

# **Die Zelle**

## **Unterrichtsbeispiele**







# Inhalt

|   |   |
|---|---|
| Vorwort.....  | <b>Fehler! Textmarke nicht definiert.</b> |
| Inhalt.....   | 5   |
| 1. Didaktische Vorüberlegungen .....                        | 7   |
| 2. Das System ZELLE .....                                   | 9   |
| 2.1 Das Grundgerät .....                                    | 9   |
| 2.2 Die Meßmodule .....                                     | 11  |
| 2.2.1 Meßmodul Lärm.....                                    | 11  |
| 2.2.2 Meßmodul Temperatur/1.....                            | 12  |
| 2.2.3 Temperatur/2 - NTC-Sensor.....                        | 13  |
| 2.2.4 Meßmodul Licht.....                                   | 14  |
| 2.2.5 Meßmodul Gas .....                                    | 15  |
| 2.3 Software .....  | 16  |
| 3. Einführung in die computergestützte Meßtechnik.....      | 19  |
| 3.1 Grundlagen.....   | 19  |
| 3.2 Temperaturmessung mit dem Monitorprogramm .....         | 21  |
| 3.2.1 Meßreihen.....  | 23  |
| 3.3 Auswertung mit WORKS .....                              | 25  |
| 3.4 Messen und beobachten mit PRISMA.....                   | 26  |
| 3.5 Messen und auswerten mit MEDA .....                     | 28  |
| 3.5.1 Verwendung eines nichtlinearen Sensors.....           | 30  |
| 3.6 Experimentieren mit UNIMESS .....                       | 31  |
| 4. Lärmmessungen.....                                       | 33  |
| 4.1 Meßtechnische Grundlagen .....                          | 33  |
| 4.2 Lärmmessung mit ZELLÄRM.....                            | 35  |
| 4.3 Direktanzeigende Messungen .....                        | 36  |
| 4.4 Registrierung und Darstellung zeitlicher Verläufe ..... | 37  |
| 4.5 Ortsunabhängige Pegelmessungen.....                     | 38  |
| 4.6 Auswertungen .....                                      | 39  |
| 4.7 Automatisch ausgelöste Messungen .....                  | 41  |
| 5. Ortsunabhängige Erfassung von Meßdaten .....             | 43  |
| 5.1 Klimabeobachtungen .....                                | 43  |
| 5.2 Messung der Luftverschmutzung.....                      | 45  |
| 5.3 Mobile Meßdatenerfassung mit dem Fahrrad .....          | 47  |
| 6. Batterien und Solarzellen.....                           | 51  |
| 6.1 Messung der Ladungskapazität.....                       | 51  |
| 6.2 Ladeverhalten von Akkus .....                           | 55  |
| 6.3 Nutzung der Solarenergie.....                           | 56  |
| 7. Versuche mit zusätzlichen Sensoren .....                 | 57  |
| 7.1 Potentiometer .....                                     | 57  |
| 7.2 Piezo-Wandler.....                                      | 57  |
| 7.3 Gleichstrommotor als Drehzahlsensor .....               | 58  |
| 7.4 Induktionsspulen .....                                  | 58  |

|  |     |
|--|-----|
| 7.5 Labormeißgeräte mit Schreiberausgang.....        | 58  |
| 8. Die Prozeßsprache .....                           | 59  |
| 8.1 Pascal-Speicheroszilloskop.....                  | 65  |
| 8.2 Transistor-Meißprogramm .....                    | 66  |
| 8.3 Bargraphanzeige in Q-Basic .....                 | 68  |
| 9. Programmierung in SIMPEL.....                     | 71  |
| 9.1 Sprachumfang und Strukturen .....                | 71  |
| 9.2 Einführung in die Programmierung mit Simpel..... | 75  |
| 9.2.1 Parameterübergabe .....                        | 76  |
| 9.2.2 Programmschleifen.....                         | 78  |
| 9.3 Anwendungsbeispiele .....                        | 80  |
| 9.3.1 Ampelsteuerung .....                           | 80  |
| 9.3.2 Minimum/Maximum-Meißwerterfassung .....        | 81  |
| 9.3.3 Unterbrochene Meißreihen.....                  | 84  |
| 9.3.4 Zählerprogramme .....                          | 87  |
| 9.3.5 Triggerung von Meißreihen.....                 | 89  |
| 9.3.6 Zeitgesteuerte Messungen .....                 | 91  |
| 9.3.7 Schrittmotoransteuerung .....                  | 93  |
| 9.3.8 I <sup>2</sup> C-Bus-Ansteuerung .....         | 95  |
| 10. Einführung in die Hardware eines Computers.....  | 99  |
| 10.1 Binärzahlen .....                               | 100 |
| 10.2 Blockstruktur und Speicher.....                 | 101 |
| 10.3 Binäre Codierung .....                          | 103 |
| 10.4 Assemblerprogrammierung.....                    | 104 |
| 10.5 Register, Rechenbefehle und Zählschleifen.....  | 105 |
| 10.6 Unterprogramme .....                            | 109 |
| 10.7 Analyse von Simpel-Programmen .....             | 111 |
| Anhang A: Anschlüsse der Zelle .....                 | 115 |
| Anhang B: Funktionsübersichten der Software .....    | 117 |
| B.1 Monitorprogramm .....                            | 117 |
| B.2 PRISMA .....                                     | 120 |
| B.3 ZELLÄRM .....                                    | 125 |
| B.4 MEDA .....                                       | 128 |
| B.5 Das Programmpaket UNIMESS.....                   | 131 |
| B.6 Übersicht der Funktionen von UNIAUS .....        | 135 |
| B.7 Der SIMPEL-Compiler .....                        | 137 |
| Anhang C: 8048-Maschinenbefehle .....                | 139 |
| Literaturverzeichnis .....                           | 141 |

## 1. Didaktische Vorüberlegungen

Für den Einsatz von Computern im Unterricht haben sich in den letzten Jahren folgende Schwerpunkte herausgebildet:

Im Bereich der Informations- und Kommunikationstechnischen Grundbildung steht der Computereinsatz selbst im Mittelpunkt. Schüler und Schülerinnen sollen mit üblichen Rechneranwendungen und ihren gesellschaftlichen Implikationen vertraut werden.

Im Informatikunterricht wird der Computer zur Einführung in die Grundlagen der Programmierung benutzt. Meist geht man dabei von einer Hochsprache aus.

Aspekte der Hardware des Computers und seiner Peripherie werden schwerpunktmäßig im Technikunterricht behandelt. Dazu gehört auch der Einsatz des Computers in technischen Systemen wie Steuerungen, Regelkreisen usw.

Im Bereich des naturwissenschaftlichen Fachunterrichts tritt der Computer in erster Linie als Hilfsmittel auf. Der Einsatz des Rechners soll etablierte Inhalte nicht verdrängen, sondern sie unterstützen. Der Einsatz ist dann sinnvoll, wenn eine Arbeitserleichterung z.B. durch verbesserte Meßtechnik erreicht werden kann oder neue bzw. vertiefte Fragestellungen bearbeitet werden können, die ohne den Rechner nicht oder nur mit großem Aufwand angegangen werden können.

Speziell beim Einsatz des Rechners zum Messen muß deutlich gemacht werden, daß die klassischen Meßverfahren nicht ersetzt, sondern nur erweitert und ergänzt werden. Deshalb darf der Rechner nicht vorschnell eingesetzt werden. Erst nach der gründlichen Einführung klassischer Meßgeräte (z.B. Voltmeter, Thermometer usw.) soll ein Computer als Meßgerät eingesetzt werden, wobei zunächst die Ähnlichkeit der Meßverfahren deutlich werden soll. In einem weiteren Schritt können dann spezifische Stärken der rechnergestützten Meßtechnik benutzt werden: automatisierte Meßreihen und verbesserte Auswertemethoden.

Alle diese Einsatzmöglichkeiten gelten für jedes Computersystem, wobei spezifische Schwächen und Stärken beachtet werden müssen. Die Zelle wurde primär für die Durchführung ortsunabhängiger Messungen entwickelt. Meßwerte werden außerhalb des Schulgebäudes gesammelt und am PC ausgewertet. Damit ist ihr primärer Einsatzbereich der naturwissenschaftliche Unterricht. Die Erfassung von Umweltdaten reicht aber auch unmittelbar in den gesellschaftlichen Bereich und damit in Fragestellungen der IKG herein.

Die Zelle ist primär ein Computer-Interface und Datenerfassungsgerät. Da sie aber über ein eigenes Computersystem mit freier Programmierbarkeit verfügt, kann auch die Hardware oder die Programmierung zum Thema des Unterrichts werden.





## 2. Das System ZELLE

Die Zelle ist ein batteriebetriebenes, intelligentes Interface mit eigenem Prozessor. Das Betriebsprogramm unterstützt ortsunabhängige Messungen an vier Meßkanälen bei einer Länge zwischen einigen Millisekunden und einem Tag. Meßwerte werden im Speicher der Zelle gesammelt und später in den PC übertragen. Die Verbindung beider Geräte erfolgt über die serielle Schnittstelle, so daß prinzipiell jeder Computer als Hostrechner benutzt werden kann. Software ist neben MS-DOS-Rechnern auch für den Macintosh verfügbar.

Das Gesamtsystem besteht aus dem Grundgerät, optionalen Meßmodulen (z.B. Für Temperatur, Lärmpegel und Beleuchtungsstärke) und aus der zugehörigen Software auf unterschiedlichen Anwendungsebenen.

Das Gerät läßt sich auch als direktes Interface für den PC einsetzen. Meß-, Steuer- oder Regelprogramme für den PC verwenden die Zelle dann als Ein-/Ausgabeeinheit.

Die Möglichkeit, mit der Zelle als eigenständigem Computer Programme zu laden und auszuführen, erlaubt die Anwendung als autonome Steuerungseinheit. Einmal geladene Programme bleiben beliebig lange einsatzbereit. Programme können in der Steuersprache Simpel geschrieben werden, für die ein Compiler erhältlich ist.

### 2.1 Das Grundgerät

Die Entwicklung der Zelle ging von einem kompletten, aber kleinen Mikrocomputer aus, der schon ausreichende Meßeingänge und digitale Ein/Ausgänge haben sollte. Außerdem wurde darauf geachtet, den Stromverbrauch bei ausreichender Arbeitsgeschwindigkeit möglichst klein zu halten und gleichzeitig ein möglichst preiswertes System zu bauen.

Die Zelle verfügt über vier Meßkanäle mit eigenen Anschlußbuchsen und einen Sammelanschluß. Verschiedene Meßwandler erlauben die Erfassung nicht-elektrischer Größen. Alle Einstellungen und die Auswertung von Meßergebnissen erfolgen über den PC, der über die RS232-Schnittstelle mit der Zelle verbunden werden kann. Die folgende Abbildung zeigt die Anschlüsse und Bedienelemente:

Abb. 2.1: Anschlüsse und Bedienelemente

1. Batteriefach für vier Mignonzellen (Batterien oder NC-Akkus)
2. Serielle Schnittstelle, verbindet die Zelle über das serielle Kabel mit dem PC
3. Anschluß für externe Stromversorgung, 5V stabilisiert.
4. RESET-Taster, setzt den Speicher der Zelle auf Null zurück oder startet ein Programm der Zelle neu.
5. START-Taster, löst Messungen aus
6. EIN/AUS-Schalter, in Stellung AUS werden Meßdaten und Programme gepuffert.
7. Kontroll-LED, zeigt laufende Messungen an.
8. Sammelanschluß für analoge und digitale Ein/Ausgänge und Betriebsspannung
9. Einzelanschlüsse für die Meßkanäle 1 bis 4

**Das System hat folgende technische Daten:**

|                  |  |
|------------------|--|
| Prozessor:       | 80C39, ROM-lose CMOS-Version aus der 8048-Familie  |
| Betriebssystem:  | Größe 1K, über Jumper J1 und J2 sind bis zu vier Versionen im 8-K-EEPROM wählbar   |
| RAM:             | 8K, davon 6K für Daten, 1K für User-Programme  |
| AD-Wandler:      | 4 Kanäle, 8 Bit, 0...2,5V, geschützt bis +/-10V  |
| Geschwindigkeit: | 1 ms/Messung bis 1 min/Messung   |
| Versorgung:      | Batteriespannung 4V...6V, keine Spannungsregelung<br>ca. 15mA im Betrieb, 1µA in Stellung "AUS"<br>(Datenpufferung)                                |
| Anschlüsse:      | vier Analogeingänge, 0-2,5V, 8 Bit<br>vier digitale Ein/Ausgänge D1...D4<br>ein einzelner digitaler Ein/Ausgang St<br>Systembus (=Expansions-Port) |
| Bedienelemente:  | EIN/AUS-Schalter, RESET-Taster, START-Taster,<br>eine LED zeigt A/D-Wandlungen.  |
| Expansions-Port: | Daten- und Adreßbus für externe Peripherie unter<br>256 Adressen kann über den RAM-Sockel<br>angeschlossen werden.                                 |

Das Betriebssystem der Zelle unterstützt alle Standardaufgaben im Bereich der Serienmessungen. Dem System werden dazu Parameter der Messung (Anzahl, Kanäle, Intervallzeit) mitgeteilt, wonach es die Messung autonom durchführt.

Neben der Anwendung als autonomes Meßdaten-Erfassungssystem kann die Zelle aber auch als normales Interface arbeiten. Alle wichtigen Elemente sind auch direkt vom PC aus steuerbar. Man kann daher die eigentliche Intelligenz eines Programms auch auf die PC-Seite verlegen.

Es gibt zahlreiche Anwendungen, die mit den Standardlösungen des Betriebssystems nicht durchgeführt werden können. Deshalb hat die Zelle die Möglichkeit, Programme nachzuladen.

Nachgeladene Programme der Zelle sind Maschinenprogramme in der Maschinensprache der 8048-Familie. Sie können z.B. mit einem 8048-Cross-Assembler erstellt werden. Einfacher ist der Umgang mit der speziell für die Zelle entwickelten Steuersprache Simpel.

Nach dem Einschalten befindet sich die Zelle in einem von drei möglichen Betriebsmodi:

- Warten auf Kommandos vom PC
- Warten auf Start einer Meßserie
- Starten eines nachgeladenen Programms

Die Betriebsmodi werden durch die Anwendersoftware gesetzt und umgeschaltet.

## 2.2 Die Meßmodule

Zum System Zelle gehören speziell entwickelte Meßmodule. Sie lassen sich mit den Einzelanschlüssen des Grundgerätes verbinden, um nicht-elektrische Größen zu erfassen. Bis zu vier unterschiedliche Meßmodule lassen sich gleichzeitig einsetzen.

Alle Meßmodule sind speziell für den tragbaren Einsatz ausgelegt. Sie erhalten ihre Stromversorgung vom Grundgerät und sind daher auf geringen Leistungsbedarf ausgelegt.

Jedes Meßmodul hat eine spezifische Übertragungskennlinie, die im folgenden als mathematische Beziehung zwischen Meßgröße und Ausgangsspannung angegeben wird. Die Spannung wird von der Zelle in Schritte von 0,01V aufgelöst, so daß sich 256 Quantisierungsstufen zwischen 0 und 2,55V ergeben.

### 2.2.1 Meßmodul Lärm

Das Meßmodul Lärm setzt den Schallpegel in eine Spannung um. Es kann direkt an der Zelle betrieben werden, die auch die Stromversorgung des Moduls mit übernimmt.

Das Modul enthält folgende Funktionsgruppen: Das Kondensatormikrofon gibt eine der Schallintensität proportionale Wechselspannung ab, die durch den Vorverstärker verstärkt wird. Es folgt ein Filter zur Bewertung der Intensität nach der genormten A-Kurve, die der Empfindlichkeit des menschlichen Ohrs angepaßt ist. Das gefilterte Signal durchläuft einen logarithmischen Gleichrichter, so daß die Ausgangsspannung logarithmisch mit der Schallintensität wächst. Sie ist damit proportional zum bewerteten Schallpegel in dB(A). Die Kalibrierung kann durch ein internes Trimpoti erfolgen.

#### Technische Daten:

|                   |                      |
|-------------------|----------------------|
| Betriebsspannung: | 4V...6V              |
| Stromaufnahme:    | ca. 5mA              |
| Meßbereich:       | 3dB(A) .... 100dB(A) |
| Ausgangsspannung: | ca. 1,0V ... 2,5V    |

#### Übertragungsfunktion:

$$L = (100 * (U/V)^2) - 25 \text{ dB(A)}$$

Abb. 2.2: Übertragungskennlinie des Meßmoduls Lärm

### 2.2.2 Meßmodul Temperatur/1

Das Meßmodul Temperatur/1 setzt die Temperatur in eine proportionale Spannung um. Es kann direkt an der Zelle eingesetzt werden und wird von dieser mit Spannung versorgt.

Das Modul besteht aus einem Temperatursensor und einem Meßverstärker. Die Ausgangsspannung steigt mit jedem Grad um 20mV. Die Kalibrierung des Moduls kann durch ein internes Trimpoti erfolgen.

#### Technische Daten:

|                   |                 |
|-------------------|-----------------|
| Betriebsspannung: | 4V...6V         |
| Stromaufnahme:    | ca. 5mA         |
| Meßbereich:       | -25°C ... 102°C |
| Auflösung:        | 0,5K            |
| Ausgangsspannung: | 0V...2,55V      |

#### Übertragungsfunktion:

$$\text{Temperatur} = (100 \cdot (U/V) / 2) - 25 \text{ °C}$$

Abb. 2.3: Kennlinie des Meßmoduls Temperatur/1

### 2.2.3 Temperatur/2 - NTC-Sensor

Das Meßmodul Temperatur/2 setzt die Temperatur in eine Spannung um. Es kann direkt an der Zelle eingesetzt werden und wird von dieser mit Spannung versorgt.

Das Modul besteht aus einem gekapselten NTC-Temperatursensor, der mit einem Festwiderstand einen Spannungsteiler bildet. Die Ausgangsspannung steigt mit der Temperatur nicht-linear an. Eine Linearisierung erfolgt softwaremäßig.

#### Technische Daten:

|                   |                 |
|-------------------|-----------------|
| Betriebsspannung: | 4V...6V         |
| Stromaufnahme     | : ca. 0,1mA     |
| Meßbereich:       | -20°C ... +60°C |
| Auflösung:        | ca. 0,5K        |
| Meßfehler:        | max ca. ±0,5°C  |
| Ausgangsspannung: | ca. 0,3V...2,0V |

Der verwendete NTC-Sensor weist einen Widerstand von 10kΩ bei 25°C auf und garantiert eine hohe Meßgenauigkeit auch ohne Kalibrierung. Er befindet sich in Reihenschaltung mit einem Festwiderstand von 10kΩ an der Referenzspannung der Zelle von 2,55V. Die folgende Tabelle zeigt die Abhängigkeit des Widerstandes und der Ausgangsspannung von der Temperatur:

| Temperatur/°C | Widerstand/kΩ | Spannung/V |
|---------------|---------------|------------|
| -20           | 67,74         | 0,328      |
| -10           | 42,45         | 0,486      |
| 0             | 27,28         | 0,684      |
| 10            | 17,96         | 0,912      |
| 20            | 12,09         | 1,154      |
| 25            | 10,00         | 1,275      |
| 30            | 8,313         | 1,392      |
| 40            | 5,828         | 1,611      |
| 50            | 4,161         | 1,801      |
| 60            | 3,021         | 1,958      |

Die Beziehung zwischen Sensorwiderstand und Meßspannung läßt sich aus den Gesetzen der Reihenschaltung von zwei Widerständen ableiten:  $U = 2,55V * 10k\Omega / (10k\Omega + R)$ . Die Referenzspannung kann hier exakt mit 2,55V eingesetzt werden, weil praktisch nur ein Widerstandsverhältnis gemessen wird, Fehler der Referenzspannung also wieder herausfallen.

Abb. 2.4: Die nichtlineare Kennlinie des NTC-Sensors

### 2.2.4 Meßmodul Licht

Das Meßmodul setzt die Beleuchtungsstärke in eine Spannung um. Es kann direkt an der Zelle eingesetzt werden und wird von dieser mit Spannung versorgt.

Das Modul besteht aus einem Lichtsensor und einem logarithmischen Meßverstärker zur Vergrößerung des Meßbereichs.

Das Maximum der spektralen Empfindlichkeit des Sensors liegt bei 550 nm, also beim Maximum der Empfindlichkeit des menschlichen Auges.

#### Technische Daten:

|                   |                         |
|-------------------|-------------------------|
| Betriebsspannung: | 4V...6V                 |
| Stromaufnahme:    | ca. 5mA                 |
| Meßbereich:       | ca. 1 lx ... 200 000 lx |
| Ausgangsspannung: | 0,1V...2,55V            |

#### Übertragungsfunktion:

$$\text{Beleuchtungsstärke } E = 0,09 \cdot 10^{\frac{(100 \cdot (U/V) / 40)}{10}} \text{ lx}$$

Eine eventuelle Kalibrierung des Moduls erfolgt softwaremäßig durch Korrektur des Faktors 0,09.

Abb. 2.4: Kennlinie für die Messung der Beleuchtungsstärke

Für einige Anwendungen ist die Energiedichte des Lichts wichtiger als die Beleuchtungsstärke. Das Verhältnis zwischen beiden Größen hängt von der spektralen Zusammensetzung des

Lichts zusammen. Beim Licht einer Glühlampe mit der Farbtemperatur 2856K (Normlicht A nach DIN 5033) beträgt der Umrechnungsfaktor:

$$\frac{\text{Beleuchtungsstärke in Lx}}{\text{Energiedichte in W/m}^2} = 21$$

Für Sonnenlicht mit der Farbtemperatur von ca. 6000K gilt der folgende Faktor:

$$\frac{\text{Beleuchtungsstärke in Lx}}{\text{Energiedichte in W/m}^2} = 100$$

Abb. 2.5: Kennlinie für die Messung der Beleuchtungsstärke

Das bedeutet z.B. für volles Sonnenlicht, daß bei einer Energiedichte von 1000 W/m<sup>2</sup> eine Beleuchtungsstärke von 100000 lx gemessen wird. Allgemein kann mit dem Meßmodul die Energiedichte des Sonnenlichts durch folgende Umrechnung bestimmt werden:

$$\text{Energiedichte} = 0,09 / 100 \cdot 10^{\frac{(100 \cdot (U/V) / 40)}{10}} \text{ W/m}^2$$

Abb. 2.6: Kennlinie für die Messung der Energiedichte

### 2.2.5 Meßmodul Gas

#### Beschreibung der Hardware

Das Meßmodul Gas dient zur Untersuchung von Luftverschmutzungen durch brennbare Gase wie z.B. Methan, Kohlenmonoxid, Benzin usw. Es verwendet einen Gas-Sensor auf Halbleiterbasis. Das Halbleitermaterial (dotiertes Zinkoxid) wird elektrisch beheizt und wird bei Anwesenheit eines oxidierendes Gases teilweise reduziert, so daß sein elektrischer Widerstand sich verringert. Der Sensor reagiert auf alle brennbaren bzw. oxydierenden Gase wie Wasserstoff, Kohlenmonoxid, Erdgas, Benzin und andere Kohlenwasserstoffe. Der Sensor wird über die Batterie der Zelle beheizt. Er wird in Reihenschaltung mit einem Festwiderstand von  $30\text{k}\Omega$  an der Referenzspannung der Zelle betrieben, so daß die Meßspannung mit steigender Luftverschmutzung sinkt.

#### Technische Daten:

|                   |                                      |
|-------------------|--------------------------------------|
| Betriebsspannung: | 4V...6V                              |
| Stromaufnahme:    | ca. 150mA (Heizstrom)                |
| Meßbereich:       | abhängig vom Meßgas (siehe Diagramm) |
| Ausgangsspannung: | ca. 0,1V ... 2,4V                    |
| Anwärmzeit:       | ca. 1 min                            |

Die Umrechnung der gemessenen Sensorspannung in den Sensorwiderstand ergibt sich aus der Reihenschaltung zu  $R = 30\text{k}\Omega * (2,55\text{V} - U) / U$ . Der so bestimmte Sensorwiderstand stellt ein Maß für die komplexe Belastung der Luft mit verschiedenen Kohlenwasserstoffen und anderen brennbaren Gasen dar. Bei Belastung mit nur einem bekannten Gas kann die Konzentration nach dem Diagramm in Abb. 2.8 aus dem Sensorwiderstand bestimmt werden.

Abb. 2.7: Der Sensorwiderstand als Funktion der Meßspannung

Abb. 2.8: Die relative Änderung des Sensorwiderstands in Abhängigkeit von der Gaskonzentrationen (Quelle: Datenblatt TGS822 der Firma Figaro)

## 2.3 Software

Die speziell für die Zelle entwickelte Software umfaßt zunächst einige Meßprogramme mit ausgeprägtem Werkzeugcharakter. Die Programme Monitor, Prisma und Zellärm sind von der Benutzerführung und den verwendeten Dateistrukturen her zueinander kompatibel.

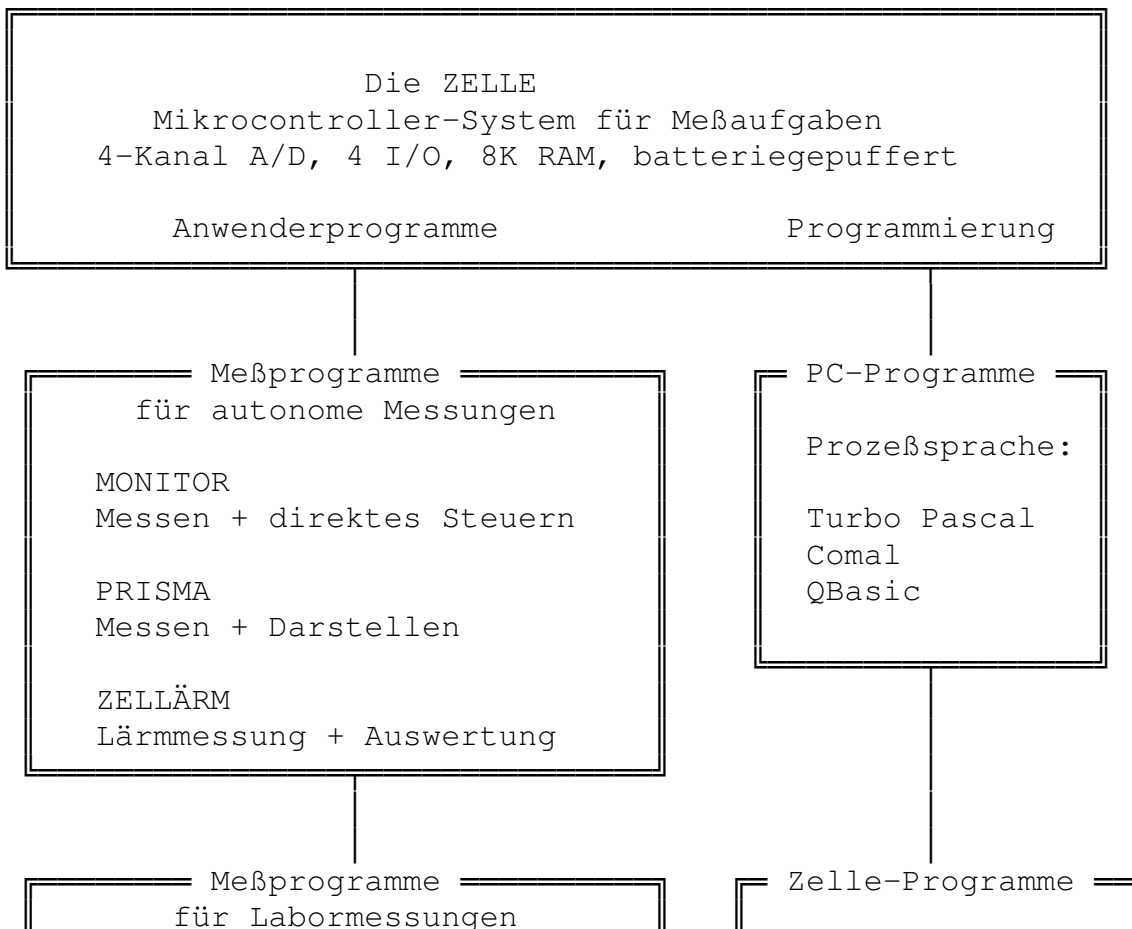
Die Programme Unimeß (Schwerpunkt Chemieunterricht) und Meda (Schwerpunkt Physik) unterstützen mehrere unterschiedliche Interfaces und zeichnen sich durch fachspezifische Auswertemethoden aus.

Auf der Ebene der Programmierwerkzeuge gibt es die Prozeßsprache als Spracherweiterung für Turbo Pascal und Comal. Sie soll den Umgang mit der Zelle auf der Ebene der Hochsprache erleichtern. Es handelt sich um eine Sammlung von Prozeduren und Funktionen, die den Datenaustausch zwischen Zelle und Hostrechner regeln. Für QBasic wird ein Teil der Prozeßsprache in diesem Handbuch vorgestellt.

Autonom in der Zelle als Computer lauffähige Programme können in der Compilersprache Simpel entwickelt werden. Simpel ist eine "kleine" Programmiersprache zur Entwicklung selbständig lauffähiger Programme in der Zelle. Der verwendete Prozessor aus der 8048-Familie ist für Ein-Chip-Anwendungen gedacht und wird normalerweise in 8048-Assembler programmiert. Um einem größeren Kreis von Anwendern die Entwicklung eigener Programme zu ermöglichen, wurde der Simpel-Compiler entwickelt.

Die Sprache bietet einen guten Kompromiß in Bezug auf Speicherbedarf, Ausführungsgeschwindigkeit und Übersichtlichkeit.

Die folgende Übersicht zeigt die für die Zelle verfügbare Software, und zwar auf der einen Seite Anwenderprogramme, auf der anderen Seite Programmierwerkzeuge.





MEDA (Schwerpunkt Physik)  
Messen + Auswerten

UNIMESS (Schwerpunkt Chemie)  
Messen + Auswertungen

8048-Assembler

SIMPEL

Teil A Messungen mit der Zelle



### 3. Einführung in die computergestützte Meßtechnik

Die fortschreitende Automatisierung erfaßt alle Lebensbereiche von Konsumgütern über die industrielle Produktion bis zu Wissenschaft und Forschung. Der Einsatz von Computern hat gerade in der Meßtechnik völlig neue Möglichkeiten geschaffen. So ist z.B. ein dichtes Netz vollautomatischer Umwelt-Meßstationen über die Bundesrepublik verteilt. Die Fülle der anfallenden Daten und ihre Auswertung mit Computern führt zu verbesserten Einschätzungen möglicher Gefahren und erforderlicher Maßnahmen des Umweltschutzes.

