

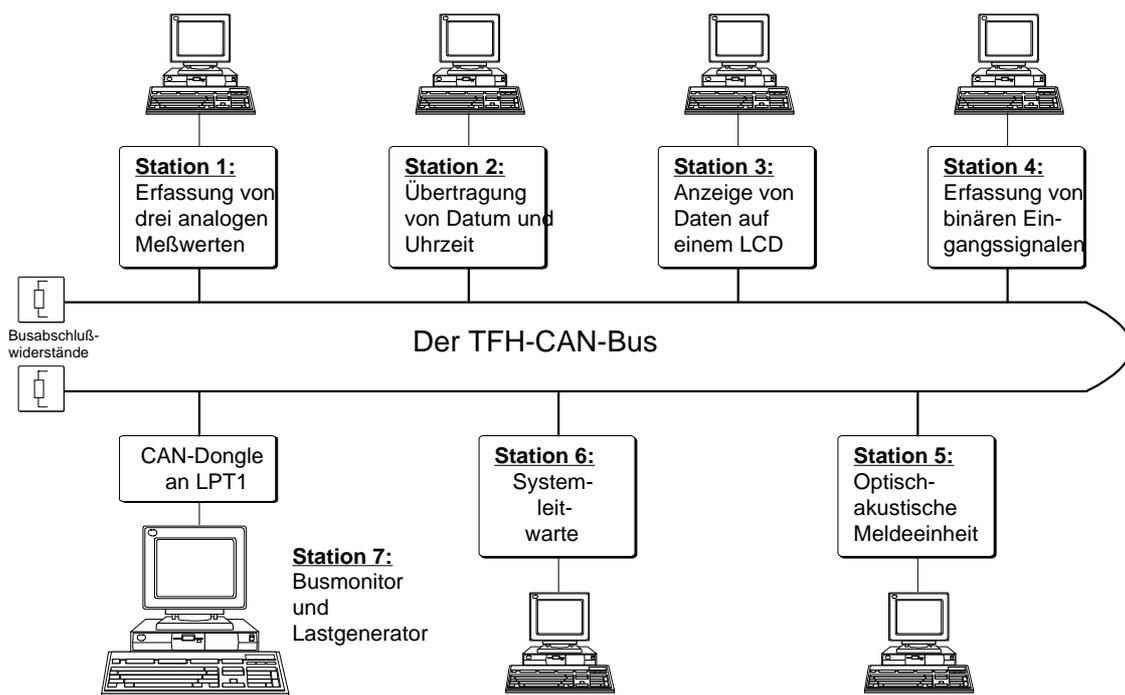
# *Das TFH-CAN-Bus- Experimental-System*

Intelligente, dezentrale Datenkommunikation  
für den Praktiker

**Bernd vom Berg, Peter Groppe**

Nachdem Sie sich in die Grundlagen des CAN-Busses eingearbeitet, die Hardware aufgebaut und sich mit der Programmierung des CAN-Controllers vertraut gemacht haben, stellen wir Ihnen nun als Höhepunkt unserer CAN-Artikel-Serie ein kleines CAN-Bus-Experimentalsystem vor, das sehr einfach und dennoch leistungsfähig aus den Komponenten des **80C537er-TFH-Mikrocontroller Systems für Lehre und Ausbildung** zusammengestellt werden kann.

In der **Abb.1** sehen Sie den Grundaufbau dieses Systems mit seinen sieben Stationen:



**Abb.1: Das TFH-CAN-Bus-Experimentalsystem**

In diesem System werden die wichtigsten Aufgaben, die ein Feldbussystem grundsätzlich erfüllen muß, realisiert:

- Erfassung von analogen Meßwerten und deren Verteilung
- Einlesen und Ausgeben von digitalen (binären) Informationen: Erfassung von Tastendrücken und Ansteuerung von akustischen Meldern.
- Einspeisung komplexer Datensätze: Uhrzeit und Datum als gemeinsame Zeitbasis für das gesamte System.
- Darstellung komplexer Systeminformationen auf einem LC-Display.
- Programmierung einer zentralen Leitstation (Systemleitwarte) am Bus, auf der alle relevanten Systeminformationen dargestellt werden.

