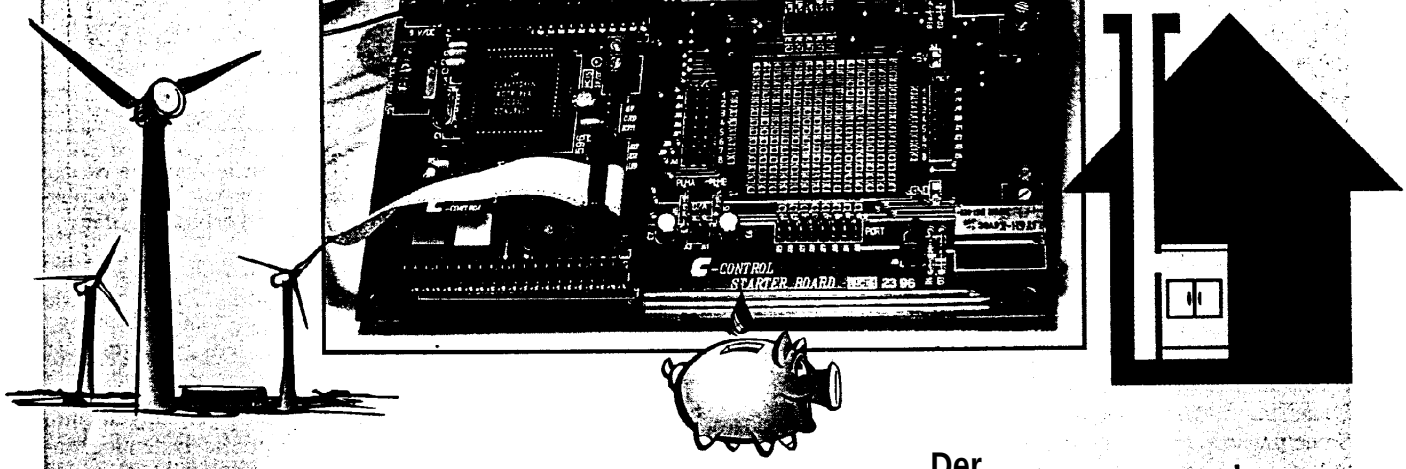
Stichwort:
C-Control

■ Eine eingeschränkte Programmiermöglichkeit mit Grafiksymbolen haben wir im *E•A•M* 8/96 beschrieben.

Den Einsatzfall bestimmt die Praxis und Sie das Programm:

BASIC als Basis



LW Die .universelle Steuereinheit für fast jeden Fall

- **Vom Anwender jederzeit neu programmierbar**
- **BASIC als Prog**rammiersprache ist **leicht**erlernbar
- **Zusatzmodule** erweitern **die Einsatzpalette**
- **Durch umfangreiche Hilfestellung schnell zum Ziel**

Steckbrief: **Auch für (Programmier-)Anfänger geeignet**

Funktion:	Selbst pr rammietbare Mikrocomputer-Steuereinheit mit zusä t zlichem Hardware-Experimentierboard
Eingang:	DCF77Aktivantenne
Schnittstelle:	RS232 für PC-Anschluß (u.a. zum Programmieren)
Mikrocontroller	*Motorola* **MC68HC05B6** mit 6 KB maskenprogrammiertem Betriebssystem
Anwenderspeicher:	**Microschip 24C65 EEPROM** mit 8 KB Umfang
Ports:	8 Analogeingänge 0.. .5 V, 8 bit Auflösung' **16** digitale Ein- bzw. Ausgänge (umprogrammierbar) 2 Quasi-Analogausgänge (PWM mit **1953 Hz)**
Anzeigen:	je eine LED (DCF77, Programm laden bzw. ausführen)
Abmessungen:	80 x 50 mm (Steuercomputer) 160 x 100 mm (Starterboard mit 5-V-Stabi, Relais u.a.)
Stromversorgung:	5 V±10% (µC); ca. 9 V (Starterboard)
Stromaufnahme:	ca. 30 **mA** (µC); **<10 mA ohne** LEDS **und** RS232
Preis:	ca. 99,95 DM (µC inkl. Schnittstellenkabel **und** Software
Starterboard:	ca. 69,95 DM (beides zusammen im Set; ca. **149,90)**

Der Umgang mit Mikroprozessoren im allgemeinen und die Programmierung ganz speziell sind für viele hartgesottene Elektroniker ein Greuel. Das liegt nicht zuletzt daran, daßviele Programmierer aus ihrer Tätigkeit eine obskure Wissenschaft machen, und daß eigene Versuche mangels richtiger Unterstützung kläglich gescheitert sind. Das ist hier grundsätzlich anders, denn mit diesem Set kann sich auch der Software-Neuling rasch einarbeiten.' Das Geheimnis liegt in der Programmiersprache BASIC, die von iedermann/frau schnell und problemlos erlernbar ist.

Stichwort:
BASIC

■ Diese speziell für Anfänger geschaffene Programmiersprache haben wir kurz im *E•A•M* 2/97 vorgestellt.

C-Control-BASIC-Steuercomputer (1)

Maßgeschneiderte Anwendungen

Der **C-Control-BASIC-Steuercomputer** ist ein eigenständiger Baustein für den universellen Einsatz in der Steuer-, Meß- und Regeltechnik; die Platine ist nur **80 x 50** mm groß **(Bild 1).** Welche Aufgabe dieser Baustein übernimmt, hängt von der jeweiligen Programmierung ab, die der Anwender durchführt; diverse Anwendungsbeispiele gehören zum Lieferumfang. Durch Umladen des Anwenderprogramms läßt sich die Einheit an nahezu jede beliebige Aufgabe anpassen.

Der Anwender erstellt seine Programme in der populären und leicht erlernbaren Programmiersprache **BASIC,** was sich auf einem separaten PC abspielt. Nach dem Austesten wird das Programm automatisch in eine Form übersetzt, die der CC-BASIC-µC *(Control Unit)* „verstehen" und ausführen kann. Dieses **Maschinenprogramm** wird dann vom PC in die Control Unit geladen, und kann dort per Knopfdruck ausgeführt werden.