Die computergestützte Meßwert-Erfassung und -Verarbeitung bietet sich als Thema einer Unterrichtseinheit an, weil sie einerseits zahlreiche Auswirkungen auf die moderne Gesellschaft hat und weil sie andererseits leicht mit traditionellen Lehrinhalten verbunden werden kann. Welche Messungen im einzelnen durchgeführt werden, hängt sicher vom unterrichtlichen Zusammenhang der Einheit ab. Die hier exemplarisch vorgestellte Temperaturmessung läßt sich leicht durch andere Themen ersetzen.

Ziel einer solchen Unterrichtseinheit muß es sein, einerseits die grundlegenden Prinzipien der Meßdatenverarbeitung kennenzulernen, andererseits aber auch Nutzen, Vorteile und Grenzen dieser Verfahren zu erkennen.

#### 3.1 Grundlagen

Jede elektrische Messung einer nicht-elektrischen Größe benötigt zunächst einen geeigneten Sensor, also ein Gerät, dessen elektrische Eigenschaften sich in Abhängigkeit von der Meßgröße ändern. Im Fall der Temperaturmessung verwendet man z.B. temperaturabhängige Widerstände, die in einer geeigneten Schaltung eine temperaturabhängige Spannung liefern. Die Spannung kann z.B. einen Meßverstärker durchlaufen und dann durch ein Meßgerät angezeigt werden.

Ersetzt man das Meßgerät durch einen Computer, dann ergeben sich neben der reinen Anzeige erweiterte Möglichkeiten der Registrierung und Verarbeitung. Übliche Computer sind allerdings in sich geschlossene Systeme, die Informationen nur über Tastatur und Diskettenlaufwerke aufnehmen können. Deshalb ist ein zusätzliches Interface erforderlich, das den Computer mit der physikalischen Außenwelt verbindet. Kernstück des Interfaces ist der Analog-Digital-Wandler (A/D-Wandler), also ein Bauteil, das die Eingangsspannung in digitale Informationen umsetzt. Die Spannung des Sensors wird also in Zahlen übersetzt, die dann z.B. umgerechnet und angezeigt werden können.

Das Prinzip der Meßwerterfassung mit dem Computer wird im folgenden Bild vereinfacht dargestellt. Der Meßverstärker ist hier weggelassen, weil einerseits viele Sensoren eine genügend hohe Spannung liefern und weil man sich den Verstärker im Sensor integriert denken kann.

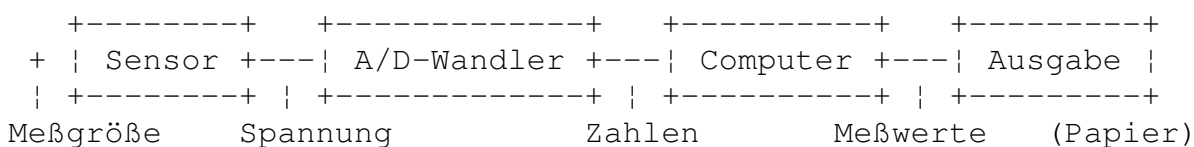


Abb. 3.1: Prinzip der Meßwerterfassung mit dem Computer

Messungen mit Computern müssen nicht grundsätzlich und in jeder Hinsicht konventionellen (analogen) Meßverfahren überlegen sein. Einer der möglichen Schwachpunkte des Systems ist der A/D-Wandler. Er ordnet jeder Eingangsspannung eine Zahl zu. Dabei wird im allgemeinen der gesamte Meßbereich in eine endliche Anzahl von Stufen unterteilt, zwischen denen der reale Meßwert eingeordnet wird. Der Wandler in der Zelle teilt z.B. den Meßbereich von 0V bis 2,55V in 255 Stufen mit einer Breite von jeweils 0,01V ein. Kleinere Unterschiede werden also nicht erkannt. Für eine Temperaturmessung mit dem Meßmodul

Temperatur/1 bedeutet dies, daß der gesamte Meßbereich von  $-25^{\circ}\text{C}$  bis  $+102,5^{\circ}\text{C}$  in 255 Stufen mit einer Auflösung von 0,5 Grad eingeteilt wird. Diese Auflösung genügt z.B. für Wetterbeobachtungen, während sie für andere Messungen unter Umständen zu grob ist.

### Abb. 3.2: Digitalisierung der Temperatur

Eine weitere Einschränkung ergibt sich daraus, daß Messungen mit Computern nicht kontinuierlich sind, sondern punktuell ausgeführt werden. Für die Beobachtung veränderlicher Größen müssen Serien von Einzelmessungen durchgeführt werden. Das Zeitintervall zwischen Einzelmessungen kann in weiten Grenzen festgelegt werden. Dabei muß allerdings die tatsächliche Änderungsgeschwindigkeit der Meßgröße berücksichtigt werden.

Soll bei einer Messung der gesamte Verlauf einer Größe mit der erreichbaren Auflösung genau erfaßt werden, dann darf sich die Meßgröße zwischen zwei Meßpunkten um maximal eine Quantisierungsstufe ändern. Während der gesamten Meßdauer müssen also entsprechend viele Einzelmessungen durchgeführt werden. Da aber jedes Meßergebnis als ein Zahlenwert digital gespeichert werden muß, ergibt sich ein großer Speicherbedarf. Deshalb ist man oft zu Kompromissen gezwungen: Das Meßintervall wird verlängert, und Zwischenwerte werden interpoliert.

Die Interpolation von Meßwerten findet da ihre Grenzen, wo zu schnelle Meßwertschwankungen auftreten. Tastet man eine schnell veränderliche Größe durch Einzelmessungen zu langsam ab, dann treten Fehlmessungen auf. Nach der Abtast-Theorie muß die Abtastfrequenz mindestens doppelt so hoch sein, wie die höchste im Meßsignal vorkommende Einzelfrequenz. Da die Zelle maximal 1000 Messungen in einer Sekunde ausführen kann, liegt die theoretische Grenzfrequenz des Eingangssignals bei 500 Hz. Sprachsignals sind damit z.B. nicht mehr erfaßbar.

### Abb. 3.3: Abtastfehler: Die Messung gibt nicht mehr den tatsächlichen Spannungsverlauf wieder.

In jedem Einzelfall müssen also die Auflösung, die Zeitintervalle und der Speicherbedarf einer Messung bedacht werden. Die Zelle verfügt über einen Speicher für ca. 6000 Meßwerte. Bei der höchsten Meßgeschwindigkeit von 1000 Einzelmessungen/Sekunde beträgt die Dauer der Messung also maximal 6 Sekunden. Da sich die Meßintervalle jedoch bis auf eine Minute vergrößern lassen, können auch Meßzeiten von bis zu vier Tagen erreicht werden. Allerdings können dann Ereignisse unerkannt bleiben, die zwischen zwei Einzelmessungen im Abstand von einer Minute fallen.

Die folgende Unterrichtsreihe verwendet die Zelle zunächst mit dem Monitorprogramm. Dieses einfache Programm wird zusammen mit der Zelle ausgeliefert und verursacht damit keine zusätzlichen Kosten. Fast alle wichtigen Grundversuche sind mit dem Monitorprogramm durchführbar. Der geringere Komfort kann bei einer Einführung didaktische Vorteile mit sich bringen, weil der Anwender zu einer eigenen Interpretation der unverarbeiteten Daten gezwungen wird.

Die Einführung erfolgt hier mit dem Temperatursensor, weil dieser eine lineare Übertragungskennlinie hat und einfache Versuche ermöglicht.

### 3.2 Temperaturmessung mit dem Monitorprogramm

Das Monitorprogramm eignet sich für Messungen mit beliebigen Sensoren, wobei allerdings grundsätzlich die Spannung am Meßeingang der Zelle angezeigt wird. Eine Umrechnung in die tatsächlich gemessene Größe muß vom Anwender geleistet werden und erfordert z.B. den Einsatz eines externen Auswerteprogramms.

Die grundsätzliche Funktion des Meßmoduls Temperatur/1 wird bei direkter Messung im Menü Ein/Aus deutlich. Am verwendeten Eingang (z.B. Kanal 1) steht eine Spannung, die sich in Abhängigkeit von der Temperatur ändert. Die Spannung wird mit einer Auflösung von 0,01V angezeigt. Oft wird eine zwischen zwei Stufen (z.B. 1,01V und 1,02V) wechselnde Anzeige beobachtet. Dieses typische Verhalten jedes A/D-Wandlers weist darauf hin, daß die Meßspannung gerade auf der Grenze zwischen zwei Quantisierungsstufen liegt.

Abb. 3.4: Direktanzeige der Spannung: Am ersten Analogeingang ( $U_{\text{Ein}(0)}$ ) wird die Sensorspannung gemessen.

Die direkte Beobachtung der Spannung verdeutlicht die grundsätzliche Funktion jedes Sensors: Die Meßgröße wird in eine elektrische Größe (Spannung) umgesetzt, damit sie über den Computer gemessen werden kann. Die gemessene Spannung muß in einem zweiten Schritt wieder in eine Temperatur umgerechnet werden. Im Unterricht kann nun entweder zuerst die Übertragungsfunktion ermittelt werden, oder es wird mit der Soll-Kennlinie des Sensors gearbeitet.

Da der Sensor eine lineare Übertragungsfunktion hat, genügen zwei bekannte Meßpunkte um die Übertragungsfunktion zu gewinnen. Durch zwei Vergleichsmessungen mit einem genauen Thermometer kann der Nullpunkt und die Steigung der Kennlinie ermittelt werden. Das Ergebnis läßt sich als Gleichung oder als Diagramm darstellen.

#### Soll-Übertragungsfunktion:

$$\text{Temperatur} = (50 * \text{Spannung} / \text{V} - 25) \text{ } ^\circ\text{C}$$

Entscheidet man sich dafür, auf eine eigene Kalibrierung des Sensors zu verzichten, dann kann man das fertige Diagramm der Übertragungsfunktion des Sensors direkt zur Interpretation der Meßwerte einsetzen. Typische, einführende Arbeitsaufträge wären:

- Miß die Sensorspannung bei der Umgebungstemperatur und bei Berührung des Sensors.
- Lies die Sensorspannungen für Temperaturen von 20°C, und 25°C aus dem Diagramm.
- Finde die zugehörigen Temperaturen der Meßspannung 1V, 1,5V und 2V.
- Vergleiche die ermittelte Temperatur mit dem Meßwert eines genauen Thermometers.

Abb. 3.5: Aus der Sensorkennlinie kann für die Sensorspannung 0,91V die Temperatur 20,5°C ermittelt werden

Die direkte Messung der Sensorspannung mit der Zelle und dem Meßmodul Temperatur ist der Messung mit einem direktanzeigenden Thermometer an Übersichtlichkeit und Auflösung in den meisten Fällen weit unterlegen. Außerdem ist die Genauigkeit der digitalen Messung nicht von vornherein gleich der Auflösung, sondern es treten neben den Quantisierungsstufen zusätzliche Toleranzen auf. Zwei mögliche Fehler können sich addieren: Zum einen hat der A/D- Wandler einen Meßfehler, und zum anderen ist auch die Temperatur/Spannungs-Umsetzung des Sensors fehlerbehaftet. Um die Grenzen der Meßgenauigkeit bewußt zu machen, empfiehlt es sich, nach Möglichkeit genaue Vergleichsmessungen durchzuführen. Damit kann

die Fehlerbestimmung und die Korrektur der angegebenen Übertragungskennlinie thematisiert werden.

### 3.2.1 Meßreihen

Der Vorteil des Computereinsatzes liegt in erster Linie in der Möglichkeit, Meßserien automatisch durchzuführen und auszuwerten. Mit dem Monitorprogramm werden dazu die gewünschten Parameter der Messung eingestellt:

- Kanal 1..4: Auswahl der aktiven Kanäle
- Blocklänge: Anzahl der Einzelmessungen
- Meßintervall: Zeit zwischen zwei Einzelmessungen

Ein Beispiel für die Einstellung einer Kurzzeitmessung von einer Minute ist:

- Kanal 1, Blocklänge=1000, Meßintervall=60ms
- (Meßzeit =  $1000 * 60\text{ms} = 60\text{s} = 1\text{min}$ )

Mit dieser Einstellung lassen sich maximal 6 Messungen (=Meßblöcke) nacheinander auslösen, da etwa 6000 Einzelmeßwerte in der Zelle gespeichert werden können. Jede Messung wird mit der Starttaste der Zelle ausgelöst. Die laufende Messung kann am Blinken der LED erkannt werden. Bei jeder Einzelmessung blinkt die LED einmal kurz auf, was zu einem Flackern oder mehr oder weniger kräftigen Dauerleuchten bei schnellen Messungen führt. So ist auch das Ende eines Meßblocks erkennbar. Eine Schülergruppe kann mit der obigen Einstellung bis zu sechs unterschiedliche Situationen untersuchen, also z.B.:

- Erwärmung des Sensors durch Handwärme
- Abkühlung des Sensors
- Abkühlung des befeuchteten Sensors
- Erwärmung an einem Heizkörper
- Abkühlung einer Kaffeetasse
- Erwärmung im Sonnenlicht

Nach der Messung müssen die Meßwerte von der Zelle in den PC übertragen werden. Für den Auslesevorgang muß angegeben werden, wieviele Blöcke übertragen werden sollen. Die Anzahl richtet sich nach der Zahl der tatsächlich durchgeführten Messungen.

Nach dem Einlesen lassen sich die Meßwerte am Bildschirm betrachten. Eine Möglichkeit ist dabei die Ausgabe in Listenform.

Abb. 3.6: Ausschnitt aus einer Liste von Meßwerten

Für jeden Zeitpunkt wird hier der erfaßte Meßwert ausgegeben. Bei relativ umfangreichen Messungen mit mehr als 100 Meßpunkten zeigt es sich, daß eine übermäßige Datenflut für eine Interpretation der Meßwerte eher hinderlich ist. Eine mögliche Übung könnte es sein, jeden zehnten Meßwert einer ausgedruckten Werteliste in ein Diagramm zu übertragen, so daß der Gesamtverlauf deutlich wird. (Um eine Liste mit dem Monitorprogramm ausdrucken zu können, müssen die Meßwerte zunächst mit Datei/Konvertieren in eine Textdatei geschrieben werden.)

Gerade bei Temperaturmessungen zeigt sich oft eine sehr geringe Änderungsgeschwindigkeit der Meßwerte. In diesem Zusammenhang ist eine Diskussion der optimalen Anzahl von Meßpunkten sinnvoll. Je nach Meßdauer und je nach Art der vorkommenden Schwankungen kann die Anzahl drastisch reduziert werden. Das Meßintervall braucht nicht wesentlich kürzer zu sein als die Zeit, in der sich der Meßwert um eine Quantisierungsstufe ändern kann.

Die zeitaufwendige Erstellung eines Diagramms von Hand führt zum Wunsch nach einer automatischen Erstellung von Meßwertdiagrammen durch den Computer. Gerade die schematischen Tätigkeiten lassen sich

einfach und nutzbringend automatisieren. Das Monitorprogramm stellt eine einfache grafische Ausgabe bereit. Die Meßpunkte lassen sich bei Einkanalmessungen mit Linien verbinden und so linear interpolieren. Die Zeitachse des Diagramms ist in Prozent der Meßzeit beschriftet.

Abb. 3.7: Ein Diagramm des Monitorprogramms

Die Diskussion der erhaltenen Meßergebnisse erfordert eine Interpretation der gemessenen Spannungen. Eine mögliche Lösung ist es, die Meßwerte in Diagrammform auszudrucken und dann eine neue Achsenbeschriftung anzubringen.

Abb. 3.8: Ein selbstbeschriftetes Diagramm

Der Einsatz von Computern bringt eine erhebliche Vereinfachung bei Langzeitmessungen über mehrere Stunden oder Tage, weil Daten vollautomatisch erfaßt werden können. Mit dem Einsatz der Zelle werden Langzeitmessungen besonders einfach, weil einerseits kein PC durch die Messung im Dauerbetrieb arbeiten muß und weil sich der Meßort flexibel wählen läßt. Autonome Langzeitmessungen können mit dem Monitorprogramm ausgeführt werden. Eine 24-Stunden-Messung kann z.B. mit den folgenden Parametern gestartet werden:

Kanal 1, Blocklänge=6000, Meßintervall=14400ms  
(Meßzeit =  $6000 * 14400\text{ms} = 86400\text{s} = 24\text{h}$ )

Mögliche Fragestellungen sind:

- Welchen Verlauf zeigt die Außentemperatur über einen ganzen Tag?
- Welchen Verlauf nimmt die Raumtemperatur in der Schule an Wochentagen und an Wochenenden?
- Wie ändert sich die Temperatur im Klassenraum bei Anwesenheit von 30 Schülern?

Die Messung der Temperatur stellt nur eine mögliche Anwendung dar. Genauso lassen sich andere Sensoren exemplarisch einsetzen, um Grundlagen der computergestützten Meßwerterfassung zu thematisieren. Sinnvoll ist auch die direkte Spannungsmessung oder der gleichzeitige Einsatz mehrerer Sensoren.



### 3.3 Auswertung mit WORKS

Weiterführende Auswertungen und eine ansprechende Präsentation der Meßwerte können z.B. mit dem in Schulen verbreiteten Programm WORKS erreicht werden. Daten vom Monitorprogramm müssen dazu erst einmal exportiert werden. Unter dem Menüpunkt Datei/Konvertieren wird eine Datei im Textformat erzeugt, die in die Tabellenkalkulation von WORKS eingelesen werden kann. Die exportierte Tabelle enthält eine Spalte für die Meßzeit in ms und eine Spalte für jeden Meßkanal.

Die folgende Beispielmessung erfaßte die Temperatur an der Unterseite einer Kaffeetasse. Die Meßdaten wurden mit dem Monitorprogramm erfaßt und dann zunächst zur Sicherheit abgespeichert. Die Datei KAFFEE.DAT verwendet das Datenformat der Zelle und hat eine Länge von ca. sechs Kilobyte. Durch das Konvertieren entsteht die Datei KAFFEE.TAB im Tabellenformat. Diese von WORKS lesbare Datei sollte in das WORKS-Verzeichnis kopiert und in KAFFEE.WKS umbenannt werden. Das erste Einlesen in WORKS benötigt etwas mehr Zeit, weil die Daten in ein internes Format gebracht werden müssen.

Abb. 3.9: Die konvertierten Daten in KAFFEE.TAB

Die Meßwerte sind hier als Rohdaten gespeichert, also als ganze Zahlen im Bereich 0 bis 255. Der größte Wert 255 entspricht der größten Meßspannung von 2,55V. In dieser Form lassen sich die Daten von der Tabellenkalkulation in Works direkt einlesen.

Abb. 3.10: Die in WORKS eingelesenen Daten

Die beiden Spalten der Tabelle müssen nun zunächst umgerechnet werden. Spalte 1 enthält die Zeit in Millisekunden. Für die endgültige Darstellung ist die Zeiteinheit Minuten günstiger. Deshalb wird in Spalte 3 eine neue Zeitreihe durch die folgende Umrechnung erzeugt.

$$=ZS (-2) / 60000$$

Diese Formel muß in die gesamte Spalte kopiert werden. In ähnlicher Weise läßt sich in Spalte 4 eine umgerechnete Datenreihe der Temperatur in °C erzeugen. Die Formel dazu enthält die Übertragungsfunktion des Temperatursensors:

$$=ZS (-2) / 2-25$$

Die gesamte Tabelle enthält nun in den Spalten 3 und 4 das umgerechnete Meßergebnis.

Abb. 3.11: Die bearbeitete Tabelle

Diese Daten lassen sich sehr einfach grafisch darstellen, indem die Spalten 3 und 4 jeweils markiert und der X- bzw. Y-Datenreihe des Diagramms zugeordnet werden. Zusätzlich läßt sich der Diagrammtyp, Achsenbeschriftungen, Gitterlinien und eine Bildüberschrift eingeben.

Abb. 3.12: Die grafische Ausgabe der Meßwerte

Die Auswertung der Daten mit einer Tabellenkalkulation bringt zahlreiche zusätzliche Möglichkeiten. So ist es z.B. sehr einfach, Teilbereiche der Messung auszuwählen und darzustellen. Außerdem lassen sich bei Bedarf auch komplexere mathematische Auswerteverfahren durchführen.

### 3.4 Messen und beobachten mit PRISMA

Das Programm Prisma erleichtert Messungen mit beliebigen Sensoren. Meßwerte werden hier automatisch interpretiert und mit den richtigen Einheiten angezeigt. Wurde in einer Unterrichtsreihe zunächst eine Einführung mit dem Monitorprogramm gegeben, dann lassen sich anschaulich verschiedene Stufen der Automatisierung von Meßwerterfassungen demonstrieren. Zahlreiche Arbeitsschritte, die bisher mühsame Kleinarbeit erforderten, werden nun vom Programm übernommen. Prisma eignet sich daher besonders für umfangreichere Untersuchungen.

Je nach Situation der Lerngruppe und Sachzusammenhang einer Unterrichtsreihe kann es im Sinne einer didaktischen Reduktion sinnvoll sein, die eher technischen Einzelheiten der Meßwerterfassung wie Digitalisierung, Übertragungskennlinien und Fehlerdiskussion von der näheren Betrachtung auszuschließen. Dann bietet Prisma sich für einen einfachen und möglichst handlungsorientierten Einstieg in die computergestützte Meßtechnik an.

Prisma verwendet Definitionsdateien für jeden Sensor. Durch Zuordnung eines Meßkanals und eines Sensors werden alle Meßwerte in die erfaßte Größe umgerechnet. Die Definitionsdateien enthalten Zuordnungstabellen, die durch zusätzliche Hilfsprogramme softwaremäßig kalibriert werden können. Eine eventuell erforderliche Kalibrierung braucht nur einmal durchgeführt zu werden, so daß man auf eine Thematisierung von Meßfehlern weitgehend verzichten kann.

Der erste Einstieg kann mit Hilfe der direktanzeigenden Messung erfolgen. Der Temperaturfühler wird z.B. an den ersten Meßkanal gesteckt, und entsprechend wird die Definitionsdatei TEMP1.DEF dem ersten Kanal zugeordnet. Mit der Wahl des Menüpunkts Messen/Multimeter erhält man dann eine Großanzeige des aktuellen Meßwerts.

Abb. 3.13: Großanzeige der Temperatur in Prisma

Prisma speichert alle erfaßten Meßwerte während der direkten Messung. Die Meßdauer kann eingestellt werden. Mit der Standardeinstellung 10 Sekunden bleibt nach kurzer Zeit der letzte Meßwert in der Anzeige stehen. Sollen Temperaturänderungen beobachtet werden, dann muß vor der Messung eine geeignete Meßdauer festgelegt werden. Prisma bietet eine Reihe von Standard-Meßzeiten, so daß auf die weiterhin mögliche freie Einstellung meist verzichtet werden kann.

Abb. 3.14: Menü der einstellbaren Meßzeiten

Die erfaßten Meßwerte können dann z.B. als Liste ausgegeben werden. Eine direkte Ausgabe an einen Drucker oder in eine Datei ist möglich. So lassen sich Meßwerte für die spätere Weiterverarbeitung festhalten.

Abb. 3.15: Ausgabe der erfaßten Meßwerte in Listenform

Die erfaßten Meßwerte lassen sich auch grafisch darstellen. Die Achsenbeschriftungen passen sich dabei automatisch den aktuellen Einstellungen an. Zusätzlich kann eine Bildüberschrift eingegeben werden.

Abb. 3.16: Ein Diagramm in Prisma

Im Gegensatz zum Monitorprogramm, das nur eine nachträgliche grafische Ausgabe ermöglicht, bietet Prisma auch die Möglichkeit, laufende Messungen gleichzeitig grafisch darzustellen. Online-Messungen

bleiben besonders gut überschaubar, weil Änderungen sofort in ihrer Tendenz sichtbar werden. Das unterstützt die Möglichkeiten direkter Beobachtungen. Eine mögliche Aufgabe wäre:

- Untersuche die Abkühlung eines auf der Heizung erwärmten Geldstücks
  - a) in ruhender Luft,
  - b) wenn es angeblasen wird,
  - c) wenn es befeuchtet wird,
  - d) wenn es befeuchtet wurde und angeblasen wird.

Mehrere Messungen können dabei in einem Diagramm übereinander dargestellt werden. Auch bei etwas unterschiedlicher Anfangstemperatur lassen sich die unterschiedlichen Änderungsraten gut erkennen. Die Meßergebnisse erlauben die Diskussion der physikalischen Hintergründe: Wärmeleitung, Wärmestrahlung, Konvektion und Verdunstungswärme.

Abb. 3.17: Vergleichende Messungen in einem Diagramm

Neben der direkten Messung besteht auch mit Prisma die Möglichkeit von offline-Messungen. Die Zelle wird dabei zunächst am PC mit den gewünschten Parametern versehen, dann getrennt und zum Meßort gebracht. Nach erfolgter Messung muß die Zelle wieder mit dem PC verbunden werden, um die Meßwerte auszulesen. Prisma liest dabei auch die Zuordnung der Sensoren, so daß die korrekte Darstellung der Meßwerte sichergestellt ist.

Autonome Messungen mit der Zelle vereinfachen die Durchführung von Langzeitmessungen und ermöglichen Messungen außerhalb des Schulgebäudes. Anwendungsbeispiele für autonome Messungen wurden schon weiter oben genannt.

Eine kritische Wertung der bisherigen Versuche kann zu folgenden Ergebnissen kommen:

- Der Computer ist bei einfachen Messungen einem herkömmlichen Thermometer nicht überlegen.
- Jeder der bisherigen Versuche kann auch mit Thermometern durchgeführt und von Hand ausgewertet werden.
- Der Computer kann den Menschen von langwierigen und fehlerträchtigen Routinearbeiten entlasten.
- Mit Computern werden Beobachtungen unterstützt und vereinfacht, so daß mehr Zeit für die Deutung der Ergebnisse bleibt.

Ein weiter Grund, Computer für Messungen einzusetzen ist, daß die Meßwerte mit Hilfe des Rechners ausgewertet werden können. Prisma bietet zwar nur sehr einfache Werkzeuge, die jedoch das Prinzip verdeutlichen können. So lassen sich z.B. für eine Messung Minimum, Maximum und Mittelwert bestimmen.

Abb. 3.18: Auswertung der Extremwerte und des Mittelwerts

Eine sehr anwendungsbezogene Unterrichtsreihe zur Meßwerterfassung mit Computern führt ohne viele theoretische Vorüberlegungen zu praktischen Fragestellungen. Damit ergibt sich die Möglichkeit, auch komplexere Untersuchungen mit zwei oder mehr Sensoren durchzuführen. Prisma unterstützt die gleichzeitige Darstellung von bis zu vier Meßkanälen mit unterschiedlichen Sensoren. So lassen sich gegenseitige Abhängigkeiten untersuchen.

Einige Sensoren der Zelle, wie z.B. der Temperatursensor/2 für Eintauchmessungen oder der Gassensor verwenden eine nichtlineare Übertragungskennlinie. Während eine Interpretation der Meßwerte mit einfacheren Programmen sehr schwierig ist, sind diese Sensoren durch die automatische Meßwertumformung in Prisma genauso einfach handhabbar wie lineare Sensoren.

### 3.5 Messen und auswerten mit MEDA

Das Meß- und Auswerteprogramm Meda unterstützt verschiedene Interface-Systeme, darunter auch die Zelle. Das Programm vereint in sich die Vorteile einer Tabellenkalkulation und eines Meßwert-Erfassungssystems. Es unterstützt zahlreiche Auswerteverfahren, so daß es in erster Linie für genaue quantitative Untersuchungen eingesetzt wird.

Meßwerte werden immer gleichzeitig als Tabelle und als Diagramm geführt. Die erfaßte Meßgröße ist immer die Spannung. Andere Meßgrößen lassen sich durch eine nachträgliche Umrechnung der Meßwerte darstellen.

Meda verwendet die Zelle als stationäres Interface für eine relativ langsame Meßwernerfassung. Übliche Meßintervalle liegen im Bereich einer Sekunde bis zu Minuten. Typische Laborversuche können dabei direkt am Bildschirm verfolgt werden. Die Vorteile der umfangreichen Auswertung können auch für Offline-Messungen und schnelle Messungen genutzt werden, wenn Meßdaten vom Monitorprogramm über eine Datei importiert werden.

Die Dauer einer Messung braucht in Meda nicht vorher bekannt zu sein, weil die Zeitachse des Meßdiagramms sich laufend selbst anpaßt. Vor einer Messung muß das Interface und die Anzahl der Meßkanäle gewählt werden. Außerdem ist das Meßintervall einzustellen.

Abb. 3.19: Meda-Einstellungsmenü für direkte Messungen

Ein typischer Versuch zur Temperaturmessung kann z.B. die Erwärmung und Abkühlung eines Metallstücks untersuchen. Ein kleines Messingplättchen wird mit Stativmaterial in guten thermischen Kontakt mit dem Temperatursensor gebracht. Die Messung soll bei Umgebungstemperatur gestartet werden. Das Metall wird dann mit einem Feuerzeug, einer Kerze oder einem LötKolben bis auf etwa 50 Grad erwärmt, um dann langsam wieder abzukühlen.

Abb. 3.20: Tabelle und Diagramm einer Messung

Um die gemessenen Sensorspannungen in die Temperatur umzurechnen, wird der erfaßte Datensatz mit der Funktion Rechnen/Y-Achse umgerechnet.

Die Umrechnungsgleichung lautet:  $Y = Y * 50 - 25$

Unter Graph/Diagramm kann eine neue Beschriftung für die Y-Achse eingegeben werden.

Abb. 3.21: Die in Temperaturen umgerechneten Meßwerte

Die eigentliche Auswertung der Messung erfolgt abschnittsweise für die Erwärmung und die Abkühlung. Zuvor müssen die Meßwerte jedoch abgespeichert werden, weil durch das Ausschneiden von Teilbereichen der Datensatz verändert wird. Ein Abschnitt der Messung kann unter Maussteuerung ausgeschnitten werden, wobei der gewählte Teilbereich zugleich in der Tabelle erscheint. Zunächst wird die Erwärmungsphase betrachtet.

Abb. 3.22: Der Teilbereich der Erwärmung

Der Bereich zeigt in erster Näherung eine Gerade. Die Funktion Rechnen/Regressionen zeigt eine gute Übereinstimmung für die lineare Regression. Die berechnete Geradengleichung zeigt eine Steigung der

Temperatur um 0,393 Grad pro Sekunde.