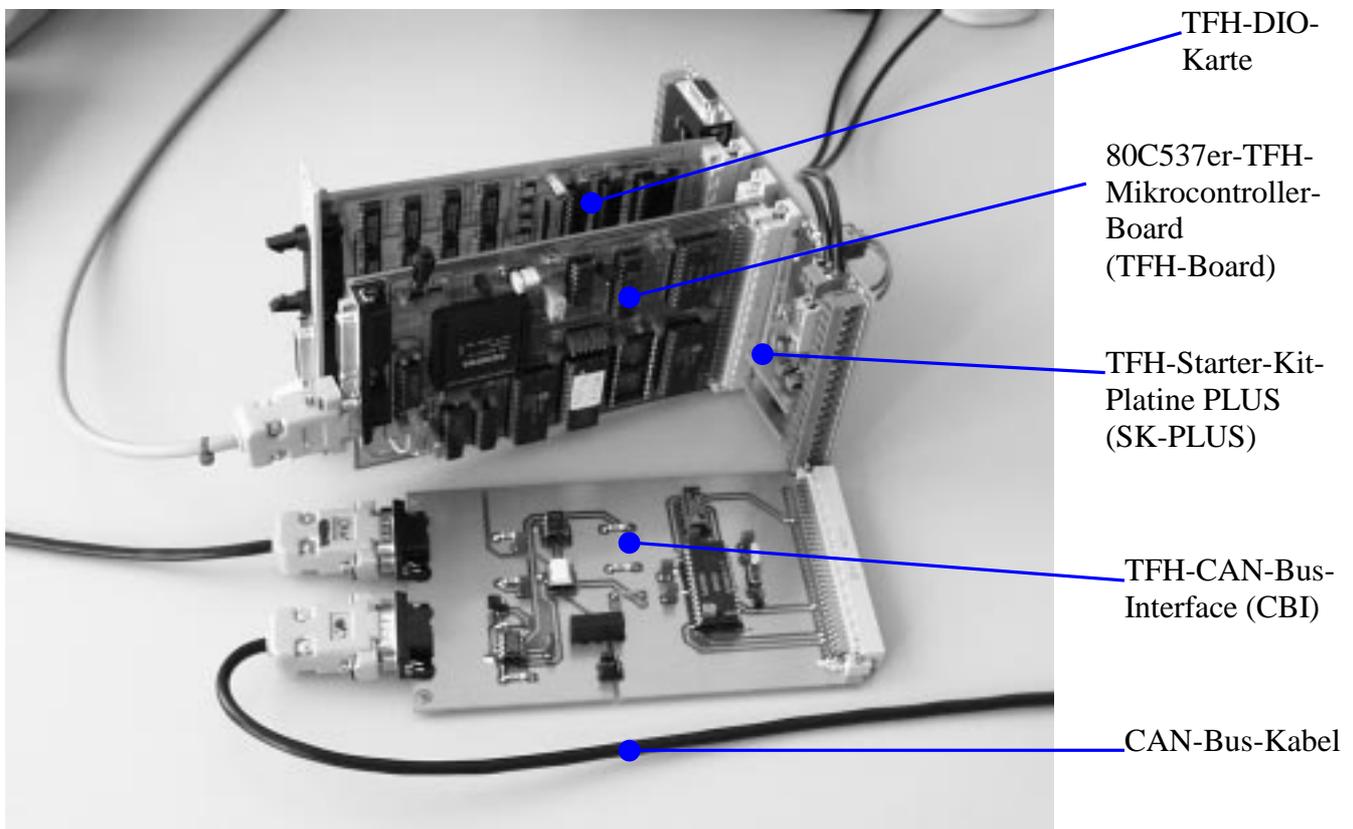
- Integration einer Busmonitorstation, die den gesamten Frameverkehr auf dem Bus überwacht bzw. darstellt und die als Lastgenerator fungiert, d.h. die einen zusätzlichen Frameverkehr zwischen "Phantom-Stationen" auf dem Bus erzeugen kann.

In der Realität können die einzelnen Feldbus-Stationen durchaus einige hundert Meter voneinander verteilt angeordnet sein, in unserem Laboraufbau beträgt die Gesamtausdehnung dieses Bussystems jedoch nur ca. 30 m, wobei die Datenübertragungsrate 50 kBit/s beträgt.