Denkbare Anwendungsfälle für diesen Steuerungscomputer sind z.B. eine intelligente Alarmanlage, die speziell an die betreffende Umgebung angepaßt ist, die Steuerzentrale einer Heizungsanlage oder ein überwachungs- und Datenerfassungssystem; die Einsatzmöglichkeiten sind nahezu unbegrenzt.

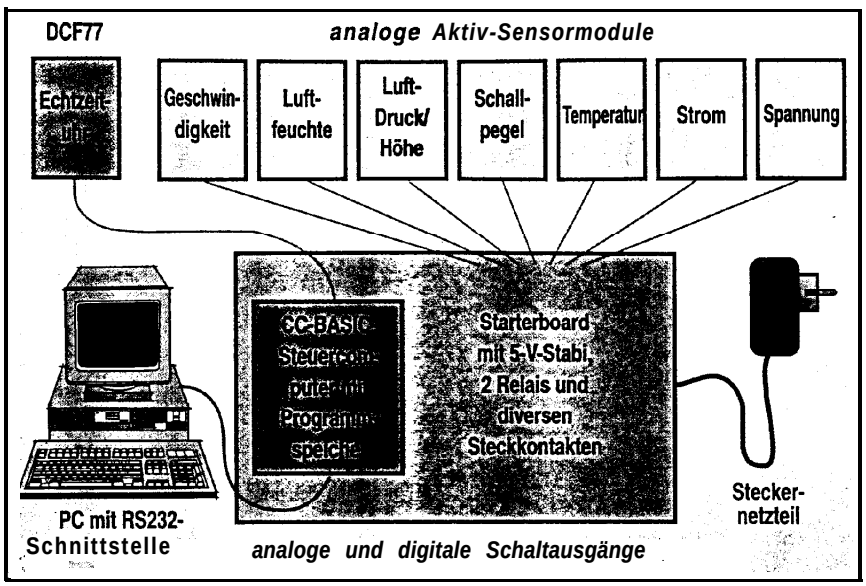


Bild 2: Die eigentliche Zentrale ist der Steuercomputer, der mit den peripheren Baugruppen in Verbindung steht.

Das Herzstück der Control Unit ist ein **Ein-Chip-Mikrocontroller** des Typs **MC68HC05B6** von *Motorola.* Dieser Controller führt das vom Anwender erstellte (oder mitgelieferte) Programm aus, sobald man den Start-Knopf betätigt. Das Programm selbst befindet sich in einem nichtflüchtigen Speicher **(EEPROM)** mit 8 KBytes Umfang (vgl. Ü36 im *E•A•M* 4/94).

Um dieses Programm in ausführbare Instruktionen umzusetzen, benötigt der Mikrocontroller ein eigenes **Betriebssystem;** dieses ist in einem Chip-internen Speicher mit 6 KBytes Umfang untergebracht.

Für den Datenaustausch mit dem PC besitzt die Platine eine **serielle RS232-Schnittstelle** (vgl. Heft 8/94, S. 64 und 65). Der Schnittstellenbaustein **MAX232** setzt die 5-V-Logikpegel in Plus/Minus-Spannungen um, wie sie für die serielle Datenübertragung benötigt werden (vgl. 'Interessantes IC' 5/94).

An einer dreipoligen Stiftleiste kann man eine Aktivantenne zum Empfang des **Zeitzeichensenders DCF77** anschließen; durch Einbeziehung der sich selbst korrigierenden Uhrzeit- und Datum-Information 'kann man sämtliche Schaltvorgänge sekundengenau programmieren, auf Wunsch auch auf Wochen, Monate oder gar Jahre im **voraus.** Die Dekodierung des empfangenen Zeitzeichens ist eine der Aufgaben des Betriebssystems, das empfangslose Zeiten mit einer internen Digitaluhr überbrückt.

Drei **Leuchtdioden** signalisieren den Systemzustand: Die grüne informiert über den DCF77-Funkuhrenempfang; die gelbe leuchtet während der Programmausführung (nach dem Druck auf den gelben START-Taster), und die rote ist u.a. während der Programmübertragung vom PC aktiv.

