

---

## Multi-Multimeter

Gunther Zielosko

### 1. Was wir wollen

Eine sehr anspruchsvolle Aufgabe für Elektroniker ist wohl die gleichzeitige Erfassung und Darstellung mehrerer Messwerte über lange Zeiten. Ein Beispiel wäre die Beobachtung von Lade- und Entladezyklen eines Akkus. Dabei sind z.B. die jeweilige Akkuspannung, der Lade- und Entladestrom, die Kapazität, Temperatur und andere Daten über die Zeit interessant. Wenn wir uns die gängigen Mittel für ein solches Vorhaben einmal genauer anschauen, finden wir eigentlich bei allen „ein Haar in der Suppe“.

- Es fängt schon bei der Anzahl der Messstellen an. Ein typischer Oszillograf z.B. hat meist nur einen bzw. zwei Kanäle, Geräte mit vier Kanälen sind bereits ausgesprochen teuer. Noch teurer wird es, wenn wir die Messwerte dann auch noch protokollieren, d.h. z.B. am Computer auswerten wollen.
- Das zweite Problem stellt die lange Messzeit dar. Wenn es um Stunden geht, sind praktisch alle Oszillografen überfordert.
- Man könnte die vier A/D-Eingänge des BASIC-Tigers<sup>®</sup> nutzen. Hierbei haben wir eine Auflösung von 10 Bit (= 1024 Counts), einen Meßbereich von 0...5V und einen Massebezug der Messungen, was oftmals Anpassungen an die jeweilige Messaufgabe erforderlich macht. Sollen z.B. Ströme gemessen werden, braucht man niederohmige Shunt-Widerstände, um die Messung nicht zu verfälschen. Dabei sind aber die zu messenden Spannungen meist sehr klein und es wird ein Verstärker notwendig. Negative Spannungen können nur über entsprechende Wandler erfasst werden.
- Auch eine Erweiterung mit einem A/D-Wandlerbaustein aus dem Sortiment von Wilke Technology oder anderen Mehrfach-A/D-Wandlern ist aus den genannten Gründen nicht immer eine Alternative.
- Multimeter mit Datenschnittstelle zum Computer (meist noch RS232) haben da verschiedene Vorteile. Sie bereiten die Messwerte bereits auf (z.B. Spannung, Strom, Widerstand, Temperatur und dazu die entsprechenden Messbereiche). Zudem haben sie meist eine sehr hohe Genauigkeit. Einfache Multimeter mit Datenschnittstelle sind dabei noch erschwinglich. Schwierig wird es allerdings, wenn mehrere Messdaten parallel erfasst und aufgezeichnet werden sollen. Wenn überhaupt, haben die meisten Computer nur noch ein oder zwei serielle Schnittstellen. Die zu einem solchen Multimeter mitgelieferte Daten-Erfassungs-Software ist in der Regel nur einmal aufrufbar, d.h. auch bei mehreren vorhandenen RS232-Schnittstellen kann nur ein Messwertverlauf dargestellt werden.

Was also können wir tun, um Messdaten

- über lange Zeiten,
- mit relativ geringen Kosten,
- auf mehreren Kanälen,
- potentialfrei,
- angepasst an den jeweiligen Messwert (Spannung, Strom usw.) und Messbereich
- und hinreichend genau

zu erfassen und protokollieren? Wir haben erkannt, dass gängige Multimeter alle Eigenschaften haben, die unsere Aufgabe lösen könnten. Wenn wir mehrere davon in unterschiedlichen Messbereichen, zur Messung unterschiedlicher Größen und potentialfrei gleichzeitig betreiben und z.B. in den Computer einlesen könnten, wäre dies die Lösung.

Die Grundlagen für die Anwendung von Multimetern zur Datenerfassung mit dem BASIC-Tiger® haben wir schon im Bericht Nr. 036 behandelt - was liegt näher, als die Erkenntnisse von dort auf unsere neue Aufgabe anzuwenden?

Noch fehlen uns aber am PC eine ausreichende Anzahl Schnittstellen sowie die entsprechende „Multikanal-Software“.