Die einzelnen CAN-Stationen sind dabei aus den verschiedenen Komponenten des **80C537er-TFH-Mikrocontroller-Systems für Lehre und Ausbildung** aufgebaut, [1,2,3]. Eingesetzt werden, **Abb.2**:

- das 80C537er-Mikrocontroller-Board (TFH-Board),
- das TFH-CAN-Bus-Interface (CBI),
- verschiedene Zusatzkarten aus dem TFH-System, hier: die "Digitale I/O-Karte" (DIO) und die "Mensch-Maschine-Schnittstellen-Karte" (MMS)

Verbunden werden diese Karten untereinander durch entsprechende Back-Plane-Platinen, z.B. mit insgesamt drei Steckplätzen (TFH-Starter-Kit-Platine (SK) PLUS).



**Abb.2: CAN-Bus-Station aufgebaut aus den Komponenten: 80C537er-TFH-Board, TFH-CAN-Bus-Interface, TFH-DIO-Karte und TFH-Starter-Kit-Platine PLUS**

Die in Abb.1 an den CAN-Stationen noch eingezeichneten PC's sind die Entwicklungsrechner für die Stationssoftware (natürlich ist ein einziger PC völlig ausreichend !) und dienen im Betrieb oder während der Testphase lediglich als "dumme Terminals" zur Anzeige von wichtigen, stationsinternen Daten. Im endgültigen Einsatz des CAN-Busses können sie ersatzlos entfallen.

Sowohl die Betriebssoftware als auch die Applikationssoftware sind in unseren Anwendungen in Pascal51 geschrieben, [2,3,4]. Selbstverständlich kann die gesamte CAN-Bus-Station auch in einer anderen 8051er-Sprache programmiert werden:

- Programmierung in Basic52, [7].
- Programmierung in C51: hierzu finden sich reichhaltige Hinweise (Flußdiagramme und Programmausschnitte) in den Applikationsunterlagen der Firma Philips, [6].

Das von uns erstellte Software-Paket **can.pas** (**can.hex**) stellt dem Anwender die folgenden Möglichkeiten zur Verfügung, **Tab..1**:

- Betrieb des SJA1000 im BasicCAN-Modus
- Interaktive Parametrierung des SJA1000:
  - Festlegung der Datenübertragungsrate im Bereich von 5 kBit/s bis zu 1 MBit/s.
  - Vorgabe eines festen (Default)Sendeframes mit variabler Datenfeldlänge und variablem Inhalt der Datenbytes.
  - Einstellung des Akzeptanz-Empfangsfilters: Festlegung des Akzeptanz-Codes und der Akzeptanz-Maske.
  - Festlegung eines RTR-Antwort-Frames.
- Aussendung von Frames:
  - Aussendung von Frames mit festen Datenwerten.
  - Aussendung des zuvor parametrierten (Default)Frames.
  - Aussendung eines RTR-Frames ( $\equiv$  RTR-Anforderung an eine andere Station)
- Empfang von Frames:
  - Fehlerüberprüfung und Darstellung der Fehlerart.
  - Darstellung der Inhalte der empfangenen Frames
  - Automatische Auswertung von empfangenen RTR-Frames und Aussendung des zuvor parametrierten RTR-Antwort-Frames.

**Tab.1: Die Möglichkeiten der CAN-Betriebssoftware can.pas**

Mit diesem Programm ist es jetzt sehr einfach möglich, die CAN-Bus-Interfaces in Verbindung mit dem jeweiligen Mikrocontroller-System zu testen, ein CAN-Bus-System schrittweise aufzubauen und in Betrieb zu nehmen.

Nachfolgend möchten wir Ihnen nun die einzelnen Stationen des TFH-CAN-Bus-Experimentalsystems mit ihren Funktionen näher vorstellen:

### **Station 1: Erfassung analoger Meßwerte**

(Hardware-Komponenten: TFH-Board, CBI, SK-PLUS)

An drei der zwölf analogen Eingänge des 80C537ers werden Meßwerte durch angeschlossene Potentiometer simuliert. Die eingestellten Spannungen werden gewandelt (Wertebereich: 0 ... 255) und zyklisch alle 750 ms über den CAN-Bus gesendet.

Gleichzeitig wird der Meßwert des zweiten Kanals überwacht und bei Überschreitung des oberen Grenzwertes von 200 wird ein Frame gesendet, das in der Station 5 die Alarmsirene aktiviert. Bei Unterschreitung des Wertes 200 wird die Sirene durch einen anderen Frame wieder ausgeschaltet.