Abb. 3.23: Das Regressionsmenü

Im Menüpunkt Graph/Diagramm kann die lineare Regression gewählt werden, um eine Ausgleichsgerade mit einzuzeichnen. Dies erleichtert eine optische Einschätzung der Abweichungen vom Geradenverlauf.

Die Abkühlungsphase der Messung zeigt eine asymptotische Annäherung an die Umgebungstemperatur. Es ist daher sinnvoll, den ausgeschnittenen Datenbereich auf die Umgebungstemperatur zu normieren, indem durch eine Umrechnung der Y-Werte die Umgebungstemperatur von 22°C subtrahiert wird. Das Diagramm zeigt dann die Temperaturdifferenz zur Umgebungstemperatur.

Abb. 3.24: Die umgerechnete Abkühlungskurve

Die Regressionsrechnung zeigt nun eine gute Übereinstimmung für die exponentielle Funktion. Nach der Theorie müßte bei reiner Wärmeleitung ein exakter exponentieller Temperaturverlauf beobachtet werden. Die mit eingezeichnete exponentielle Regressionskurve zeigt deutliche Abweichungen der tatsächlichen Meßwerte. Der steilere Temperaturabfall bei höheren Temperaturen ist auf den Einfluß von Wärmestrahlung und Konvektion zurückzuführen.

Abb. 3.25: Gleichzeitiges Einzeichnen der Regressionsgleichung

### 3.5.1 Verwendung eines nichtlinearen Sensors

Die hier gezeigte Messung verwendete den Temperatursensor 1 der Zelle mit einer linearen Übertragungskennlinie. Für Temperaturmessungen in Flüssigkeiten ist der NTC-Sensor besser geeignet. Ein Nachteil ist seine nichtlineare Kennlinie. Im mittleren Teil zwischen etwa 0°C und 45° kann sie jedoch näherungsweise als linear betrachtet werden. In die Kennlinie können Schüler und Schülerinnen eine geeignete Ausgleichsgerade einzeichnen und dann Steigung und den Durchstoßpunkt durch die Y-Achse bestimmen. Abb. 3.26 zeigt ein bearbeitetes Diagramm. Die Geradengleichung wurde mit

$$y = 0,43 x - 30$$

ermittelt. Diese Funktion kann nun benutzt werden, um Meßwerte des Sensors in die Temperatur umzurechnen.

Abb. 3.26: Ausgleichsgerade in der Kennlinie des NTC-Sensors

Die grafische Bestimmung der Näherungsgeraden läßt sich auch mit dem Computers durchführen. Meda ist ein geeignetes Werkzeug zum Auffinden einer linearen Näherungsfunktion. Dazu werden zunächst die theoretischen Wertepaare Spannung-Temperatur (s. Kap. 2.3.3) in die Tabelle eingetragen. Läßt man sich die lineare Regression mit eintragen, dann läßt sich durch Ausschneiden von Kurvenabschnitten die Linearisierung optimieren. Abb. 3.27 zeigt das Ergebnis für den Bereich 0°C bis 40°C.

Abb. 3.27: Lineare Regression mit Meda

Für den Bereich 0°C bis 40°C läßt sich also in guter Näherung die lineare Funktion

$$y = 0.441 * x - 30.556$$

verwenden.

Meßwerte des NTC-Sensors können mit dieser Funktion in Temperaturen umgerechnet werden. Dazu wird unter dem Menüpunkt Rechnen/Y-Achse diese Funktion eingegeben. Abb. 3.28 zeigt eine so mit Meda durchgeführte Messung mit dem NTC-Sensor.

Abb. 3.28: Temperaturmessung mit dem NTC-Sensor

### 3.6 Experimentieren mit UNIMESS

Das Programmpaket Unimess des Arbeitskreises Computer im Chemieunterricht enthält die Meßwerterfassung für sehr viele unterschiedliche Interfaces und unterstützt zahlreiche Auswertemethoden, die schwerpunktmäßig im Chemieunterricht eingesetzt werden. Die Zelle wird im Direktmodus als Analoginterface mit einem bis zu vier Eingangskanälen unterstützt.

Durch Einträge in das zugehörige Installationsprogramm lassen sich sehr viele Sensoren an das Programm anpassen. Dabei muß das verwendete Interface, die Schnittstelle, der Eingangskanal, der Meßbereich, die Einheit, ein Umrechnungsfaktor und ein eventueller Offset angegeben werden. Während der Messung erhält der Anwender immer die richtigen Meßwerte und Einheiten. Im Normalfall ist die Installation von Sensoren Aufgabe des Lehrers. Wurde allerdings in einem Kurs intensiv mit Übertragungskennlinien verschiedener Sensoren gearbeitet, dann bietet sich die Installation als Anwendung und Vertiefung im Unterricht an.

Die folgende Tabelle zeigt Einträge für die Installation der Zelle als Meßgerät für Spannungen und Temperaturen mit dem Temperatursensor 1 (0 bis 100°C) und dem Temperatursensor 2 (NTC-Sensor, hier 0 bis 50°C). Für den NTC-Sensor wird die angenäherte lineare Funktion aus Kap. 3.5 benutzt.

Abb. 3.29: Das Installationsmenü zu Unimess

Die eigentliche Messung erfolgt für Einkanalmessungen mit dem Programm Unimess 1. Nach der Wahl der Meßgröße kann die Messung in Großziffern oder als Diagramm mit Linien oder Punkten gewählt werden. Die Einzelmessungen erfolgen entweder zeitgesteuert nach einer gewählten Interwallzeit oder als einzeln ausgelöste Messungen. Einzelauslösungen dienen im Chemieunterricht z.B. den Titrationsmessungen, bei denen zwischen den Messungen jeweils ein bestimmtes Volumen einer Flüssigkeit zugegeben wird.

Bei zeitgesteuerten Messungen muß eine Gesamtmeßdauer mit angegeben werden. Sie kann jedoch überschritten werden, ohne daß die zusätzlichen Meßwerte verlorengehen. Die folgende Messung zeigt die Erwärmung des NTC-Sensors durch Handwärme.

Abb. 3.30: Ein Meßergebnis mit Unimess

Die Meßwerte können editiert, gespeichert, ausgedruckt oder in einem Diagramm verarbeitet werden. Das Diagramm kann beschriftet und ausgedruckt werden. Für umfangreiche Auswertungen müssen die Daten gespeichert und dann mit dem Auswerteprogramm Uni-Aus weiterverarbeitet werden.





## 4. Lärmmessungen

Die Lärmbelastung am Arbeitsplatz und am Wohnort stellt heute eine ernstzunehmende Gefährdung dar. Auswirkungen des Lärms auf Gesundheit und Leistungsfähigkeit, Entstehung und Vermeidung von Lärm, Lärmschutzmaßnahmen und umweltpolitische Konzepte sind daher relevante Themen im Unterricht geworden. Schon seit einigen Jahren werden Unterrichtseinheiten zum Thema Lärm durchgeführt (vgl. [2]). Mit der Zelle und ihrem Lärm-Meßmodul stehen tragbare, netzunabhängige Geräte zur Verfügung, die sich problemlos auch außerhalb der Schule und für Langzeitmessungen einsetzen lassen.

### 4.1 Meßtechnische Grundlagen

Lärmpegel werden in dB(A) gemessen. Dies entspricht etwa der veralteten Einheit Phon. Bei 0 dB(A) liegt die Hörschwelle des Menschen. Ein Motorrad erzeugt in einem Meter Abstand ca. 80 dB(A), und die Schmerzgrenze liegt bei etwa 120 dB(A).

Ein Pegel ist immer der Logarithmus des Verhältnisses einer Meßgröße zu einer Vergleichsgröße. Er wird meist in Dezibel (dB) angegeben. Die Angabe "Der Pegel liegt um 10 dB höher" bedeutet, daß die 10-fache Leistung vorliegt. Allgemein gilt für einen Pegel L:

$$L = 10 \log P/P_0 ; P = \text{Leistung}, P_0 = \text{Vergleichsleistung}$$

Einer Steigerung der Leistung um das 100-fache entspricht also eine Verstärkung von 20 dB. Jeder Pegel läßt sich in ein Leistungsverhältnis umrechnen.

| Pegel | Leistungsverhältnis |
|-------|---------------------|
| 0 dB  | 1                   |
| 1 dB  | 1,26                |
| 3 dB  | ca. 2               |
| 10 dB | 10                  |
| 20 dB | 100                 |
| 30 dB | 1 000               |
| 40 dB | 10 000              |
|       | u. s. w.            |

Für die Bewertung von Schallpegeln im Zusammenhang mit der Belastung von Menschen hat man die menschliche Hörschwelle als Vergleichspegel gewählt. Diese ist aber sehr stark von der Frequenz abhängig. Bei etwa 1000 Hz ist das Ohr am empfindlichsten, darunter und darüber nimmt die Empfindlichkeit ab. Ein Meßgerät für Schallpegel muß Geräusche nach ihrer Frequenz so bewerten wie das menschliche Ohr. Der geforderte Frequenzgang ist genormt und heißt A-Kurve. Daher stammt die bei Lärmpegeln verwendete Bezeichnung dB(A).

Ein Unterschied von 10 dB wird vom Menschen etwa als eine Verdopplung der Lautstärke empfunden. Der gesamte Pegelumfang des Ohrs von ca. 120 dB wird nur mit Schwierigkeiten von einem Meßgerät erreicht.

Reale Messungen des Lärmpegels werden heute meist mit direkt anzeigenden Meßgeräten durchgeführt. Geräuschpegel schwanken aber im allgemeinen recht stark. Deshalb mißt man eine gewisse Zeit lang in kurzen Intervallen von z.B. 5 s und schreibt jeweils den größten gemessenen Pegel eines Intervalls auf. Die so erhaltenen Meßwerte können dann ausgewertet werden.

Bei Langzeitmessungen ist insbesondere der minimale, mittlere und maximale Pegel sehr aussagekräftig. Die Auswertung dieser Pegel ist genormt:

- L95 wird als Minimalwert verwendet. Es ist der Pegel, der von 95 % aller Pegel überschritten wird.
- L01 wird als Maximalwert verwendet. Es ist der Pegel, der nur noch von 1 % aller Pegel überschritten wird.  
Die Festlegung von L01 und L95 sondert "Ausreißer" in den gemessenen Pegeln aus.
- Leq wird als Mittelwert verwendet. Es ist der energieäquivalente Mittelwert, also der Pegel, der im Mittel die gleiche Schallenergie enthält.

Trägt man L01, L95 und Leq über einen ganzen Tag lang auf, dann lassen sich Aussagen über Umfang und Art der Lärmbelastung machen.

Das Programm Zellärm erfaßt Serien von 500 bis 6000 Meßwerten in regelmäßigen Intervallen. Jeder einzelne Meßwert ist der maximale Pegel in seinem Intervall. Neben der Darstellung von Pegelverläufen über die Meßzeit werden die Auswertung von Häufigkeitsverteilungen in Abhängigkeit von Pegeln und die Ermittlung von L01/L95/Leq ermöglicht.

## 4.2 Lärmmessung mit ZELLÄRM

Das Programm Zellärm erlaubt die Lärmmessung und -auswertung auf unterschiedliche Art. Im Online-Betrieb werden die Meßdaten gleichzeitig registriert und angezeigt, wobei für die Ausgabe zwischen Diagramm, Großanzeige mit Bargraph und Liste gewählt werden kann. Im Offline-Betrieb kann die Zelle vom PC getrennt werden, um außerhalb des Unterrichtsraums Messungen durchzuführen. Die in der Zelle gespeicherten Meßdaten werden dann später in den PC eingelesen und ausgewertet. Die Auswertung umfaßt neben der Darstellung als Diagramm und Liste auch die Häufigkeitsverteilung nach Pegeln und die Ermittlung der Pegel L01 und L95, also derjenigen Pegel, die nur noch von 1% bzw. 95% der Meßwerte im betrachteten Zeitraum überschritten werden. Schließlich können auch die energieäquivalenten Mittelwerte  $L_{eq}$  bestimmt werden.

Vor einer Messung muß die Meßdauer eingestellt werden. Zellärm stellt dazu Standardeinstellungen zwischen 10 Sekunden und 24 Stunden bereit. Die Voreinstellung beträgt 10s.

| Meßzeit     | Einzelmessungen | Intervallzeit |
|-------------|-----------------|---------------|
| 10 s        | 1000            | 10 ms         |
| 1 min       | 1000            | 60 ms         |
| 6 min       | 1000            | 360 ms        |
| 1 h         | 6000            | 600 ms        |
| 6 h         | 6000            | 3,6 s         |
| 24 h        | 6000            | 14,4 s        |
| 12 x 10 s   | 12 x 500        | 20 ms         |
| 12 x 1 min  | 12 x 500        | 120 ms        |
| 12 x 10 min | 12 x 500        | 1,2 s         |

### 4.3 Direktanzeigende Messungen

Eine Unterrichtsreihe zum Thema Lärm sollte zuerst einen Eindruck von der Pegelmessung und dem verwendeten Maß dB(A) vermitteln, indem Meßwerte direkt angezeigt werden. Zellärm bietet dazu eine digitale Großanzeige mit Bargraph. Sinnvoll ist zunächst eine Meßdauer von 10 Sekunden. Erfahrungsgemäß werden Schüler und Schülerinnen durch die direkte Anzeige des Lärmpegels zu spontanem Rufen und Pfeifen angeregt. Der letzte gemessene Geräuschpegel am Ende der Meßdauer bleibt in der Anzeige stehen.

Abb. 4.1: Direktanzeige des Lärmpegels

Um ein Gefühl für die gemessenen Geräuschpegel zu vermitteln, können Probemessungen in unterschiedlichen Situationen durchgeführt werden:

- größtmögliche Ruhe im Unterrichtsraum
- normales und angestregtes Sprechen
- angeregte Diskussion
- Blättern in Heften und Büchern
- Störgeräusche bei offenem und geschlossenem Fenster

Es zeigt sich, daß schon nach kurzer Übungszeit die Pegel unterschiedlicher Geräusche recht genau geschätzt werden können. Eine besonders interessante Übung ist die größtmögliche Ruhe. Es zeigt sich, daß praktisch niemals vollkommene Ruhe herrscht. Jedes leise Knarren eines Stuhles und jedes entfernte Gespräch wird deutlich angezeigt. Es ist kaum möglich, die Empfindlichkeitsschwelle des Meßgeräts von ca. 30 dB(A) in der Schule zu erreichen. Die uns allezeit umgebenden Geräuschpegel werden so bewußt gemacht.

Nach einer praktischen Einführung empfiehlt sich ein theoretischer Exkurs zur Messung von Lärmpegeln. Je nach Jahrgangsstufe kann auf den logarithmischen Charakter der Pegelmessung eingegangen werden. Jede Verzehnfachung der Schalleistung erhöht den Pegel um 10 dB. Von der Hörschwelle des Menschen (0dB(A)) bis zur Schmerzgrenze (ca. 120 dB(A)) liegen also 12 Zehnerpotenzen. Das Lärm-Meßmodul enthält einen elektronischen Logarithmierer und erreicht damit einen Meßbereich von 30 dB(A) bis 100 dB(A). Das verwendete A-Filter paßt den Frequenzgang des Meßgeräts dem des menschlichen Ohrs an. Bei etwa 1kHz besteht die größte Empfindlichkeit, weit darüber und darunter nimmt sie stark ab.

Die Auswirkungen von Lärm auf das Wohlbefinden und die Gesundheit müssen diskutiert werden. Praktische Versuche lassen sich zur Konzentrationsfähigkeit bei unterschiedlichen Geräuschpegeln durchführen. Ausgearbeitete Vorschläge dazu finden sich in [2].

## 4.4 Registrierung und Darstellung zeitlicher Verläufe

Jede Einzelmessung gibt ein punktuelles Meßergebnis wieder. Belastungen durch hohe Lärmpegel sind jedoch sehr stark von der Dauer und vom Verlauf der Pegel abhängig. Hohe Pegel sind dann besser zu ertragen, wenn sie nur selten auftreten. Ein Gewöhnungseffekt ist andererseits nur bei etwa gleichbleibenden Pegeln zu erwarten. Lärmpegel müssen daher über längere Zeiträume gemessen und registriert werden.

Es kann sinnvoll sein, Meßergebnisse einer direkten Messung von Hand zu notieren und z.B. in Diagrammen auszuwerten. Diese sehr aufwendige Arbeit führt zu dem Gedanken, daß solche Routineaufgaben nach Möglichkeit automatisiert werden sollten.

Alle mit Zellärm direkt gemessenen Pegel einer Messung werden im Computer registriert. Sie lassen sich z.B. als Liste am Bildschirm ausgeben oder ausdrucken.

Abb. 4.2: Meßwerte in Listenform

Eine solche Liste kann dann von Hand in ein Diagramm übertragen werden. Die große Zahl der Meßpunkte zwingt aber dazu, eine Auswahl zu treffen, indem z.B. nur jeder zwanzigste Pegel benutzt wird.

Auch diese Arbeit kann automatisiert werden. Zellärm stellt eine grafische Ausgabe von Pegeln bereit. Diagramme lassen sich am Bildschirm betrachten und ausdrucken.

Abb. 4.3: Diagramm der gemessenen Lärmpegel

Direkte Messungen lassen sich nicht nur in Großanzeige, sondern auch in Listenform oder in Diagrammform darstellen. Insbesondere die Direktmessung bei gleichzeitiger Erzeugung eines Diagramms bietet eine übersichtliche Form der Meßwerterfassung, weil der Gesamtverlauf über den Meßzeitraum im Blick bleibt. Einzelereignisse lassen sich diskreten "Peaks" zuordnen und direkt vergleichen.

Abb. 4.4: Eine Direktmessung mit gleichzeitiger Diagrammausgabe

Zahlreiche Versuche können im Klassenraum durchgeführt werden, um Verfahren und Auswertung der Lärmmessung einzuüben.

- Bestimmung des Bereichs üblicher Pegel im Unterricht
- Untersuchung der zeitlichen Lärmverteilung: Lassen sich z.B. einzelne Sprecher aus dem Grundpegel an Geräuschen hervorheben?
- Welcher "Störabstand" ist für eine gute Verständlichkeit erforderlich?
- Welche Lärmpegel stören die Konzentrationsfähigkeit?

## 4.5 Ortsunabhängige Pegelmessungen

Zellärm unterstützt die Fähigkeit der Zelle, Meßwerte im tragbaren Einsatz zu erfassen. Nach der Einstellung der Meßzeit kann die Zelle vom PC getrennt werden. Am eigentlichen Meßort läßt sich eine Messung mit der Starttaste auslösen. So lassen sich z.B. Lärmpegel an verschiedenen Straßen einer Stadt erfassen.

Nach der Rückkehr vom Meßort müssen die Meßwerte in den PC übertragen werden. Sie stehen dann zur weiteren Auswertung bereit. Die Meßwerte sollten zunächst auf einer Diskette gespeichert werden, um spätere Vergleiche anstellen zu können. Eine umfangreiche Meßdatensammlung kann späteren speziellen Auswertungen dienen.

Die Auswahl der Meßzeiten richtet sich nach der jeweiligen Aufgabe und den praktischen Möglichkeiten der Durchführung. Neben der Einstellung der Meßdauer ist auch die Wahl einer aufgeteilten Messung mit zwölf einzelnen Meßblöcken möglich. Damit lassen sich mehrere Messungen durchführen, die gemeinsam in den PC übertragen werden. Es können im wesentlichen drei Typen von Messungen unterschieden werden:

- Registrierung zeitlich begrenzter Vorgänge: Einfahrender Zug, überfliegendes Flugzeug, Pausenlärm. Das Meßgerät wird hierbei in der Hand gehalten, die Messung dauert zwischen 10 s und einer Stunde.
- Langzeitmessung an einem Ort: Verkehrslärm an einer Straße, vom Fenster eines Wohnhauses gemessen. Hierbei beträgt die Meßzeit vorzugsweise 24 Stunden.
- Untersuchung einzelner Lärmquellen: Vergleich unterschiedlicher Fahrzeuge im Stand und bei der Vorbeifahrt. Wahl von 12 Teilmessungen von z.B. 10s.

Abb. 4.5: Verkehrslärm an einer vielbefahrenen Straße

## 4.6 Auswertungen

Zellärm bietet verschiedene Möglichkeiten der Auswertung von Meßdaten. Neben der einfachen Ausgabe in Listenform und als Diagramm kann die Verteilung der Einzelmesswerte auf Größenklassen ausgewertet werden. Dabei steht eine Darstellung in Listenform und als Diagramm zur Verfügung. Zur Verdeutlichung des Verfahrens kann die Verteilung auch aus einer Liste von Meßwerten durch Abzählen gleicher Pegel von Hand erzeugt werden.

Abb. 4.6: Auswertung der Pegelverteilung für die Messung nach Abb. 4.5

Abb. 4.7: Grafische Darstellung der Pegelverteilung für den Verkehrslärm

Abb. 4.8: Pegelverteilungen für 12 vergleichende Kurzzeitmessungen

Die Diskussion der Häufigkeitsverteilung nach Pegeln kann zu einer Unterscheidung verschiedener Geräuschsituationen führen. Das Beispiel zeigt die typische Situation des Verkehrslärms, für den ein flaches Maximum und eine relativ breite Pegelverteilung beobachtet werden kann. Selbst wenn alle Fahrzeuge den gleichen Maximalpegel lieferten, führte die laufende Änderung der Distanz zur Lärmquelle und die zufällige Verteilung der Anzahl von Fahrzeugen zu dieser breiten Verteilung. Ganz anders wäre z.B. die typische Situation eines Vortrags: Es müßten zwei deutliche Maxima für Sprechphasen und Sprechpausen des Vortragenden beobachtet werden. Liegen beide Maxima relativ nahe beieinander, dann muß von störenden Hintergrundgeräuschen ausgegangen werden.

Die Häufigkeitsverteilung kann als Zwischenstufe für die Auswertung der Extrempiegel L01 und L95 herangezogen werden. Bei einer Anzahl von 6000 Einzelmessungen findet man L01 als den Pegel, der nur noch von 60 Einzelergebnissen übertroffen wird. In der Liste nach Abb. 4.6 ist das der Pegel 58 dB(A). L95 ist entsprechend der Pegel, der nur noch von maximal 300 Einzelergebnissen unterboten wird, im obigen Beispiel 79 dB(A).

Während die extremen Pegel noch leicht ohne Hilfe des Programms ausgewertet werden können, ist die Auswertung des energieäquivalenten Mittelwerts  $Leq$  praktisch nur noch mit Programmhilfe möglich. Zellärm wertet  $Leq$  zusammen L01 und L95 bei kurzen Messungen über den gesamten Meßzeitraum aus, während bei längeren Messungen abschnittsweise gerechnet wird. Bei 24-Stunden-Messungen wird z.B. stundenweise ausgewertet.

Abb. 4.9: Pegelauswertung zum Verkehrslärm

Zur Beurteilung der Lärmbelastung an einem Ort muß in erster Linie der Äquivalentpegel betrachtet werden. Die Belastung an einem Wohnort muß über den Tagesverlauf erfaßt werden. Die folgende Messung zeigt ein typisches Beispiel. Hier wurde in einem Wohngebiet in der Stadt gemessen.

Abb. 4.10: Ergebnis einer 24-Stunden-Messung

Eine Unterrichtsreihe zum Thema Lärm muß sich auch mit Lärmschutzmaßnahmen beschäftigen. Verschiedene Maßnahmen können durch Vergleichsmessungen auf ihre Wirksamkeit hin untersucht werden. Mögliche Fragestellungen sind:

- Welche Dämpfung bewirken einfache Fensterscheiben oder Termopanescheiben?
- In welchem Maße vermindern Hecken den Lärm?
- Wie wirksam sind verschiedene Lärmschutzwälle an Autobahnen?
- Welche Unterschiede zeigen moderne Flugzeuge gegenüber älteren bei Start und Landung?



## 4.7 Automatisch ausgelöste Messungen

Die Vorteile der Automatisierung in der Meßwerterfassung werden schon bei Langzeitmessungen sehr deutlich. Spezielle Untersuchungen können trotzdem sehr aufwendig werden, wenn jede Messung von Hand ausgelöst werden muß. Der Einsatz eines Computers bietet jedoch auch die Möglichkeit, Messungen nach vorgegebenen Kriterien automatisch auszulösen. Damit wird eine weitere Stufe der Automatisierung gezeigt.

Bei jeder Messung mit Zellärm ist grundsätzlich ein autonomes Steuerprogramm in der Zelle aktiv. Es steuert den Zeitablauf der Messung und sucht aus den gewählten Meßintervallen das Pegel-Maximum. Die Meßdaten werden dabei zugleich in der Zelle gespeichert und an den PC gesendet.

Während die grundlegenden Aufgaben des Steuerprogramms festliegen, kann insbesondere die automatische Auslösung von Messungen nach unterschiedlichen Kriterien realisiert werden. Zellärm enthält drei Steuer-Optionen, die im Auswahl-Menü als J, K und L geführt werden:

- J: Nach einem Start mit der Starttaste werden 12 Kurzzeitmessungen von je 10 s im Abstand von jeweils 10 min ausgelöst. So erhält man kurze Proben aus einem Zeitraum von ca. zwei Stunden.
- K: Es werden 12 Messungen von jeweils 1 min ausgelöst, wenn Pegel ab 90 dB(A) auftreten. Damit lassen sich wiederholt auftretende Spitzenbelastungen z.B. durch Fluglärm erfassen.
- L: Das Programm startet eine 10-Sekunden-Messung, die immer dann wiederholt wird, wenn ein lauterer Geräusch festgestellt wird. So kann man das lauteste Ereignis in einem beliebigen Zeitraum erfassen.

Die Steuerprogramme sind in Simpel geschrieben und sind im Quelltext zugänglich. Damit können interessierte Arbeitsgruppen auch eigene Ideen verwirklichen und spezielle Steuerprogramme entwickeln.



## 5. Ortsunabhängige Erfassung von Meßdaten

Neben der vielfach bereits fest etablierten Unterrichtsreihe zum Thema Lärm bieten sich zahlreiche Betätigungsfelder für weitere Messungen von Umweltdaten. In den Bereich des Umweltschutzes gehören die Messungen der Luftqualität, Messungen an Gewässern und die Langzeitmessung der Radioaktivität. Daneben unterstützen reine Naturbeobachtungen z.B. den Biologieunterricht oder die Wetterkunde. Ein weiterer Bereich beschäftigt sich mit technischen Systemen außerhalb des Labors. So lassen sich z.B. Messungen an bewegten Fahrzeugen unterschiedlichen Fachbereichen zuordnen.

### 5.1 Klimabeobachtungen

Langzeitmessungen des Klimageschehens stellen eine typische Anwendung der Zelle dar. Hier wird zunächst eine Untersuchung des Zusammenhangs von Sonneneinstrahlung und Temperaturänderung vorgestellt. Für die Messungen wird Prisma eingesetzt.

Als Sensoren werden das Meßmodul Temperatur\_1 (Meßbereich  $-20^{\circ}\text{C}\dots+107^{\circ}\text{C}$ ) an Kanal 1 und das Meßmodul Belichtungsmesser (Meßbereich 1lux...100000lux) an Kanal 2 gewählt.

Der Temperatursensor wird an einer schattigen Stellen im Freien aufgestellt. Dagegen kann der Lichtsensor frei aufgestellt werden. Das ist jedoch nur dann sinnvoll, wenn ein Ort zugänglich ist, an den im Laufe des Tages keine Schatten von Gebäuden oder Bäumen fallen. Bei der vorliegenden Messung wurde eine indirekte Lichtmessung hinter dem Fenster durchgeführt. Der Sensor wurde dabei von diffusem Licht getroffen.

Um die Messung vorzubereiten, wird in Prisma eine Zuordnung der Meßkanäle zu den Sensoren vorgenommen. TEMP1.DEF wird Kanal 1 zugeordnet, während LICHT\_I.DEF für Kanal 2 gewählt wird. Der Temperatursensor besitzt auch noch die Datei TEMP1\_30.DEF mit einem reduzierten Meßbereich. Der Lichtsensor hat ebenfalls zwei Dateien. Während LICHT\_I.DEF die Beleuchtungsstärke I in Lux zuordnet, kann man mit LICHT\_E die Leistungsdichte E der Lichtstrahlung in  $\text{W}/\text{m}^2$  messen. Da bei einer Messung im Schatten ohnehin nur ein Teil der Sonnenenergie gemessen werden kann, wird die Messung in Lux gewählt.

Um den Meßaufbau zu überprüfen, sollte zunächst eine kurze online-Messung durchgeführt werden. Dazu wird die Meßzeit 5 min gewählt. Die eigentliche Probemessung kann z.B. mit der gut sichtbaren, großen Digitalanzeige erfolgen. Die eigentliche Messung erfolgt offline mit einer Meßdauer von 24 Stunden.

Abb. 5.1: Lufttemperatur und Sonneneinstrahlung

Das Ergebnis der Messung erlaubt eine Diskussion des Wettergeschehens an diesem sonnigen Tag im Juni. Man erkennt die gegenüber der Einstrahlung verzögerte Erwärmung. Die leichten Schwankungen der Beleuchtungsstärke deuten auf eine geringe Bewölkung hin.

Interessante Versuche bieten sich bei der Messung der Bodentemperatur an. In den Bereich der Biologie fallen Untersuchungen der Lebensmöglichkeiten für Kleintiere im Winter. In welcher Bodentiefe lassen die Temperaturen ein Überleben zu? Welche Temperaturen werden in größeren Laubhaufen gehalten? Diese Untersuchungen könnten im Zusammenhang mit der Erarbeitung umweltgerechter Empfehlungen für Kleingärtner durchgeführt werden.

Der Themenbereich Heizung, Wärmeisolierung und Umweltschutz läßt sich durch reale Messungen an Gebäuden unterstützen. Interessant ist z.B. die gleichzeitige Messung von Innen- und Außentemperatur. Der NTC-Temperatursensor eignet sich wegen seines gekapselten Aufbaus und seiner langen, dünnen Zuleitung besonders für solche Messungen. Abb. 5.2 zeigt das Ergebnis einer Messung über 24 Stunden an einem Dezembertag an einem Pavillon einer Schule. Ein Sensor wurde neben dem Gebäude in die Erde gesteckt,

um möglichen Bodenfrost zu erkennen. Die Zelle befand sich mit dem zweiten Sensor im Gebäude, nahe des Fensters auf einer elektrischen Nachtspeicherheizung.