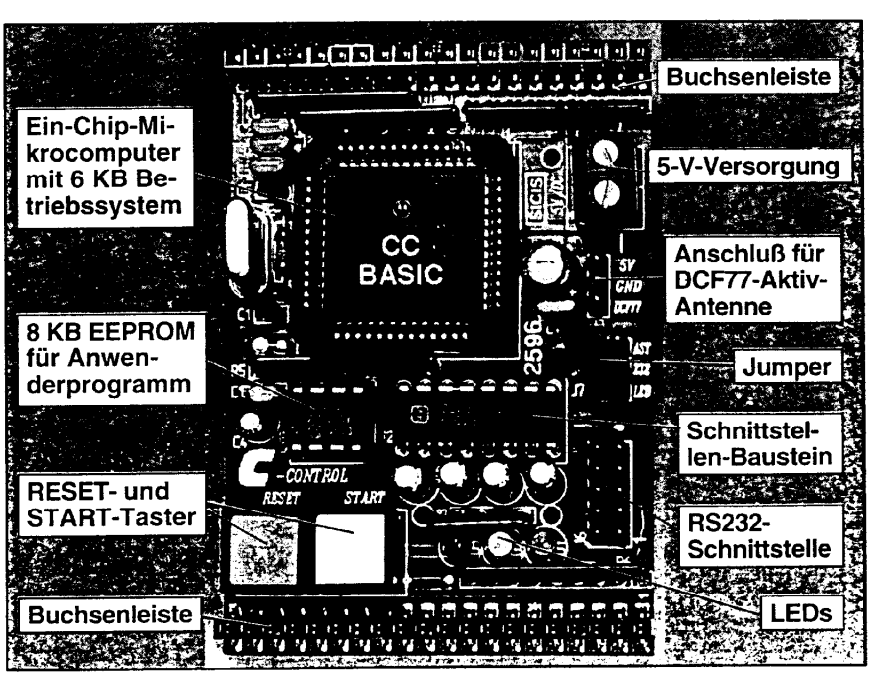
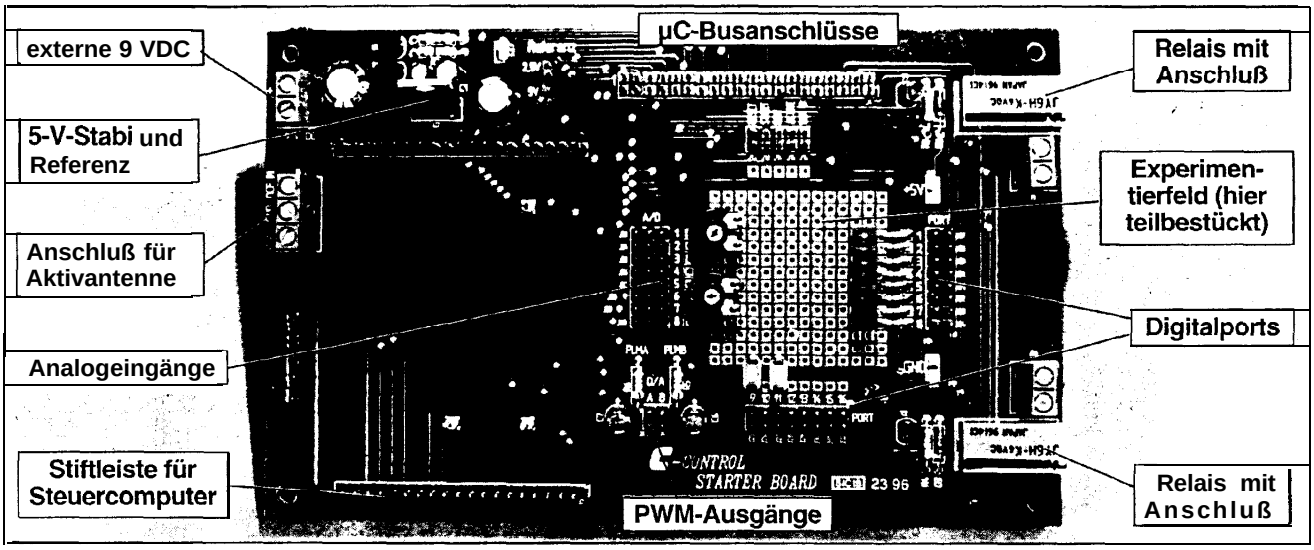


Bild 1: Der Steuercomputer bildet eine eigenständige, funktionsfähige Einheit; über die beiden Buchsenleisten kommuniziert er mit externen Elementen.

Stichwort: COM1 und COM2

■ Die COM1-Buchse ist meist mit der Maus belegt; bei 25poliger COM2-Buchse wird ein Adapter benötigt.

C-Control-BASIC-Steuercomputer (1)



Auf der Platine der Control Unit sind außerdem noch drei **Jumper** zur Systemkonfiguration vorgesehen. Zwei dieser Brücken erlauben das Abtrennen der LEDs bzw. des Schnittstellenbausteins von der Versorgungsspannung, um die Stromaufnahme von normalerweise 30 mA auf unter 10 mA zu senken. Wenn die dritte Brücke gesteckt ist, wird unmittelbar nach dem Anlegen der Speisespannung ein automatischer Programmstart ausgelöst.

An den beiden 20poligen **Buchsenleisten** liegen sämtliche Systemsignale an; außerdem sind hier alle Ein- und Ausgangsleitungen herausgeführt. Das System verfügt über **8 analoge Eingänge, 2 quasi-analoge Ausgänge** (pulsweitenmoduliert) und **16 frei als Ein- oder Ausgang** definierbare Digitalports. Über diese E/A-Leitungen findet der Datenaustausch mit der Umgebung statt; so können hierüber beispielsweise analoge Meßwerte eingelesen oder digitale Schaltsignale ausgegeben werden **(Bild 2).**

Der Steuercomputer benötigt zur Versorgung eine **stabilisierte Gleichspannung** von 5 V±10%. Die Einheit allein nimmt nur 30 mA auf (s.o.), aber bei Beschaltung mit externen Komponenten muß natürlich die Belastung durch die peripheren Komponenten mit berücksichtigt werden.

Bild 4: Die Nabelschnüre zum PC; zur Anpassung zwischen 9- und 25poliger COM-Buchse dient u.U. ein Adapter.

Passend zum Steuercomputer wird ein **Starterboard** im Europaformat angeboten **(160 x** 100 mm; **Bild** 3). Hier sind zwei 20polige Stiftleisten zur Aufnahme der Control-Unit vorgesehen (vgl. Bild 5). Außerdem sind sämtliche Systemsignale einschließlich der Ein/Ausgangsleitungen an Steckstiften zugänglich, so daß man für eigene Versuche und Entwicklungsarbeiten eine ideale Basis hat.

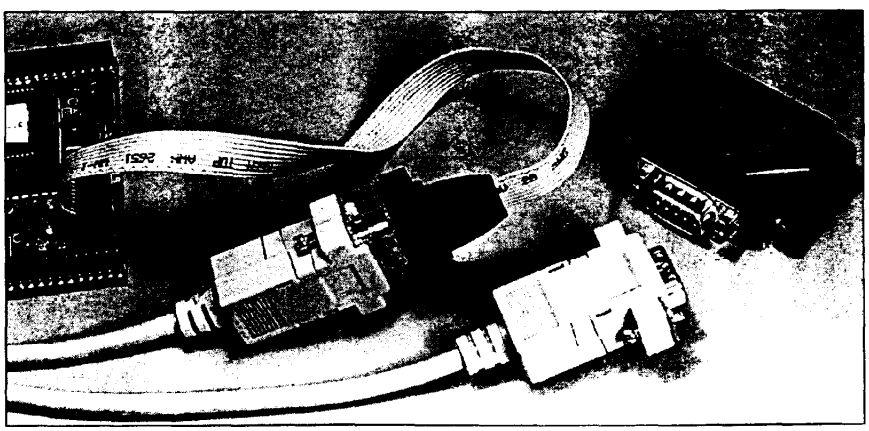
Auf einem freien Feld von Lötpunkten hat man außerdem die Möglichkeit, kleine Experimentierschaltungen aufzubauen. Diese Karte kann von einem unstabilisierten **9-V-Steckernetzteil** gespeist werden; ein 5-V-Stabi erzeugt die Versorgung für die Control-Unit. Zwei RC-Glieder glätten die von den PWM-Ausgängen gelieferten Rechtecksignale, so daß hier zwei separate Gleichspannungen für Steuerungsaufgaben verfügbar sind.