## **2. Die Lösung: SER2 - Serielle Schnittstellen durch Software**

BASIC-Tiger® und Tiger-BASIC® bieten, wie wir wissen, für fast alle Probleme eine Lösung. Warum sollte nicht das kleine Multitalent die Sammlung und Aufbereitung der Daten von z.B. vier Multimetern übernehmen und sie dann in geeigneter Form an den PC übergeben? Vielleicht ist auch die zusätzliche Nutzung der vier internen A/D-Wandler möglich, so dass am Ende vier hochgenaue, potentialfreie Messkanäle über vier Multimeter und vier 10-Bit massebezogene Messkanäle zur Verfügung stehen?

Aber auch beim BASIC-Tiger® haben wir zunächst das Problem, dass es nur zwei „offizielle“ serielle Schnittstellen gibt. Aber da war doch noch was...

Richtig, es gibt den SER2-Treiber, der serielle Schnittstellen durch Software realisieren kann. Kann man da vier zusätzliche serielle Schnittstellen gewinnen? Tatsächlich – prinzipiell scheint das zu gehen. Schauen wir uns einmal diesen Gerätetreiber genauer an, am besten die Definitionen des Device-Driver-Handbuches:

*Dieser Device-Treiber ermöglicht die asynchrone serielle Ein- und Ausgabe auf internen I/O-Pins. Die Schnittstelle ist rein softwaremäßig realisiert. Bei der Installation des Treibers wird durch den Dateinamen festgelegt, an welchen Pins die serielle Eingabe und Ausgabe stattfindet. Die Baudrate ist durch die Einstellung des TIMERA sowie den Prescaler des Device-Treibers festgelegt.*

*Der Treiber kann auf individuelle Anforderungen eingestellt werden:*

- *RxD + TxD: Kanäle einzeln zu-/abschaltbar.*
- *RxD + TxD: jeweils mit 256 Byte FiFo-Puffer ausgestattet. (mit neuem Treiber einstellbar!)*
- *RxD + TxD: mit Flusskontrolle: RTS / CTS zu-/abschaltbar.*
- *TxD: RS-485 Buszugriffs-Kontrolle TE zu-/abschaltbar.*
- *Data-Bits: Datenformat: 1...8 Bits.*
- *Parity-Bit: No, Even, Odd, Mark, Space.*

- *Baudraten: quasi-stufenlose Baudraten über TIMERA und Prescaler im Bereich von 12 kBd-Senden (bei 1 x TxD) bis zu 3 Bd.*
- *Pegel: TRUE + INVERSE Pegel möglich für RS-232 mit/ohne Leitungs-Treiber.*
- *PINs: RxD, TxD, RTS, CTS und TE lassen sich auf fast jeden beliebigen I/O-Pin des Tigers legen.*
- *Kanäle: bis zu 8 serielle Kanäle (RxD / TxD in beliebiger Mischung).*

Das klingt gut, leider gibt es für diesen Supertreiber aber auch Beschränkungen wie diese:

*Beachte: SER2\_XX.TDD belastet die CPU um ein Vielfaches mehr als ein Treiber wie SER1B\_XX.TDD, da für jedes einzelne Bit mehrere System-Task-Aufrufe erfolgen. Beim Einsatz dieses Treibers sollte daher berücksichtigt werden:*

- *SER2 nur einsetzen, wenn noch freie CPU Leistung verfügbar ist.*
- *Gesamt-Baudrate aller RxD und TxD Kanäle nicht zu hoch wählen:*
- *Bei derzeitigen Modulen: Summe = max. ca. 10 kBaud.*
- *z.B: 5 x TxD a 1200 Bd oder: 1 x RxD + TxD a 4800 Bd.*
- *Bei hoher CPU-Auslastung kann die Debug-Funktion beeinträchtigt werden.*

**Noch etwas Wichtiges. Im Laufe der Zeit wurden verschiedene Device-Driver von Wilke Technology verbessert, an neue Bedingungen angepasst und Fehler an ihnen beseitigt. Betroffen davon ist auch der Device-Driver SER2. Nach diversen Fehlversuchen mit dem alten Treiber aus Tiger-BASIC<sup>®</sup> 5.01 lief es mit dem aktuellen Treiber dann problemlos. Zur Sicherheit legen wir den funktionierenden Treiber SER2\_80\_R1.TDD für die gewählte Konfiguration mit bei.**

Versuchen wir es also. Eigentlich wollten wir ausgehend vom Bericht Nr. 036 wieder Multimeter vom Typ DT-9062 von DIGITEK benutzen. Mühsam haben wir damals die komplizierte Datenverschlüsselung aufgeklärt und die Software dafür entwickelt. Allerdings ist die Firma Digitek im Internet „umgezogen“ und kann nun über:

<http://www.digitek.com.hk/index.html>

erreicht werden. Auch das DT-9062 ist bei ELV nicht mehr erhältlich, dafür gibt es den Nachfolger DT-4000ZC, der vom Autor mit der hier beschriebenen Soft- und Hardware getestet wurde, das DT-9062 funktioniert ebenfalls:

<http://www.elv.de/Main.asp?Menue=Shop&SessionId=00205407560330761134>

Die Bestelldaten bei ELV sind:

RS-232-Digital-Multimeter DT-4000 ZC  
Artikel-Nr.: 68-459-86                      39,95 €

### Technische Daten

Gleichspannung:                      400mV/4/40/400V; 0,5%, 600V; 0,8%  
Wechselspannung:                    4/40/400V; 0,8%, 600V; 1,2%, (40-400Hz)  
Gleichstrom:                            400uA/4mA; 2%, 40/400mA; 1,5%, 4/10A; 2%  
Wechselstrom:                        400uA/4mA; 2,5%, 40/400mA; 2%, 4/10A; 2,5%, (40-400 Hz)

Widerstand:	400 Ohm/4/40/400 kOhm/4 MOhm; 1%, 40 MOhm; 2%
Kapazität:	40nF; 3%, 400nF/4/40uF; 2%, 100uF; 3%
Frequenz:	10/100/1k/10k/100kHz/1M/10MHz; 0,02%
Tastverhältnis:	0,1-99%, 2%
Temperatur (DT 4000 ZC):	integrierter Sensor: 0 bis 40°C; +/-3°C
Externer Sensor:	-50 bis 200°C; +/-(-0,75%+3°C), 200 bis 750°C; +/-(-1,5%+3°C)
Batterien:	2 x Mignon/AA
Abmaße (H x B x T):	163 x 80 x 44 mm inkl. Holster
Gewicht:	500 g inkl. Batterien



*Bild 1 Das DT-4000ZC von Digitek mit 4000 Counts (Anzeige max. 3999)*

Nun versuchen wir, mit Hilfe des SER2-Treibers mehrere dieser Geräte gleichzeitig abzufragen. Die Baudraten des DT-9062 und des DT4000ZC sind auf 2400 Bd festgelegt, damit könnten wir trotz der oben beschriebenen Grenzen des Treibers gerade 4 Geräte bedienen ( $4 \times 2400 = 9600$ ).

Überlegen wir weiter. Die zeitlichen Abläufe sind im Gegensatz zum Treiber SER1 (übrigens nicht zu verwechseln mit der Schnittstelle SER1!), bei dem die beiden Schnittstellen direkt durch den internen Prozessor gesteuert werden, bei SER2 durch einen weiteren Device-Driver, nämlich TIMERA vorgegeben. Wir müssen also vor der Benutzung von SER2 den Treiber TIMERA installieren und programmieren, d.h. eine Frequenz vorgeben. Zum sicheren Betrieb sollte ein möglichst genaues Vielfaches der Baudrate benutzt werden, wobei sich der

Faktor 3 als Kompromiss zwischen Prozessorbelastung und Genauigkeit bewährt hat. Diese Betriebsweise wird als Oversampling bezeichnet. Für uns bedeutet das:

3 x 2400 Bd → 7200 Hz

Im Handbuch finden wir unter TIMERA keine ganz genau passende Frequenz. Das macht aber nicht viel aus, da vom Multimeter nur relativ kurze Sequenzen kommen, bei denen die sich summierenden Ungenauigkeiten noch nicht kritisch sind. Eine relativ gut passende Frequenzwahl wäre:

Frequenz	Faktor (Bereich 2)	
7.267	86	
7.183	87	→ liegt am nächsten an 7200 Hz
7.102	88	

#### *Auszug aus der Frequenztabelle des Device-Drivers TIMERA*

Damit ergeben sich die Installations-Anweisungen für die beiden Treiber wie folgt:

```
INSTALL_DEVICE #TA, "TIMERA.TDD",2,87 ' 7183 Hz fuer 2400 baud

INSTALL_DEVICE #SER2,"SER2_80_R1.TDD", & 'R1 legt die Buffer-Groesse fest
8, & ' Datenbits
0, & ' Paritaet 0=keine
1, & ' invert 0=true, 1=invers (da wir keine "echten" RS232-Schnittstelle benoetigen, "1")
0, & ' tx Prescaler
3, & ' rx Oversample (3 mal hoehere Frequenz)
1, & ' reserviert, immer 1
0 ' Handshake, 0=kein
```