Abb. 5.2: Innen- und Außentemperatur

Eine Diskussion der Meßergebnisse zeigt, daß an diesem Tag kein Bodenfrost auftrat. Auch am Tage fand jedoch keine Erwärmung über acht Grad statt. Die Nachtspeicherheizung wurde offenbar zwischen fünf Uhr und acht Uhr am Morgen aufgeladen. Überraschend ist, daß es am ganzen Tag nur eine Ladeperiode gibt. Es erscheint jedoch sinnvoll, daß zum Unterrichtsbeginn um acht Uhr die höchste Temperatur der Heizung erreicht ist, um die Räume vorzuwärmen.

## 5.2 Messung der Luftverschmutzung

Die Luftverschmutzung ist eines der gravierendsten Umweltprobleme unserer Zeit. Messungen mit der Zelle und dem Meßmodul Gas können das Bewußtsein für diese Problematik schärfen und Diskussionen über umweltbewußtes Handeln anregen.

Der Widerstand des Gassensors stellt ein Maß für die komplexe Luftbelastung dar und kann für vergleichende Messungen herangezogen werden. Für das Meßprogramm Prisma steht eine Definitionstabelle zur Anzeige des Widerstands in Kiloohm (GAS.DEF) bereit, so daß Messungen mit dem Meßmodul Gas einfach durchführbar sind. Der Heizstrom des Sensors von ca. 150mA bringt es mit sich, daß möglichst mit NC-Akkus in der Zelle gearbeitet werden sollte. Sie weisen eine ausreichende Spannungskonstanz und eine sichere Meßdauer von ca. zwei Stunden auf (vgl. Kap. 6).

Der Einsatz des Meßmoduls Gas im Unterricht sollte zunächst durch eine Untersuchung der allgemeinen Eigenschaften des Sensors vorbereitet werden. Abb. 5.3 zeigt das typische Einschaltverhalten, wenn die Zelle, und damit der Heizstrom des Sensors, vor der Messung für einige Minuten ausgeschaltet war. Der Sensor wird mit dem Aufheizen zunächst sehr niederohmig und erreicht dann nach wenigen Minuten einen Gleichgewichtszustand. Ein neuer Sensor benötigt zuerst eine längere Zeit, bis er reproduzierbare Ergebnisse liefert.

Abb. 5.3: Einschaltverhalten des Sensors

Interessante Versuche im Unterrichtsraum können die Reaktion auf Lösungsmittel von Klebstoffen, Filzstiften oder andere Schadstoffe untersuchen. Um vergleichbare Ergebnisse zu erhalten, müssen gleiche Versuchsbedingungen festgelegt werden. So könnte man vereinbaren, daß in einem Pappkarton eine Fläche von einem Quadratcentimeter mit dem Klebstoff bestrichen wird. Der Sensor wird dann in den Karton gebracht.

Abb. 5.4 zeigt eine Messung mit zwei unterschiedlichen Klebstoffen.

Abb. 5.4: Nachweis der Lösungsmittel von zwei Klebstoffen

In ähnlicher Weise lassen sich Versuche mit Feuerzeuggas, Alkohol und Tabakrauch durchführen. Messungen der Raumluft über längere Zeiträume können Aufschluß über Belastungen und erforderliche Lüftungszeiten geben. Die Empfindlichkeit des Gas-Sensors reicht an die des menschlichen Geruchssinns heran. Anders als unsere Nase kennt er jedoch keinen Gewöhnungseffekt, es werden also auch geringere, aber andauernde Konzentrationen angezeigt.

Besondere Gefahren gehen vom Straßenverkehr und von Industrie-Emissionen aus. Mit der Zelle lassen sich Luftbelastungen an beliebigen Orten erfassen. Ein mögliches Unterrichtsprojekt könnte die Erstellung einer Belastungskarte des umliegenden Wohngebiets zum Ziel haben. Verkehrsreiche Straßen müßten dabei auch auf ihre zeitliche Belastungsverteilung untersucht werden.

Abb. 5.5: Abgasbelastung für einen Fußgänger bei geringer Verkehrsdichte

Abb. 5.5 zeigt eine Messung während eines Fußwegs durch Stadtgebiet zu einer Zeit geringer Verkehrsdichte. Die einzelnen Phasen lassen sich bestimmten Straßen zuordnen. Ein deutlicher Peak geht auf eine Ampelkreuzung zurück. Stehende Autos mit laufendem Motor werden hier als besondere Luftverschmutzer deutlich.

Mit dem Meßmodul Gas lassen sich Langzeitmessungen wegen des erhöhten Strombedarfs nur mit einer zusätzlichen Stromversorgung durchführen. Dazu läßt sich ein Akkumulator höherer Kapazität oder ein stabilisierte Netzgerät einsetzen. Abb. 5.6 zeigt das Ergebnis einer 24-Stunden-Messung an einer vielbefahrenen Straße an einem Wochentag. Zu den Zeiten des größten Berufsverkehrs kommt es hier regelmäßig zu Staus mit besonderen Belastungen. Die Aufstellung der Meßanordnung (Zelle, Meßmodul Gas und Zusatzakku) erfolgte am Abend zuvor. Die Messung wurde mit der in Prisma enthaltenen automatischen Schaltuhrfunktion zeitverzögert um Mitternacht gestartet.

Abb. 5.6: Langzeitmessung an einer Hauptverkehrsstraße

Obwohl im allgemeinen eine Mischung verschiedener Gase erfaßt wird, lassen sich viele weitergehende Untersuchungen anstellen, die eine vertiefte Auseinandersetzung mit der Problematik der Luftverschmutzung ermöglichen:

- Welchen Einfluß hat die Höhe einer Wohnung über der Straße?
- Ist die Höhe eines Kinderwagens für die Belastung des Babys entscheidend?
- Ist eine Filterwirkung durch Gärten oder Wälder nachweisbar?
- Lassen sich Industrieanlagen als Verschmutzer lokalisieren?
- Werden nachts heimlich Schadstoffe entlassen?
- Treten bei bestimmten Windrichtungen höhere Belastungen auf? Wenn ja, woher stammen sie?
- Unter welchen Bedingungen tritt Smog auf?

Die Unterscheidung zwischen Kraftfahrzeugen und anderen Emissionsquellen läßt sich oft durch die Korrelation mit den Ergebnissen einer gleichzeitigen Lärmmessung erreichen.

Eine Kalibrierung des Sensors für die Messung von Konzentrationen bestimmter Gase ist prinzipiell möglich. Das Datenblatt des Meßmoduls enthält ein Diagramm der Widerstandsänderung in Abhängigkeit von der Konzentration für verschiedene Gase (vgl. Abb. 2.8). Damit lassen sich spezielle Definitionstabellen für Prisma aufbauen. Zwischen einzelnen Exemplaren der Sensoren existiert eine gewisse Streuung im Grundwiderstand. Sie kann berücksichtigt werden, wenn einzelne Meßpunkte im Labor durch definierte Gasmischungen kalibriert werden.

### 5.3 Mobile Meßdatenerfassung mit dem Fahrrad

Das Fahrrad eignet sich sowohl als Hilfsmittel zur mobilen Meßdatenerfassung als auch als Studienobjekt an ganz verschiedenen Orten im Unterricht. Vom Studium kinematischer und mechanischer Probleme im Physikunterricht über ergometrische Untersuchungen im Biologieunterricht bis hin zu verkehrspolitischen Fragen reicht ein weites Spektrum unterschiedlichster Fragestellungen. Von besonderem Vorteil ist die Tatsache, daß sehr viele Schüler das Fahrrad täglich benutzen und daß es auch unter den Lehrern wieder mehr Anhänger findet.

Die Zelle läßt sich problemlos auf dem Fahrrad mitführen und mobil für verschiedenste Meßwerterfassungen einsetzen. Die Auswertung der Meßwerte geschieht nach der Fahrt am PC in der Schule.

Für sehr viele Fragestellungen ist die Erfassung der Momentangeschwindigkeit des Fahrrads erforderlich. Übliche Tachometer arbeiten entweder rein mechanisch oder über am Rad angebrachte Impulsgeber. Setzt man den Dynamo als Meßwertgeber für die Geschwindigkeit ein, dann werden keine zusätzlichen Elemente benötigt. Die Frequenz der erzeugten Wechsellspannung ist ein direktes Maß für die Geschwindigkeit.

Der Umsetzungsfaktor der Dynamofrequenz in die Geschwindigkeit beträgt konstant etwa 5,5km/h / 100Hz. Damit kann man Frequenzen bis maximal 1kHz erwarten. Das Signal kann nach Spannungsbegrenzung durch eine Zenerdiode direkt über den ersten Digitaleingang der Zelle eingelesen werden. Der Dynamo darf ohne Verfälschung der Meßwerte belastet oder unbelastet betrieben werden.

#### Abb. 5.7: Anschluß des Dynamos an die Zelle

Die Frequenzmessung ist im Betriebssystem der Zelle nicht vorgesehen, kann jedoch in Simpel programmiert werden. Die Programmierung braucht selbst nicht Gegenstand des Unterrichts zu sein, wenn man auf das compilierte Programm zurückgreift. Es liegt dem Monitorprogramm der Zelle bei (FREQUENZ.BIN).

Änderungen am Programm können leicht durchgeführt werden, um z.B. den Meßzeitraum oder die Zahl der Kanäle den Bedürfnissen anzupassen. Grundlagen der Programmierung in Simpel und auch spezielle Zählerprogramme werden in Kap. 9 vorgestellt. Hier sollen nur einige kurze Erklärungen gegeben werden.

Listing 5.1 zeigt das kommentierte Programm. Es erreicht eine Grenzfrequenz von 2,5kHz und eine Auflösung von 10Hz. Die Programmierung ist zeitkritisch, da zehn Abtastvorgänge etwa gleichmäßig auf etwas weniger als eine Millisekunde verteilt werden müssen. Der Befehl Warte erzeugt jeweils eine Verzögerung bis zur nächsten abgelaufenen Millisekunde. Er sorgt hier für die Genauigkeit der Torzeit.

Datensätze der Zelle haben immer einen Parameterblock von zehn Bytes, der die aktuellen Einstellungen enthält. Er wird von einem auslesenden Programm ausgewertet, um die Daten den einzelnen Meßkanälen und Zeiten korrekt zuzuordnen. Die hier gewählten Einstellungen sind: Ca. 1500 Messungen an 4 Kanälen im Abstand von einer Sekunde.

#### **;Programm zur Frequenzmessung, Eingang D1,**

```
;Frequenz bis 2,5kHz, Torzeit 100ms
;Zusätzliche Analogmessung an A1...A3
;1500 Messungen, je 1s
```

```
Prozedur flank     ;zählt Impulse
LiesMerk 3         ;Merkspeicher 3 lesen,
Plus 1            ;erhöhen
Merk 3            ;und wieder abspeichern
ProzedurEnde
```

```
Prozedur high     ;wertet "high" aus
```

```

LiesMerk 2          ;alter Zustand
WennX=0 flanken    ;gewechselt?
X 1
Merk 2              ;neuer Zustand 1
ProzedurEnde

Prozedur low        ;wertet "low" aus
X 0
Merk 2              ;neuer Zustand 0
ProzedurEnde

Prozedur trigger    ;wertet D1 aus
EinPort             ;D1...D4 lesen
AND 16              ;D1 maskieren
WennX>0 high        ;wenn D1=1 : "high"
EinPort             ;D1...D4 lesen
AND 16              ;D1 maskieren
WennX=0 low         ;wenn D1=0 : "low"
ProzedurEnde

Prozedur frequenz   ;zählt 1ms lang
A 10                ;Zähler A laden
Amal trigger        ;10 mal "trigger"
Warte 1              ;bis zum Ende der ms
ProzedurEnde

Prozedur torzeit    ;steuert Messungen
X 0
Merk 2              ;Zustand = 0
Merk 3              ;Impulszähler = 0
B 100
Bmal frequenz       ;100ms frequenz
LiesMerk 3          ;Ergebnis lesen
SchreibeRAM         ;abspeichern
Warte 150           ;150ms
Messen 1            ;A1 messen
SchreibeRAM         ;abspeichern
Warte 250           ;250ms
Messen 2            ;A2 messen
SchreibeRAM         ;abspeichern
Warte 250           ;250ms
Messen 3            ;A3 messen
SchreibeRAM         ;abspeichern
Warte 250           ;250ms
ProzedurEnde

Anfang              ;Hauptprogramm:
AnfangRAM           ;Parameterblock:
SchreibeRAM 6       ;6*256 Messungen
SchreibeRAM 0
SchreibeRAM 1       ;Kanäle 1...4
SchreibeRAM 2
SchreibeRAM 3
SchreibeRAM 4
SchreibeRAM 3       ;Intervallzeit 1000ms
SchreibeRAM 228
Immer torzeit       ;endlos, bis RAM voll
Ende

```



## Listing 5.1: Das Frequenz-Meßprogramm zur Geschwindigkeitserfassung

Das Programm wurde zur Registrierung der Geschwindigkeit während einer kurzen Fahrt verwendet. Abb. 5.8 zeigt den gesamten Verlauf. Die Auswertung erfolgte mit WORKS. Im Diagramm sind deutlich drei Wartezeiten an Ampeln zu erkennen.

## Abb. 5.8: Geschwindigkeits-Zeit-Diagramm einer Fahrt

Durch numerische Integration mit den Mitteln einer Tabellenkalkulation läßt sich leicht die Strecke ermitteln. Abb. 5.9 zeigt das Weg-Zeit-Diagramm der Fahrt.

## Abb. 5.9: Das mit WORKS ermittelte Weg-Zeit-Diagramm der Fahrt

Die genaue Auswertung kann zahlreiche Daten liefern:

- Die Fahrtzeit betrug 435 Sekunden.
- Die Wartezeiten betragen insgesamt 83s.
- Durchschnittsgeschwindigkeit war  $3,6\text{m/s} = 13,1\text{km/h}$
- Durchschnitt ohne Wartezeiten war  $4,5\text{m/s} = 16,1\text{km/h}$
- Die maximale Geschwindigkeit war  $8,8\text{m/s} = 32\text{km/h}$
- Die Gesamtstrecke betrug 1578m
- Die erste Teilstrecke betrug 136m,

Während der Fahrt wurde gleichzeitig die Spitzenspannung des unbelasteten Dynamos gemessen. Abb. 5.10 zeigt, daß sie recht gut dem Verlauf der Geschwindigkeit folgt. Genauere Untersuchungen zeigen, daß die Abweichungen vom linearen Zusammenhang zwischen Frequenz und Spannung im unbelasteten Zustand des Dynamos relativ gering sind.

## Abb. 5.10: Dynamospannung während der Fahrt

Oft sollen Meßgrößen nicht in Abhängigkeit von der Geschwindigkeit, sondern ortsabhängig aufgenommen werden. Im Rahmen einer Unterrichtsreihe zum Umweltschutz könnte z.B. versucht werden, eine Belastungskarte der Stadt für Lärm und Luftverschmutzung zu erstellen. Setzt man den Dynamo des Fahrrads als Wegstreckenaufnehmer ein, so lassen sich Messungen automatisch nach vorgegebenen Streckenintervallen ausführen. Die Meßwerte sind dann leicht einer Stadtkarte zuzuordnen.

Wählt man Streckenabschnitte von 10 Metern, dann müssen vor jeder neuen Messung etwa 650 Impulse des Dynamos abgewartet werden. Um eine hohe Genauigkeit der Streckenmessung zu erreichen, muß für jedes Fahrrad ein individueller Umsetzfaktor ermittelt werden. Listing 5.2 zeigt ein Meßprogramm für vier Kanäle.

**; Taktgesteuerte Vierkanalmessung**

```
Prozedur high          ;wertet "high" aus
X 1
Merk 2                ;neuer Zustand 1
ProzedurEnde
```

```
Prozedur low          ;wertet "low" aus
X 0
Merk 2                ;neuer Zustand 0
ProzedurEnde
```

```

Prozedur trigger      ;wertet D1 aus
EinPort              ;D1...D4 lesen
AND 16              ;D1 maskieren
WennX>0 high        ;wenn D1=1 : "high"
EinPort              ;D1...D4 lesen
AND 16              ;D1 maskieren
WennX=0 low         ;wenn D1=0 : "low"
ProzedurEnde

Prozedur 200takte
A 200
Amal trigger
ProzedurEnde

Prozedur streckemessen
B 3
Bmal 200takte      ;600 Takte
B 50
Baml trigger      ;+ 50 Takte
ProzedurEnde

Prozedur messungen
Messen 1           ;A1 messen
SchreibeRAM        ;abspeichern
Messen 2           ;A2 messen
SchreibeRAM        ;abspeichern
Messen 3           ;A3 messen
SchreibeRAM        ;abspeichern
Messen 4           ;A4 messen
SchreibeRAM        ;abspeichern
streckemessen     ;Dynamotakte abwarten
ProzedurEnde

Anfang             ;Hauptprogramm:
AnfangRAM         ;Parameterblock:
SchreibeRAM 5      ;5*256+220=1500 Messungen
SchreibeRAM 220
SchreibeRAM 1      ;Kanäle 1...4
SchreibeRAM 2
SchreibeRAM 3
SchreibeRAM 4
SchreibeRAM 0      ;Intervallzeit 10ms
SchreibeRAM 6      ;entspricht 10m
Immer messungen   ;endlos, bis RAM voll
Ende

```

### Listing 5.2: Ein Programm zur taktgesteuerte Messung

Das Programm führt bis zu 1500 Messungen im Abstand von 10 Metern durch. Es ist also eine Gesamtstrecke von 15 km möglich. Die Meßwerte lassen sich z.B. mit Prisma einlesen und auswerten. Die Zeitachse ist dabei durch eine Steckenachse zu ersetzen, wobei eine Sekunde für einen Kilometer steht.

## 6. Batterien und Solarzellen

Der Verbrauch von Batterien und Akkus für Geräte der Konsumelektronik nimmt ständig zu und wird zu einem entscheidenden Faktor der Umweltbelastung. Der verantwortungsbewußte Umgang mit Batterien muß daher auch in der Schule thematisiert werden. Neben handlungsbezogenen Zielen ergeben sich auch Zugänge zu einem vertieften physikalischen und technischen Verständnis. Darüberhinaus stellt die Messung an Batterien einen exemplarischen Fall der automatisierten Meßwerterfassung dar, so daß sich das Thema auch als Einstieg in die computergestützte Meßtechnik anbietet.

### 6.1 Messung der Ladungskapazität

Die am häufigsten eingesetzten Batterien sind Mignonzellen. Der Anwender muß sich dabei zwischen verschiedenen Bauformen und Fabrikaten entscheiden. Neben einfachen Zink-Kohle-Zellen werden die verbesserten Alkali-Mangan-Batterien mit einer höheren Ladung angeboten. Eine weitere Alternative stellen Nickel-Cadmium-Akkus dar. Eine Entscheidung muß sowohl wirtschaftliche Fragen als auch den Umweltschutz berücksichtigen. Ein objektiver Vergleich der "Leistungsfähigkeit" verschiedener Zellen kann über die Messung der Ladungskapazität erfolgen. Die Zelle wird dabei unter definierten und vergleichbaren Bedingungen komplett entladen. Durch eine Messung der Stromstärke über die Zeit läßt sich nach einer zeitlichen Integration die Ladung ermitteln. Ein Ni-Cd-Akku mit einer Kapazität von 0,5 Ampèrestunden sollte z.B. 10 Stunden lang einen Strom von 0,05 A aufrechterhalten können.

Eine praktikable Messung unter realistischen Bedingungen kann mit einer Entladung durch einen konstanten Lastwiderstand durchgeführt werden. Üblicherweise wird dabei die Zellenspannung erfaßt. Die Stromstärke läßt sich daraus mit  $I=U/R$  ermitteln. Da die Stromstärke mit zunehmender Entladung der Zelle abnimmt, muß die auswertende Software in der Lage sein, das Flächenintegral unter einer beliebigen Meßkurve zu berechnen. In der Praxis hat sich dabei das Programm Unimesh besonders bewährt.

Die ersten Versuche sollten mit Akkus ausgeführt werden, weil eine eventuelle Fehlmessung nach dem erneuten Laden wiederholt werden kann. Die folgende Messung zeigt die Entladung eines gebrauchten Akkus mit der Nennkapazität 600 mAh an einem 10- $\Omega$ -Lastwiderstand. Es wurde ein Meßintervall von 60 s gewählt. Die Entladekurve zeigt das für NC-Akkus typische Verhalten: Die Spannung wird sehr konstant bei etwa 1,2V gehalten, um dann relativ schnell abzufallen, wenn der Akku entladen ist. Die Messung dauerte ca. 6 Stunden.

Abb. 6.1: Entladekurve eines Ni-Cd-Akkus

Die Auswertung erfolgte mit dem Programm UNI-Aus. Für die Integration muß ein Kurvenabschnitt festgelegt werden. Um vergleichbare Bedingungen auch für Batterien zu gewährleisten, wurde bis zu einer Entladung auf eine Zellenspannung von 0,9 V integriert. Der Rechner liefert dabei ein Integral von 19000 Vs. Die Division durch 10  $\Omega$  ergibt die Ladung von 1900 As = 1900 C. Mit der Einheitenumformung 3600 As = 1 Ah ergibt sich für den Akku eine Ladung von 0,53 Ah. Der Wert ist geringfügig geringer als vom Hersteller angegeben. Dies kann einerseits auf das Alter des Akkus zurückzuführen sein. Andererseits wird die Nennkapazität für eine zehnstündige Entladung angegeben. Eine schnellere Entladung führt grundsätzlich zu einer geringeren Kapazität.

Das Programm Unimesh unterstützt die Zelle nur für online-Messungen. Vor allem bei sehr leistungsfähigen Batterien ergeben sich dabei unzumutbar lange Meßzeiten. Besser ist es daher, Meßwerte offline mit der Zelle zu erfassen, um sie dann an das Meßprogramm zu übertragen. Der einfachste Weg einer Datenübertragung besteht darin, gewonnene Meßwerte durch ein kleines Simpel-Programm so an den PC zu

übertragen, als wären sie direkt gemessen. Das Simpel-Programm muß dabei die von Unimess kommenden Abfragen der Analogeingänge abfangen und mit Meßwerten aus dem Speicher der Zelle beantworten. Das folgende Listing zeigt dieses Hilfsprogramm:

```
;Programm Unimess-Offline
;gespeicherte Meßwerte werden an Unimess wie
;online-Meßwerte gesendet

Prozedur antworten
EinRS232           ;Kanalnr. abwarten
LiesRAM
AusRS232           ;Meßwert senden
ProzedurEnde

Prozedur horchen
EinRS232           ;7: Kommando zum Auslesen
Y 7
WennX=Y antworten
ProzedurEnde

Anfang
AnfangRAM
A 8                ;Parameter überlesen
Amal LiesRAM
Immer horchen
Ende
```

#### Listing 6.1: Ein Hilfsprogramm zur Datenübertragung an Unimess

Die eigentliche Messung kann z.B. über das Monitorprogramm gestartet werden. Die Parameter sind dabei: 1.Kanal, 60000 ms/Messung, 6000 Messungen. Während der Messung sollte zusätzlich ein Voltmeter angeschlossen werden, um den Stand der Entladung zu erkennen. Nach dem Ende der Messung muß das Übertragungsprogramm in die Zelle geladen und gestartet werden. Für die eigentliche Übertragung soll in Unimess eine Messung im Sekundenraster eingestellt werden. Bei der Einstellung der Beschriftungen des Diagramms kann die Zeiteinheit Sekunden durch Minuten ersetzt werden. Die Übertragung der Meßwerte erfolgt also um den Faktor 60 schneller als die Messung. Das Übertragungsprogramm ist so geschrieben, daß es durch keine RESET-Funktion des PC gestoppt werden kann, so daß es auch nach dem Initialisierungsversuch durch Unimess aktiv bleibt. Ein Anschalten des Programms ist nur durch die Starttaste nach einem Reset der Zelle möglich. Man kann die Einstellung der offline-Messung und das Übertragungsprogramm parallel in der Zelle bereithalten. Die Umschaltung erfolgt durch verschieden langes Festhalten der Starttaste nach dem Einschalten der Zelle oder nach einem Betätigen der Reset- Taste:

- 2 Sekunden: Messung
- 4 Sekunden: Übertragungsprogramm

Die folgende Messung an einer einfachen Zink-Kohle-Batterie wurde nach dem beschriebenen Verfahren offline aufgenommen und mit Unimess ausgewertet. Die Meßdauer betrug 6 ½ Stunden. Die Entladekurve zeigt eine wesentlich schlechtere Spannungskonstanz. Auch unterhalb von 0,9 V findet eine weitere, langsame Entladung statt. Bis zu einer Zellenspannung von 0,9 V wurde eine Ladung von 0,51 Ah abgegeben. Damit ist die einfache Batterie dem Akku in jeder Beziehung unterlegen.

#### Abb. 6.2: Entladekurve einer Zink-Kohle-Batterie

Die besten Ergebnisse liefern Alkali-Mangan-Zellen. Bei der erwarteten wesentlich längeren Enladungszeit

tritt für offline-Messungen das Problem auf, daß Unimessex maximal 901 Meßwerte verarbeitet und die Zelle keine längeren Meßintervalle als ca. eine Minute ausführen kann. Die Meßdauer wäre damit auf 15 Stunden begrenzt. Um die verschiedenen Batterietypen unter gleichen Bedingungen zu testen, darf keine schnelle Entladung durch einen kleineren Lastwiderstand durchgeführt werden. Eine Lösung des Problems besteht darin, zwar im Minutenabstand zu messen, bei der Übertragung jedoch jeden zweiten Meßwert zu übergehen. Dies gelingt durch ein leicht modifiziertes Übertragungsprogramm:

**;Programm Unimessex-Offline - Version 2**

```
;gespeicherte Meßwerte werden an Unimessex wie
;online-Meßwerte gesendet
;jedes zweite Byte wird überlesen
```

Prozedur antworten

```
EinRS232          ;Kanalnr. abwarten
LiesRAM
AusRS232          ;Meßwert senden
LiesRAM           ;Meßwert überlesen
ProzedurEnde
```

Prozedur horchen

```
EinRS232          ;7: Kommando zum Auslesen
Y 7
WennX=Y antworten
ProzedurEnde
```

Anfang

```
AnfangRAM
A 8               ;Parameter überlesen
Amal LiesRAM
Immer horchen
Ende
```

**Listing 6.2: Übertragungsprogramm mit Datenreduktion**

**Abb. 6.3: Meßergebnis für eine Alkali-Batterie**

Das Meßergebnis zeigt eine deutlich höhere Ladung von 2,53 Ah. Außerdem ist die Spannungskonstanz wesentlich besser als bei der Zink-Kohle-Batterie. Beide Batterietypen wurden in einer quecksilberfreien Ausführung gekauft, so daß sie in Bezug auf ihre Umweltverträglichkeit vergleichbar sind. Die folgende Übersicht stellt die Ergebnisse der Untersuchung zusammen:

| Typ        | Kapazität | Spannungskonstanz | Kosten   | aufladbar |
|------------|-----------|-------------------|----------|-----------|
| Zink-Kohle | 0,51 Ah   | schlecht          | ca. 1,-  | nein      |
| Alkali     | 2,53 Ah   | gut               | ca. 1,80 | nein      |
| Ni-Cd-Akku | 0,53 Ah   | sehr gut          | ca. 4,00 | 1000 mal  |

Im Vergleich schneidet die Alkali-Batterie gegenüber der Zink-Kohle-Batterie so gut ab, daß sie in den meisten Fällen die bessere Wahl darstellt. Ein weiterer Vorteil dieses Typs ist die wesentlich geringere Selbstentladung und die verbesserte Auslaufsicherheit. Alkali-Zellen sollten daher vor allem auch dort eingesetzt werden, wo Geräte mit geringem Stromverbrauch jahrelang mit einem Batteriesatz auskommen, also z.B. in Taschenrechnern oder Meßgeräten.

Das Preis-Leistungsverhältnis eines Ni-Cd-Akkus erscheint gegenüber der Alkali-Batterie etwa um den Faktor 100 günstiger, wenn man von etwa 1000 Aufladungen ausgeht. Ein Nachteil ist aber die größere Selbstentladung, so daß sich der Einsatz auf häufig gebrauchte Geräte mit einem Strombedarf oberhalb ca. 10

mA beschränkt. Außerdem muß sichergestellt sein, daß Akkus zuverlässig entladen und geladen werden, da sonst ihre Lebensdauer erheblich sinkt. Die Gesamtbilanz muß auch die Umweltverträglichkeit berücksichtigen. Durch den hohen Gehalt an Schwermetall (Cadmium) stellt dieser Akkutyp eine mögliche Umweltgefährdung dar. Mit entscheidend ist dabei der Stand der Wiederverwertungstechnik. Bisher wird vielfach nur der Stahlanteil zurückgewonnen, der problematische Cadmium-Anteil jedoch deponiert.

Im Rahmen eines größeren Projekts könnten Schüler und Schülerinnen unterschiedliche Batterie-Fabrikate vergleichend untersuchen. Auch neue und alternative Bauformen sollten dabei einbezogen werden. So ist z.B. der neue Nickel-Hydrid-Akku in bezug auf Ladung und Umweltverträglichkeit dem Ni-Cd-Akku weit überlegen. Für spezielle Aufgaben hat sich auch die Ladungsspeicherung in Kondensatoren bewährt, so daß auch hier vergleichende Messungen sinnvoll erscheinen.

## 6.2 Ladeverhalten von Akkus

Das richtige Ladeverfahren ist mit entscheidend für die Lebensdauer eines Akkus. Bei Ni-Cd-Akkus treten dabei folgende Probleme auf:

- Eine Tiefentladung schädigt den Akku und kann zu einem Zellschuß führen.
- Eine erhebliche Überladung schädigt den Akku durch Überhitzung und übermäßigen Innendruck.
- Werden Akkus mit unterschiedlicher Vorgeschichte gemeinsam entladen oder geladen, dann besteht die Gefahr der Tiefentladung oder der Überladung.

Das Standard-Ladeverfahren besteht darin, den Akku für 14 Stunden mit I-10 zu laden also mit dem Strom, den der geladene Akku bei zehnstündiger Entladung liefern kann. Bei einer Kapazität von 500mAh müßte z.B. mit 50mA geladen werden. War der Akku zuvor nicht vollständig entladen, dann tritt eine Überladung auf. Bei dieser geringen Stromstärke ist sie jedoch relativ ungefährlich.

Moderne Ladegeräte verwenden verschiedene Verfahren für eine schnellere und sicherere Ladung. Dafür geeignete Akkus müssen "schnelladefähig" sein, also einen höheren Ladestrom aushalten können. Ein Ladeverfahren besteht z.B. darin, den Akku zuerst automatisch vollständig zu entladen, um dann beim Laden die richtige Ladungsmenge zuzuführen. Eine sichere Entladung muß die Tiefentladung vermeiden, also z.B. bei einer Zellenspannung von 0,9V abgebrochen werden. Die korrekte Ladungsmenge ergibt sich aus der Nennkapazität des Akkus. Neuere Ladeverfahren versuchen jedoch, auch an der Ladespannung eine vollständige Aufladung zu erkennen, um ein Überladen zu vermeiden.

