

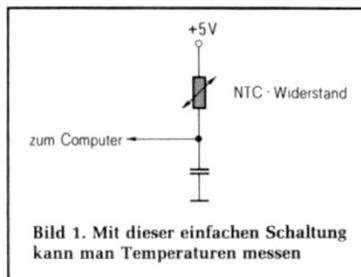
Dr. Heiner Schwarze

CBM und EMUF in der Ausbildung

Die technische Realisierung mikroprozessorgesteuerter Geräte unterscheidet sich sowohl im Äußeren als auch in der Programmierung von den in der Ausbildung üblichen Bildschirmcomputern. Am Beispiel der Temperaturmessung wird eine Anwendung des CBM-Computers stufenweise zu einer praxisnahen Problemlösung mit dem Einplatinencomputer EMUF weiterentwickelt. Bei diesem Vorgehen erhält man einen Einblick in die Funktion eines Mikroprozessors und in mögliche Anwendungen.

Der zunehmende Einsatz von Mikroprozessoren in Haushaltsgeräten, in der Fahrzeugherstellung [1] und am Arbeitsplatz führt zu der Zielsetzung, an einfachen Beispielen die Arbeitsweise der Mikroprozessoren in diesem Anwendungsbereich zu veranschaulichen. Es geht dabei weniger um eine vollständige Darstellung der Möglichkeiten, die sich durch diese Technik ergeben, sondern darum, einen ausgewählten Aspekt zu verdeutlichen. Der Beitrag wendet sich damit eher an interessierte Laien auf dem Gebiet der Mikroprozessortechnik als an erfahrene Programmierer. Als Ausgangspunkt für die folgenden Anwendungen soll ein CBM- oder Apple-Computer gewählt werden. In Schulen oder anderen Ausbildungsstätten stehen diese oder ähnliche Computer zur Verfügung, eine Vertrautheit mit der Programmierung in Basic ist weitgehend vorhanden.

Unter den genannten Voraussetzungen



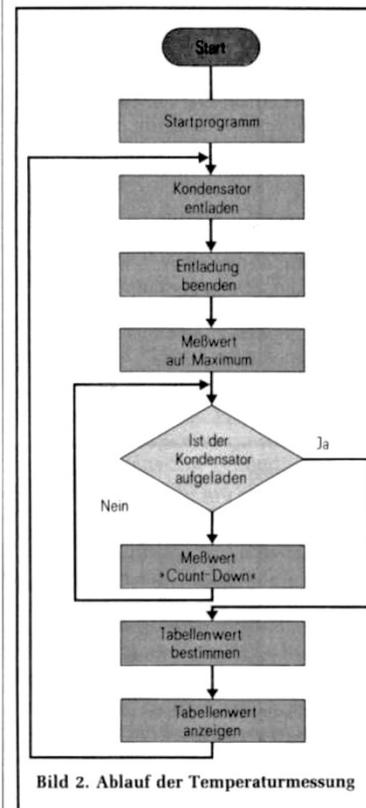
könnte man in folgender Weise vorgehen: Als erstes wird der Computer mit externen Bauelementen zusammengesetzt, einfache Beispiele für Daten- ein- und -ausgabe lassen sich erarbeiten. In einem zweiten Schritt werden Beschränkungen durch die Programmiersprache zumeist in Form von Zeitproblemen verdeutlicht. Damit erscheint eine Übertragung von Programmteilen in die Assemblersprache, also in eine mikroprozessorspezifische, schnelle Sprache sinnvoll. Anschließend wird der eingeschlagene Weg dadurch fortgesetzt, daß man das gesamte Programm in Assembler formuliert. Auf diese Weise kann auf die Sprache Basic verzichtet werden, ein Minimalsystem wie der „EMUF“ läßt sich zur Problemlösung einsetzen. Erst damit lassen sich dann charakteristische Aspekte von Mikroprozessor-Anwendungen wie Preiswürdigkeit und Flexibilität überzeugend demonstrieren.

Die Temperatur wird mit einfachen Mitteln gemessen

Die Messung der Temperatur und eine sich daraus ergebende Steuerung findet man vielfach in der Praxis, besonders aktuell zur Zeit in der Steuerung von Heizungsanlagen. Dieses Beispiel soll ausgearbeitet werden, stellvertretend für die Messung auch anderer Größen wie Helligkeit, Druck, Schall und Spannung. Abhängig vom jeweiligen Parameter ändert sich der Widerstand des Meßfühlers, bei der Temperaturmessung der ei-