Bild 3: Auf dem Starterboard sind sämtliche Signale des Steuercomputers leicht zugänglich sortiert.

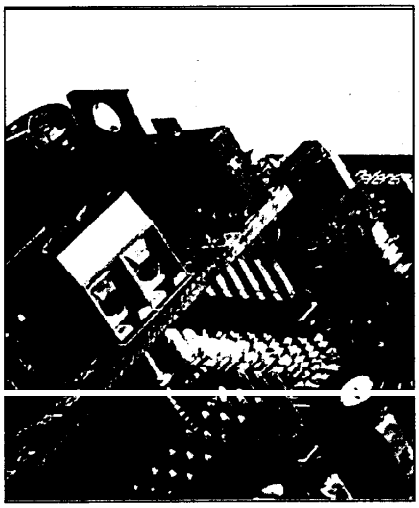
Außerdem besitzt das Starterboard **zwei Relais** mit vorgeschalteten Treibertransistoren, um größere Lasten potentialfrei schalten zu können. Über Steckbrücken lassen sich diese Schaltausgänge mit einem Ausgangsport verbinden.

Zum Anschluß der DCF77-Aktivantenne ist eine dreipolige **Schraubklemme** vorhanden. Auf dem Board kann außerdem die Referenzspannung für die Analogeingänge umgeschaltet werden (zwischen 2,5 V und 5,0 V).

Die Verbindung zwischen Steuercomputer und PC wird über ein neunpoliges Kabel plus Steckadapter hergestellt **(Bild** 4). Bei Bedarf kann noch ein Zwischenstecker von 9polig- auf 25polig-D-Sub erforderlich sein.



C-Control-BASIC-Steuercomputer (1)



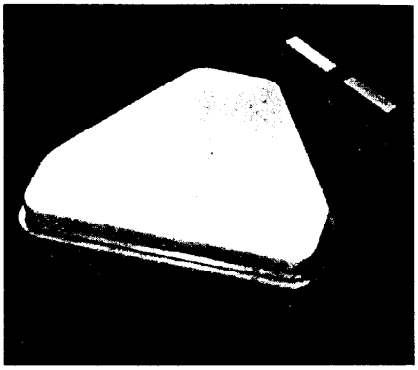


Bild 5 (links): Der Steuercomputer wird auf das Starterboard aufgesteckt.

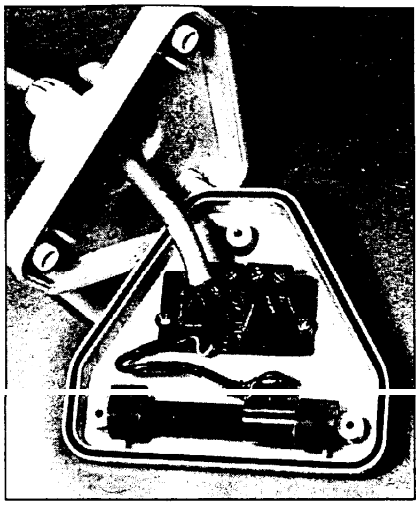


Bild 6 (oben u. rechts): Die Aktivantenne kann auch draußen montiert werden.

Durch drei Kodierungsstifte kann beim Zusammenstecken von Steuercomputer und Starterboard kein um 180" verdrehtes Falscheinsetzen erfolgen. Die Stiftleisten des Boards brauchen einfach nur in die entsprechenden Bohrungen der Control-Unit gesteckt zu werden **(Bild** 5). Es versteht sich, daß dabei die Versorgungsspannung ausgeschaltet ist.

Bei der Aktivantenne handelt es sich um einen DCF77-Empfangsbaustein, der ausgangsseitig das digitalisierte Zeittelegramm im 5-V-Pegel liefert **(Bild** 6). Durch den Einbau in ein wetterfestes Gehäuse (IP 54) läßt sich diese Baugruppe auch im Freien montieren, wenn der Empfang aufgrund von Stahlbeton oder störverseuchter Umgebung nicht möglich sein sollte. Das Antennengehäuse läßt sich um 360" drehen und kann so optimal ausgerichtet werden; das Anschlußkabel wird von einer Zugentlastung abgefangen.

Ein/Ausgabe

Alle diejenigen, die im Umgang mit elektronischen Bauteilen und Schaltungen vertraut sind (mit der *Hardware*), haben beim Zugang zur Mikroprozessortechnik einen erheblichen Vorteil: Es bereitet ihnen absolut keine Verständnisprobleme, mit einem auf HIGH liegenden Logik-Ausgang eine LED oder einen Transistor einzuschalten; das schließt den Einbau etwaiger Strombegrenzungswiderstände oder Freilaufdioden natürlich mit ein.

Wer von der Programmierung herkommt (von der *Software*-Seite), hat erfahrungsgemäß Probleme damit, daß ein bestimmtes Bit in einem Befehl oder der Befehl selbst sichtbare Auswirkungen in der Schaltung hervorruft. Dennoch sind Schwellenängste von beiden Seiten absolut unbegründet, zumal es wie im vorliegenden Fall so fundierte Unterstützung gibt.