Verwendete (gesendete) Frame-Identifizier:

- 100h für den Meßwertframe (Datenfeldlänge: 3 Bytes)
- 402h für das Einschalten der Sirene (Datenfeldlänge: 0 Bytes)
- 403h für das Ausschalten der Sirene (Datenfeldlänge: 0 Bytes).

### **Station 2: Übertragung von Uhrzeit und Datum**

(Hardware-Komponenten: TFH-Board, CBI, SK-PLUS)

Die Informationen aus dem sich auf dem TFH-Board befindlichen RTC-Chip (Uhrzeit und Datum) werden ausgelesen und alle 800 ms zyklisch auf den CAN-Bus übertragen.

Verwendete (gesendete) Frame-Identifizier:

- 300h für die Übertragung der Uhrzeit (Datenfeldlänge: 6 Bytes)
- 310h für die Übertragung des Datums (Datenfeldlänge: 7 Bytes)

Alternativ kann diese Station auch mit einer Baugruppe zum Empfang und zur Aufbereitung von DCF77-Signalen ausgerüstet werden, so daß dann im gesamten System immer die korrekte "Atom-Zeit" zur Verfügung steht, [4].

### **Station 3: Anzeige von Daten auf einem LC-Display**

(Hardware-Komponenten: TFH-Board, MMS, CBI, SK-PLUS, LC-Display)

Diese Busstation dient zur reinen Visualisierung von Daten aus dem System und zwar werden angezeigt:

- die drei analogen Meßwerte der Station 1
- die Ein-/Ausschaltzustände für die Alarmsirene und die Glocke der Station 5

Mit anderen Worten: diese Station empfängt zwar alle Frames auf dem Bus (keine Akzeptanzfilterung eingeschaltet), wertet aber nur diejenigen mit den Identifiern 100h, 401h bis 403h aus, d.h. stellt deren Informationen auf dem LC-Display dar. Eigene Frames werden von dieser Station nicht ausgesendet.

#### **Station 4: Erfassung binärer Eingangssignale**

(Hardware-Komponenten: TFH-Board, MMS, CBI, SK-PLUS, Matrix-Tastatur)

Diese Station erfaßt mit einer Zykluszeit von 200 ms drei Tastendrucke von einer angeschlossenen Matrixtastatur, die dann in Form unterschiedlicher Frames auf dem CAN-Bus publiziert werden.

Verwendete (gesendete) Frame-Identifizier:

- 200h: Aktivierung von Taste 0 (Datenfeldlänge: 0 Byte)
- 201h: Aktivierung von Taste 1 (Datenfeldlänge: 0 Byte)
- 202h: Aktivierung von Taste 2 (Datenfeldlänge: 0 Byte)

Gleichzeitig werden über die Tasten 0 und 1 noch die Sirene und die Glocke der Station 5 aktiviert (mittels gesendeter Frames mit den Identifiern 400h ... 403h).

#### **Station 5: Optisch-akustische Meldeeinheit**

(Hardware-Komponenten: TFH-Board, DIO, CBI, SK-PLUS, Sirene, Glocke)

An dieser Station sind zur Zeit zwei Signalgeber (eine Sirene und eine Glocke) angeschlossen, die bei bestimmten Ereignissen über den CAN-Bus aktiviert werden können, z.B. die Sirene bei der Überschreitung kritischer Meßwertgrenzen (von der Station 1 ausgelöst) oder die Glocke beim Anfang und Ende der Mittagspause (angesteuert von der Station 2 aus). Diese Einheit empfängt also nur Frames und wertet diese aus.

Verwendete (empfangene) Frame-Identifizier:

- 400h: Glocke ein (Datenfeldlänge: 0 Byte)
- 401h: Glocke aus (Datenfeldlänge: 0 Byte)
- 402h: Sirene ein (Datenfeldlänge: 0 Byte)
- 403h: Sirene aus (Datenfeldlänge: 0 Byte)

#### **Station 6: Systemleitwarte**

(Hardware-Komponenten: TFH-Board, CBI, SK-PLUS)