Einfache Versuche mit konstanten Ladeströmen lassen sich z.B. mit einem Netzgerät aus der Physiksammlung, einem Ampèremeter und einer kleinen Glühlampe als Vorwiderstand durchführen. Der positive Temperaturkoeffizient des Glühfadens führt zu einer Nachregelung des Stroms bei steigender Ladespannung und damit zu einem relativ konstanten Ladestrom. Die folgende Messung wurde an einem Ni-Cd-Akku mit 500mAh bei einem Ladestrom von 170mA durchgeführt. Neben der Zellenspannung wurde auch die Zelltemperatur gemessen. Der Versuch wurde mit Prisma durchgeführt.

Abb. 6.4: Spannung und Temperatur des Akkus beim Laden

Das Ergebnis zeigt zunächst einen charakteristischen Anstieg der Ladespannung. Kurz vor dem Erreichen der vollständigen Ladung steigt die Spannung nicht mehr. Nach 3.75 Stunden zeigt sich eine kleine Verringerung der Ladespannung. Gleichzeitig steigt zu dieser Zeit die Zelltemperatur am stärksten an. Es läßt sich also deutlich erkennen, daß die Zelle nach 3.75 Stunden vollständig geladen war. Jede weitere Energiezufuhr führte nur zur Erwärmung.

Die insgesamt aufgenommene Ladung läßt sich leicht bestimmen:

$$Q = I \cdot t = 3.75\text{h} \cdot 170\text{mA} = 638 \text{ mAh.}$$

### 6.3 Nutzung der Solarenergie

Solarzellen werden bevorzugt zusammen mit Akkus eingesetzt. Während der Akku bei Sonnenschein geladen wird, kann die gespeicherte Energie zu beliebigen Zeiten eingesetzt werden. Die Planung einer Solaranlage erfordert zunächst eine Prognose der zu erwartenden Energie. Hierzu werden Langzeitmessungen durchgeführt. Das grundsätzliche Vorgehen zeigt der folgende Versuch:

Eine Dünnschicht-Solarbatterie mit der Leistungsangabe 12V, 50mA soll zum Laden von vier Mignonakkus mit einer Kapazität von 500mAh eingesetzt werden. Trotz der hohen Leerlaufspannung darf die Solarbatterie direkt mit den Akkus verbunden werden, da sie sich im wesentlichen wie eine Konstantstromquelle verhält, die Spannung also auf die Ladespannung der Akkus von ca. 5V absinkt. Die maximale Stromstärke von 50mA bei vollem Sonnenlicht ist ebenfalls problemlos. Zum Schutz gegen Entladung bei geringem Lichteinfall sollte allerdings eine Siliziumdiode in Reihe geschaltet werden.

Eine erste grobe Überschlagsrechnung zeigt, daß bei zehnstündigen Sonnenschein eine Ladung von 500mAh zu erwarten wäre. Im Sommer könnten der Akkus also theoretisch an einem Tag geladen werden. Tatsächlich gibt es jedoch viele Gründe für schlechtere Ergebnisse:

- Beim Laden entstehen Verluste von ca. 40%.
- Das Sonnenlicht wird durch Wolken abgeschattet.
- Viele mögliche Aufstellorte werden zeitweise abgeschattet.
- Ohne eine Nachführung trifft das Sonnenlicht meist nur schräg auf die Solarzellen.

Die wirkliche Ladungsbilanz kann nur für einen gegebenen Ort durch Langzeitmessungen ermittelt werden. Dazu fügt man in den Ladestromkreis einen Shuntwiderstand von z.B.  $50\Omega$  ein. Sein Spannungsabfall kann mit der Zelle gemessen werden, um den Ladestrom zu erfassen.

Die folgende Messung zeigt den Verlauf des Ladestroms an einem durchschnittlichen, teilweise bewölkten Tag. Die Messung wurde am frühen Morgen gestartet und nach 12 Stunden beendet. Die Solarbatterie war dabei an der Innenseite eines Fensters befestigt. Der Aufstellort war relativ ungünstig in Bezug auf den Winkel zur Sonne. Außerdem war er zeitweise durch Gebäude abgeschattet. Dementsprechend gering war die erhaltene Ladung von nur 12 mAh. Die Ladedauer für einfache Akkus beträgt unter diesen Bedingungen mehr als einen Monat.

Abb. 6.5: Die gemessene Ladestromkurve eines Tages

Interessierte Kurse könnten sich intensiv mit der Optimierung einer kleinen Solaranlage beschäftigen. Verbesserungen lassen sich durch Aufstellung an einer Gebäude-Südseite, nach Möglichkeit sogar im Freien erzielen. Zusätzlich kann mit dem vertikalen Aufstellwinkel experimentiert werden.



## 7. Versuche mit zusätzlichen Sensoren

Zahlreiche Fragestellungen in Naturwissenschaft und Technik lassen sich mit sehr einfachen und preiswerten Sensoren oder mit ohnehin in den naturwissenschaftlichen Sammlungen vorhandenen Geräten untersuchen. Im folgenden werden Sensoren und Anwendungsbeispiele für die Zelle vorgestellt. Sie sollen eine Spannung im Bereich bis 2,5V abgeben. Die Eingänge der Zelle sind extrem hochohmig, so daß Sensoren kaum belastet werden.

### 7.1 Potentiometer

Ein einfaches Potentiometer eignet sich zur Messung von Winkeln. Es wird als Spannungsteiler an der Referenzspannung der Zelle betrieben. Günstige Widerstände liegen im Bereich 10 k $\Omega$  bis 100 k $\Omega$ . Drahtpotis zeichnen sich meist durch geringere Reibungswiderstände aus.

Potis eignen sich zur Untersuchung mechanischer Schwingungen. Das folgende Bild zeigt die Schwingungen eines Pendels, das direkt mit der Potiachse verbunden wurde.

Abb. 7.1: Schwingungen eines Pendels

Mit der Zelle lassen sich auch Schwingungen in der Natur untersuchen, so z.B. die durch Wind angeregten Schwingungen eines Baumes. Das Poti wird dazu mit einer Schnurrolle versehen und über eine am Baum befestigte Schnur bewegt. Die Meßergebnisse lassen die Diskussion möglicher Sturmschäden zu.

Ähnliche Untersuchungen lassen sich z.B. zu Schwingungen an Kraftfahrzeugen durchführen. Denkbar ist auch eine Untersuchung der Beinbewegung eines Radfahrers oder eines Läufers.

Das Prinzip des Spannungsteilers läßt sich auch sinnvoll für alle Arten nichtlinearer Widerstände einsetzen, wenn sie zusammen mit einem Festwiderstand an die Referenzspannung gelegt werden. Mit LDRs läßt sich auf diese Weise einfach die Helligkeit messen, mit NTCs oder PTCs die Temperatur und mit Feldplatten das magnetische Feld.

### 7.2 Piezo-Wandler

Piezo-Wandler werden als preiswerte Schallgeber eingesetzt. Umgekehrt geben sie aber bei Druckschwankungen erhebliche Spannungen ab. Die Empfindlichkeit ist am größten für Biegeschwingungen. Voraussetzung für die Umsetzung niedriger Frequenzen ist der hochohmige Anschluß. Abb. 7.2 zeigt den Anschluß an die Zelle. Da der Sensor Wechselspannungen abgibt, wird er mit der halben Referenzspannung vorgespannt.

Abb. 7.2: Anschluß eines Piezo-Wandlers an die Zelle

Der Wandler besitzt eine ausreichende Empfindlichkeit, um den Pulsschlag eines Menschen zu untersuchen. Dazu wird er mit der Rückseite z.B. gegen den Unterarm gedrückt. Das folgende Diagramm zeigt eine Messung, die online mit Prisma aufgenommen wurde.

Abb. 7.3: Pulsschlag, gemessen mit dem Piezo-Wandler

Mit einem geeigneten Gewichtsstück auf dem Sensor lassen sich seismische Erschütterungen registrieren. In Gebäuden lassen sich oft Schwingungen durch Gehbewegungen nachweisen.

Abb. 7.4: Bodenschwingungen in einem Gebäude

Auch fallende Regentropfen lassen sich mit diesem Sensor registrieren. Eine mögliche Anwendung liegt in der Messung der Niederschlagsmenge. Dazu muß versucht werden, die Größe der einzelnen Tropfen über die Impulshöhe zu erfassen.

### 7.3 Gleichstrommotor als Drehzahlsensor

Der einfache Gleichstrommotor ist ein brauchbarer Sensor für Winkelgeschwindigkeiten bzw. Drehzahlen. Die Leerlaufspannung des Motors ist streng proportional zur Drehgeschwindigkeit. Motoren aus Kassettenrecordern sind wegen ihres leichten Laufs besonders geeignet.

Abb. 7.5: Induktionsspannung eines frei auslaufenden Motors

Eine Anwendung des Sensors ist die Erfassung von Windgeschwindigkeiten mit Hilfe eines Windrades. Das Meßgerät läßt sich mit einem Tachometer kalibrieren, indem es bei Windstille aus dem fahrenden Auto gehalten wird. Wenn die Übertragungsfunktion bekannt ist, kann eine Definitionsdatei für Prisma erstellt werden.

### 7.4 Induktionsspulen

Magnetische Wechselfelder lassen sich leicht durch die Induktionsspannung in einer Spule erfassen. Die Zelle kann zwar keine Wechselfeldspannung messen, durch den unipolaren Arbeitsbereich kommt es jedoch praktisch zu einer Gleichrichtung, weil nur die positiven Halbwellen gemessen werden. Deshalb kann man auf einen Meßgleichrichter verzichten. Bei Amplituden über 0,5V kann es allerdings zu einer Beeinflussung der anderen Meßkanäle, und damit zu Meßfehlern bei Mehrkanalmessungen kommen.

Über die magnetischen Streufelder elektrischer Geräte lassen sich leicht Einschaltzeiten überwachen. Das folgende Meßergebnis wurde am Transformator eines temperaturgeregelten LötKolbens aufgenommen. Als Induktionsspule wurde eine handelsübliche Pickup-Spule mit Saugnapf für Telefonverstärker eingesetzt. Die induzierte Spannung ist auch ohne Verstärkung ausreichend, um direkt von der Zelle gemessen zu werden. Zur besseren Darstellung wurde das Minimum/Maximum-Erfassungsprogramm nach Kap. 9.3.2 verwendet.

Abb. 7.6: Messung am Transformator eines geregelten LötKolbens

Die Stärke des Streufelds läßt eine eindeutige Unterscheidung der drei möglichen Zustände (ausgeschaltet, Lastbetrieb, Leerlauf) zu. Das Regelverhalten des LötKolbens ist deutlich zu erkennen. Ähnliche Versuche lassen sich z.B. auch am Motor eines Kühlschranks durchführen. So läßt sich z.B. die Einschaltdauer eines Kühlschranks unter verschiedenen Bedingungen wie Umgebungstemperatur, Häufigkeit des Öffnens usw. untersuchen.

### 7.5 Labormeßgeräte mit Schreiberausgang

Viele Labormeßgeräte für Temperatur, pH-Wert, Magnetfelder, elektrische Felder, Ladungen oder kleinste Spannungen besitzen Analogausgänge zum Anschluß eines Schreibers. Sie können direkt mit der Zelle verbunden werden, wenn sie über einen geeigneten Ausgangsspannungsbereich verfügen.

Die korrekte Darstellung der Meßwerte erfordert die Kenntnis der Übertragungskennlinie des Meßgeräts, die z.B. bei der nachträglichen Umrechnung in Meda oder in einer Tabellenkalkulation verwendet wird. Meßgeräte mit Schreiberausgang lassen sich in Unimess nach einer Installation verwenden. Für die Verwendung in Prisma kann eine passende Definitionsdatei erstellt werden.

Das folgende Meßergebnis wurde mit einem pH-Meter und Unimess erstellt. Die Untersuchung zeigt den pH-Wert von Leitungswasser. Der scheinbare Anstieg ist wahrscheinlich auf eine Temperaturänderung während der Messung zurückzuführen. Genauere Untersuchungen müßten mit einer gleichzeitigen Temperaturerfassung durchgeführt werden.

Abb. 7.7: Messung des pH-Wertes von Leitungswasser

## Teil B Programmierung der Zelle

### 8. Die Prozeßsprache

Die Zelle kann als nicht-intelligentes PC-Interface eingesetzt werden, das von einem PC-Programm angesteuert wird. Der Datenaustausch zwischen PC und Zelle findet dabei über die serielle Schnittstelle statt.

Die Programmierung in einer Hochsprache wird durch die Prozeßsprache erleichtert. Sie entlastet den Anwender von hardwarespezifischen Detailoperationen und ermöglicht damit auch Schülerinnen und Schülern die Lösung komplexer Aufgaben. Die Prozeßsprache besteht aus einer Sammlung von Prozeduren und Funktionen, die in Turbo Pascal als Unit zur Verfügung steht oder in das Programm eingebunden werden kann. Die Prozeßsprache steht in gleicher Weise für Comal zur Verfügung. Für Qbasic werden hier Teile der Prozeßsprache vorgestellt.

Das Konzept der Prozeßsprache wurde vom Modul-Bus-System übernommen. Die Zelle bietet gegenüber dem Modul-Bus oder der Compact-Box durch ihren Prozessor und ihren internen Speicher erweiterte Möglichkeiten.

Die einzelnen Sprachelemente bestehen aus einem Befehlswort und oft mehreren Parametern. Im Folgenden sollen die wichtigsten Elemente erläutert werden.

#### Grundwortschatz der Prozeßsprache

##### **AEIN**

Typ: Funktion  
 Rückgabe: Wert (0-255)  
 Parameter: - Eingang (0-3)

##### **Beschreibung:**

Der Meßbereich der analogen Eingänge ist 0 - 2,55 V. An welchem der vier Eingänge die Spannung gemessen wird, wird durch den Übergabeparameter "Eingang" bestimmt. Die Funktion AEIN liefert einen Wert zwischen 0 und 255, wobei der Zahl 0 die Spannung 0 V und der Zahl 255 die Spannung 2,55 V entspricht.

##### **UEIN**

Typ: Funktion  
 Rückgabe: Realzahl  
 Parameter: - Eingang (0-3)

##### **Beschreibung:**

Die Zelle besitzt vier Eingänge, die durch den Übergabeparameter "Eingang" angewählt werden können. Diese Funktion liefert die gemessene Spannung in Volt.

## DEZAUS

Typ: Anweisung

Parameter: - Wert (0-15)

### **Beschreibung:**

Alle digitalen Anschlüsse werden als Ausgänge definiert und durch diese Anweisung gesetzt. Sie sind von 0 bis 3 numeriert. Der Ausgabewert errechnet sich aus der Summe der Stellenwerte ( $2^{\text{Nummer}}$ ) für die zu setzenden Ausgänge. Jede Zahl zwischen 0 (kein Ausgang gesetzt) und 15 (alle Ausgänge gesetzt) entspricht einem bestimmten Muster von gesetzten und nicht gesetzten Ausgängen. Durch den Befehl DEZAUS wird das jeweilige Muster gesetzt, das dem Wert entspricht.

## BINAUS

Typ: Anweisung

Parameter: - Bitmuster ("I", "O", "X", "T")

### **Beschreibung:**

Der Übergabeparameter "Bitmuster" ist eine vierstellige Zeichenkette, wobei jeder Stelle ein bestimmter Ausgang zugeordnet ist. Das erste Zeichen (links) steuert den Ausgang 3, das letzte (rechts) den Ausgang 0. Diese Anweisung definiert alle digitalen Anschlüsse als Ausgänge. Vier Zeichen sind erlaubt und haben folgende Funktion:

"I" - der Ausgang wird gesetzt

"O" - der Ausgang wird gelöscht

"X" - der Ausgang wird nicht verändert

"T" - der Ausgang wird invertiert

## DEZEIN

Typ: Funktion

Rückgabe: Wert (0-15)

Parameter: - keine -

### **Beschreibung:**

Alle digitalen Anschlüsse werden als Eingänge definiert und über diese Funktion abgefragt. Jeder Eingang besitzt eine Nummer zwischen 0 und 3. Der gelesene Wert ergibt sich aus der Summe der Stellenwerte ( $2^{\text{Nummer}}$ ) für die gesetzten Eingänge. Als Ergebnis liefert diese Funktion somit eine Zahl, die zwischen 0 (kein Eingang gesetzt) und 15 (alle Eingänge gesetzt) liegt.

## BINEIN

Typ: Funktion

Rückgabe: Zeichenkette (4-stellig aus "I", "O")

Parameter: - keine -

### **Beschreibung:**

Diese Funktion definiert alle digitalen Anschlüsse als Eingänge und gibt eine vierstellige Zeichenkette bestehend aus "I" und "O" zurück. Ein "I" bedeutet dabei, daß ein Eingang gesetzt ist, während ein "O" auf einen nicht gesetzten Eingang hinweist. Aus der Stellung innerhalb der Zeichenkette kann man erkennen, um welchen Eingang es sich handelt. Das erste Zeichen (links) entspricht dem Eingang 3, das letzte Zeichen (rechts) dem Eingang 0.

## INIT

Typ: Anweisung

Parameter: - keine

### **Beschreibung:**

Diese Anweisung bereitet den Datenaustausch zwischen Rechner und Interface vor. Sie muß aufgerufen werden, bevor die Zelle das erste Mal angesprochen wird.

## WARTE

Typ: Anweisung

Parameter: - Sekunden

### **Beschreibung:**

Obwohl diese Anweisung direkt nichts mit der Prozeßdatenverarbeitung zu tun hat, wurde sie in den Sprachumfang aufgenommen, da sie sich in zahlreichen Fällen als sehr nützlich erwiesen hat. Durch sie wird die weitere Programmausführung für die in Sekunden angegebene Zeit unterbrochen. Der Übergabeparameter ist eine Realzahl.

## Erweiterter Sprachumfang der Prozeßsprache

### STAUS

Typ: Anweisung  
Parameter: Zustand (0,1)

#### **Beschreibung:**

Mit dieser Anweisung wird der Anschluß ST als Ausgang definiert und entsprechend dem Übergabeparameter "Zustand" auf 1 oder 0 gesetzt.

### STEIN

Typ: Funktion  
Rückgabe: Wahrheitswert  
Parameter: - keine -

#### **Beschreibung:**

Durch diese Funktion wird der Anschluß ST als Eingang definiert und gelesen. Ist ST gesetzt, wird WAHR, ansonsten FALSCH zurückgegeben.

### TASTER

Typ: Funktion  
Rückgabe: Wahrheitswert  
Parameter: - keine -

#### **Beschreibung:**

Diese Funktion liest die Start-Taste. Ist sie gedrückt, wird WAHR, ansonsten FALSCH zurückgegeben.

### UBAT

Typ: Funktion  
Rückgabe: Realzahl  
Parameter: - keine -

#### **Beschreibung:**

Diese Funktion gibt die momentane Batteriespannung zurück. Voraussetzung für eine exakte Messung ist, daß der Eingang A4 unbeschaltet ist.

## BLOCKDEF

Typ: Anweisung

Parameter: - Anzahl (1-6136)  
- Eingänge (1-15)  
- Zeit (0 - 65535)

### Beschreibung:

Mit dieser Funktion wird eine Serienmessung vorbereitet. Der erste Parameter legt die Anzahl der Meßwerte fest. Wenn der Speicherplatz ausreicht, können mehrere Meßserien gespeichert werden.

Durch den Parameter "Eingänge" wird festgelegt, welche Eingänge bei der Messung berücksichtigt werden sollen. Jedem Eingang ist eine Zahl zu geordnet: A1=1, A2=2, A3=4, A4=8. Der anzugebende Wert ergibt sich aus der Addition der Zahlen. "Zeit" bestimmt den Abstand zwischen zwei Messungen in Millisekunden.

Die Messung selbst wird durch die Starttaste oder die Anweisung BLOCKSTART (s.u.) ausgelöst. Sofern die Gesamtzahl der Einzelmessungen kleiner als der verfügbare Speicherplatz ist, können mit der Starttaste noch weitere Blöcke gemessen werden.

## BLOCKSTART

Typ: Anweisung

Parameter: - keine -

### Beschreibung:

Diese Anweisung startet eine durch BLOCKDEF definierte Serienmessung.

Sofern noch Platz im Speicher ist, können weitere Messungen mit der Starttaste ausgelöst werden. Die Anweisung BLOCKSTART kann immer nur eine Meßserie auslösen. Erneutes Aufrufen von BLOCKSTART überschreibt die alten Meßwerte im Speicher.

## BLOCKEIN

Typ: Anweisung

Parameter: - Anzahl

### Beschreibung:

Liest eine durch "Anzahl" definierte Zahl von Blöcken der Serienmessungen aus und schreibt sie in das Array Dat\_Puf. Blocklänge und Anzahl der Kanäle werden automatisch erkannt. Der Wert von "Anzahl" sollte nach den mit BLOCKDEF gesetzten Einstellungen und damit nach der Zahl der möglichen Meßserien festgelegt werden, die nacheinander mit der Starttaste ausgelöst werden konnten.

## PROLADEN

Typ: Anweisung

Parameter: - Dateiname

### Beschreibung:

Lädt ein User-Programm mit dem angegebenen "Dateinamen" von der Diskette/Festplatte in den Programm-Bereich der Zelle.

## **PROSTART**

Typ: Anweisung

Parameter: - keine -

### **Beschreibung:**

Diese Anweisung startet ein USER-Programm. Die gleiche Wirkung hat auch das Drücken der Reset-Taste.



## 8.1 Pascal-Speicheroszilloskop

Listing 8.1 zeigt die Anwendung der Prozeßsprache für ein einfaches Einkanal-Speicheroszilloskop. Jede Meßserie wird hier über die Tastatur des PC ausgelöst, sofort eingelesen und grafisch dargestellt.

### Program Speicherossi\_Demo;

```
Uses CRT, Graph, Zelle;
```

```
var Ch : Char;
```

```
procedure Messung;
```

```
begin
```

```
  BlockDef (512,1,1);      {512 Messungen an A0 bei 1ms/Messung}
```

```
  BlockStart;             {vom PC aus starten}
```

```
  Warte (0.6);            {Messung abwarten}
```

```
  BlockEin(1);            {Meßwerte in Dat_Puf einlesen}
```

```
end;
```

```
procedure Darstellung;
```

```
var Graphmode, Graphdriver, n : Integer;
```

```
begin
```

```
  GraphDriver := Detect;
```

```
  InitGraph (GraphDriver, GraphMode, '');
```

```
  ClearDevice;
```

```
    Line (50,40,562,40);      Line (562,40,562,296);
```

```
    Line (562,296,50,296);    Line (50,296,50,40);
```

```
    for n := 1 to 511 do
```

```
      Line (51+n, 296-Dat_Puf[n+11], 50+n, 296-Dat_Puf[n+10]);
```

```
      Repeat until KeyPressed;
```

```
  CloseGraph;
```

```
end;
```

```
begin
```

```
  Init;                    {RS232 und Zelle initialisieren}
```

```
  Repeat
```

```
    Messung; Darstellung;
```

```
    Ch := ReadKey;
```

```
  until Ch = chr(27);
```

```
end.
```

Listing 8.1: Ein einfaches Speicher-Oszilloskop in Turbo Pascal

Abb. 8.1: Ein Bildschirmausdruck des Meßprogramms

## 8.2 Transistor-Meßprogramm

Die Programmierung in einer Hochsprache bietet die Möglichkeit, Meßwerte während der Messung gleichzeitig zu verarbeiten. Ein typische Anwendung zeigt Listing 8.2. Hier wird ein NPN-Transistor untersucht. Aus der Spannungsmessung an vier Meßpunkten werden mehrere Meßgrößen abgeleitet und am Bildschirm in einer einfachen grafischen Darstellung des Meßaufbaus dargestellt. Abb. 8.2 zeigt einen Bildschirmausdruck, der zugleich den Meßaufbau verdeutlicht. Die Eingangsspannung der Schaltung wird über ein Potentiometer vorgegeben, so daß man jeden beliebigen Arbeitspunkt einstellen und die davon abhängigen Größen sofort beobachten kann. Die Zelle verfügt über extrem hochohmige Eingänge, so daß auch bei geringen Basisströmen in der Größenordnung von  $0,1\mu\text{A}$  keine Verfälschung durch das Meßgerät auftritt.

Das Programm kann als Trainingsprogramm zum Verständnis der Funktion eines Transistors eingesetzt werden. Außerdem bietet sich die Möglichkeit an, mit dem Programm Meßreichen unterschiedlicher Parameter aufzunehmen.

### Program NPN;

```
uses CRT, Zelle;
var Ui, Ueb, Uec, Ib, Ic, Ur, U, V : Real;
    Ch : Char;
begin
ClrScr;
Init;
writeln ('
                               +-----+ ');
writeln ('                               +----+   |   +-----+   ');
writeln ('                               | 1 |   +-----+   ');
writeln ('                               |kΩ |   | Ur = |   ');
writeln ('                               +----+   |   +-----+   ');
writeln ('                               |   |   |   |   ');
writeln ('                               +-----+ +-----+   ');
writeln ('                               | Ic = |   |   |   ');
writeln ('                               |   |   |   |   ');
writeln ('                               | V = |   |   |   ');
writeln ('                               |   |   |   |   ');
writeln ('                               +-----+ +-----+   ');
writeln ('                               |   |   |   |   ');
writeln ('                               C +-----+   ');
writeln ('                               +-----+   ');
writeln ('                               |   |   |   |   ');
writeln ('                               |   |   |   |   ');
writeln ('                               +-----+   ');
writeln ('                               | Ib = |   B |   |   |   ');
writeln (' o -----|100 kΩ +-----| +-----+   ');
writeln (' | +-----+   |   |   |   |   |   |   |   |   ');
writeln (' |   |   |   |   |   |   |   |   |   |   ');
writeln (' +-----+   |   |   |   |   |   |   |   |   ');
writeln (' |   |   |   |   |   |   |   |   |   |   ');
writeln (' |   |   |   |   |   |   |   |   |   |   ');
writeln (' |   |   |   |   |   |   |   |   |   |   ');
writeln (' +-----+   |   |   |   |   |   |   |   |   ');
write (' o -----+-----+   ');
repeat
repeat
    Ui := UEin (0);
    Ueb := UEin (1);
    Uec := UEin (2);
    U := UEin (3);
    GotoXY (3,23); write (Ui:5:2,' V');
    GotoXY (30,23); write (Ueb:5:2,' V');
    GotoXY (62,21); write (Uec:5:2,' V');
    Ur := U - Uec;
    GotoXY (62,6); write (Ur:5:2,' V');
    Ib := (Ui - Ueb) / 0.1;
    GotoXY (23,18); write (Ib:4:1,' μA');
    Ic := (U - Uec) / 1;
```

```
GotoXY (48,10); write (Ic:5:2,' mA');
GotoXY (15,7); write (U :5:2,' V ');
if Ib>0 then V := Ic / (Ib / 1000) else V := 0;
GotoXY (16,9); write (V:4:0 );
delay (200);
until KeyPressed;
Ch := ReadKey;
if Ch= ' 'then begin
  Repeat until KeyPressed;
  Ch := ReadKey;
end;
until Ch = chr(27);
end.
```

**Listing 8.2:** Meßprogramm zur Erfassung von Spannungen, Strömen und der Stromverstärkung an einem Transistor

**Abb. 8.2:** Messung an einem Silizium-Kleinsignaltransistor

### 8.3 Bargraphanzeige in Q-Basic

Basic stellt nach wie vor ein geeignetes Werkzeug zur Lösung kleiner Aufgaben dar. Oft kann auf Grundkenntnisse von Schülern und Schülerinnen zurückgegriffen werden, wenn es darum geht, schnell und ohne viel Aufwand eine kleine Meßaufgabe zu lösen.

Seit der Einführung des Betriebssystems MS DOS 5.0 wird statt des bisherigen GW-Basic das neuere und verbesserte Q-Basic ausgeliefert. Deshalb soll hier zumindest ein kleiner Teil der Prozeßsprache auch für Q-Basic vorgestellt werden.

Listing 8.3 zeigt ein Programm zur Realisierung eines vierkanaligen Bargraph-Meßgeräts. Während die Balkenanzeige laufend aktualisiert wird, erfolgt nur nach einem Betätigen der Leertaste eine digitale Meßwertausgabe. Der Benutzer hat daher die Möglichkeit, momentane Meßwerte einzufrieren und gleichzeitig den weiteren Verlauf zu beobachten.

Das Programm benötigt neben der Initialisierung und Rücksetzung der Zelle nur die Funktion Aein zur analogen Eingabe. Zusätzlich wurden jedoch noch die Sprachelemente zur digitalen Ein- und Ausgabe aufgenommen, um gleichzeitig mit den wichtigsten Elementen der Prozeßsprache die Möglichkeit digitaler Steuerungen zu eröffnen.

```

DECLARE SUB dezaus (dwert!)
DECLARE FUNCTION DezEin ()
DECLARE FUNCTION Aein! (Kanal!)
DECLARE SUB SystemReset ()
DECLARE SUB Init ()
DECLARE SUB Sende (Zeichen!)
DECLARE FUNCTION Empfang! ()

Init
SystemReset
SCREEN 7
FOR n = 0 TO 3
  LINE (0, 25 + 40 * n)-(255, 25 + 40 * n)
  FOR i = 0 TO 5
    LINE (i * 50, 22 + 40 * n)-(i * 50, 25 + 40 * n)
  NEXT i
NEXT n
FOR n = 1 TO 4
  LOCATE 5 * n, 1
  PRINT "0    0.5    1    1.5    2    2.5"
NEXT n
DO
  a$ = INKEY$
  FOR n = 0 TO 3
    u = Aein(n)
    LINE (0, 10 + n * 40)-(u, 20 + n * 40), 2, BF
    LINE (u + 1, 10 + n * 40)-(256, 20 + n * 40), 0, BF
  NEXT n
  IF a$ = " " THEN
    FOR n = 0 TO 3
      u = Aein(n)

```

```

        LOCATE 3 + 5 * n, 35
        PRINT u / 100
    NEXT n
END IF
IF a$ = CHR$(27) THEN
    SCREEN 0
    END
END IF
LOOP
SCREEN 0
END

FUNCTION Aein! (Kanal!)
    Sende 7
    Sende (Kanal! + 1)
    Aein! = Empfang
END FUNCTION

SUB dezaus (dwert!)
    dwert = (dwert AND 15) * 16 + 15
    Sende (11)
    Sende (dwert)
END SUB

FUNCTION DezEin
    Sende 12
    DezEin = INT(Empfang / 16)
END FUNCTION

FUNCTION Empfang!
    d$ = INPUT$(1, #1)
    Empfang! = ASC(d$)
END FUNCTION

SUB Init
    OPEN "COM2:9600,N,8,2,CS,DS" FOR RANDOM AS #1
    SystemReset
END SUB

SUB Sende (Zeichen)
    PRINT #1, CHR$(Zeichen);
END SUB

SUB SystemReset
    FOR n = 1 TO 10
        Sende 0
    NEXT
END SUB

```

Listing 8.3: Ein Bargraph-Meßprogramm in Q-Basic

Abb. 8.3: Vier analog und digital angezeigte Meßkanäle

## 9. Programmierung in SIMPEL

Simpel ist eine einfache Programmiersprache für Meß- und Steuerungsaufgaben mit der Zelle. Die grundsätzliche Arbeitsweise eines Steuerrechners läßt sich relativ leicht mit Simpel demonstrieren. Auch jüngere Schülerinnen und Schüler finden erfahrungsgemäß einen leichten Zugang zu dieser Sprache.