Diese Einstellung erzeugt eine zusätzliche serielle Schnittstelle, wobei formell die folgende Pinzuordnung gilt:

Treiber-Name	TxD (out)	RxD (in)	CTS (in) or TE (out)	RTS (out)
SER2_80_R1.TDD	L80	L81	L82	L83

Die Eintragung (0, & ' tx Prescaler) in obigem Installationsaufruf bedeutet, dass Tx abgeschaltet worden ist, die Eintragung (0 ' Handshake, 0=kein) bedeutet, dass keine Handshake-Signale benutzt werden. Für unsere Anwendung benötigen wir nur einen Eingang RxD, also Pin L81. Sind damit die übrigen Pins von SER2\_80\_R1.TDD (L80, L82, L83) für andere Nutzung verloren? Klares Nein – alle nicht benötigten I/O's lassen sich weiterhin beliebig verwenden. Noch besser – sie können auch als weitere serielle Eingänge eingesetzt werden. Damit ist alles klar, wir werden also 4 serielle Schnittstellen SER2 installieren und für jede einen Eingang verwenden, diese können nach obigen Erkenntnissen durchaus hintereinander liegen. Wie werden dann aber im Programm die verschiedenen SER2-Schnittstellen voneinander unterschieden? Wir können für jede Schnittstelle einen Namen vergeben, also etwa nach folgendem Schema:

```
INSTALL_DEVICE #50,"SER2_80_R1.TDD" usw. (die folgenden Eintraege hier weggelassen)
INSTALL_DEVICE #51,"SER2_81_R1.TDD" usw. "
INSTALL_DEVICE #52,"SER2_82_R1.TDD" usw. "
INSTALL_DEVICE #53,"SER2_83_R1.TDD" usw. "
```

Zu beachten ist noch, dass die Device Nummern nicht mit bereits vergebenen Nummern in den Include-Dateien kollidieren. Die hier gewählten Nummern 50 bis 53 erfüllen diese Bedingung.

Im Ergebnis dieser Vorbereitungen haben wir vier zusätzliche serielle Schnittstellen, die in diesem Falle allerdings nur zum Empfang auf den Pins 81, 82, 83 und 84 benutzt werden können.

### 3. Die Schaltung

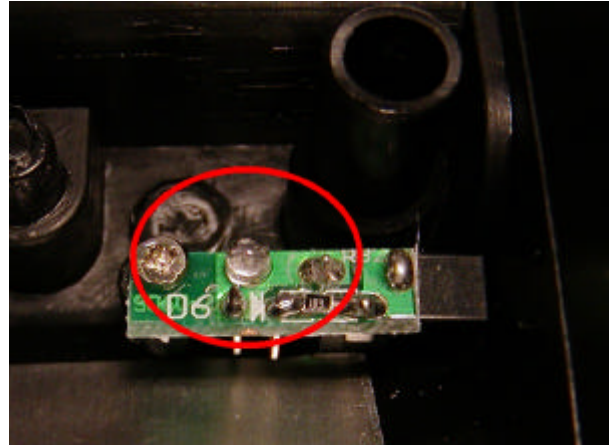
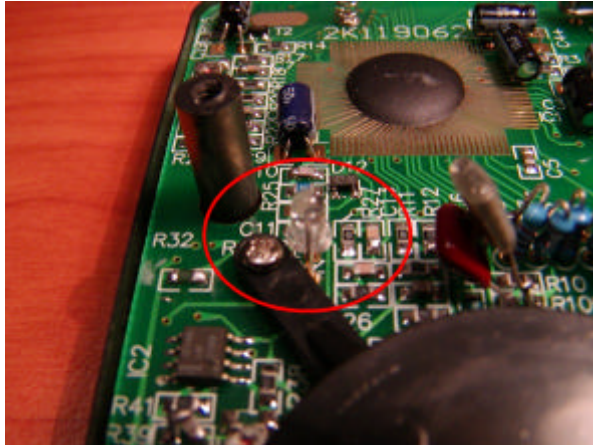
**Die Übertragung der Messwerte erfolgt im Multimeter zunächst über einen Optokoppler und dann über eine etwas ungewöhnliche serielle Schnittstelle. Die galvanische Trennung des Messgerätes vom PC oder anderen Auswertegeräte ist sowohl für die Sicherheit von Personen gegen elektrischen Schlag als auch für die oft erforderliche Erdfreiheit der Messung erforderlich. Die interne Schnittstellenanordnung im Multimeter sollte also in keinem Falle verändert werden, da die Optokoppler-Strecke wesentlicher Bestandteil des Isolationsschutzes des Gerätes ist. Nur im Originalzustand werden die technischen Daten des Gerätes auch bezüglich der Sicherheit garantiert.**