nes NTC-Widerstands. Da ein Computer im allgemeinen nur auf Spannungspegel reagiert, besteht die Aufgabe der externen Bauelemente darin, die Widerstandsänderung des NTC-Widerstands in eine Spannungsänderung am Anschluß des Computers umzuwandeln. Dies geschieht sehr einfach mit Hilfe der Schaltung in Bild 1: Über den NTC-Widerstand wird ein Kondensator aufgeladen. Erreicht die Spannung am Kondensator einen Wert von beispielsweise 3,5 V, dann liest der Computer vom Anschluß statt „0“ zu Beginn der Aufladung den Wert „1“ ein. Ein Maß für die Temperatur ist die Zeit, in der sich der Kondensator zum Umschaltwert auflädt. Bei hoher Temperatur ist der Widerstandswert des NTC-Widerstands gering, der Strom zum Kondensator groß und die Ladezeit entsprechend gering. Bei niedriger Temperatur ist die Zeit zum Erreichen der Schaltspannung vergleichsweise groß.

Mit einem Mikroprozessor läßt sich relativ leicht aus dem Wert der Ladezeit der zugehörige Temperaturwert gewinnen: Zu jedem Zeitwert wird durch eine Kali-



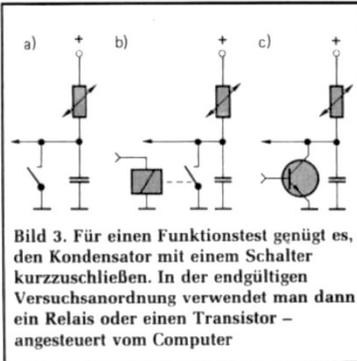


Bild 3. Für einen Funktionstest genügt es, den Kondensator mit einem Schalter kurzzuschließen. In der endgültigen Versuchsanordnung verwendet man dann ein Relais oder einen Transistor – angesteuert vom Computer

brierung der zugehörige Temperaturwert bestimmt und gespeichert. Anschließend wird nach jedem Meßvorgang der gespeicherte Temperaturwert ausgewählt und angezeigt. Für den Programmablauf ergibt sich damit das in Bild 2 dargestellte Flußdiagramm. Der Kondensator muß zu Beginn jeder Messung entladen werden. Dies könnte man, um den Ablauf der Messung zu erklären, erst einmal mit einem Schalter

von Hand durchführen. Im folgenden läßt sich dann ein vom Mikroprozessor gesteuertes Relais oder ein Schalttransistor verwenden (Bild 3).

Nach einer kurzen Wartezeit wird die Entladung beendet, indem der Kurzschluß des Kondensators aufgehoben wird. Dann beginnt der eigentliche Meßvorgang. Die Variable „Meßwert“ wird auf einen Wert von beispielsweise 250 gesetzt. Anschließend wird geprüft, ob der Kondensator aufgeladen ist, genauer: ob an dem Anschluß des Computers der Wert „1“ liegt. Zu Beginn der Messung wird der Kondensator kaum aufgeladen sein, der Meßwert wird um 1 auf 249 heruntergezählt und die Abfrage, ob der Anschluß auf „1“ liegt, wiederholt. Bei einem hohen Temperaturwert ist der Kondensator relativ schnell aufgeladen, der Meßwert wird nicht sehr häufig heruntergezählt und hat in dem Moment, in dem der Kondensator die Schaltspannung erreicht hat, einen großen Wert. Zu einem großen Meßwert gehört also auch ein großer Temperaturwert. Der genaue Zusammenhang ist in einer Tabelle gespeichert, wobei für eine ausreichende Meßgenauigkeit mehreren Meßwerten

derselbe Temperaturwert zugeordnet ist. Eine mögliche Zuordnung wäre:

Meßwerte	Temperaturwerte
200	
199	32 °C
198	
197	31 °C
196	

Als Teil des Programms wird also eine Tabelle gespeichert, wobei in dem gezeigten Beispiel der 197te Tabellenwert zum Meßwert 197 den Wert 31 °C für die Temperatur angibt. Mit der Anzeige des Tabellenwerts ist der Meßvorgang beendet.