Die eigene Programmierung mit Simpel kann aus unterschiedlichen Unterrichtssituationen erwachsen:

- Ein technisch orientierter Kurs kann Simpel für die Lösung von Steuerungsproblemen einsetzen. Beispiele: Ampelsteuerung, Steuerung von Bewegungsmodellen, Regelkreise usw.
- Im naturwissenschaftlichen Unterricht werden besondere Meßprobleme mit Simpel lösbar, wobei nicht die Sprache selbst, sondern das Ergebnis im Vordergrund steht. Beispiele: Triggerung von Messungen, Zeitgesteuerte Auslösung von Messungen, Vorauswertung von Meßergebnissen usw.
- Ein Informatikkurs kann allgemeine Strukturen exemplarisch mit Simpel behandeln, wobei die Programmierung eines autonomen Systems zusätzliche Motivationen bieten kann. Beispiele: Programmierung eines Barcode-Lesers, Morseprogramme, serielle Datenübertragung usw.

Die folgende allgemeine Einführung soll einen Überblick über die Sprache vermitteln.

### 9.1 Sprachumfang und Strukturen

Simpel kennt Prozeduren. Alle Variablen sind global und vom Typ Byte. Parameter werden in globalen Variablen an Prozeduren übergeben. Alle Rechenoperationen verwenden die Variablen X und Y, wobei Ergebnisse immer in X erscheinen. Kontrollstrukturen sind Zählschleifen, bedingte Prozeduraufrufe und bedingte Programmschleifen.

Die Sprache Simpel basiert auf nur wenigen, einfachen Regeln:

- Simpel arbeitet wie ein Taschenrechner mit nur einer Variablen X.
- Rechenoperationen können zusätzlich eine Variable Y verwenden.
- Es sind nur ganze Zahlen im Bereich 0...255 erlaubt.
- Zahlen können dezimal, hexadezimal oder binär geschrieben werden.
- Zahlen können in 20 Merkspeichern zwischengespeichert werden.
- Bis zu ca. 6000 Zahlen können im Datenspeicher (RAM) aufgehoben werden.
- Befehle können zu Prozeduren mit eigenem Namen zusammengefaßt werden.
- Jede Programmzeile besteht aus maximal zwei Worten (Befehlen, Zahlen..).

**Befehlsübersicht:****Laden/Speichern**

|                 |  |    |
|-----------------|--|----|
| X Zahl          | Eine Zahl in X eingeben (z.B. X 100)   |    |
| Y Zahl          | Eine Zahl in Y eingeben                |    |
| XnachY          | Den Wert von X nach Y kopieren         |    |
| YnachX          | Den Wert von Y nach X kopieren         |    |
| TauscheXY       | Die Werte in X und Y tauschen          |    |
| Merk Nummer     | X in einen Merkspeicher (1...20) laden |    |
| LiesMerk Nummer | Einen Merkspeicher nach X auslesen     |    |
| LiesMerk(X)     | Merkspeicher Nummer X auslesen         |    |
| SchreibeRAM     | X ins RAM schreiben                    | 1) |
| LiesRAM         | Einen Wert aus dem RAM lesen           |    |
| AnfangRAM       | RAM an den Anfang setzen               |    |

(Die RAM-Operationen erfolgen immer automatisch in aufsteigender Folge. Beim Erreichen der Obergrenze bricht das gesamte Programm automatisch ab.)

**Eingabe/Ausgabe**

|                |  |    |
|----------------|--|----|
| Messen Kanal   | Über Analogeingang (Kanal 0...3) Spannung messen | 1) |
| AusPort Zahl   | Ausgabe einer Zahl über den Port                 | 1) |
| EinPort        | Einlesen einer Zahl vom Port                     |    |
| StAn           | Steuerport setzen                                |    |
| StAus          | Steuerport zurücksetzen                          |    |
| EinSt          | Steuerport einlesen                              |    |
| AusI/O Adresse | Ausgabe über I/O-Adresse 0...255                 |    |
| EinI/O Adresse | I/O-Adresse 0...255 auslesen                     |    |
| AusRS232 Zahl  | Serielle Ausgabe zum Hostcomputer                | 1) |
| EinRS232       | Serieller Empfang vom Hostcomputer               |    |

- 1) Die Parameter zu diesen Befehlen können wahlweise in X oder als Konstante Übergeben werden. X 15 ... AusPort und AusPort 15 haben die selbe Wirkung. Zahlen dürfen auch hexadezimal (z.B. FFh) oder binär (z.B. 11110000b) angegeben werden.

**Mathematische Operationen**

|            |                              |
|------------|------------------------------|
| Plus Zahl  | X mit einer Zahl addieren    |
| PlusY      | X mit Y addieren             |
| Minus Zahl | Von X eine Zahl subtrahieren |
| Minus Y    | Y von X subtrahieren         |

**Logische Grundfunktionen**

|                |  |
|----------------|--|
| AND Zahl       | X UND Zahl                                       |
| ANDY           | X UND Y  |
| OR Zahl        | X ODER Zahl                                      |
| ORY            | X ODER Y   |
| XOR Zahl       | X EXKLUSIV-ODER Zahl                             |
| XORY           | X EXKLUSIV-ODER Y                                |
| NOT            | NICHT X (invertieren)                            |
| LinksSchieben  | Bits in X um eine Stelle nach links verschieben  |
| RechtsSchieben | Bits in X um eine Stelle nach rechts verschieben |



TauscheHalbbytes    Bit 0...3 mit Bit 4...7 tauschen

### Prozeduren

Prozedur Name    Anfang einer eigenen Prozedur  
 ProzedurEnde    Ende der eigenen Prozedur  
 ProzedurVerlassen    Eine Prozedur verlassen

(Der Name der Prozedur wird wie ein Befehl behandelt. Er kann auch in weiteren Prozeduren verwendet werden. Maximal sind 8 geschachtelte Aufrufe erlaubt. Ein Name darf 20 Zeichen lang sein. Prozedurnamen dürfen wahlweise mit Konstanten aufgerufen werden, wobei X geladen wird.)

### Bedingte Prozeduraufrufe

WennX=Y Name    Aufruf der Prozedur Name, wenn X und Y gleich  
 WennX>=Y Name    Aufruf, wenn X größer oder gleich Y  
 WennX<Y Name    Aufruf, wenn X kleiner als Y  
 WennX<>Y Name    Aufruf, wenn X ungleich Y  
 WennX=0 Name    Aufruf, wenn X gleich Null  
 WennX>0 Name    Aufruf, wenn X größer als Null  
 WennTaste Name    Aufruf, wenn Taste gedrückt  
 WennNichtTaste Name    Aufruf, wenn Taste nicht gedrückt

(Außer Prozeduren dürfen auch alle Befehle bedingt aufgerufen werden, die aus nur einem Wort bestehen, also keinen Parameter brauchen.)

### Zählschleifen (Zähler A...H)

A Zahl    Schleifenzähler A mit einer Zahl laden  
 XnachA    Schleifenzähler A mit X laden  
 Amal Name    A-mal Prozedur Name aufrufen  
 AnachX    Schleifenzähler A auslesen  
 Immer Name    Endlosschleife

(Es gibt insgesamt acht Schleifenzähler, mit denen man Zählschleifen aufbauen kann. Außer Prozedurnamen können auch Ein-Wort-Befehle aufgerufen werden.)

### Bedingte Programmschleifen

SolangeNichtFertig    Schleifenanfang  
 ...    (Programmzeilen)  
 SolangeEnde    Schleifenende  
 Fertig    Den Schleifendurchlauf verhindern  
 NichtFertig    Den Schleifendurchlauf freigeben  
 SolangeX=0    Schleifenbedingung: X gleich Null  
 SolangeX>0    Schleifenbedingung: X größer Null  
 SolangeTaste    Schleifenbedingung: Taste gedrückt  
 SolangeNichtTaste    Schleifenbedingung: Taste nicht gedrückt

(In einer Schleife dürfen beliebig viele Programmzeilen stehen. Schleifen dürfen achtfach geschachtelt werden. Die Befehle Fertig und NichtFertig wirken nur auf die SolangeNichtFertig-Schleife, während die übrigen Schleifen direkt X bzw. den Zustand der Starttaste abfragen.)

### Programmstruktur

Anfang    Anfang des Hauptprogramms  
 Stop    Einfrieren des Programmzustandes  
 Ende    Ende des Hauptprogramms

|            |   |    |
|------------|---|----|
| ;Text      | Kommentartext                           |    |
| Warte Zahl | Zahl mal 1ms warten                     | 1) |
| Nichts     | 2,5µs warten                            |    |
| Uhrstart   | Timer starten für "Warte"               |    |
| Code Text  | Maschinencode oder Hextabellen einfügen |    |

(Der Text nach dem Befehl Code muß in hexadezimaler Form eingegeben werden. Als Trennzeichen zwischen Bytes ist das Komma erlaubt, also Code FF00FF oder Code FF,00,FF.)

## 9.2 Einführung in die Programmierung mit Simpel

Jedes Simpel-Programm besteht wie ein Pascal-Programm aus Prozeduren und einem Hauptprogramm. Prozeduren können im Hauptprogramm und in weiteren Prozeduren genau wie vordefinierte Befehle und Funktionen aufgerufen werden. Damit die Namen der Prozeduren leichter vom vorhandenen Sprachumfang unterschieden werden können, werden alle Prozedurnamen klein geschrieben, während alle Befehle auch Großbuchstaben enthalten. Der Compiler unterscheidet grundsätzlich nicht zwischen Groß- und Kleinbuchstaben, sondern die Unterscheidung dient allein der besseren Lesbarkeit der Programme. Kommentare können an jeder Stelle eingefügt werden und müssen mit einem Semikolon beginnen.

Der grundsätzliche Aufbau eines Simpel-Programms wird im Programm "Blinklicht" gezeigt. Es verwendet die Leuchtdiode der Zelle an der niederwertigsten Leitung des Ports. Da die LED gegen +5V angeschlossen ist, muß die Leitung zurückgesetzt werden, um die LED anzuschalten.

Die Befehle **AusPort** und **Warte** können optional mit einem Parameter versehen werden. Fehlt die Angabe eines Zahlenwerts hinter dem Befehl, wird der aktuelle Wert in X übergeben. Zahlenwerte können binär, hexadezimal oder dezimal angegeben werden.

Das eigentliche Hauptprogramm enthält nur eine Endlosschleife, in der die Prozedur blink immer wieder aufgerufen wird. Nach einem Neustart und nach jedem Reset beginnt das Programm mit einem Sprung zum Anfang des Hauptprogramms.

```
;Programm Blinklicht, steuert LED der Zelle an
;Die Bits 0...3 des Ports sind intern mit dem A/D-Wandler
;verbunden, das Bit 0 zusätzlich mit der LED
;Die Portanschlüsse D1...D4 werden über die Bits 4...7
;angesteuert.
```

```
Prozedur blink
AusPort 11111110b      ;oder FEh, oder 254, LED an
Warte 250              ;250 ms
AusPort 11111111b      ;oder FFh, oder 255, LED aus
Warte 250              ;250 ms
ProzedurEnde
```

```
Anfang
  Immer blink          ;Endlosschleife
Ende
```

Listing 9.1: Ein Blinkprogramm in Simpel

### 9.2.1 Parameterübergabe

Soll eine Prozedur vom Hauptprogramm oder von einer anderen Prozedur einen oder mehrere Parameter erhalten, dann muß bedacht werden, daß grundsätzlich alle Variablen, also X, Y, die Merkspeicher und die Zähler A bis H global verfügbar sind. In diesem Punkt verhält sich Simpel ähnlich wie Basic. Man muß daher genau beachten, welche Prozedur welche Variablen verändert.

In vielen Fällen bietet sich X zur Parameterübergabe an. Prozedurnamen können grundsätzlich mit einem Wert aufgerufen werden, der dann vor dem Aufruf in X geladen wird. Werden viele Parameter benötigt, bieten sich die Merkspeicher an.

Das folgende Programm zur Grenzwertüberwachung demonstriert die Parameterübergabe mit Hilfe der Merkspeicher. Es wird empfohlen, die Merkspeicher in aufsteigender Folge (Merk 1, Merk 2 usw.) zur Parameterübergabe zu benutzen. Das Programm soll laufend die Spannung am ersten Analogeingang überwachen und im Falle einer Überschreitung von 1 V einen Warnton über einen externen Piezo-Schallwandler ausgeben, dessen Frequenz und Länge sich mit dem Ausmaß der Überschreitung ändert. Die Prozedur pieps erhält in Merk 1 die Verzögerungszeit für die Tonerzeugung. Sie wird in der Prozedur warnton aus dem aktuellen Meßwert berechnet, der in Merk 2 aus der Prozedur überwachen übernommen wurde.

Zusätzlich zur Parameterübergabe sollte man von der Möglichkeit Gebrauch machen, den X-Wert in jeder Prozedur zu sichern und wiederherzustellen, die X intern verändert, nicht aber einen Wert in X ausgeben soll. Jede Prozedur muß einen eigenen Merkspeicher benutzen. Es wird empfohlen, dazu die Merkspeicher von Merk 20 an in absteigender Folge einzusetzen. Im Programm "Grenzwert" könnte man ohne Sicherung von X auskommen, weil X bei jeder Messung neu gewonnen wird. Die Sicherung ist aber trotzdem sinnvoll, weil die geschriebenen Prozeduren dann in späteren Programmen problemloser eingesetzt werden können. Auch alle vordefinierten Befehle lassen X und Y unverändert, sofern sie nicht Parameter in X übergeben.

#### **;Programm Grenzwertüberwachung an Kanal 1**

```
;Ein Piezo-Schallwandler ist zwischen GND und dem Anschluß
;D4 anzuschließen
```

```
Prozedur pieps
```

```
Merk 20                ;X sichern
AusPort 01111111b     ;D4 low (Piezo-Summer)
LiesMerk 1             ;Parameter lesen
Warte                 ;Wartezeit
AusPort 11111111b     ;D4 high
LiesMerk 20           ;X wiederherstellen
ProzedurEnde
```

```
Prozedur warnton
```

```
Merk 19                ;X sichern
LiesMerk 2             ;Parameter lesen
Minus 99               ;Grenzwert subtrahieren
Merk 1                 ;Parameter für pieps
A 255                  ;Schleifenzähler laden
Amal pieps             ;255 mal pieps
LiesMerk 19           ;X wiederherstellen
ProzedurEnde
```

```
Prozedur überwachen
```

```
Merk 18           ;X sichern
Messen 1          ;Spannung Messen
Merk 2            ;Parameter für warnton
Y 100             ;Vergleichswert 100
WennX>=Y warnton ;warnton, wenn >= 1V
LiesMerk 18       ;X wiederherstellen
ProzedurEnde
```

```
Anfang
Immer überwachen ;Endlosschleife
Ende
```

**Listing 9.2:** Parameterübergabe an Prozeduren

## 9.2.2 Programmschleifen

In den bisherigen Beispielprogrammen wurde nur die Endlosschleife (Immer) verwendet. Es gibt daneben auch noch zwei andere Schleifentypen. Am Beispiel des Programms "Meßserien" werden beide vorgestellt. Die Wiederholschleife mit den acht möglichen Schleifenzählern A bis H kann eine Prozedur oder einen Befehl bis zu 255 mal aufrufen. In der Prozedur meßserie wird so eine Zählschleife für 100 Messungen programmiert. Dieser Schleifentyp benötigt immer dann eine eigene Prozedur für den Inhalt der Schleife, wenn ein Befehl mit zusätzlichem Parameter oder mehrere Befehle ausgeführt werden sollen.

Die zweite Schleifenkonstruktion, nämlich die der Solange-Schleifen, benötigt keine weitere Prozedur. In der Prozedur warte59900ms wird sie angewandt, um die ganze Wartezeit mit nur einer Prozedur erzeugen zu können. Hier wird die SolangeX>0-Schleife verwendet. X selbst stellt dabei das Zählregister dar. In anderen Programmen kann als Prüfbedingung für die Solange-Schleife der Zustand der Starttaste (SolangeTaste) oder das allgemeine Flag "Fertig" (SolangeNichtFertig) dienen.

Der gesamte Zeitablauf des Programms wird durch die Warte-Befehle gesteuert. Sie erzeugen ein genaues Zeitraster, indem jeweils nur bis zum Ablauf der nächsten Millisekunde gewartet wird. Die Befehle Messen und SchreibeRAM benötigen zusammen weniger als eine Millisekunde, so daß die Prozedur kanal1 insgesamt in genau einer Millisekunde abläuft. Der genaue Beginn des Zeitrasters wird durch den Befehl Uhrstart im Hauptprogramm festgelegt. Der gesamte Ablauf wird durch eine Endlosschleife gesteuert. Sobald allerdings nach ca. einer Stunde der verfügbare Speicherplatz im RAM erschöpft ist, bricht der Befehl SchreibeRAM das ganze Programm ab. Wenn dies nicht erwünscht ist, muß die gesamte Datenmenge durch eine weitere Zählschleife begrenzt werden.

Das Programm stellt ein Beispiel dafür dar, wie eigene Meßprogramme über die Möglichkeiten des Betriebssystems der Zelle hinausgehen können. Die fest implementierten Meßfunktionen erlauben zwar das Festlegen von Meßserien mit hundert Messungen im Millisekundenraster. Sie müssen jedoch einzeln über die Starttaste ausgelöst werden, da eine automatische Wiederholfunktion im Betriebssystem fehlt.

**;Programm Meßserien, einmal pro Minute werden**

;100 Meßwerte im Abstand 1ms gewonnen

```
Prozedur warte59900ms ;wartet 60s - 100ms
Merk 20                ;X sichern
X 239
SolangeX>0             ;239 mal
  XnachY               ;X sichern
  Warte 250            ;250 ms (verändert X)
  YnachX               ;X wiederherstellen
  Minus 1
SolangeEnde
Warte 150              ;plus 150ms
LiesMerk 20           ;X wiederherstellen
ProzedurEnde
```

```
Prozedur kanal1
Merk 19                ;X sichern
Messen 1              ;Meßwert holen
SchreibeRAM           ;abspeichern
Warte 1               ;Millisekundenraster
LiesMerk 19          ;X wiederherstellen
ProzedurEnde
```

```

Prozedur meßserie
Merk 18           ;X sichern
A 100            ;
Amal kanal1      ;100 Meßwerte
warte59900ms     ;Minutenraster
LiesMerk 18      ;X wiederherstellen
ProzedurEnde

```

```

Anfang
AnfangRAM
SchreibeRAM 0    ;Meßparameter:
SchreibeRAM 100 ;100 Messungen
SchreibeRAM 1    ;Kanal 1 an
SchreibeRAM 0    ;Kanal 2 aus
SchreibeRAM 0    ;Kanal 3 aus
SchreibeRAM 0    ;Kanal 4 aus
SchreibeRAM 0    ;Meßzeit
SchreibeRAM 0    ;0 ms + 1ms/Kanal
Uhrstart
Immer meßserie  ;Endlosschleife
Ende

```

### Listing 9.3: Programmschleifen zur Meßwerterfassung

Meßwerte werden immer sequentiell im RAM abgelegt. Bei Verwendung mehrerer Kanäle werden die Daten der Eingänge geschachtelt. Um die gewonnenen Meßwerte später eindeutig zuordnen und auswerten zu können, sollen acht Meßparameter den eigentlichen Meßwerten vorangestellt werden:

1. Highbyte,
2. Lowbyte der Blocklänge
3. Kanal 1 (1/0)
4. Kanal 2 (2/0)
5. Kanal 3 (3/0)
6. Kanal 4 (4/0)
7. Highbyte,
8. Lowbyte der Meßzeit minus Kanalzahl

In dieser Form wird ein Datensatz kompatibel zum Datenformat der Programme Monitor, Prisma oder Zellärm.

## 9.3 Anwendungsbeispiele

Die folgenden Anwendungsbeispiele aus unterschiedlichen Bereichen sollen einen Überblick über typische Aufgabenstellungen und ihre Lösungen mit der Steuersprache Simpel geben. Charakteristisch für alle vorgestellten Anwendungen ist der sinnvolle Einsatz der Zelle als autonome Meß- und Steuerungseinheit. Weitere Anregungen findet man in der Simpel-Programmsammlung [6].

### 9.3.1 Ampelsteuerung

Die Ampelsteuerung eignet sich als einführendes Beispiel in die Steuerungstechnik. Die Zelle kann ein einfaches Ampelmodell mit drei Leuchtdioden und Vorwiderständen ohne zusätzliche Leistungstreiber ansteuern. Ein Anforderungskontakt wird über die Starttaste realisiert. Die einmal programmierte Zelle kann als autonome Steuerungseinheit verwendet werden, so daß ein starker Realitätsbezug deutlich wird.

```
;Steuerung einer Ampel mit Anforderungskontakt
;D1..D3 steuern direkt LEDs in umgekehrter Logik
;Die LEDs werden mit 470 Ohm gegen +5V angeschlossen
```

```
Prozedur warte0,1s      ;Warteprozedur
Warte 100
ProzedurEnde
```

```
Prozedur ampel
AusPort 11101111b      ;Rot
SolangeNichtTaste      ;Warten auf Anforderung
SolangeEnde
A 10
Amal warte0,1s         ;1s
AusPort 11011111b      ;Gelb
A 10
Amal warte0,1s         ;1s
AusPort 10111111b      ;Grün
A 50
Amal warte0,1s         ;5s
ProzedurEnde
```

```
Anfang
Immer ampel            ;Endlos
Ende
```

Listing 9.4: Steuerungsprogramm für eine Ampel

Die Ampelsteuerung steht exemplarisch für ähnliche Aufgaben. Ein technisch orientierter Kurs kann mit Hilfe geeigneter Leistungstreiber und Relaisstufen eine Lichtsteueranlage für Diskos entwickeln. Über angeschlossene Potentiometer kann z.B eine Geschwindigkeitsregelung und eine Programmauswahl für verschiedene Lichtmuster erfolgen.



### 9.3.2 Minumum/Maximum-Meßwerterfassung

Bei Langzeitmessungen von Größen, die sich schnell verändern können, besteht die Gefahr, wichtige Ereignisse zu übersehen, wenn sie zwischen zwei Einzelmessungen auftreten. Für Lärmmessungen werden daher grundsätzlich die Maximalwerte im Meßintervall erfaßt. In anderen Fällen können Minimalwerte genauso wichtig sein. Das folgende Programm erfaßt daher Minima und Maxima. Die Messwerterfassung erfolgt an vier Kanälen im Abstand von jeweils einer Sekunde. Die tatsächliche Abtastung der Meßkanäle findet alle vier Millisekunden statt. Für jedes Meßintervall werden acht Datenbytes gespeichert, und zwar zuerst die Maxima der vier Kanäle und dann die Minima. Die grafische Darstellung der Meßwerte ergibt dadurch unterschiedlich dicke Linien, die die beobachtete Streuung der Meßwerte widerspiegeln. Dieses Programm wurde weiter oben in Kap 7.2 angewandt, um seismische Schwingungen zu erfassen. Abb. 7.4 zeigt ein Meßergebnis, das mit Prisma dargestellt wurde.

#### **;Vierkanalmessung, Minima und Maxima**

;1 Messung benötigt 8 Bytes

Prozedur vierkanal ;benötigt 4ms

Messen 1

Merk 9 ;Zwischenspeicher

TauscheXY

LiesMerk 1 ;Maximum Kanal 1

WennX<Y YnachX

Merk 1

LiesMerk 9 ;Meßwert zurückholen

TauscheXY

LiesMerk 5 ;Minimum Kanal 1

WennX>=Y YnachX

Merk 5

Warte 1

Messen 2

Merk 9 ;Zwischenspeicher

TauscheXY

LiesMerk 2 ;Maximum Kanal 2

WennX<Y YnachX

Merk 2

LiesMerk 9 ;Meßwert zurückholen

TauscheXY

LiesMerk 6 ;Minimum Kanal 2

WennX>=Y YnachX

Merk 6

Warte 1

Messen 3

Merk 9 ;Zwischenspeicher

TauscheXY

LiesMerk 3 ;Maximum Kanal 3

WennX<Y YnachX

Merk 3

LiesMerk 9 ;Meßwert zurückholen

TauscheXY

LiesMerk 7 ;Minimum Kanal 3

```
WennX>=Y YnachX
Merk 7
Warte 1
Messen 4
Merk 9 ;Zwischenspeicher
TauscheXY
LiesMerk 4 ;Maximum Kanal 4
WennX<Y YnachX
Merk 4
LiesMerk 9 ;Meßwert zurückholen
TauscheXY
LiesMerk 8 ;Minimum Kanal 4
WennX>=Y YnachX
Merk 8
Warte 1
ProzedurEnde
```

```
Prozedur meßpunkt
X 0 ;Initialisierung
Merk 1 ;Maximum Kanal1
Merk 2 ;Maximum Kanal2
Merk 3 ;Maximum Kanal3
Merk 4 ;Maximum Kanal4
X 255
Merk 5 ;Minimum Kanal1
Merk 6 ;Minimum Kanal2
Merk 7 ;Minimum Kanal3
Merk 8 ;Minimum Kanal4
A 248 ;248 * 4ms =
Amal vierkanal ;992 ms messen
LiesMerk 1 ;8ms speichern
SchreibeRAM
Warte 1
LiesMerk 2
SchreibeRAM
Warte 1
LiesMerk 3
SchreibeRAM
Warte 1
LiesMerk 4
SchreibeRAM
Warte 1
LiesMerk 5
SchreibeRAM
Warte 1
LiesMerk 6
SchreibeRAM
Warte 1
LiesMerk 7
```

```
SchreibeRAM
Warte 1
LiesMerk 8
SchreibeRAM
Warte 1
ProzedurEnde

Anfang
AnfangRAM
SchreibeRAM 5 ;5*256+220=1500 Messungen
SchreibeRAM 220
SchreibeRAM 1 ;Kanäle 1...4
SchreibeRAM 2
SchreibeRAM 3
SchreibeRAM 4
SchreibeRAM 3 ;3*256+228+4=1000ms
SchreibeRAM 228
Immer meßpunkt
Ende
```

**Listing 9.5:** Meßprogramm für Minima und Maxima

### 9.3.3 Unterbrochene Meßreihen

Meßwerterfassungen an verschiedenen Orten und über längere Zeiträume machen es oft erforderlich, die Zelle in der Zwischenzeit auszuschalten. Im Normalfall wird die Messung dabei zurückgesetzt, so daß jeder neue Meßblock die alten Meßwerte überschreibt. Um trotzdem mit einem Batteriesatz mehrere Einzelmessungen auf einen längeren Zeitraum verteilen zu können, kann das Verhalten der Zelle durch ein kleines Simpel-Programm nach Listing 9.6 geändert werden.

Die Zelle besitzt im unteren Kilobyte ihres RAMs einen weitgehend ungenutzten Speicherbereich. Das Betriebssystem verwendet nur die letzte Adresse (03FF) zum Speichern des aktuellen Betriebsmodus. Prisma verwendet zusätzlich einige Bytes zum Speichern der gewählten Sensoren. Hier wird nun die Adresse 03FE benutzt um zu vermerken, wieviele Meßblöcke bereits gefüllt sind.

Wird die Zelle nach der ersten Messung erneut eingeschaltet, dann wird diese zusätzlich gespeicherte Information zunächst aus dem RAM gelesen, um entsprechend viele Datenadressen zu überspringen. So gelangt die neue Messung direkt hinter die zuletzt ausgeführte. Bei vier Eingangskanälen und 240 Messungen pro Block können insgesamt sechs Messungen ausgeführt werden.

Die Befehle SchreibeRAM und LiesRAM benutzen den Speicher der Zelle immer sequenziell, d.h. ein Adreßzeiger wird jeweils automatisch erhöht. AnfangRAM stellt den Adreßzeiger auf den Anfang des Meßdatenspeichers bei Adresse 0800. Der Adreßzeiger selbst ist unter den Adressen der Merkspeicher 29 (Highbyte) und 30 (Lowbyte) zugänglich. Deshalb können die Prozeduren "blockschreiben" und "blocklesen" durch Verändern dieses Zeigers auf die Adresse 03FE zugreifen.