Die Hardware der äußeren Schaltung ist vergleichsweise harmlos. Die für die Ankopplung der vier Multimeter relevanten Leitungen sind softwareseitig festgelegt. Wie schon im Bericht Nr. 036 beschrieben, funktionieren die Multimeter DT-9062 bzw. DT-4000 ZC zwar an Standard-RS232-Schnittstellen, dennoch muss im Falle BASIC-Tiger® ein wenig getrickst werden. Am Pin 2 der weiblichen SUB-D-Buchse am Kabel des Messgerätes kommt das serielle Signal an. Es bietet sich an, unserer Schaltung auch 4 SUB-D-Stecker zu spendieren. Also geht das Signal auf deren Pin 2 weiter an das entsprechende I/O-Pin des Tigers. Ein zweiter Draht vom Multimeter liegt auf Pin 4 der SUB-D-Buchse. Dieser ist die Stromversorgung des Optokopplers im Multimeter. Bei normalen RS232-Schnittstellen kommt im Empfangsfall (des PC's) hier ein positives statisches Signal an. Das müssen wir „künstlich“ erzeugen, indem wir über einen Schutzwiderstand eine Verbindung zu Vcc herstellen. Masse gibt es hier nicht, da das Multimeter ohnehin galvanisch getrennt von der übrigen Welt sein soll. Immer, wenn die serielle Schnittstelle des Gerätes etwas sendet, schaltet die Fotodiode rein optisch je nach logischem Pegel eine Verbindung zu Vcc unserer Schaltung oder wird hochohmig. Damit die hochohmige Strecke auch zu einem logischen Pegel führt, braucht jede Datenleitung noch einen Widerstand zu Masse.

Noch etwas! Die Lage der logischen Signale vom Multimeter ist an echte RS232-Standard-Schnittstellen angepasst. Wir haben aber an den 4 I/O-Pins L81 - L84 nur TTL-Pegel. Die für serielle Schnittstellen überall (auch im Plug-and-Play-Lab und in den diversen BASIC-Tiger®-Ausführungen intern) verwendeten Bausteine MAX232 wandeln diese Pegel nicht nur von 0V/5V auf -15V/+15V, sondern negieren sie auch noch. Damit alles wieder stimmt, muss das Signal also irgendwo noch invertiert werden. Glücklicherweise kann man das einfach durch einen Parameter beim Aufruf des SER2-Treibers erledigen, wir erinnern uns:

```
1, & ' invert 0=true, 1=invers (da wir keine "echten" RS232-Schnittstelle benoetigen, "1")
```