Anfangs mag es noch problematisch erscheinen, wenn dieselbe Leitung wahlweise Ein- oder Ausgang sein kann; bei herkömmlicher Logik liegt die Übertragungsrichtung normalerweise fest. Betrachtet man die Sache näher, dann erweist sich die Programmierbarkeit als großer Vorteil: Man kann per Anweisung bestimmen, welche der E/A-Leitungen Eingang und welche Ausgang sind.

Das muß nicht einmal für immer gelten: Sogar während der Programmausführung läßt sich die Übertragungsrichtung eines Pottbits ändern, um einen Pin doppelt auszunutzen. So hat z.B der µC aus der UHF-Schaltstufe im *E•A•M* 5/96 einige E/A-Leitungen doppelt genutzt, um zwischen der Zustandsanzeige (Ausgänge für LEDs) und der Tastatur (Bediener-Eingaben) hin und herzuschalten; durch die schnelle Umschaltfolge merken weder Auge noch Hand des Anwenders etwas davon.

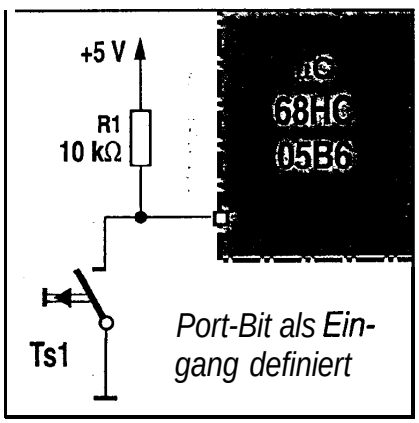


Bild 7: Einen digitalen Eingang kann man ganz einfach mit einem Taster beschalten; der Pull-up ist auf der Platine.

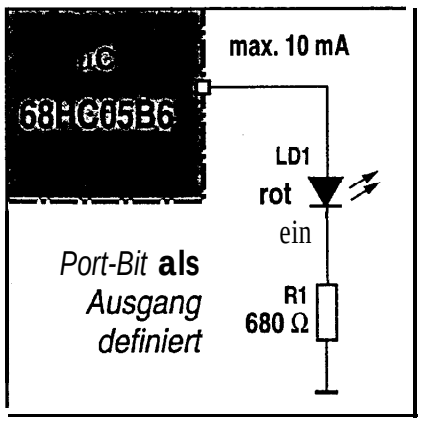


Bild 8: Der Zustand eines digitalen Ausgangs läßt sich z.B: mit einer Leuchtdiode plus Vorwiderstand anzeigen.

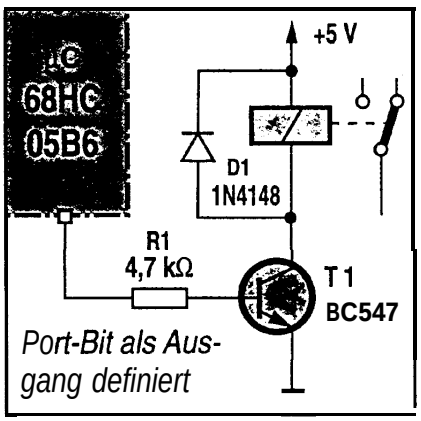


Bild 9: Zum Schalten größerer Lasten verwendet man ein Relais mit Treibertransistor; das Relais ist ein 6-V-Typ.

Stichwort: µP und µC

■ Diese Kürzel für 'Mikroprozessor' bzw. 'Mikrocomputer' erläutern Ihnen die Blickpunkte im *E•A•M* 3/93.

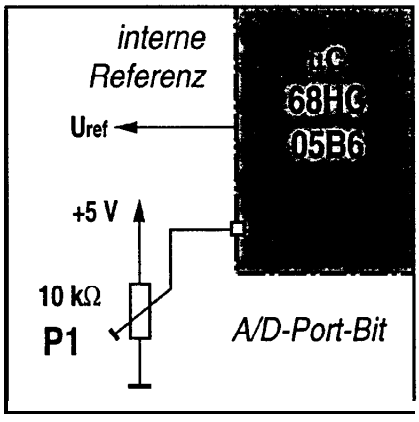


Bild 10: Zu Demonstrationszwecken können Sie an einen Analogeingang einfach ein Trimmpoti anschließen.

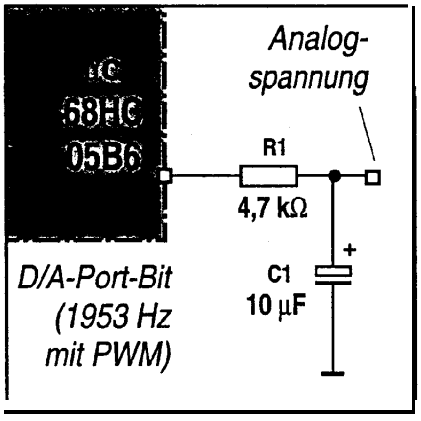


Bild 11: Zur Analogausgabe stehen zwei pulsweitenmodulierte Ausgänge mit 1,9 kHz und 5-V-Amplitude zur Verfugung.

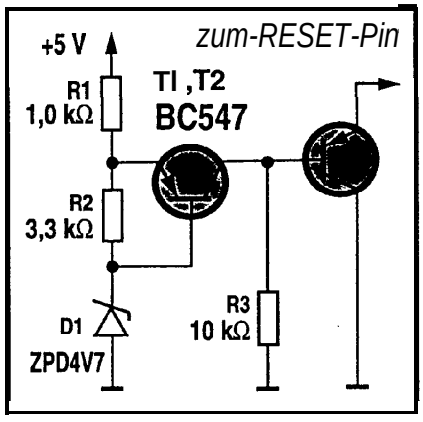


Bild 12: Diskret aufgebauter Spannungswächter zum automatischen Neustart nach einem Spannungsausfall.