An der Darstellung des Programmablaufs erkennt man, daß viele technische Daten der verwendeten Bauelemente nicht angesprochen werden. Es ist auch nicht das Ziel dieses einführenden Versuchs, die Aufladungszeit in Abhängigkeit von der Temperatur zu berechnen, was ja zu einer genaueren Analyse der verwendeten Bauelemente führen muß. Es soll vielmehr eine Meßreihe mit einem geeichten Thermometer durchgeführt werden, bei der für eine vorgegebene

```

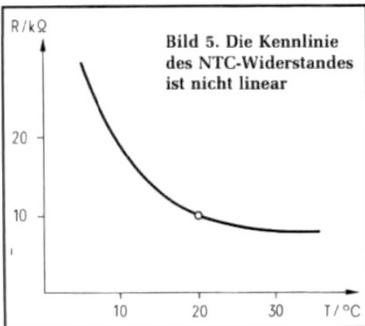
122 : PRINT"●RCBM MISST TEMPERATUR!"
124 : PRINT"□BITTE BAUEN SIE FOLGENDE
126 : PRINT"SCHALTUNG AUF:"
128 : PRINT
130 : PRINT"+ )
132 : PRINT"
134 : PRINT"
136 : PRINT"
138 : PRINT"
140 : PRINT"
142 : PRINT"BIT 0
144 : PRINT"
146 : PRINT"
148 : PRINT"BIT 1 /
150 : PRINT"
152 : PRINT"
154 : PRINT"
156 : PRINT"
158 : PRINT
160 : PRINT"WENN SIE FERTIG SIND,"
162 : PRINT"GEBEN SIE BITTE RF_EIN!"
164 : GET A#: IF A#<>"F"THEN 164
166 :
168 REM--AUS- EINGABE VORBEREITEN-----
170 :
172 : AUSGABE=59471
174 : REM-AUS- UND EINGABE UEBER
176 : EINGABE=59471
178 : REM-SPEICHER 59471 (USER-PORT).
180 : RICHTUNG=59459
182 : REM-SPEICHER 59459 DEFINIERT
184 : REM-DIE BITS FUER AUSGABE.
186 :
188 : POKE RICHTUNG,2+4+8+16+32+64+128
190 : REM ANSCHLUSS 0 AUF EINGABE
192 : REM ANSCHLUSS 1-7 AUSGABE
194 :
196 REM++MESSUNG+++++
198 :
200 REM-C UEBER BIT 1 ENTLADEN-----
202 :
204 : POKE AUSGABE,2
206 : REM-BIT 1 (2*1=2) AUF "1"
208 : FOR I=1 TO 1000: NEXT
210 : REM-WARTESCHLEIFE

212 :
214 REM-ENTLADUNG UEBER BIT 1 BEENDEN
216 :
218 : POKE AUSGABE,0
220 : REM-BIT 1 (2*1=2) AUF "0"
222 :
224 REM-MESSWERT BESTIMMEN UEBER BIT 0
226 :
228 : MESSWERT = 250
230 : REM-MAXIMUM DEFINIEREN
232 : IF PEEK(EINGABE) = 1 THEN 248
234 : REM-WENN BIT 1 AUF "1", DANN
236 : REM-IST C AUF 3,5V AUFGELADEN.
238 : REM-SPRUNG ZUR AUSWERTUNG!
240 : MESSWERT = MESSWERT -1
242 : GOTO 232
244 : REM-SPRUNG ZUR ABFRAGE BIT 0
246 :
248 REM-ANZEIGEWERT AUS DER TABELLE---
250 :
252 : FOR INDEX = 0 TO MESSWERT
254 : READ ANZEIGE
256 : NEXT INDEX
258 : REM-ZU JEDEM MESSWERT IST IN
260 : REM-DER TABELLE DER WERT FUER
262 : REM-DIE TEMPERATUR GESPEICHERT.
264 : RESTORE
266 : REM-ZEIGER AUF TABELLENSTART
268 :
270 REM-ANZEIGE AUF DEM BILDSCHIRM----
272 :
274 : PRINT
276 : PRINT"□DIE TEMPERATUR:";ANZEIGE;
278 : PRINT"□GRAD C"
280 :
282 : PRINT"●●";
284 : REM"DER CURSOR WIRD ZUM BEGINN
286 : REM"DES AUSDRUCKS GEFUEHRT UND
288 : REM"UEBERSCHREIBT INH BEI DER
290 : REM"NAECHSTEN MESSUNG!
292 :
294 GOTO 200
296 : REM-C UEBER BIT 1 ENTLADEN
298 : REM-NAECHSTE MESSUNG
300 :

302 REM-TABELLE DER ANZEIGEWERTE-----
304 :
306 : REM-DIE INDEXWERTE <70 UND > 196
308 : REM-SIND OHNE BEDEUTUNG.
310 : REM-DER MESSBEREICH IST DAMIT
312 : REM-AUF 0 GRAD < T < 34 GRAD
314 : REM-FESTOEGLEIT.
316 :
318 DATA 0,0,0,0,0,0,0,0,0,0,0
320 DATA 0,0,0,0,0,0,0,0,0,0,0
322 DATA 0,0,0,0,0,0,0,0,0,0,0
324 DATA 0,0,0,0,0,0,0,0,0,0,0
326 DATA 0,0,0,0,0,0,0,0,0,0,0
328 DATA 0,0,0,0,0,0,0,0,0,0,0
330 : REM 60-69
332 DATA 0,0,0,0,0,0,0,0,0,0,0
334 : REM 70-79
336 DATA 8,8,8,8,8,8,8,8,8,8,8
338 : REM 80-89
342 : REM 90-99
344 DATA 9,9,9,10,10,10,10,10,10,10
346 : REM 100-109
348 DATA 10,10,11,11,11,11,11,11,11,11
350 : REM 110-119
352 DATA 11,11,12,12,12,12,12,12,12,12
354 : REM 120-129
356 DATA 13,13,13,13,13,13,13,14,14,14
358 : REM 130-139
360 DATA 14,14,14,14,14,15,15,15,15,15
362 : REM 140-149
364 DATA 15,16,16,16,16,16,17,17,17,17
366 : REM 150-159
368 DATA 17,18,18,18,18,19,19,19,19,20
370 : REM 160-169
372 DATA 20,20,20,21,21,21,22,22,22,22
374 : REM 170-179
376 DATA 23,23,23,24,24,24,25,25,25,26
378 : REM 180-189
380 DATA 26,26,27,27,27,28,28,28,29,29
382 : REM 190-199
384 DATA 30,30,30,31,31,32,32,32,33,33
386 : REM 200-209
388 DATA 34,34,34,34,34,34,34,34,34,34
390 : REM 210-219
392 DATA 99,99,99,99,99,99,99,99,99,99