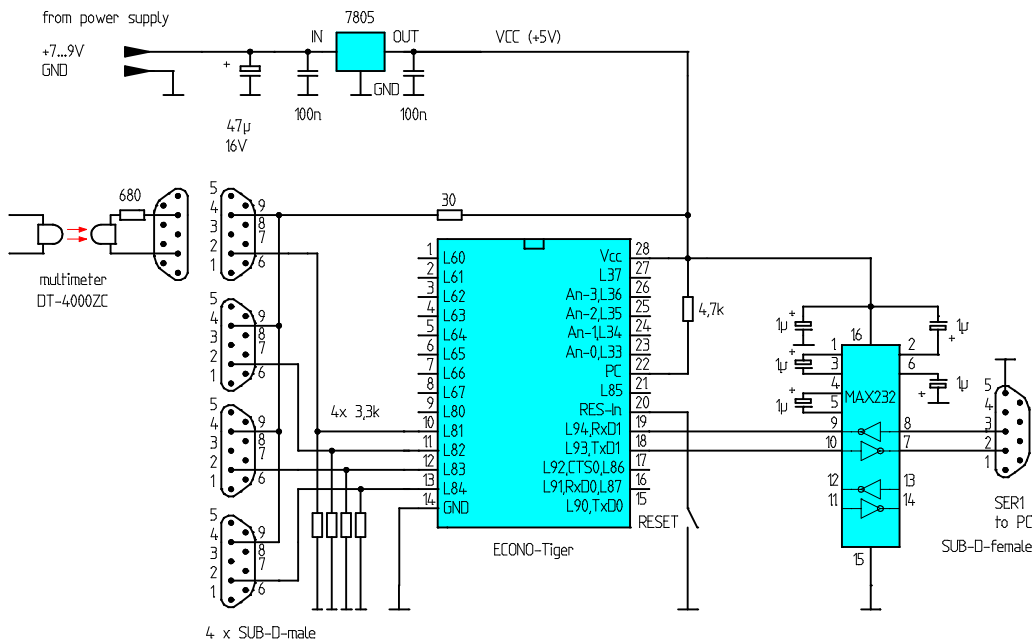
Damit kann es losgehen, die folgenden Bilder zeigen Details der Signalstrecke im Multimeter und die komplette Schaltung mit den für die Multimeter-Ankopplung relevanten Verbindungen:



*Bild 2 Infrarot-Sender (hier im DT-9062)... Bild 3 ...und der abgesetzte Empfänger*

Damit die eingesammelten Daten weiter verwendet werden können, kann ein Display oder/und eine weitere, diesmal „echte“ serielle Schnittstelle SER1 vorgesehen werden. Ein eventuelles Display wird wie gewohnt an den Port 6 sowie die erforderlichen Steuerleitungen des Port 3 angeschlossen.

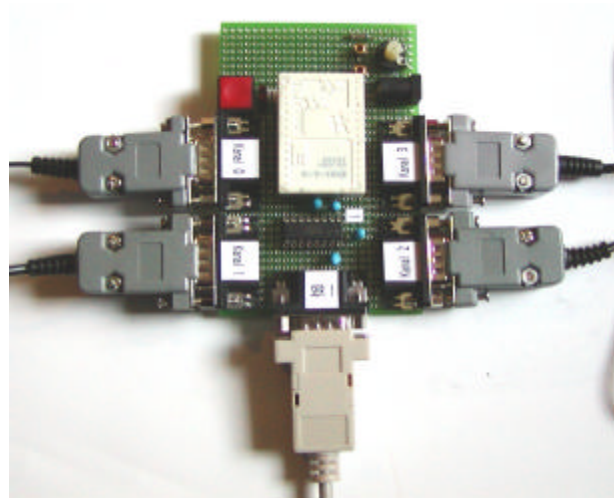
Auch die serielle Schnittstelle wird gemäß der schon oft praktizierten Schaltung realisiert. Ein MAX232 wandelt die SER1-Signale so um, dass sie ein PC verstehen kann. Falls Sie einen BASIC-Tiger® mit eingebauten RS232-Wandlern verwenden wollen, können Sie den PC natürlich direkt an SER1 anschließen. Für die serielle Schnittstelle zum PC benutzen wir 9600 Baud.



*Bild 4 Ankopplung von vier Multimetern an einen Tiny-Tiger-Economy ENN-4/4*



*Bild 5 3x DT-4000 ZC und 1x DT-9062 im Einsatz*



*Bild 6 Die „Verteilerstation“*

#### 4. Tiger-Software

Klar, dass wir ein Programm für den Tiger brauchen. Das Programm MULTI01.TIG organisiert die 4 zusätzlichen seriellen Schnittstellen, fragt ab, ob sich in den 4 Eingangspuffern Daten befinden und liest diese aus. Die kryptischen Daten – wir erinnern uns, dass das DT-9062 bzw. das DT-4000 ZC lediglich die Informationen der einzelnen Segmente sendet – werden automatisch aufbereitet und zu sinnvollen Informationen



---

umgewandelt. Dann gehen sie auf zwei Wegen nach außen, einmal über das optional anzuschließende Display und zum anderen auf die serielle Schnittstelle SER1. Die Daten aller 4 Multimeter werden nacheinander über diese Schnittstelle ausgegeben, so dass Auswerteprogramme auf dem PC die 4 Multimeter „gleichzeitig“ beobachten können.