Die Festlegung eines E/A-Kanals erfolgt im Programm mit der **DEFINE**-Anweisung (vgl. folgende Seiten und Mini-Poster). Im einfachsten Fall kann man ein **digitales Eingangssignal** manuell über einen Taster ändem und den LOW-Zustand „abfragen" **(Bild 7).**

Die simpelste denkbare Ausgabe ist die Ansteuerung einer Leuchtdiode **(Bild** 8). Hierbei ist allerdings zu beachten, daß jeder Ausgang maximal mit 10 mA belastet werden darf. **Kurzschluß bzw. Überlast können den Controller beschädigen oder gar zerstören!** Deshalb muß ein Widerstand zur Strombegrenzung eingefugt werden.

Wenn die Treiberleistung nicht ausreicht, muß man sie mit einem Transistor verstärken. Beim Anschluß eines **Relais** ist auf die Obligate Freilaufdiode zu achten, die die Induktionsspitzen „vernichtet"; andernfalls stirbt der Transistor **(Bild 9).**

Beim Betrieb eines Eingabekanals als **Analogeingang** erfolgt im Controller die Analog/Digital-Umsetzung, was mit einer Auflösung von 8 bit geschieht **(Bild 10).** Als erforderliche Referenzspannung kann die 5-V-Versorgung benutzt werden. Während der Umsetzung wird die Signalquelle mit ca. 10 µA belastet.

Zur Erzeugung **analoger Ausgangsspannungen** besitzt der Controller zwei PWM-Ausgänge, die mit jeweils einem externen RC-Glied abgeschlossen werden müssen **(Bild 11).** Je nach Tastverhältnis des Rechtecksignals entsteht eine leicht wellige Gleichspannung zwischen 0.. .5 V, die man bei Bedarf über einen OpAmp auskoppelt.

Bei laufender Programmausführung darf die Versorgungsspannung nicht ausfallen, weil dadurch der EEPROM-Inhalt durcheinandergeraten könnte. Um das zu verhindern, kann man die im **Bild 12** gezeigte **RESET-Schaltung** für Unterspannung einsetzen.

C-Control-BASIC-Steuercomputer – die Zielsetzung

Die Mikroprozessortechnik ist aus dem heutigen Leben nicht mehr wegzudenken. Das liegt nicht etwa an einer Modeerscheinung, sondern an den damit 'verbundenen;': gravierenden Vorteilen. Auch im privaten -oder Hobby-Bereich lassen sich Mikro-Computer sehr vorteilhaft einsetzen, wie zahlreiche Bauanleitungen aus unserem Magazin belegen.

Leider ist die Programmierung eines µPs eine zeitaufwendige Angelegenheit, die nicht nur ein spezielles Wissen, sondern auch noch eine teilweise recht aufwendige Geräteausstattung erfordert (vgl. Beitrag im *E•A•M* 1/97). Mit dem CC-BASIC-Steuer-Computer können nun auch diejenigen ihren eigenen Mikrocomputer programmieren, die über keine speziellen Programmierkenntnisse verfugen.

Der Grund dafür liegt in der Programmiersprache, für die das populäre; leicht erlernbare **BASIC** gewählt wurde. Die dabei. 'verwendeten Begriffe sind anschaulich und ohne weiteres verständlich; die Einarbeitung in BASIC geschieht fast spielerisch, wobei auch der Fortgeschrittene noch genügend Möglichkeiten hat, sein Wissen auszubauen und zu vertiefen.": Anschauliche Programmbeispiele den mit der *Hardware* mitgeliefert.

Die Erstellung des BASIC-Anwenderprogramms erfolgt auf einem ganz normalen Heimcomputer. Das kann wahlweise unter DOS oder Windows nichst (NT bzw. Win95) erfolgen. Beim Einsatz von Windows hat man sogar die Möglichkeit, das Programm „trocken" zu testen, d.h. die komplette Ausführung auf dem PC zu simulieren.

Weil das Anwenderprogramm fast in „Umgangssprache" geschrieben: ist, müssen die Anweisungen umgesetzt werden, damit sie der Mikrocontroller des CC-BASIC-Steuercomputers verstehen kann. Diese Übersetzung des *Quelltextes in* die *Maschinensprache* übernimmt vollautomatisch ein mitgeliefertes Hilfsprogramm, der sogenannter ***Compiler*** (gespr. *Kompailer*).

Erst das Ergebnis dieser Umsetzung wird in den Programmspeicher, der Control-Unit übertragen, um dort ausgeführt zu werden. Mit dieser „unteren" Ebene aber hat der Anwender nichts mehr zu tun. Er „unterhält" sich mit dem Computer ausschließlich in einer dem Menschen näheren, höheren Programmiersprache, was auf dem Umweg über den PC und ein Übersetzerprogramm abläuft. ■

■ Wie die einzelnen Temperaturskalen zusammenhängen, erfahren Sie in den Blickpunkten von *E•A•M* 7/96.

C-Control-BASIC-Steuercomputer (1)

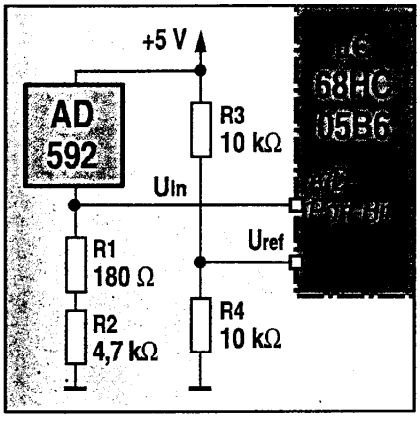


Bild 13: Eine einfache Schaltung zur Temperaturmessung; die Eingangsspannung U_{In} darf maximal U_{ref} erreichen.

AD592 liefert 1 µA/Kelvin
$U_{In} \le U_{ref}$ = 2,50 V
$U_{In} = I \cdot (R1+R2)$
$I \le U_{In}/(R1+R2) = 2{,}50\ V/4{,}88\ k\Omega$
I ≤ 512 µA: Temperatur ≤ 512 K
512 K = digitalisierte 2-5-0 [V]
Umrechnungsfaktor: 2 K/Digit
Temperatur: +20°C = 293 K
Strom @ 20°C = 293 µA
Spannung @ 20°C = 1-4-3 [V]
errechnete Temperatur ≈ 286 K

Bild 14: Diese Schritte liegen dem Rechengang zugrunde, der den digitalisierten Temperaturwert verarbeitet.