```

Bild 4. Dieses Basic-Programm entspricht dem Ablauf von Bild 2



ne Temperatur der Meßwert angezeigt und notiert wird. Mit den so gewonnenen Werten läßt sich die Tabelle der nach Meßwerten geordneten Temperaturwerte aufstellen und in das Programm übernehmen.

Es besteht im Rahmen einer Vertiefung die Möglichkeit, die gemessene Abhängigkeit zu untersuchen, die Kennlinie des NTC-Widerstands zu erklären und die Aufladung des Kondensators mathematisch zu beschreiben. Es ist jedoch freigestellt, wie weit man diese Betrachtungen führen will, die im folgenden nicht vorausgesetzt werden.

Geringe Basic-Vorkenntnisse genügen

Bei einigen Basic-Vorkenntnissen benötigt man zusätzlich Angaben über den „User-Port“, der Schnittstelle zwischen CBM-Computer und angeschlossenem Meßaufbau. Im Handbuch [2] findet man die folgenden Angaben:

Adresse des Ausgabe- und Eingabespeichers: 59471

Adresse des Datenrichtungsspeichers: 59459

Für die Temperaturmessung benötigt man die Kontakte Bit 0, Bit 1 und Masse des „User-Ports“, außerdem den +5-V-Anschluß des Recorder-Steckverbinders.

Zu Beginn der Signalausgabe muß im Programm die Richtung der Datenübertragung für jedes einzelne Bit festgelegt werden. Eine „1“ in einem Bit des Richtungsspeichers bewirkt eine Ausgabe durch das entsprechende Bit am „User-Port“. Zur Temperaturmessung wird Bit 0 zur Eingabe, Bit 1 zur Ausgabe verwendet. Der Befehl POKE 59459,2 schaltet durch die Wertigkeit der Bits (Tabelle 1) das Bit 1 auf Ausgabe. Im Programmausdruck (Bild 4) wird dieser Zusammenhang noch einmal verdeutlicht: Die Zeilen 122...166 enthalten die Schaltung mit den vorgesehenen Anschlüssen zum CBM-Computer. Die in

den PRINT-Anweisungen hervorgehobenen Zeichen dienen zur Steuerung des Cursors, wie auch in [3] ausführlich dargestellt. Der folgende Programmteil bereitet die Eingabe und Ausgabe über den „User-Port“ vor. Durch den Befehl in Zeile 188

```
POKE RICHTUNG,
2+4+8+16+32+64+128
```

werden die Bits 1...7 auf Ausgabe geschaltet. Dabei ist nicht klar ersichtlich, warum die Bits 2...7, die nicht weiter verwendet werden, ebenfalls umzuschalten sind. Der Grund ist eine vereinfachte Programmierung der Abfrage von Bit 0: Wenn ein Bit nämlich auf Ausgabe geschaltet ist, liefert es auf eine Abfrage mit PEEK (Eingabe), entsprechend PEEK (59471), den Wert „0“. Wenn also nur Bit 0 als Eingang geschaltet ist, kann das Abfrageergebnis nur „0“ oder „1“ (je nach Spannung an Bit 0) lauten.