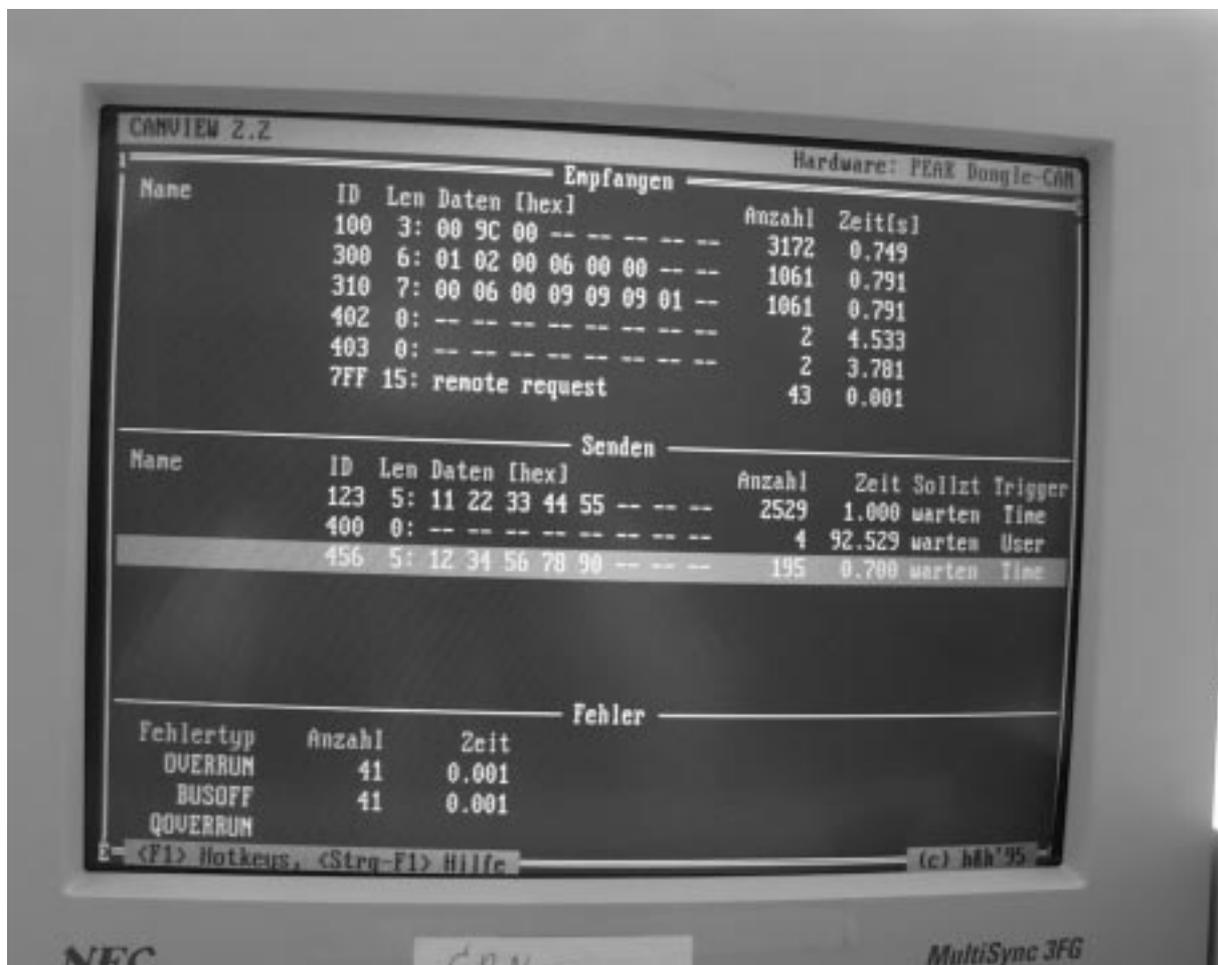
In dieser Station werden alle relevanten Frames empfangen, aufbereitet und auf dem PC-Monitor visualisiert: die Uhrzeit und das Datum, die erfaßten Meßwerte, die gedrückten Tasten, die Aktivierung von Glocke und Sirene und statistische Daten über die Anzahl der insgesamt empfangenen Frames, die Anzahl der aufgetretenen Datenübertragungsfehler und die Anzahl der Frames mit falschen Datenfeldlängen.

Mit anderen Worten: diese Station wertet alle Frames auf dem Bus aus, ohne jedoch selber welche auszusenden.