#### **;Unterbrochene Meßreihen**

;4 Kanäle, je 60s, 240 Messungen

Prozedur blockschreiben

```
Merk 2          ;X sichern
LiesMerk 29     ;Adreßzeiger sichern
Merk 3
LiesMerk 30
Merk 4
X 3             ;Adresse 03FD einstellen
Merk 29
X 254
Merk 30
LiesMerk 2     ;X zurücklesen
SchreibeRAM    ;an die Adresse 3FD
LiesMerk 3     ;Adreßzeiger wiederherstellen
Merk 29
LiesMerk 4
Merk 30
ProzedurEnde
```

Prozedur blocklesen

```
X 3             ;Adresse 03FD einstellen
Merk 29
X 254
Merk 30
LiesRAM
ProzedurEnde
```

```

Prozedur vierkanal
Messen 1
SchreibeRAM
Warte 1
Messen 2
SchreibeRAM
Warte 1
Messen 3
SchreibeRAM
Warte 1
Messen 4
SchreibeRAM
Warte 1
Warte 246      ;Gesamtzeit 250ms
ProzedurEnde

```

```

Prozedur blockmessen
SolangeNichtTaste
SolangeEnde
LiesMerk 1
Plus 1          ;Block erhöhen
Merk 1
blockschreiben ;Block speichern
A 240
Amal vierkanal
ProzedurEnde

```

```

Prozedur erstemessung
SolangeNichtTaste
SolangeEnde
AnfangRAM
SchreibeRAM 0      ;Parameterblock:
SchreibeRAM 240    ;240 Messung
SchreibeRAM 1      ;alle Kanäle
SchreibeRAM 2
SchreibeRAM 3
SchreibeRAM 4
SchreibeRAM 0      ;246ms + 4ms
SchreibeRAM 246
blockmessen
ProzedurEnde

```

```

Prozedur übergehen
B 240              ;einen Block weiter:
Bmal LiesRAM      ;240 Bytes * 4 Kanäle
B 240
Bmal LiesRAM
B 240

```

```
Bmal LiesRAM
B 240
Bmal LiesRAM
ProzedurEnde
```

```
Anfang
X 0
Merk 1          ;Block 0
AnfangRAM
LiesRAM
LiesRAM          ;zweites Byte = 0?
WennX=0 erstemessung
blocklesen       ;wieviele schon gemessen?
Merk 1          ;enthält aktuellen Block
AnfangRAM
A 8
Amal LiesRAM     ;Parameterblock überlesen
LiesMerk 1
XnachA
Amal übergehen  ;Speicherbereich aufsuchen
immer blockmessen
Ende
```

Listing 9.6: Programm zur Durchführung mehrerer Messungen über längere Zeiträume

Jedes neue Starten des Programms fügt weitere Meßdaten an die bisherigen an. Für den allerersten Start muß jedoch eine eindeutige Bedingung dafür festgelegt werden, daß der Speicher neu beschreiben werden soll. Als Bedingung wurde hier festgelegt, daß der Datenspeicher zuvor gelöscht worden ist. Das Meßprogramm überprüft dann die zweite Speicherzelle im Datenspeicher. Wenn sich hier ein Nullbyte findet, wird der erste Meßblock ausgeführt. Um den Speicher zu löschen, benötigt man ein kleines Zusatzprogramm nach Listing 9.7.

**;Programm RAMNEU.SIM, löscht das Daten-RAM**

```
Prozedur leeren
X 0
SchreibeRAM
ProzedurEnde
```

```
Anfang
AnfangRAM
immer leeren
Ende
```

Listing 9.7: Löschen des Meßwertspeichers der Zelle

Zur Vorbereitung einer Meßreihe muß immer zuerst das Löschmodul und dann das Meßprogramm geladen und gestartet werden. Prinzipiell ist es jedoch möglich, die Benutzung zu vereinfachen, indem man ein selbstmodifizierendes Programm schreibt. Es sollte den Speicher nur beim ersten Start löschen und sich dann so verändern, daß jeder weitere Start in die normale Meßfunktion führt.

### 9.3.4 Zählerprogramme

Ein Digitalzähler hat im allgemeinen die drei Grundfunktionen Start, Stop und Reset. Grundsätzlich wird die Gesamtzahl der Ereignisse im Meßzeitraum erfaßt. Nachträglich kann nicht mehr festgestellt werden, ob innerhalb des Meßzeitraums unterschiedliche Häufigkeiten auftraten. Wurde z.B. die Anzahl der Autos an einer Straße über 24 Stunden gezählt, dann läßt sich aus dem Ergebnis zwar die durchschnittliche Verkehrsdichte berechnen, Unterschiede zwischen verkehrsreichen und verkehrsarmen Zeiten werden jedoch nicht mehr sichtbar.

Ganz andere Möglichkeiten ergeben sich, wenn man einen Computer als Zähler einsetzt. Nun kann man nämlich beliebig Zwischenergebnisse speichern bzw. die Messung in kurze Beobachtungszeiträume aufteilen. Einzelne Bereiche lassen sich später beliebig zu größeren Einheiten zusammenfassen, so daß die Meßzeiträume erst bei der Auswertung festgelegt werden müssen.

Das folgende Programm realisiert einen Handzähler, der über die Starttaste der Zelle bedient wird und minutenweise Ergebnisse speichert.

Die korrekte Funktion ist daran erkennbar, daß jedesmal die Leuchtdiode der Zelle aufleuchtet, solange man auf die Starttaste drückt. In der Terminalfunktion des Simpel-Compilers wird nach jeweils einer Minute die Anzahl der gezählten Ereignisse sichtbar, wenn die Zelle weiterhin mit dem PC verbunden bleibt. Das laufende Programm speichert seine Ergebnisse gleichzeitig im RAM der Zelle, so daß autonome Meßwerterfassungen mit anschließender Datenübertragung möglich sind.

Das Programm liefert Daten, die sich mit dem Monitorprogramm oder mit Prisma wie übliche Serienmessungen auslesen lassen. Dabei werden alle Einstellungen in Bezug auf Meßdauer und Anzahl der Messungen unverändert so gelassen, wie sie bei der letzten Serienmessung eingestellt wurden. Sie können also z.B. vom Monitorprogramm aus festgelegt werden. Für einstündige Beobachtungszeiträume gelten die folgenden Parameter: Kanal 1, Meßintervall 60000ms, Blocklänge 60.

**;Ereigniszähler: Eingabe über Starttaste oder Pin 2 am Sammelstecker**  
;Ergebnisse werden minütlich erfasst und abgespeichert

```
Prozedur flanken    ;zählt Impulse
LiesMerk 3          ;Merkspeicher 3 lesen,
Plus 1             ;erhöhen
Merk 3             ;und wieder abspeichern
ProzedurEnde
```

```
Prozedur high      ;wertet "high" aus
LiesMerk 2         ;alter Zustand
WennX=0 flanken   ;gewechselt?
X 1
Merk 2             ;neuer Zustand 1
ProzedurEnde
```

```
Prozedur low       ;wertet "low" aus
X 0
Merk 2             ;neuer Zustand 0
ProzedurEnde
```

```
Prozedur trigger   ;wertet D0 aus
Taste
```

```

WennX>0 high           ;wenn D0=1 : "high"
Warte 1
Taste
WennX=0 low           ;wenn D0=0 : "low"
Warte 1
ProzedurEnde

Prozedur 250ms
A 125
Amal trigger
ProzedurEnde

Prozedur 60s
X 0
Merk 3
B 240
Bmal 250ms
LiesMerk 3           ;Ergebnis lesen
SchreibeRAM         ;abspeichern
LiesMerk 3
AusRS232
ProzedurEnde

Anfang               ;Hauptprogramm:
AnfangRAM           ;Parameterblock:
A 8
Amal LiesRAM
Immer 60s           ;endlos, bis RAM voll
Ende

```

#### Listing 9.8: Ein programmierter Digitalzähler

Eine mögliche Anwendung des Programms liegt z.B. in einer Untersuchung zur Verkehrsdichte. Schüler beobachten eine Straße und drücken für jedes Fahrzeug einmal auf die Starttaste der Zelle. Möglich ist auch die selektive Erfassung nur einer Fahrtrichtung oder bestimmter Fahrzeugtypen. Die so erfaßten Daten können z.B. mit dem Monitorprogramm der Zelle eingelesen und in Tabellenform ausgegeben werden. Zusätzlich ist es oft sinnvoll, die Daten mit externen Programmen weiter auszuwerten.

Das Zählprogramm erfaßt Ereignisse über die Start-Taste. Ohne Änderung ist aber auch eine vollautomatische Erfassung möglich, da der Anschluß dieser Taste von außen zugänglich ist. An Pin 2 (Taste) und Pin 4 (Masse) des Sammelsteckers kann je nach Aufgabe z.B. eine Lichtschranke, ein Trittschalter oder ein Reedkontakt angeschlossen werden.

Möglich ist auch die Anwendung als Digitalzähler für ein Geiger-Müller-Zählrohr. Die Impulse müssen als TTL-Impulse vorliegen und mindestens 2ms lang sein.

Zwei weitere Zählprogramme wurden schon in Kap. 5.3 im Zusammenhang mit Messungen am Fahrraddynamo vorgestellt.



### 9.3.5 Triggerung von Meßreihen

Das Meßprogramm Zellärm stellt die Möglichkeit bereit, Messungen nach eigenen Vorstellungen automatisch auszulösen. Grundsätzlich erfolgt jede Messung in Zellärm über ein in der Zelle aktives Simpelprogramm. Es übernimmt die Zeitsteuerung und bildet aus vielen Meßwerten im Millisekundentakt jeweils den Maximalwert im Meßintervall.

Spezielle Messungen können ausgeführt werden, indem man eigene Steuerprogramme in Simpel schreibt und unter den Optionen 1 bis 3 bereithält. Die Programme müssen in kompilierter Form unter dem Namen J.BIN, K.BIN und L.BIN im Programmverzeichnis von ZELLAERM.EXE stehen.

Während die grundlegenden Aufgaben des Steuerprogramms festliegen, kann insbesondere die automatische Auslösung von Messungen nach unterschiedlichen Kriterien realisiert werden.

Die Steuerprogramme der frei gestaltbaren Optionen 1 bis 3 müssen sich in Bezug auf Meßzeiten und Umfang jeweils an eine der Standardeinstellungen A bis I halten und einen entsprechenden Parameterblock schreiben. Die gewählten Parameter eines optionalen Steuerprogramms werden dem Programm Zellärm nur durch den Parameterblock mitgeteilt.

Das folgende Programm befindet sich auf der Zellärm-Programmdiskette:

#### **;K.SIM als Beispiel für Option 2: Getriggerte**

```
;Auslösung von 12 1-min-Messungen bei
;Triggerleveln von 90 dB(A)
```

```
Prozedur Maximalwert
```

```
Messen 1
```

```
wennX>=Y XnachY ; Y erhält Maximalwert
```

```
warte 1
```

```
ProzedurEnde
```

```
Prozedur Messung
```

```
Y 0
```

```
D 118
```

```
Dmal Maximalwert ; 118 ms
```

```
YnachX
```

```
SchreibeRAM ; benötigt ca 0,3 ms
```

```
Warte 1 ; füllt Millisekunde
```

```
YnachX
```

```
AusRS232 ; benötigt ca 0,8 ms
```

```
Warte 1 ; füllt Millisekunde
```

```
ProzedurEnde ; insgesamt 120 ms
```

```
Prozedur 100mal
```

```
B 100
```

```
Bmal Messung
```

```
ProzedurEnde
```

```
Prozedur Trigger
```

```
NichtFertig
```

```
SolangeNichtFertig
```

```
Messen 1
```

```
Y 230 ; Vergleichswert 90 dB(A)
WennX>=Y Fertig
Warte 10
Messen 1 ; Kontrollmessung
WennX<Y Nichtfertig
SolangeEnde
ProzedurEnde

Prozedur Block
Trigger ; Auslösung über 90 dB(A)
C 5
Cmal 100mal
ProzedurEnde

Anfang
AnfangRAM ; Parameterblock:
SchreibeRAM 1 ; 12 * 500*120 ms
SchreibeRAM 244 ; 1*256+244 = 500
SchreibeRAM 1 ; nur Kanal 1
SchreibeRAM 0
SchreibeRAM 0
SchreibeRAM 0
SchreibeRAM 0
SchreibeRAM 119 ; 0*256ms+119ms+1ms=120ms
A 12
Amal Block
Ende
```

**Listing 9.9: Steuerprogramm für getriggerte Lärmessungen**

### 9.3.6 Zeitgesteuerte Messungen

Langzeitmessungen über 24 Stunden sollten von 0 Uhr bis 24 Uhr eines Tages laufen. Damit die Messung aber schon vor Mitternacht eingerichtet werden kann, muß eine verzögerte Auslösung der Messung erfolgen. Dazu dient das Programm WARTE.SIM.

Das Programm soll eine Messung auslösen, die z.B. über das Monitorprogramm oder Prisma eingerichtet wurde. Nachdem alle Parameter der Messung in die Zelle übertragen wurden, muß das Schaltuhrprogramm geladen und gestartet werden. Der Start wird durch ein kurzes Blinken der Leuchtdiode angezeigt. Darauf kann die Zelle sich selbst überlassen werden.

Das Programm stellt einfache Prozeduren für Verzögerungen von einer Sekunde, einer Minute und einer Stunde bereit. Der eigentliche Start der Messung erfolgt durch einen Sprung ins Betriebssystem der Zelle. Normalerweise werden dort zuerst die aktuellen Einstellungen der Meßserie aus dem RAM in acht Register des Prozessors kopiert. Dann wartet die Zelle auf einen Tastendruck, um mit der Messung zu beginnen. Der Tastendruck sollte hier aber gerade vermieden werden. Deshalb werden die Einstellungen schon im Simpel-Programm aus dem RAM in die Register 53 bis 60 des Prozessors kopiert. Sie lassen sich als Merkspeicher 21 bis 28 ansprechen. Der Einsprung ins Betriebssystem übergeht dann die Abfrage der Starttaste. Das Beispielprogramm verzögert die Messung um 12 Stunden und 30 Minuten. Diese Einstellung läßt sich im Hauptprogramm leicht verändern. Auf der Prisma-Programmdiskette befinden sich die compilierten Schaltuhrprogramme für einige häufig benötigte Zeiten.

```
Prozedur 100ms
Warte 100
ProzedurEnde
```

```
Prozedur 1s
E 10
Emal 100ms
ProzedurEnde
```

```
Prozedur 1min
F 60
Fmal 1s
ProzedurEnde
```

```
Prozedur 1h
G 60
Gmal 1min
ProzedurEnde
```

```
Prozedur starten
AnfangRAM
LiesRAM      ;Einstellungen
Merk 21      ;kopieren
LiesRAM      ;in Register 65..73
Merk 22
LiesRAM
Merk 23
LiesRAM
Merk 24
```

```
LiesRAM
Merk 25
LiesRAM
Merk 26
LiesRAM
Merk 27
LiesRAM
Merk 28
Code 44,45      ; jmp 0245
ProzedurEnde
```

```
Anfang
Ausport 254     ;LED an
Warte 200
Ausport 255     ;LED aus
A 12
Amal 1h         ;12 h
A 30
Amal 1min       ;30 Min
starten
Ende
```

**Listing 9.10:** Eine programmierbare Schaltuhr für Meßreihen

### 9.3.7 Schrittmotoransteuerung

Das Darlington-Treiberarray ULN2803 eignet sich zur Ansteuerung von Schrittmotoren durch die Zelle. Das folgende Programm verwendet einen Schrittmotor als Anzeigeelement für einen Analogwert an A1.

Abb. 9.1: Anschluß eines Schrittmotors an die Zelle

**;4-Bit-Schrittmotorsteuerung über D0...D3 (SCHRITT.SIM)**  
;Anschluß über ULN2803-Treiber

Prozedur links

```
LiesMerk 1           ;aktuellen Zustand laden
LinksSchieben       ;nächste Stellung
Merk 1
OR 00001111b       ;Bits 0 ... 3 hochsetzen
AusPort             ;ausgeben
Warte 5
ProzedurEnde
```

Prozedur rechts

```
LiesMerk 1           ;aktuellen Zustand laden
RechtsSchieben     ;nächste Stellung
Merk 1
OR 00001111b       ;Bits 0 ... 3 hochsetzen
AusPort             ;ausgeben
Warte 5
ProzedurEnde
```

Prozedur größer

```
LiesMerk 2           ;Schrittzähler erhöhen
Plus 1
Merk 2
rechts              ;ein Schritt rechts
ProzedurEnde
```

Prozedur kleiner

```
LiesMerk 2           ;Schrittzähler verkleinern
Minus 1
Merk 2
links               ;ein Schritt links
ProzedurEnde
```

Prozedur anzeigen

```
Messen 1            ;Kanal 1 messen
XnachY
LiesMerk 2           ;Schrittzähler laden
Plus 1
WennX>=Y kleiner   ;dem Meßwert nachführen
LiesMerk 2
WennX<Y größer
```

ProzedurEnde

Anfang

X 10001000b ;Initialisierung des

Merk 1 ;Ausgabezustands

X 10

Merk 2 ;Nullpunkt-Offset

A 250

Amal links ;linker Anschlag: Nullpunkt

Immer anzeigen ;laufend dem Meßwert nachführen

Ende

**Listing 9.11: Ansteuerung eines Schrittmotors als Analoganzeige**

### 9.3.8 I<sup>2</sup>C-Bus-Ansteuerung

Der I<sup>2</sup>C-Bus (Inter-IC-Bus) ist ein Bussystem zur Verbindung mehrerer ICs untereinander. Er wird meist zum internen Datenaustausch zwischen ICs innerhalb eines Geräts benutzt. Der besondere Vorteil liegt in der Verwendung von nur zwei Leitungen, einer Datenleitung und einer Taktleitung. Unterschiedliche Bausteine werden über Adressen angewählt, die ebenfalls über den Bus gesendet werden. Meist wird ein Mikrocontroller als zentrales Steuerungselement eingesetzt, das mehrere Peripheriebausteine ansteuert.

Es gibt zahlreiche Bausteine, wie Digitalports, AD/DA-Wandler, Anzeigentreiber, Uhrenbausteine und Speicher, die das I<sup>2</sup>C-Busprotokoll beherrschen. Sie lassen sich gut für externe Hardware-Erweiterungen der Zelle einsetzen. Außerdem werden diese Bausteine inzwischen vermehrt für eigenständige PC-Interfaces und Funktionsmodelle eingesetzt. Eine mit dem PC realisierte Steuerungsaufgabe kann dann nachträglich für die Zelle angepaßt werden, so daß autonome und ortsunabhängige Systeme entstehen.

I<sup>2</sup>C-Bausteine verfügen über hardwaremäßig einstellbare Adressen, so daß mehrere Bausteine des gleichen Typs am Bus betrieben werden können. Unterschiedliche Bausteine unterscheiden sich in ihrer Grundadresse. Für weitere Einzelheiten des Busprotokolls sei auf die Literatur verwiesen [7].

Das folgende Programm stellt alle Grundroutinen für die Ansteuerung des I<sup>2</sup>C-Bus vor. In der Prozedur "testen" wird ein Portbaustein PCF8574 angesprochen. Die an den Baustein übertragenen Daten werden bei offenen Portanschlüssen unverändert zurückgelesen. Abb.9.2 zeigt den Anschluß des ICs.

Abb. 9.2: Anschluß eines Portbausteins PCF8574 an die Zelle

**;I<sup>2</sup>C-Bus: die beiden höchstwertigen Bits des Ports**

;sind SCL (D7) und SDA (D6)

```
Prozedur I2C_Init
AusPort 11111111b
ProzedurEnde
```

```
Prozedur I2C_Start
AusPort 10111111b ;SDA=0
AusPort 00111111b ;SCL=0
ProzedurEnde
```

```
Prozedur I2C_Stop
AusPort 00111111b ;SDA=0, SCL=0
AusPort 10111111b ;SCL=1
AusPort 11111111b ;SDA=1
ProzedurEnde
```

```
Prozedur Acknowledge
AusPort 00111111b ;SDA=0, SCL=0
AusPort 10111111b ;SCL=1
AusPort 00111111b ;SCL=0
```

```
Prozedur keinAcknowledge
AusPort 01111111b ;SDA=1, SCL=0
AusPort 11111111b ;SCL=1
AusPort 01111111b ;SCL=0
```

ProzedurEnde

Prozedur ausgeben

```

Merk 20           ;Zwischenspeicher
X 8
Merk 19           ;Zähler für 8 Bits
SolangeX>0
LiesMerk 20
AND 128           ;Bit 7 maskieren
RechtsSchieben   ;Nach Bit 6 (SDA)
Plus 00111111b
AusPort           ;SDA setzen
Plus 10000000b
AusPort           ;SCL=1
Minus 10000000b
AusPort           ;SCL=0
LiesMerk 20
LinksSchieben    ;nächstes Bit
Merk 20
LiesMerk 19
Minus 1           ;Zähler - 1
Merk 19
SolangeEnde      ;8 mal
AusPort 01111111b ;SDA=1
AusPort 11111111b ;SCL=1
;EinPort         ;Ack vom IC lesen
;And 64          ;nur zu Testzwecken
;AusRS232        ;"0", wenn IC antwortet
AusPort 01111111b ;SCL=0
ProzedurEnde

```

Prozedur einlesen

```

AusPort 01111111b ;SDA=1, SCL=0
X 8
Merk 19           ;Zähler für 8 Bits
SolangeX>0
AusPort 11111111b ;SCL=1
LiesMerk 20
LinksSchieben
XnachY
EinPort
AND 64           ;SDA lesen
LinksSchieben
LinksSchieben    ;Bit 0
PlusY
Merk 20          ;sichern
AusPort 01111111b ;SCL=0
LiesMerk 19
Minus 1          ;Zähler - 1

```



```
Merk 19
SolangeEnde      ;8 mal
LiesMerk 20
ProzedurEnde

Prozedur testen
I2C_Start
ausgeben 64      ;Adresse 32, schreiben
LiesMerk 1
Plus 1           ;hochzählen
Merk 1
ausgeben         ;Daten an PCF8574
I2C_Stop
I2C_Start
ausgeben 65      ;Adresse 32, lesen
einlesen         ;Daten von PCF8574
AusRS232
keinAcknowledge
I2C_Stop
ProzedurEnde

Anfang
I2C_init
Immer testen
Ende
Listing 9.12: Ansteuerung des I2C-Portbausteins PCF8574
```



## 10. Einführung in die Hardware eines Computers

Die grundlegende Funktion eines Computers läßt sich in der Schule immer nur in Teilbereichen klären. Beginnt man mit der Funktion logischer Schaltkreise, so wird es kaum möglich sein, alle Ebenen bis zur Programmierung in einer Hochsprache lückenlos zu behandeln. Der hier vorgeschlagene Kurs behandelt im wesentlichen die mittleren Ebenen der Hard- und Software. Die genaue Funktion logischer Grundschaltungen sowie das Innenleben des Mikroprozessors wird weitgehend ausgeklammert. Der Mikroprozessor selbst wird also als black box nur in seinen grundlegenden Funktionen behandelt, indem die Wirkung einzelner Maschinenbefehle untersucht wird. Über die Programmierung in Assembler hinaus kann der Aufbau einer einfachen Compilersprache exemplarisch für die Sprache Simpel vermittelt werden.

Es erscheint sinnvoll, die Zelle für einen solchen Kurs zu verwenden, weil sie einerseits eine sehr geringe Komplexität aufweist und weil sie andererseits eine gute Motivation zur Auseinandersetzung mit der hardwarenahen Programmierung bietet, da sich so autonom lauffähige Programme entwickeln lassen. Der in der Zelle verwendete Mikrocontroller aus der 8048-Familie wird auch in der technischen Realität für kleine Steuerungsaufgaben eingesetzt und in Assembler programmiert. Beispiele dafür sind Steuerungen für Ampeln, Waschmaschinen oder Videorecorder. Auch der Personalcomputer selbst bietet ein Beispiel für den Einsatz dieses Prozessors: In der Tastatur sitzt üblicherweise ein 8048-Prozessor, dessen Aufgabe darin besteht, gedrückte Tasten zu erkennen und an den PC zu melden.

Die gesamte Unterrichtsreihe kann entweder von den Hardwaregrundlagen oder von komplexen Anwendungen ausgehen. Je nach Zielsetzung sind also zwei Wege möglich:

- Die Mikroprozessor-Grundlagen führen über den Aufbau einer einfachen Hochsprache zu den Anwendungen,
- oder aus komplexen Anwendungen ergeben sich Fragestellungen nach den Grundlagen.

## 10.1 Binärzahlen

Ein Unterrichtsreihe über die Hardware eines Computers erfordert zunächst eine theoretische Behandlung von Zahlensystemen. Grundlage jedes Prozessors ist die binäre Logik, die nur mit den beiden Grundzuständen Aus (0) und An (1) arbeitet. Diese kleinste Informationseinheit wird als ein Bit bezeichnet. Einheiten von acht Bits nennt man ein Byte. Es laßt sich je nach Basis der verwendeten Zahlensystems binär, hexadezimal oder dezimal angeben. Die folgende Tabelle stellt die drei Zahlensysteme für alle Zahlen vor, die sich mit vier Bits darstellen lassen:

| binär | hexadezimal | dezimal |
|-------|-------------|---------|
| 0000  | 0           | 0       |
| 0001  | 1           | 1       |
| 0010  | 2           | 2       |
| 0011  | 3           | 3       |
| 0100  | 4           | 4       |
| 0101  | 5           | 5       |
| 0110  | 6           | 6       |
| 0111  | 7           | 7       |
| 1000  | 8           | 8       |
| 1001  | 9           | 9       |
| 1010  | A           | 10      |
| 1011  | B           | 11      |
| 1100  | C           | 12      |
| 1101  | D           | 13      |
| 1110  | E           | 14      |
| 1111  | F           | 15      |

Viele einfache Mikrocomputer, wie auch die Zelle, arbeiten mit einer Datenbreite von 8 Bits (= 1 Byte). Der Zahlenbereich reicht dann bis 255 oder hexadezimal FF. Üblicherweise werden alle Zahlenwerte hexadezimal angegeben, weil man sie dann einheitlich zweistellig schreiben kann. Den Schülerinnen und Schülern muß deutlich werden, daß der Mikroprozessor immer Binärzahlen verarbeitet, daß also die Art der Zahlendarstellung nur aus Gründen der besseren Lesbarkeit gewählt wurde.

Da die größte hexadezimale Ziffer (F) den dezimalen Wert 15 hat, ergibt sich bei 16 ein Übertrag auf die zweite Stelle. Die Hexadezimalzahl wird also folgendermaßen in eine Dezimalzahl umgerechnet:

| binär     | hexadezimal | dezimal              |
|-----------|-------------|----------------------|
| 1111 0000 | = F0        | = 15 * 16 = 240      |
| 1010 1010 | = AA        | = 10 * 16 + 10 = 170 |
| 1001 0011 | = 93        | = 9 * 16 + 3 = 147   |

Geeignete Übungen zur Umrechnung zwischen den Zahlensystemen sollen den Schülerinnen und Schülern einen klaren Eindruck der binären Zahlen vermitteln.

## 10.2 Blockstruktur und Speicher

Die Grundelemente der Zelle lassen sich in einem einfachen Blockmodell darstellen. Das Herz des Systems bildet der Mikroprozessor (CPU, Central Processing Unit). Hier findet die eigentliche Verarbeitung von Daten statt. Was genau geschehen soll, ist durch das Programm im Programmspeicher festgelegt. Es besteht aus einer Folge von Zahlen, die dem Prozessor mitteilen, was er tun soll. Verarbeitete Daten können in einem Datenspeicher abgelegt werden. Zusätzlich besteht die Möglichkeit, Daten über den Port, also äußere Anschlüsse der Zelle, oder über das Terminal, also den angeschlossenen PC auszutauschen. Über das Terminal gelangen auch Programme in die Zelle.

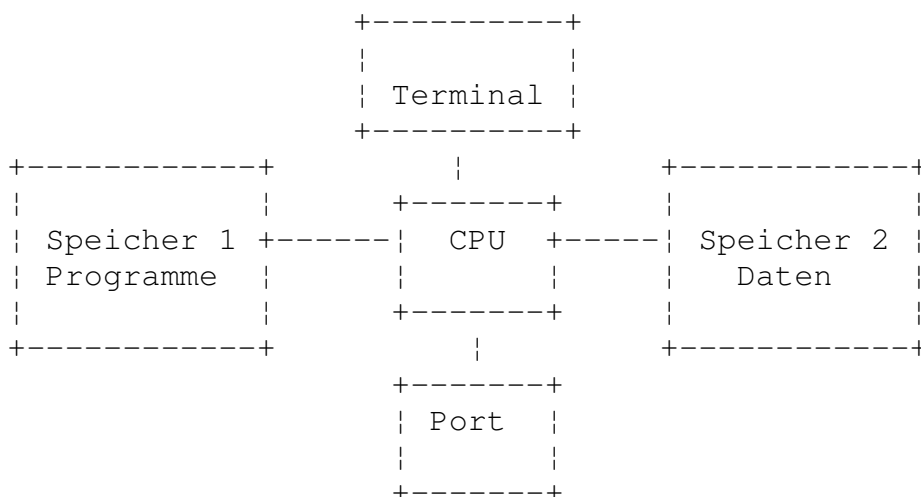


Abb. 10.1: Blockstruktur des Prozessorsystems Zelle

Die tatsächlich vorhandenen Speicher in der Zelle werden durch dieses vereinfachte Modell zur zum Teil beschrieben. Im Gegensatz zu einem Mikroprozessor wie er z.B. im C64-Homecomputer verwendet wird, enthält die Zelle einen sog. Mikrocontroller. Das ist eine integrierte Schaltung, in der sich außer einem Mikroprozessor auch noch Speicher, Ein/Ausgabe-Ports, Zeitgeber und andere Elemente befinden. Diese hohe Integration erlaubt es, sehr einfache Computer zu bauen.

- Das Betriebsprogramm der Zelle befindet sich in einem Festwertspeicher, genauer einem EPROM (ROM= Read Only Memory, EPROM = elektrisch programmierbares und durch UV-Licht lösches ROM). Dieser Speicher ist für den Anwender nicht ohne weiteres zugänglich und braucht deshalb zunächst nicht betrachtet zu werden.
- Das RAM (Random Access Memory, Speicher mit wahlfreiem Zugriff) der Zelle hat eine Größe von 8 Kilobyte (KB), also 8196 Bytes. Ein Bereich von einem KB ist für eigene Programme reserviert. Nur dieser Bereich braucht zunächst als Programmspeicher betrachtet zu werden. Weitere 6 KB dienen der Speicherung von Meßdaten. Ein RAM verliert normalerweise beim Ausschalten des Computers seinen Inhalt.  
In der Zelle bleibt es jedoch mit der Batterie verbunden, so daß Meßdaten und Programme batteriegepuffert erhalten bleiben. Der Programmspeicher der Zelle erscheint daher als ein nicht-flüchtiger Speicher.
- Der Mikrocontroller selbst enthält ein zusätzliches kleines Daten-RAM mit einer Größe von 128 Bytes. Die einzelnen Speicherzellen werden auch als "Register" bezeichnet. Sie werden für die Zwischenspeicherung von Rechenergebnissen und für besondere Funktionen des Prozessors benötigt. Dieses interne RAM des Mikrocontrollers ist flüchtig, d.h. es verliert seinen Inhalt mit dem Ausschalten der Zelle.

- Auch der Port der Zelle kann als ein Speicher betrachtet werden. Er hat eine Breite von acht Bits, d.h. man kann dort ein Byte speichern. Die Zustände der einzelnen Bits sind dabei über acht Anschlüsse herausgeführt, von denen vier dem Anwender an den Anschlußbuchsen der Zelle zur Verfügung stehen. Hier können z.B. Leuchtdioden (LED) angeschlossen werden. Die interne LED der Zelle ist ebenfalls am Port angeschlossen und erlaubt einfache Versuche.