Der Meßvorgang, beginnend in Zeile 196, folgt der Darstellung im Flußdiagramm. Zur Entladung des Kondensators wird Bit 1 auf „1“ gesetzt, der Transistor wird durchgeschaltet und schließt den Kondensator kurz. Nach einer Warteschleife wird die Entladung beendet, indem Bit 1 auf „0“ zurückgesetzt wird. Daraufhin wird in Zeile 224 eine Zählerschleife gestartet, die den Meßwert von dem Wert 250 herunterzählt. Die Abfrage in Zeile 232 prüft, ob der Kondensator aufgeladen ist und verzweigt, wenn die Bedingung erfüllt ist, zum Programmteil „Anzeigewert aus der Tabelle“. Der Meßwert gibt den Index an, unter dem der Temperaturwert in der Tabelle gespeichert ist. In der FOR-NEXT-Schleife in den Zeilen 252...256 werden die Tabellenwerte von 0 bis zum Indexwert, zum Beispiel 197, durchgegangen. Der 197te Wert der Tabelle, die Zahl 32, wird der Variablen „Anzeige“ zugeordnet und mit den nachfolgenden Programmteilen angezeigt. Anschließend wird die Messung wiederholt.

Diskussion der Schwachstellen führt zum Einplatinencomputer

Zum lauffähigen Basic-Programm gehört die Bestimmung der Tabellenwerte, wo-

Tabelle 1: Wertigkeit der einzelnen Bits

Bit	0	1	2	3	4	5	6	7
Wertigkeit	2 ⁰	2 ¹	2 ²	2 ³	2 ⁴	2 ⁵	2 ⁶	2 ⁷
	1	2	4	8	16	32	64	128
Beispiele:								
Dezimalwert 2 => Bit 1 = 1								
Dezimalwert 19 => Bit 0, Bit 1 und Bit 4 = 1								

zu einige Vorüberlegungen zu treffen sind. Wie Vorversuche zeigen, treten in einem Temperaturbereich von 10 °C...30 °C Meßwerte im Bereich von 100...200 auf. Weiterhin wird deutlich, daß bei hohen Temperaturen eine Temperaturänderung nur eine geringe Veränderung des Meßwerts zur Folge hat, da der Widerstandswert des NTC-Widerstands sich nur geringfügig verändert. Dadurch wird die Meßgenauigkeit bei hohen Temperaturen stark eingeschränkt, ein Fehler unter ± 0,5 °C ist über 40 °C kaum zu erreichen, wenn man einen Bereichsanfang von etwa 10 °C zugrunde legt.

Bei tiefen Temperaturen steigt der Widerstandswert des NTC-Widerstands entsprechend Bild 5 stark an. Die Meßgenauigkeit wird damit erhöht, allerdings verlängert sich die Meßzeit auf einige Sekunden. Aus den genannten Gründen scheint die Beschränkung auf einen Meßbereich von 8 °C bis 36 °C ein geeigneter Kompromiß zu sein. Zur Bestimmung der Tabellenwerte für den gewählten Meßbereich wird in das Programm die Zeile

```
251 PRINT „MESSWERT = “; MESS-
WERT : GOTO 200 : REM NEUE
MESSUNG
```

eingefügt. Bei vorgegebener Temperatur wird nach einer Messung der Meßwert angezeigt und mit dem Temperaturwert notiert. Die damit gewonnenen Tabellenwerte sind dann in die DATA-Anweisungen der Zeilen 318...392 einzutragen. Anschließend wird die Zeile 251 wieder gelöscht.

In einem nächsten Schritt kann man die Schaltung und das Programm nach Verbesserungen hinsichtlich der Meßgenauigkeit und der Meßzeit analysieren, was zu einer Untersuchung der Zählerschleife in den Zeilen 232...240 führt. Eine Forderung nach kürzeren Meßzeiten oder höherer Auflösung kann nur über eine schnellere Bearbeitungszeit der Zählerschleife erfüllt werden. In der Basic-Realisierung des Programms ist allerdings eine wesentliche Zeitreduzierung nicht zu erreichen. Außerdem erscheint ein Basic-Rechner für eine derartige Anwendung nur bedingt geeignet, da sich seine Möglichkeiten eher bei anders gelagerten, weniger zeitkritischen Problemen ausnutzen lassen. Eine technisch angemessenere und auch für die gewählte Anwendung besser geeignete Lösung läßt sich mit einem Minimalsystem wie dem EMUF [4, 5] realisieren. Es wird dabei auf die Programmiersprache Basic

verzichtet, was einerseits den Schaltungsaufwand erheblich reduziert und andererseits durch die mikroprozessor-spezifische Assemblersprache die Bearbeitungszeit der Befehle erheblich verkürzt.