Einfachbeispiel

Die Grundelemente der Programmierung macht man sich am besten an einem realen Beispiel klar. Wir haben dafür eine einfache **Heizungsregelung** ausgewählt, die nach dem Prinzip des Zweipunktreglers arbeitet.

Dazu muß man die Temperatur zunächst mit geeigneten Mitteln mes*sen,* den Meßwert *aufbereiten* und daraus ein *Schaltsignal* für die Heizung ableiten (vgl. auch grafische Lösung im *E•A•M* 8/96, Seite 77).

Für einfache Temperaturmessungen eignet sich der Sensor **AD 592,** der pro Kelvin einen Konstantstrom von 1 µA liefert. Dieser Strom soll über einen Widerstand R1+R2 nach Masse abfließen und dabei eine temperaturproportionale Meßspannung U_{In} hervorrufen, die an einen A/D-Eingang des Steuercomputers führt **(Bild 13). Zum** Betrieb der A/D-Eingänge muß der µC eine Referenzspannung U_{ref} zugeführt bekommen, die die Bereichsgrenze angibt. Diese Referenz beträgt im Beispiel oben 2,50 V; wir gewinnen sie durch Teilung der Versorgungsspannung mit R3/R4.

Der Dimensionierung von Referenz und R1+R2 liegen Überlegungen zugrunde, die das Programm möglichst einfach gestalten; die richtigen Vorüberlegungen sind für eine sinnvolle Programmumsetzung oftmals von entscheidender Bedeutung. Den Rechengang haben **wir** im **Bild 14** noch einmal stichwortartig zusammengefaßt:

Für die Eingangsspannung gilt $U_{In} \le U_{ref}$; das gibt die Funktion des A/D-Umsetzers vor. Diese Spannung U_{In} ergibt sich als Produkt aus Konstantstrom 1 mal Widerstand (R1+R2). Bei den eingetragenen Bauteilwerten darf der Strom demnach maximal 512 µA groß werden (2,50 V/4,88 kΩ); dieser Wert ist beim AD 592 bei einer Temperatur von 5 12 K ≈ +239°C erreicht; bei 0 K = -273°C ist 1 = 0.

Die obere Bereichsgrenze wird vom Controller intern so digitalisiert, daß als Ergebnis eine Ziffernfolge von 2-5-O [V] herauskommt (ein Komma kennt der A/D-Umsetzer nicht). Wenn diese 2-5-O gerade 512 K entsprechen (= 2 K/Digit), dann braucht man den gemessenen und digitalisierten Wert nur mit 2 zu multiplizieren (512/250 ≈ 2), um auf die Temperatur in Kelvin zu kommen.

Betrachten wir die Verhältnisse bei +20°C (= 293 K), wo der Strom 1 = 293 µA beträgt und eine Meßspannung UIn = 1,43 V erzeugt. Ein digitalisierter Meßwert von l-4-3, multipliziert mit 2, ergibt gerade 286 [K]. Bezogen auf den tatsächlichen Wert von 293 K bedeutet das einen Fehler von nur 2,3%.

Wir könnten die Heizungsregelung ohne weiteres auf Kelvin-Basis realisieren; das hätte nur den Nachteil, daß man bei der Untersuchung von Zwischenergebnissen jedesmal im Kopf umrechnen muß (wenn man den Meßwert z.B. anzeigen oder die Schwellen ändern will). Deshalb ziehen wir vom Kelvin-Wert noch 273 ab, um die **Celsius-Temperatur** zu erhalten.

Diese Temperatur brauchen wir nun nur bloß mit einer unteren und oberen Schwelle zu vergleichen (in Celsius!) und aus dem Ergebnis das Schaltsignal für die Heizung abzuleiten.

Programmtechnisch sieht das folgendermaßen aus **(Bild 15):** Zunächst müssen die Randbedingungen festgelegt werden; das passiert mit den **DE-**FINE-Anweisungen, die dem Programm u.a. mitteilen, von wo es den Meßwert einlesen und wo es das Schaltsignal ausgeben soll: A/D-Port Nr. 1 bekommt den *symbolischen Namen* **SENSOR,** und der Digitalausgang wird **HEIZUNG** getauft; das Ergebnis der Datenaufbereitung landet in einer RAM-Doppelzelle *(=Word)* namens **TEMPERATUR.**

Als Einschaltschwelle **EIN** werden 20[°C] und für **AUS** als Ausschaltschwelle 30[°C] festgelegt.

```
define sensor ad[1]        Sensoreingang am A/D-Port Nr. 1
define heizung port[1]     Schaltausgang am Digitalport 1
define temperatur word     Speicherfür Zwischenergebnis

define ein 20              Einschaltschwelle = 20°C
define aus 30              Ausschaltschwelle = 30°C

#regelung                  Markierung des Programmanfangs
 temperatur = sensor * 2 - 273
 if temperatur < ein then heizung = 1
 if temperatur > aus then heizung = 0
goto regelung              Endlosschleife: Sprung zum Programmanfang
```

Bild 15: Das BASIC-Programm für die Einfach-Heizungsregelung; für den Controller muß es noch übersetzt werden.

Stichwort: Umsetzung

■ Zum Thema Digital/Analog- und Analog/Digital-Umsetzung beginnt im *E•A•M* 2/97 eine Grundlagenreihe.

Erst jetzt beginnt das eigentliche Programm, das mit einem sogenannten La*bel* markiert wird: **#REGELUNG** kennzeichnet nicht nur den Programmanfang, sondern bildet auch ein Sprungziel, zu dem verzweigt werden kann. Die bei früheren BASIC-Dialekten üblichen Zeilennummern werden beim CC-BASIC nicht benötigt; man kann sie aber dennoch verwenden, z.B. als Label-Ersatz.

