:59:59, 31.12.1999 nach 00:00:00, 01.01.00 bei der kommenden Jahrtausendwende). Deshalb sollte der Sekundenwert zuerst ausgelesen werden. Am Ende des Auslesevorgangs sollte er dann noch einmal abgefragt und darauf überprüft werden, ob er seinen Maximalwert überschritten hat. Ist das der Fall, ist es am einfachsten, den gesamten Vorgang zu wiederholen.

***Tip: Multiplikation/ Division durch Anwendung von Schiebebefehlen***

Durch die Anwendung der bitweisen Operatoren können innerhalb eines Bytes einzelne Bits gesetzt oder rückgesetzt (ausgeblendet) werden. Ist z.B. der Zustand eines Byteports in die Variable "ventile" eingelesen worden, von dem aber nur der Zustand des Bits 1, der das Ventil 1 überwacht, interessiert, können alle anderen Bits durch die Anweisung `ventil1 = ventile AND &B00000010` ausgeblendet werden. War Bit 1 gesetzt, steht danach in der Variablen "ventil1" der Wert 2 andernfalls der Wert 0. Der Zustand aller anderen Bits ist ausgeblendet worden. Der Inhalt der Variablen "ventile" selbst bleibt erhalten.

***Tip: Übersichtliche Gestaltung von Programmen***

Ein Programm wird übersichtlicher, wenn man sehr umfangreiche Berechnungen auf mehrere Zeilen aufteilt.

## Anwendung

### ***Tip: Hyperterminal und die grafische Entwicklungsumgebung***

Ist Hyperterminal gestartet worden, kann ein CControl-Programm nur von der Control-Unit aus und nicht von der Entwicklungsumgebung gestartet und angehalten werden.

### ***Tip: Automatisches Entfernen des führenden "V " bzw.. "T" in "logbook.txt"***

Für eine Weiterverarbeitung der Meßwerte in "logbook.txt" (siehe Datenauswertung) kann das führende "V" hinderlich sein. Durch die Funktion "Ersetzen" eines einfachen Textverarbeitungsprogrammes (z.B. mit dem Windows95-Programm WordPad) lassen sich die führenden Zeichen einfach aus der Datei entfernen.

***Tip:*** Sollten Sie Schwierigkeiten mit Ihren WINCOM-Treibern haben, können Sie aus dem Verzeichnis "Treiber" auf der CD-ROM einen anderen Treiber installieren. Lesen Sie davor unbedingt die Datei "readme.txt". CONRAD Electronic übernimmt keine Garantie oder Haftung für Fremdsoftware.

## Sensoren

### Windrichtungsmesser

#### ***Tip: Multiplikation mit einer gebrochenen Zahl unter CC-BASIC***

Soll die Umrechnung innerhalb eines CC-BASIC-Programmes erfolgen, ist zu berücksichtigen, daß mit CC-BASIC nur ganze Zahlen verarbeitet werden können. Es muß also zuerst mit dem Wert 135 multipliziert und dann wieder durch 100 geteilt werden. Man erhält dann den ganzzahligen Anteil der Multiplikation als Ergebnis