Und jetzt Assembler- statt Basic-Programm

Eine weitgehende Übertragbarkeit der bereits entwickelten Programmteile wird dadurch ermöglicht, daß der EMUF zwei 8-Bit-Ports zur Verfügung stellt, also eine gegenüber dem CBM-Computer verdoppelte Zahl von Eingabe- oder Ausgabelösungen. Bei einer Verwendung als Temperaturmeßgerät kann der eine Anschluß, Port A, in einer zum „User-Port“ beim CBM entsprechenden Weise genutzt werden. Weder die Schaltung noch die Belegung der Bits müssen geändert werden. Der zweite Anschluß, Port B, wird als Ausgang zum Betrieb einer zweistelligen 7-Segment-Anzeige geschaltet, da der EMUF selbst über keine Anzeige verfügt (Bild 6).

Wie aus der Schaltung ersichtlich ist, werden insgesamt nur acht Ein-/Ausgabelösungen benötigt, so daß man auf einen Port verzichten könnte. Allerdings wird dann die Programmierung aufwendiger und die Trennung zwischen Meßaufbau und Anzeige schwieriger. Als einführendes Beispiel erscheint die dargestellte Schaltung besser geeignet; die nicht benutzten Leitungen können gegebenenfalls für Erweiterungen verwendet werden.

Das Programm für den EMUF wird in Assembler entwickelt. Für den CBM stehen mehrere Assembler zur Verfügung [6, 7]. Eine Einführung in die Programmierung findet man in einigen vor kurzer Zeit erschienenen Büchern [8, 9, 10]. Als Einführung erscheint es trotz fehlender Voraussetzungen möglich, die Funktionsweise des angegebenen Assembler-Programms zu verstehen. Die benutzten Befehle werden in Tabelle 2 erläutert.

Die Kommentare in den Programmzeilen verdeutlichen die Wirkung der Befehle. Das Ziel der Assembler-Programmierung ist es, deutlich zu machen, daß bei der vorgegebenen Anwendung als Temperaturmeßgerät oder auch ähnlichen Anwendungen die Programme in Basic und Assembler eine sehr ähnliche Struktur aufweisen. Die Programmierung in Assembler führt zu einer Einführung in die Arbeitsweise eines Mikroprozessors, in die Programm- und Datenspeicherung und schließlich in die Arbeitsweise eines mikroprozessorgesteuerten Meßgeräts, welches einer gebräuchlichen tech-

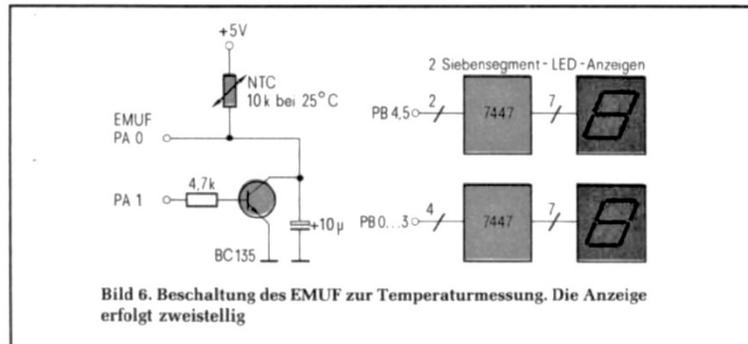


Bild 6. Beschaltung des EMUF zur Temperaturmessung. Die Anzeige erfolgt zweistellig

nischen Realisierung weitgehend entspricht.

Der Vergleich der Programme soll mit der „Aus-/Eingabe“-Vorbereitung beginnen (Bild 7). Wieder werden im Assembler-Programm die Speicheradressen für die „Ports“ festgelegt. Anschließend werden in den Richtungsspeichern die Ausgabebits auf „1“ gesetzt. Dabei wird bei Adresse 9C04 deutlich, daß durch LDA #% 11111110 und STA PAD Bit 0 auf Eingabe und die restlichen Bits auf Ausgabe geschaltet sind. Durch die Möglichkeiten der Zahldarstellungen (% für binäre und \$ für hexadezimale Darstellung) hat man neben der gebräuchlichen dezimalen Schreibweise die Möglich-

keit, die jeweils verständlichste Darstellung auszuwählen. Im Programmteil „C ENTLADEN ÜBER BIT 1“ wird dieser Vorteil wieder ausgenutzt. Die folgende Warteschleife wird über ein Unterprogramm verwirklicht, welches den Prozessor mindestens 1 ms lang beschäftigt. Hier wird der Wert 4 bei 9C13 geladen, das entspricht 4 ms Wartezeit. Anschließend wird die Entladung beendet. Auch hier wird deutlich, daß einem Basic-Befehl wie POKE AUSGABE, 0 jeweils zwei Assembler-Befehle LDA #0 und STA PA entsprechen.