In der ersten Programmzeile wird die Celsius-Temperatur berechnet; dazu wird, wie eben beschrieben, der vom Sensor gelieferte Meßwert mit 2 malgenommen, und von diesem Produkt wird 273 abgezogen; das Ergebnis wird in der Speicherzelle **TEMPERATUR** abgelegt.

In der zweiten Zeile wird geprüft, ob die Temperatur *kleiner* ist als die Einschaltschwelle **EIN.** Falls ja **(IF... THEN),** soll das Port-Bit **HEIZUNG** auf HIGH gehen; andernfalls bleibt alles beim alten, und es geht weiter mit der nächsten Programmzeile. Dort wird geprüft, ob die Temperatur größer ist als die Ausschaltschwelle; falls dies zutrifft, soll **HEIZUNG** LOW werden; andernfalls passiert nichts als das Fortschreiten zur letzten Programmzeile.

Dort steht eine Sprungsanweisung zurück zum Programmanfang **#REGELUNG,** womit eine **Endlosschleife** entsteht; nach dem Starten wird das Beispielprogramm also immer wieder durchlaufen, und **zwar sequentiell eine Zeile nach der anderen,** sofern keine Sprunganweisung auftritt. Sichtbare Änderungen treten im Beispiel nur dann auf, wenn eine der Abfragebedingungen erfüllt ist und dann ein angeschlossenes Relais umschaltet.

Folgende Dinge fallen hierbei auf:

- Wer mitgerechnet hat, wird bei 20°C auf einen Rechenwert von 286 – 273 = 13 gekommen sein; statt der tatsächlich herrschenden 20°C geht der Computer also von 13°C aus, was scheinbar ein Fehler von über 30% ist. Tatsächlich aber berechnet sich der **Fehler** aus den Kelvin-Temperaturen 286/293, womit man auf die genannten 2,3% kommt; für den geringen Aufwand ist das nicht schlecht.

Bild 17: Innenschaltung des aktiven Temperatursensors, der wesentlich aufwendiger ist als das Beispiel von Bild 13.

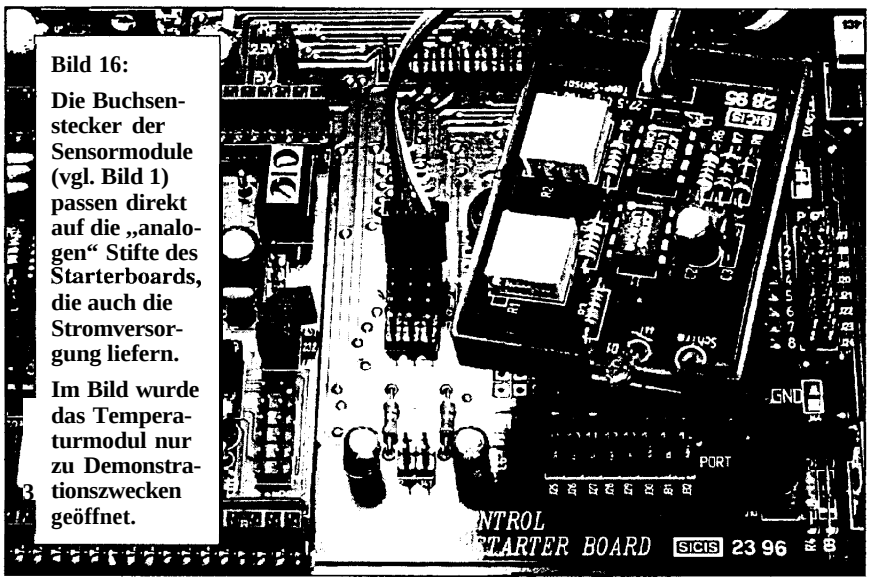


Bild 16:

Die Buchsenstecker der Sensormodule (vgl. Bild 1) passen direkt auf die „analogen" Stifte des Starterboards, die auch die Stromversorgung liefern.

Im Bild wurde das Temperaturmodul nur zu Demonstrationszwecken geöffnet.

- Um eine Schaltschwelle zu verändem, braucht man nur die entsprechende Zuweisung zu modifizieren, und zwar nur einmal, auch wenn der betreffende Wert mehrmals im Programm auftaucht; das ist einer der unschätzbaren Vorteile von **symbolischen Bezeichnungen.** Ein anderer ist die leichte Nachvollziehbarkeit von Programmen; schon der Name stellt einen sinnfälligen Zusammenbang zur Bedeutung her *(Eselsbrücke).*

- Die angeborene Scheu vor den englischen Anweisungen ist offensichtlich unbegründet; das bißchen Englisch wie IF...THEN oder GOTO u.ä. reimt sich auch der noch zusammen, der in der Schule notorisch geschwänzt hat.. .

- Das **Quellprogramm** in Bild 15 ist durchweg in Kleinbuchstaben geschrieben. Das ist ohne weiteres zulässig, während an anderer Stelle peinlich genau auf die vorgeschriebene Form (die Syntax) zu achten ist; so muß z.B. die Portnummer zwingend in eckigen Klammem stehen, weil es andernfalls beim Übersetzen **(Kompilieren) zu** einer Fehlermeldung kommt.

Wem die mangelnde Genauigkeit des Einfach-Beispiels nicht ausreicht, der muß mehr Aufwand treiben. So werden passend zum Starterboard fertige Sensormodule angeboten, die Ausgangsspannungen von 0. ..2,5 V liefern und direkt auf die Analogstifte aufgesteckt werden können **(Bilder 16** und 17).

Zur Verarbeitung der von diesem aktiven Modul gelieferten Meßwerte muß das Programm auf eine **Wertetabelle** zurückgreifen, mit der dann aber auch sehr genaue Berechnungen möglich sind. Diese und andere Tabellen sowie detaillierte Programmbeispiele gehören zum Lieferumfang der Control-Unit.