**Station 7: Busmonitor und Lastgenerator**

(Hardware-Komponenten: PC mit "CAN-Dongle")

Diese Station ist ein einzelner PC, der allerdings jetzt direkt an den CAN-Bus angeschlossen ist. Das Interface für diese Einheit besteht aus einem kommerziellen "CAN-Dongle", der einfach auf die LPT-Schnittstelle aufgesteckt wird, [5]. Mit Hilfe eines Softwarepaketes kann man so einen PC sehr leicht und preiswert "CAN-tauglich" machen. Der Anwender hat nun einerseits die Möglichkeit, den gesamten Frameverkehr auf dem CAN-Bus zu empfangen und zu analysieren, **Abb.3**, und kann andererseits auch Test- und Lastframes auf den CAN-Bus einspeisen, um zu sehen, wie die anderen Stationen auf diese Frames reagieren. Man hat damit ein unentbehrliches Test-, Analyse- und Inbetriebnahme-Hilfsmittel zur Verfügung.



**Abb.3: Die Darstellung auf der Busmonitor- und Lastgenerator-Station**

## **Der Blick nach vorne**

Die zukünftigen Weiterentwicklungen des TFH-CAN-Bus-Experimental-Systems bewegen sich in zwei Richtungen:

1. Aufbau äußerst kompakter CAN-Bus-Stationen mit dem SLIO(Serial Linked I/O)-Baustein PC82C150 von Philips. Dieser IC enthält einen kompletten CAN-Controller und eine Vielzahl von analogen und digitalen Ein- und Ausgängen direkt integriert auf einem Chip. Die Anzahl der benötigten Bauelemente zum Aufbau einer CAN-fähigen Sensor-/Aktorstation wird damit auf ein Minimum reduziert.
2. "CAN-Bus meets the Big Brother WWW": über eine äußerst kompakte Server-Station wird der CAN-Bus an das Netz der Netze angeschlossen und damit wird die erste Komponente des großen Traumes "Weltweites Messen, Steuern, Regeln und Überwachen über das Internet" verwirklicht.

## **Hinweis:**

Das hier verwendete TFH-CAN-Bus-Interface (CBI) unterscheidet sich vom Elektor-CAN-Bus-Interface (s. Elektor 11/99, S.30 ff) nur durch das Platinenformat (angepaßt an das 80C537er-TFH-Europakrten-Format) und durch den zur TFH-Backplane passenden Steckverbinder (96pol-VG-Leiste).

## **Literatur**

- [1] Bernd vom Berg, Peter Groppe  
"80C537-Einplatinencomputer"  
Elektor, Nr. 6/97
- [2] Bernd vom Berg, Peter Groppe  
"Das 8051er-Lehrbuch"  
Elektor-Verlag Aachen, 1997
- [3] O´Niel V. Som  
"Pascal für 8051 und Derivate"  
Elektor-Verlag, Aachen, 1998
- [4] Bernd vom Berg, Peter Groppe  
"Das 8051er Praktikumsbuch"  
Elektor-Verlag, Aachen, 2000

- [5] PCAN-Dongle  
Parallel zu CAN Adapter für PC's  
PEAK-Service GmbH  
[www.peak-service.com](http://www.peak-service.com)
- [6] CAN-Bus-Controller- und Transceiver-Bausteine Philips-Semiconductors:  
<http://www-us.semiconductors.philips.com/can/>  
<http://www-us.semiconductors.philips.com/can/support>
- [7] Burkhard Kainka  
"CAN-Bus-Praxis mit BASIC537"  
Elektor 1/2000

## *Und wie geht's nun weiter, wenn Sie so richtig mit dem CAN-Bus los legen wollen ?*

Für Ihren ersten Einsteig in die CAN-Bus-Technik können wir Ihnen jetzt noch vier gute Tipps mit auf den Weg geben:

### **1. Die CAN-Bus-Betriebssoftware: can.pas / can.hex**

Auf dem Elektor-Web-Server finden Sie noch unsere CAN-Betriebssoftware (can.pas / can.hex), mit der Sie Ihr CAN-Bussystem austesten und in Betrieb nehmen können.

### **2. Die CAN-Applikationssoftware für das TFH-CAN-Bus-Experimental-System**

Die Applikationssoftware (Pascal51er-Quell-Code und 8051er-Intel-HEX-File) zum Betrieb der Feldbusstationen 1 bis 6 können Sie bei der **Firma PalmTec** (s. nachfolgend) als Paket für 35,00 DM (zuzüglich Verpackung und Porto) bestellen.

### **3. Weitergehende Literatur zum diesem Thema**

finden Sie in unserem neusten Buch aus dem Elektor-Verlag:

#### **"Das 8051er Praktikumbuch"**

(s. Literatur-Liste, [4]).