Die Bestimmung des Meßwerts verläuft in bereits vertrauter Weise durch Verschiebeoperationen zwischen dem Ar-

Tabelle 2: Liste der verwendeten Assembler-Befehle

Transportbefehle von den Speichern im Prozessor A, X, Y (Akkumulator, X-Speicher, Y-Speicher) zum Programmspeicher oder Ein-/Ausgabe-Speicher		
LDA	Load Accumulator	Der Inhalt des Akkumulators wird geladen mit einer Zahl, dem Speicherinhalt von PA oder mit dem X-ten Tabellenelement, wobei X der Wert des X-Speichers ist
#%11111110	PA	
STA	Store Accumulator	Der Inhalt von A wird gespeichert, X-Speicher geladen,
LDX	Load X	Y-Speicher geladen entsprechend LDA
LDY	Load Y	
Rechenbefehle zur Subtraktion von „1“ in den Speichern A, X, Y		
SEC	Set Carry	Subtraktion von „1“ im Akkumulator, wobei das Übertragsbit gesetzt wird
LDA MESSWERT		
SBC#1	Subtract with Carry	Subtraktion von 1 im X- oder im Y-Speicher
DEX	Decrement X	
DEY	Decrement Y	
Sprungbefehle		
JMP	Jump	Sprung entsprechend „GOTO“
JSR	Jump to Subroutine	Sprung zum Unterprogramm und
RTS	Return from Subroutine	Rücksprung entsprechend „GOSUB“ und „RETURN“
BNE	Branch if not Equal	bedingter Sprung, wenn die letzte Operation einen Wert ungleich „0“ liefert
NOP	No Operation	keine Rechenoperation

schneller; die Meßgenauigkeit und die Auflösung lassen sich entsprechend steigern. Die Zahl der möglichen Anwendungen vervielfacht sich dadurch. Es können mehrere Meßstellen abgefragt werden, es können für thermostatisierte Anordnungen bei vorgewählten Temperaturreichungen Heizung oder Kühlung zugeschaltet werden. Alles zu einem angemessenen Preis mit einem technisch angemessenen Aufbau. Es fällt relativ leicht, weitere Anwendungen für solche oder ähnliche Geräte zu finden, die Meßgrößen verarbeiten und Steuerungsaufgaben übernehmen sollen. Man sollte dabei aber nicht das eigentliche Ziel dieser oder ähnlicher Einführungen in Mikroprozessoranwendungen aus den Augen verlieren. Für einen am CBM-Computer oder ähnlichen Geräten erfahrenen Basic-Programmierer läßt sich durch solche Beispiele ein weiterer

Anwendungsbereich des Computers erschließen. Dazu benötigt man einfache und überschaubare Einstiegsprobleme, die für einen Anfänger ohne Vorkenntnisse auf dem Gebiet der Assembler-Programmierung eine zielgerichtete Einarbeitung ermöglichen. Auf diesem Weg erhält man ein besseres Verständnis für die Vorgänge, die im vertrauten CBM-Computer ablaufen, und nicht zuletzt wird deutlich, in welcher Weise ein Computer ohne den vertrauten Bildschirm und ohne Tastatur Aufgaben in der Steuerung und Regelung erfüllen kann.

Literatur

- [1] Weishaupt, Walter; Fleischer, Helmut: Service-Intervall-Anzeige. mc 1982, Heft 2, Seite 57...61.
- [2] Commodore-Computer, Bedienungshandbuch, Commodore, Neu-Isenburg.
- [3] Profeld, Hans-Joachim: Komfortable Grafik mit dem CBM-Tischcomputer. mc 1982, Heft 5, Seite 48...49.
- [4] Feichtinger, Herwig: Mädchen für alles. mc 1981, Heft 2, Seite 20...23.
- [5] Das EMUF-Sonderheft. Franzis-Verlag, München, 1982.
- [6] CBM-Assembler-Disassembler. H. Stäudel, 6250 Limburg/L.
- [7] Microware-Assembler. DANN-Datentechnik, Berlin, 1981.
- [8] 6502-Programmieren in Assembler. te-wi-Verlag, München, 1981.
- [9] Programmierung des 6502, Sybex, Düsseldorf, 1982.
- [10] Programmier-Handbuch. MOS-Technology, MCDS Microcomputer-Datensysteme, Darmstadt, 1977.
- [11] Sternberg, Günther: Simulator für EPROM. mc 1982, Heft 6, Seite 36...39.
- [12] Instant-ROM. Fa. Tec-Sys, München.
- [13] Konz, Michael: CBM-EPROM-Programmer. mc 1982, Heft 3, Seite 72...76.