**Zu beachten ist allerdings, dass das Programm in der jetzigen Fassung nicht mehr in den preiswerteren Tiny-Tiger-Economy ENN-R/4 passt (zu wenig RAM!), deshalb muss ein ENN-4/4 verwendet werden.**

## 5. PC-Software

Damit man mit den sozusagen gebündelten Daten der vier Multimeter wirklich etwas Praktisches anfangen kann, muss eine Auswertung am PC ermöglicht werden. Dazu gibt es zwei Möglichkeiten:

### Direkte Auswertung der Daten

Die Hauptarbeit der Umwandlung der „unlesbaren“ Daten von den vier Multimetern und deren quasi simultane Ausgabe hat der BASIC-Tiger® bereits geleistet. An der Schnittstelle SER1 liegen alle Daten als ASCII-Text vor und können direkt vom PC aufgenommen werden. Jedes Terminalprogramm kann die Messwerte sofort anzeigen und z.B. als Text-File abspeichern. Wem das noch nicht reicht – solch ein Textfile kann natürlich auch in Excel eingelesen werden. Nach ein paar Anpassungsschritten (z.B. Dezimalpunkt in Dezimalkomma für deutsches Excel umwandeln!) können die Daten dann auch in ein Diagramm übernommen werden oder stehen zu weiteren Berechnungen in Excel zur Verfügung.

### Das PC-Programm MULTI01.TST

Beigelegt zu diesem Applikationsbericht finden Sie ein Testpoint-Runtime-Programm, das die Daten der Multimeter optisch ansprechend aufbereitet und grafisch darstellt. Sie verfahren wie im Applikationsbericht 031 dargestellt und starten die Datei SETUP.EXE aus dem ZIP-Archiv MULTI01.ZIP. Nachdem alles installiert wurde, können Sie das Programm nach dem Aufruf sofort verwenden. Bevor Sie das Programm starten (Button „Start“), sollten Sie die Schnittstelle an Ihrem PC auswählen (Selektor „Schnittstelle“) und die Hardware angeschlossen haben, da es ansonsten zu einem Time-Out-Fehler der von TestPoint geöffneten seriellen Schnittstelle (Vorauswahl COM1) kommt. Liegen Daten an, können Sie auf vier Kanälen sowohl digital, als Tabelle und auch in grafischer Form alle vier Multimeter beobachten.

Legen Sie den Schalter Betrieb auf „Stop“, wird die Messreihe angehalten. Man kann dann die Kurven oder die Tabelle sichten und weiter verwenden.

Das Programm liest die in ASCII-Format vorliegenden Daten ein, zeigt auf vier Digital-Displays die Kanal-Nr., den Messwert, den Faktor (milli, kilo, Mega usw.), die Einheit sowie alle anderen Daten des gemessenen Wertes an. Für die grafische Darstellung wird dieser

Faktor so umgerechnet, dass alle Werte eines Kanals zusammen passen, auch wenn zwischendurch automatisch der Messbereich umgeschaltet wurde. Was nämlich im Normalgebrauch ein Segen ist – die automatische Bereichsumschaltung – ist für die grafische Darstellung eher ein Fluch. Wenn Sie z.B. zunächst im Volt-Bereich messen und diese Spannung dann kontinuierlich sinkt, bis sich der mV Bereich einschaltet, springen die reinen Zahlenwerte natürlich hoch. Die grafische Anzeige muss das dann „relativieren“. Im Beispielprogramm werden die „kilos“, „millis“ usw. deshalb zunächst in reine Zahlenwerte umgeformt und grafisch dargestellt. Die Skala passt sich den Werten automatisch an. Das hat aber zur Folge, dass kleine Werte (z.B. einige Millivolt) nur dann hoch aufgelöst erscheinen, wenn es nicht zwischendurch hohe Werte (z.B. einige Volt) gegeben hat, was natürlich die Skala entsprechend an diese hohen Werte angeglichen hat.

Die aufgezeichneten Kurven können mit dem Button „Inspect“ rechts im Bild später ausgewertet werden (Zoomen, Drucken). Auch der Inhalt der Tabelle lässt sich in die Zwischenablage kopieren und anderweitig weiter bearbeiten.



Bild 7 Das Programm MULTI01.TST

Viel Spaß mit dem neuen Multi-Multimeter!