9C44 A2 01 HI1 LDX # 1 ; START ZAEHLER HI	9C72 08 08 08	.#08,#08,#08,#08,#08;25-29
9C46	9C75 08 08	
9C46 A0 42 LO1 LDY # 66 ; START ZAEHLER LO	9C77 08 08 08	.#08,#08,#08,#08,#08;30-34
9C48	9C7A 08 08	
9C48 EA WA1 NOP ; WARTESCHLEIFE	9C7C 08 08 09	.#08,#08,#09,#09,#09;35-39
9C49	9C7F 09 09	
9C49 EA NOP	9C81 09 09 09	.#09,#09,#09,#09,#09;40-44
9C4A	9C84 09 09	
9C4A EA NOP	9C86 09 09 09	.#09,#09,#09,#09,#09;45-49
9C4B	9C89 09 09	
9C4B EA NOP	9C8B 10 10 10	.#10,#10,#10,#10,#10;50-54
9C4C	9C8E 10 10	
9C4C EA NOP	9C90 10 10 10	.#10,#10,#10,#10,#11;55-59
9C4D	9C93 10 11	
9C4D 88 DEV	9C95 11 11 11	.#11,#11,#11,#11,#11;60-64
9C4E	9C98 11 11	
9C4E D0 F8 BNE WA1 ; WARTESCHLEIFE 1	9C9A 11 11 11	.#11,#11,#11,#11,#12;65-69
9C50	9C9D 11 12	
9C50 CA DEX	9C9F 12 12 12	.#12,#12,#12,#12,#12;70-74
9C51	9CA2 12 12	
9C51 D0 F3 BNE LO1 ; ZAEHLER LO	9CA4 12 12 12	.#12,#12,#12,#13,#13;75-79
9C53	9CA7 13 13	
9C53 38 SEC	9CA9 13 13 13	.#13,#13,#13,#13,#13;80-84
9C54	9CAC 13 13	
9C54 E9 01 SBC #1	9CAE 14 14 14	.#14,#14,#14,#14,#14;85-89
9C56	9CB1 14 14	
9C56 D0 EC BNE HI1 ; ZAEHLER HI	9CB3 14 14 15	.#14,#14,#15,#15,#15;90-94
9C58	9CB6 15 15	
9C58 60 RTS	9CB8 15 15 16	.#15,#15,#16,#16,#16;95-99
9C59	9CBB 16 16	
9C59	9CBD 16 16 16	.#16,#16,#16,#17,#17;100-104
9C59	9CC0 17 17	
9C59	9CC2 17 17 17	.#17,#17,#17,#17,#18;105-109
9C59	9CC5 17 18	
9C59	9CC7 18 18 18	.#18,#18,#18,#18,#19;110-114
9C59	9CCA 18 19	
9C59	9CCC 19 19 19	.#19,#19,#19,#19,#19;115-119
9C59	9CCF 19 19	
9C59	9CD1 20 20 20	.#20,#20,#20,#20,#20;120-124
9C59	9CD4 20 20	
9C59	9CD6 21 21 21	.#21,#21,#21,#21,#21;125-129
9C59	9CD9 21 21	
9C59	9CDB 22 22 22	.#22,#22,#22,#22,#22;130-134
9C59	9CDE 22 22	
9C59	9CE0 23 23 23	.#23,#23,#23,#23,#24;135-139
9C59	9CE3 23 24	
9C59	9CE5 24 24 24	.#24,#24,#24,#25,#25;140-144
9C59	9CE8 25 25	
9C59	9CEA 25 25 25	.#25,#25,#25,#26,#26;145-149
9C59	9CED 26 26	
9C59	9CEF 26 26 27	.#26,#26,#27,#27,#27;150-154
9C59	9CF2 27 27	
9C59 00 00 06 TABELLE	9CF4 28 28 28	.#28,#28,#28,#28,#28;155-159
9C59 06 06	9CF7 28 28	
9C5E 06 07 07	9CF9 29 29 29	.#29,#29,#29,#29,#29;160-164
9C61 07 07	9CFC 29 29	
9C63 07 07 07	9CFE 30 30 30	.#30,#30,#30,#30,#30;165-169
9C66 07 07	9D01 30 30	
9C68 07 07 07	9D03 31 31 31	.#31,#31,#31,#32,#32;170-174
9C6B 07 07	9D06 32 32	
9C6D 07 07 07	9D08 33 33 33	.#33,#33,#33,#39,#39;175-179
	9D0B 99 99	