Die Schülerinnen und Schüler müssen zunächst verstehen, daß einzelne Bytes in einem Speicher über "Adressen" zugänglich werden. Mit Hilfe der integrierten Entwicklungsumgebung des Simpel-Compilers können über die Funktion Hexedit/Programm Speicherbereiche des Programm-RAMs angesehen und verändert werden. Entsprechend erlaubt Hexedit/Register den Zugriff auf das Daten-RAM.

Abb. 10.2: Das Editorfenster für den Programmspeicher

Abb. 10.3: Das Editorfenster für den Registerspeicher

Einfache Übungen beinhalten zunächst das Lesen und Verändern gezielter Speicheradressen. Die Schülerinnen und Schüler können dabei den Unterschied zwischen nicht-flüchtigen und flüchtigen Speichern erkennen, indem sie eingegebene Dateninhalte nach dem Aus- und Wiedereinschalten der Zelle erneut auslesen. Der nicht-flüchtige Programmspeicher ist in seiner Wirkung mit einer Diskette vergleichbar, deren Inhalt zwar dauerhaft ist, der jedoch auch bewußt verändert werden kann.

### 10.3 Binäre Codierung

Für Schülerinnen und Schüler mag das Bild vom Computer weitgehend durch die Vorstellung einer "denkenden Maschine" geprägt sein. Die Vorstellung wird teilweise durch den Umgang mit höheren Programmiersprachen oder Anwenderprogrammen gefördert. Demgegenüber muß eine Unterrichtsreihe über die Hardware eines Computers bewußt machen, daß die einzige Verständigung zwischen Mensch und Maschine letztlich aus Binärzahlen besteht. Sie werden vom Mikroprozessor nach relativ wenigen Regeln verarbeitet. Die Programmierung besteht also darin, die Zahlen zu finden, die den Prozessor zu der gewünschten Aktion veranlassen.

Ein einfacher Zugang zum Verständnis dieser grundlegenden Tatsache besteht darin, zunächst nur vorgegebene einfache Programme einzutippen und zu starten. Die Schülerinnen und Schüler lernen dabei, daß es auf jedes einzelne Bit ankommt, und daß kleinste Fehler bereits die Funktion des Programms beeinträchtigen. Ein erstes Programm kann z.B. aus nur fünf Bytes bestehen: **23 00 39 84 00**

|      | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | A  | B  | C  | D  | E  | F  |
|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0400 | 23 | 00 | 39 | 84 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0410 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

Listing 10.1: Maschinenprogramm zum Einschalten der Leuchtdiode

Dieses Programm schaltet die eingebaute Leuchtdiode der Zelle ein. Es eignet sich auch dazu, die Wirkung des batteriegepufferten RAMs der Zelle zu erproben. Wird die Zelle bei laufendem Programm ausgeschaltet, dann führt das spätere Einschalten zu einem Neustart des Programms. Die gleiche Wirkung zeigt ein Betätigen der RESET-Taste der Zelle.

Ein etwas komplexeres Programm könnte z.B. einen Blinker mit selbstveränderlicher Blinkfrequenz sein. Das genaue Verständnis des Programms kann erst später erreicht werden.

|      | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | A  | B  | C  | D  | E  | F  |
|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0400 | BD | FF | 23 | FE | 39 | 94 | 10 | 23 | FF | 39 | 94 | 10 | 84 | 02 | 00 | 00 |
| 0410 | CD | FD | AF | BE | 60 | EE | 15 | EF | 13 | 83 | 00 | 00 | 00 | 00 | 00 | 00 |

Listing 10.2: Ein Blinkerprogramm mit wechselnder Frequenz

## 10.4 Assemblerprogrammierung

Damit Schülerinnen und Schüler verstehen können, wie der Mikroprozessor den binären Code verarbeitet, müssen zunächst einige Informationen zu internen Aufbau gegeben werden.

- Der "Akku" ist der wichtigste Speicher des Mikroprozessors. Er enthält nur ein Byte, das entsprechend der durchgeführten Aktion verändert wird.
- Der "Port" ist ebenfalls ein Speicher mit einem Byte, dessen einzelne Bits jedoch außen zugänglich sind.
- Der "Adreßzähler" ist ein Speicher, dessen Inhalt immer die Adresse des gerade bearbeiteten Bytes enthält.

Mit diesen Vorstellungen läßt sich die Funktion des ersten Programms (23 00 39 84 00) klären:

23: Dieser Befehl bewirkt, daß die Zahl in der nächsten Adresse in den Akku geladen wird.

39: Der Inhalt des Akkus wird an den Port ausgegeben. Die interne LED ist gegen die positive Betriebsspannung geschaltet und leuchtet beim Zurücksetzen von Bit 0 auf.

84: Der Adreßzähler wird mit dem folgenden Byte geladen, so daß das Programm an der angegebenen Stelle fortfährt. Im Beispiel wird ein wiederholter Sprung zum Anfang des Programms durchgeführt.

Insgesamt werden nur wenige Mikrosekunden für die Bearbeitung dieser Befehle benötigt. Aber das Programm wiederholt sich endlos, bis es abgebrochen wird. Man könnte den "Sprungbefehl" 84 auch weglassen. Aber dann würden die folgenden zufälligen Zahlen als Befehle interpretiert, und das Ergebnis wäre zufällig und könnte zu einem Absturz führen.

Um Maschinenprogramme leichter lesbar zu machen, wird die sog. Assembler-Schreibweise eingeführt. Die einzelnen Programmbytes werden zusammen mit den Adressen, Mnemonics als "Merkwörter" und zusätzlichen Kommentaren aufgeschrieben. Die Schülerinnen und Schüler sollen verstehen, daß es sich hierbei nur um eine verbesserte Schreibweise handelt, während der Mikroprozessor nach wie vor nur die eigentlichen Programmbytes verarbeitet. Das erste Programm sieht nun so aus:

| Adresse | Befehl | Operand | Mnemonic   | Kommentar  |
|---------|--------|---------|------------|------------|
| 0400    | 23     | 00      | MOV A, #0  | 0 laden    |
| 0402    | 39     |         | OUTL P1, A | ausgeben   |
| 0403    | 84     | 00      | JMP 0400   | zum Anfang |

Listing 10.3: Ein Assemblerlisting

Die Mnemonics sind abgekürzte, englische Wörter. MOV steht für move (=bewege, verschiebe). MOV A, #0 bedeutet, daß die Null in den Akku (A) gelangen soll. Links steht das Ziel, rechts die Quelle. Das Doppelkreuz (#) kennzeichnet die eingebene Zahl als Datenbyte. OUTL steht für output logical (Ausgabe binärer Daten). Die Daten sollen vom Akku kommen und zum Port 1 (P1) gelangen. JMP bedeutet jump (springe). Die Zieladresse hinter dem Befehl wird ohne Doppelkreuz geschrieben.

Die Assembler-Schreibweise ist wesentlich übersichtlicher als eine Liste von Zahlen. Man kann z.B. sehr leicht erkennen, welche Operanden man ändern könnte. Die Schülerinnen und Schüler können in einer Versuchsreihe den Operanden in Adresse 0401 ändern, um herauszufinden, welches Bit die LED der Zelle steuert. Dabei ergibt sich, daß nur Bit 0 die Leuchtdiode anschaltet, wenn es zurückgesetzt ist.

```
xxxx xxx0 : LED an
xxxx xxx1 : LED aus
```

Die oberen vier Bits sind am Sammelstecker der Zelle herausgeführt. Mit einer kleinen Zusatzhardware, z.B. einem Ampelmodell mit drei LEDs, kann die Ansteuerung dieser Steuerausgänge erprobt werden.



## 10.5 Register, Rechenbefehle und Zählschleifen

Der Mikrocontroller 80C39 enthält die acht "Arbeitsregister" R0 ... R7. Das sind Speicherzellen in den ersten acht Registeradressen. Sie lassen sich mit speziellen Befehlen laden und lesen. Die folgenden Befehle laden Register mit einem Datenbyte:

```
B8 X    MOV R0,#X    die Zahl X in R0 laden
B9 X    MOV R1,#X    die Zahl X in R1 laden
usw.
BF X    MOV R7,#X    die Zahl X in R7 laden
```

Durch ein kleines Programm kann z.B. R7 gezielt geladen werden, um die Lage dieses Registers mit Hexedit/Register im Datenspeicher des Prozessors zu erkennen:

| Adresse | Befehl | Operand | Mnemonic   | Kommentar      |
|---------|--------|---------|------------|----------------|
| 0400    | BF     | 07      | MOV R1,#07 | 07 in R7 laden |
| 0402    | 84     | 00      | JMP 00     | zum Anfang     |

Listing 10.4: Laden des Registers 7

In gleicher Weise kann auch das Register R5 lokalisiert werden. Alle anderen Register können scheinbar nicht geändert werden, weil sie auch vom Betriebsprogramm der Zelle verwendet werden.

Neben dem direkten Laden eines Registers ist auch der Datenaustausch mit dem Akku möglich. Im Folgenden wird jeweils nur der Befehl für R7 genannt. Die anderen Register lassen sich entsprechend behandeln, wenn man die zweite hexadezimale Ziffer des Befehls in 8 (R0), 9 (R1), A (R2), B (R3), C (R4), D (R5) und E (R6) ändert.

```
BF X    MOV R7,#X    die Zahl X in R7 laden
AF      MOV R7,A     den Akkuinhalt in R7 laden
FF      MOV A,R7     den Inhalt von R7 in den Akku laden
```

Im Normalfall arbeitet der Prozessor mit der "Registerbank 0" im Adreßbereich 0...7. Er kann seine acht Arbeitsregister aber auch in einen anderen Adreßbereich legen. Durch eine "Bankumschaltung" auf die "Registerbank 1" wird bewirkt, daß die Register an den Adressen 18 bis 1F im Datenspeicher stehen. Das hat für die Zelle den Vorteil, daß nun alle Register vom Betriebssystem ungestört verwendet und beobachtet werden können.

```
D5      SEL RB1     Registerbank 1 wählen
C5      SEL RB0     wieder Registerbank 0 wählen
```

Alle übrigen Adressen im internen RAM des Prozessors können nicht durch einfache Befehle, sondern nur indirekt angesprochen werden. Die indirekte Adressierung erfordert das Laden der Adresse in das Register R0. Danach kann man durch spezielle Befehle auf diese Adresse zugreifen. Der "Klammeraffe" @ in den Mnemonics (z.B. MOV @R0,A) weist darauf hin, daß nicht das Register R0 selbst gemeint ist, sondern die Speicherzelle, deren Adresse in R0 steht. Die indirekte Adressierung ist auch über R1 möglich, jedoch reicht es hier aus, mit nur einem Adreßregister zu arbeiten.

```
B8 Adr  MOV R0,#Adr  Die Adresse in R0 laden
B0      MOV @R0,#X   X in die Adresse Adr schreiben
A0      MOV @R0,A    den Akkuinhalt dort speichern
F0      MOV A,@R0    den Inhalt der Adresse laden
```

Spezielle Rechenbefehle des Prozessors verwenden meist den Akku und die Arbeitsregister. Die folgende

Auswahl reicht für zahlreiche Versuche aus:

|      |           |                                       |
|------|-----------|---------------------------------------|
| 1F   | INC R7    | den Inhalt von R7 um 1 erhöhen        |
| CF   | DEC R7    | den Inhalt von R7 um 1 verkleinern    |
| 17   | INC A     | den Inhalt des Akkus um 1 erhöhen     |
| 07   | DEC A     | den Inhalt des Akkus um 1 verkleinern |
| 03 X | ADD A, #X | Die Zahl X zum Akkuinhalt addieren    |
| 6F   | ADD A, R7 | R7 zum Akkuinhalt addieren            |
| 37   | CPL A     | alle Bits im Akku umkehren            |
| 53 X | ANL A, #X | logisch UND                           |
| 43 X | ORL A, #X | logisch ODER                          |

Im Folgenden werden einige kleine Rechenprogramme vorgestellt, die sich für Übungen eignen:

| Adresse | Befehl | Operand | Mnemonic | Kommentar  |
|---------|--------|---------|----------|------------|
| 0400    | 1F     |         | DEC R7   | R7 minus 1 |
| 0401    | 84     | 00      | JMP 0400 | zum Anfang |

Listing 10.5: Registers 7 wird laufend dekrementiert

Ergebnis: Der Inhalt von R7 wird so schnell verkleinert, daß der Inhalt beim Stop nicht vorhergesagt werden kann. Das Programm bildet also einen "Zufallsgenerator".

| Adresse | Befehl | Operand | Mnemonic    | Kommentar     |
|---------|--------|---------|-------------|---------------|
| 0400    | B8     | 20      | MOV R0, #20 | Adresse 20    |
| 0402    | F0     |         | MOV A, @R0  | laden         |
| 0403    | 03     | 07      | ADD A, #07  | plus 7        |
| 0405    | A0     |         | MOV @R0, A  | speichern     |
| 0406    | 84     | 06      | JMP 0406    | Warteschleife |

Listing 10.6: Der Inhalt der Speicheradresse 20 erhöht sich um 7.

Ergebnis: Nach jedem Start des Programms kann mit beobachtet werden, daß das Byte in Adresse 20 um 7 erhöht wurde. Der Übertrag über FF geht verloren.

| Adresse | Befehl | Operand | Mnemonic   | Kommentar       |
|---------|--------|---------|------------|-----------------|
| 0400    | 23     | 09      | MOV A, #09 | 09 laden        |
| 0402    | 37     |         | CPL A      | Bits umkehren   |
| 0403    | 03     | 03      | ADD A, #03 | plus 3          |
| 0405    | 37     |         | CPL A      | Bits umkehren   |
| 0406    | AF     |         | MOV R7, A  | in R7 schreiben |
| 0407    | 84     | 07      | JMP 0407   | Warteschleife   |

Listing 10.7: Programm für die Subtraktion 9 - 3

Ergebnis: Im Register 7 erscheint die Zahl 06. Der Prozessor besitzt keinen Subtraktionsbefehl, so daß der Additionsbefehl auf Umwegen eingesetzt werden muß.

Zählschleifen lassen sich unter Verwendung der Register R0 bis R7 bilden. Der Befehl DJNZ (Decrease and Jump if Not Zero, Verkleinere um Eins und springe, wenn das Ergebnis noch nicht Null ist) erfordert die

Angabe einer 8-Bit Zieladresse. Er erreicht grundsätzlich ein Ziel in der aktuellen "Seite", also z.B. im Bereich 0400 bis 04FF. Seitengrenzen können nicht übersprungen werden. Alle einfachen Übungen verwenden jedoch ohnehin kleine Programme mit wesentlich weniger als 256 Bytes.

EF            DJNZ   R7,Adr.            Springe (R7)-mal nach Adr.

Bevor dieser Befehl ausgeführt werden kann, muß das Register mit der Anzahl der Wiederholungen geladen werden.

| Adresse | Befehl | Operand | Mnemonic   | Kommentar       |
|---------|--------|---------|------------|-----------------|
| 0400    | BF     | 09      | MOV R7,#09 | R7 mit 09 laden |
| 0402    | 1D     |         | INC R5     | R5 plus 1       |
| 0403    | EF     | 02      | DJNZ R7,02 | 9 mal springen  |
| 0405    | 84     | 05      | JMP 0405   | Warteschleife   |

Listing 10.8: Der Inhalt von R5 wird genau neun mal um Eins erhöht.

Zählschleifen können verwendet werden, um Programme langsamer ablaufen zu lassen:

| Adresse | Befehl | Operand | Mnemonic     | Kommentar |
|---------|--------|---------|--------------|-----------|
| 0400    | 23     | 00      | MOV A,#0     |           |
| 0402    | 39     |         | OUTL P1,A    | LED an    |
| 0403    | BF     | FF      | MOV R7,#FF   |           |
| 0405    | EF     | 05      | DJNZ R7,0405 | FF mal    |
| 0407    | 23     | FF      | MOV A,#FF    |           |
| 0408    | 39     |         | OUTL P1,A    | LED aus   |
| 0409    | 84     | 09      | JMP 0409     | Ende      |

Listing 10.9: Eine Verzögerungsschleife mit R7

Ergebnis: Die LED blinkt einmal kurz auf, da die erreichte Verzögerung für 255 Schleifendurchläufe etwa 1ms beträgt.

Größere Wartezeiten erhält man mit ineinander geschachtelten Zählschleifen. Als Zähler müssen nun zwei Arbeitsregister eingesetzt werden.

| Adresse | Befehl | Operand | Mnemonic     | Kommentar     |
|---------|--------|---------|--------------|---------------|
| 0400    | 23     | 00      | MOV A,#0     |               |
| 0402    | 39     |         | OUTL P1,A    | LED an        |
| 0403    | BF     | FF      | MOV R7,#FF   | Warteschleife |
| 0405    | BE     | FF      | MOV R6,#FF   |               |
| 0407    | EE     | 07      | DJNZ R6,0407 |               |
| 0409    | EF     | 05      | DJNZ R7,0405 |               |
| 040B    | 23     | FF      | MOV A,#FF    |               |
| 040D    | 39     |         | OUTL P1,A    | LED aus       |
| 040E    | BF     | FF      | MOV R7,#FF   | Warteschleife |
| 0410    | BE     | FF      | MOV R6,#FF   |               |
| 0412    | EE     | 12      | DJNZ R6,12   |               |
| 0414    | EF     | 10      | DJNZ R7,10   |               |
| 0416    | 84     | 00      | JMP 0400     | Zum Anfang    |

Listing 10.10: Ein Blinkprogramm mit geschachtelten Zählschleifen

Ergebnis: Die LED blinkt langsam. Der Sprungbefehl am Ende des Programms bildet eine Endlosschleife.

## 10.6 Unterprogramme

Mehrfach benutzte Programmteile lassen sich vorteilhaft als Unterprogramme schreiben. Ein Unterprogramm wird mit einem CALL-Befehl aufgerufen und muß mit einem RET-Befehl abgeschlossen werden. Nach dem CALL-Befehl wird wie nach dem JMP-Befehl nur eine 8-Bit Adresse angegeben. Die höherwertigen Adreßbits sind im Maschinenbefehl selbst enthalten, d.h. es gibt jeweils einen Befehl für jede der acht möglichen Seiten. Es genügt jedoch zunächst, nur die Befehle für die Seite 4 (0400 ... 04FF) anzugeben.

|    |     |      |     |                     |
|----|-----|------|-----|---------------------|
| 94 | Adr | CALL | Adr | Unterprogrammaufruf |
| 83 |     | RET  |     | Rücksprung          |

Unterprogramme können irgendwo im Speicher stehen. Schreibt man sie an relativ hohe Adressen, dann kann das Hauptprogramm leicht erweitert werden, ohne daß das Unterprogramm geändert wird. Das folgende Programm bildet ebenfalls einen Blinker, diesmal jedoch mit einer Warteschleife im Unterprogramm "Warten"

| Adresse | Befehl | Operand | Mnemonic   | Kommentar  |
|---------|--------|---------|------------|------------|
| 0400    | 23     | 00      | MOV A, #0  |            |
| 0402    | 39     |         | OUTL P1, A | LED an     |
| 0403    | 94     | 20      | CALL 0420  | warten     |
| 0405    | 23     | FF      | MOV A, #FF |            |
| 0407    | 39     |         | OUTL P1, A | LED aus    |
| 0408    | 94     | 20      | CALL 20    | warten     |
| 040A    | 84     | 00      | JMP 0400   | Zum Anfang |

Unterprogramm "Warten"

|      |    |    |               |               |
|------|----|----|---------------|---------------|
| 0420 | BF | FF | MOV R7, #FF   | Warteschleife |
| 0422 | BE | FF | MOV R6, #FF   |               |
| 0424 | EE | 24 | DJNZ R6, 0424 |               |
| 0426 | EF | 22 | DJNZ R7, 0422 |               |
| 0428 | 83 |    | RET           |               |

Listing 10.11: Verwendung eines Unterprogramms

An dieser Stelle der Unterrichtsreihe läßt sich nachträglich das Maschinenprogramm nach Listing 10.2 disassemblieren und analysieren. Dort wird einem Unterprogramm der Schleifenwert einer Warteschleife übergeben, so daß die Wartezeit vom Hauptprogramm aus beeinflußt werden kann.

Unterprogrammaufrufe können auch Betriebssystem-Funktionen der Zelle im ROM aufrufen. So können komplexere Programme geschrieben werden, ohne daß man zu weit in technische Einzelheiten eindringen muß. Z.B. läßt sich die serielle Schnittstelle der Zelle durch einfache Unterprogrammaufrufe benutzen. Bytes können über die Terminalfunktion des Sempel-Compilers gesendet und empfangen werden.

34 00 CALL 0100 "Senden", die Zelle sendet die Zahl im Akku

34 27 CALL 0127 "Empfangen", die Zelle empfängt eine Zahl und übergibt sie im Akku

Das folgende Programm verwendet die serielle Schnittstelle für ein einfaches Rechenprogramm:

| Adresse | Befehl | Operand | Mnemonic   | Kommentar |
|---------|--------|---------|------------|-----------|
| 0400    | 34     | 27      | CALL 0127  | Empfangen |
| 0402    | 03     | 7F      | ADD A, #7F | plus 7F   |

```

0403      34      00      CALL 0100      Senden
0405      84      05      JMP  0405      Ende

```

## Listing 10.12: Datenaustausch über die serielle Schnittstelle

Ergebnis: Die empfangene Zahl wird mit 7F addiert und zurückgesandt.

Fertige Unterprogramme bezeichnet man auch als "Makrofunktionen" oder kurz "Makros". Einige nützliche Makros eignen sich speziell für Meßprogramme.

```

34 51  CALL 0151  "Messen", beim Aufruf muß der Akku den Kanal (1...4) enthalten.
34 83  CALL 0183  "RAM-Reset", den Zeiger auf den Anfang des Datenbereichs im RAM
          (0800) zurücksetzen
34 8F  CALL 018F  "Schreiben", den Akkuninhalt im RAM speichern
34 AB  CALL 01AB  "Lesen", ein Byte aus dem RAM lesen
34 C1  CALL 01C1  "Warten", eine Millisekunde lang warten

```

Das folgende Programm sammelt 100 Meßwerte von Kanal 1. Sie können z.B. vom Monitorprogramm gelesen und ausgewertet werden. Die ersten acht Adressen des Meßdaten-RAMs sollen die Parameter der Messung enthalten und werden von diesem einfachen Programm übergangen. Man kann sie vor dem Auslesen über das Monitorprogramm einstellen: Kanal 1, Blocklänge 100, Intervallzeit 100 ms.

| Adresse | Befehl | Operand | Mnemonic     | Kommentar |
|---------|--------|---------|--------------|-----------|
| 0400    | 34     | 83      | CALL 0183    | RAM-Reset |
| 0402    | BF     | 08      | MOV R7,#08   | 8 mal:    |
| 0404    | 34     | AB      | CALL 01AB    | Lesen     |
| 0406    | EF     | 02      | DJNZ R7,0404 |           |
| 0408    | BF     | 64      | MOV R7,#64   | 100 mal:  |
| 040A    | 23     | 01      | MOV A,#01    | Kanal 1   |
| 040C    | 34     | 51      | CALL 0151    | Messen    |
| 040E    | 34     | 8F      | CALL 018F    | Schreiben |
| 0410    | BE     | 64      | MOV R6,#64   | 100 mal:  |
| 0412    | 34     | C1      | CALL 01C1    | Warten    |
| 0414    | BE     | 12      | DJNZ R6,0412 |           |
| 0416    | BF     | 0A      | DJNZ R7,040A |           |
| 0418    | 84     | 18      | JMP 0418     | Ende      |

## Listing 10.13: Ein Programm zur Meßdatenerfassung

## 10.7 Analyse von Simpel-Programmen

Die Arbeit an einfachen Assemblerprogrammen vermittelt einen Eindruck von den Möglichkeiten und der Mühsal dieser Art der maschinennahen Programmierung. Sie zeigt zugleich auch die Notwendigkeit höherer Programmiersprachen.

Die Schülerinnen und Schüler sollten verstehen, daß Programme in höheren Programmiersprachen verwendet werden, weil sie dem menschlichen Denken entgegenkommen. Der Prozessor kann diese Programme jedoch nicht direkt verarbeiten, sondern sie müssen in Maschinenprogramme übersetzt (compiliert) werden oder über einen Interpreter komplexe Maschinenroutinen aufrufen. Simpel stellt eine einfache Compilersprache dar. Die compilierten Programme lassen sich leicht disassemblieren und analysieren, um die grundlegende Funktion eines Interpreters zu verstehen.

Für die Analyse bieten sich einfache Programme an, die in ihrer Funktion den besprochenen Maschinenprogrammen gleichen. Das folgende Simpelprogramm schaltet die LED der Zelle ein.

```
Anfang
X 0
AusPort
Ende
```

Listing 10.14: Ein Beispielprogramm in Simpel

Nach dem Compilieren muß es in die Zelle geladen werden. Mit Hexedit/Disassem läßt sich der Speicherinhalt der Zelle disassemblieren:

```
0400 84 02 JMP 0402
0402 23 00 MOV A, #00
0404 39 OUTL P1, A
0405 24 F4 JMP 01F4
```

Listing 10.15: Der disassemblierte Programmcode

Eine Analyse des disassemblierten Programms zeigt, daß am Anfang ein Sprungbefehl auf den Beginn des eigentlichen Programms steht. Er dient dazu, eventuelle Unterprogramme zu überspringen. Das Ende des Programms bildet ein Sprung zu der Adresse 01F4 im Betriebssystem. Dort befindet sich eine Schleife, in der das Programm verharrt, bis der Hostrechner erneut die Kontrolle über die Zelle übernimmt.

Es ist leicht zu erkennen, daß die Variable X in Simpel mit dem Akku des Prozessors identisch ist. Der Befehl Ausport wird durch einen einzigen Maschinenbefehl (39, OUTL P1,A) übersetzt.

Ein disassembliertes Programm sollte möglichst ausgedruckt und mit Kommentaren versehen werden, damit so ein vertieftes Verständnis erreicht wird.

```
0400 84 02 JMP 0402 ; zum Anfang
; Anfang
0402 23 00 MOV A, #00 ; X 0
0404 39 OUTL P1, A ; AusPort
0405 24 F4 JMP 01F4 ; Ende
```

Listing 10.16: Das kommentierte Listing

Simpel gestattet eine optionale Parameterübergabe an alle Befehle, die den Inhalt von X verarbeiten. Das obige Programm kann also auch so geschrieben werden:

```
Anfang
```

```
Ausport 0
Ende
```

### Listing 10.17: Die optionale Parameterübergabe in Simpel

Das Disassemblieren zeigt, daß die erzeugten Programmcodes exakt identisch sind. Der zu übergebende Parameter wird also zuerst in den Akku geladen.

Mit ähnlich einfachen Programmen läßt sich die Übersetzung aller Simpel-Befehle analysieren. Einen schnellen Überblick vermittelt auch die Übersetzungstabelle SIMPEL.TAB des Simpel-Compilers. Es zeigt sich, daß zahlreiche Befehle mit nur einem Maschinenbefehl übersetzt werden. Komplexere Befehle rufen Funktionen aus dem Betriebssystem auf:

```
Messen      : CALL 0151
AusRS232    : CALL 0100
EinRS232    : CALL 0127
```

In einigen Fällen werden kleinere Programmsequenzen eingefügt. So wird z.B. die Subtraktion wegen des Fehlens eines Subtraktionsbefehls im 80C39 durch folgende Bytes ersetzt (vgl. Listing 10.7):

```
Minus Zahl : BE Zahl    MOV  R6,Zahl
            37          CPL  A
            6E          ADD  A,R6
            37          CPL  A
```

Ein Merkmal vieler Hochsprachen ist die Verwendung von Prozeduren, die unter ihrem frei definierbaren Namen aufgerufen werden können. Das folgende Simpel-Programm zeigt die Verwendung von zwei Prozeduren, wobei die Prozedur "testen" die Prozedur "rechnen" aufruft. Außerdem wird hier die Immer-Schleife verwendet.

```
Prozedur rechnen
Plus 2
AusPort
ProzedurEnde

Prozedur testen
Immer rechnen
ProzedurEnde

Anfang
testen 3
Ende
```

### Listing 10.18: Verwendung von Prozeduren in Simpel

```
0400  84 0B  JMP 040B    ;zum Anfang
;Prozedur rechnen
0402  03 02  ADD A,#02    ;Plus 2
0404  39     OUTL P1,A  ;AusPort
0405  83     RET       ;ProzedurEnde
;Prozedur testen
0406  94 02  CALL 0402    ;rechnen
0408  84 06  JMP 0406    ;Immer-Schleife
040A  83     RET       ;ProzedurEnde
;Anfang
```



```

040B  23 03  MOV A,#03      ;Parameter 3
040D  94 06  CALL 0406      ;testen
040F  24 F4  JMP 01F4      ;Ende

```

Listing 10.19: Prozeduren werden als Unterprogramme übersetzt.

Die Analyse zeigt, daß die Prozeduren in der Reihenfolge ihres Auftretens als Unterprogramme übersetzt werden. Das Hauptprogramm steht hinter den Unterprogrammen und wird durch einen Sprungbefehl am Anfang des Programms erreicht. Im Gegensatz dazu beginnen Assemblerprogramme meist mit dem Hauptprogramm, das Unterprogramme an höheren Adressen aufruft.

Jeder Prozeduraufruf wird durch einen Unterprogrammaufruf mit der Adresse der Prozedur übersetzt. Am Ende einer Prozedur wird der Rücksprungbefehl eingesetzt. Eine Prozedur kann mit einem Parameter aufgerufen werden. Simpel lädt dann vor dem eigentlichen Prozeduraufruf den Akku mit dem Parameter. Im vorliegenden Beispiel wird die Zahl 3 im Akku an die Prozedur testen übergeben und von dort an die Prozedur rechnen.

Die Immer-Schleife verwendet einen Rückwärtssprung, so daß sich eine Endlosschleife ergibt. Zählschleifen werden in Simpel mit dem Schleifenbefehl DJNZ gebildet. Die Zähler A bis H sind den Registern R0 bis R7 in der zweiten Speicherbank zugeordnet. Das folgende Programm zeigt die Übersetzung der Zählschleife:

```

Prozedur rechnen
Plus 2
AusPort
ProzedurEnde

Anfang
A 8
Amal rechnen
Ende

```

Listing 10.20: Verwendung von Zählschleifen in Simpel

```

0400  84 06  JMP 0406      ;zum Anfang
;Prozedur rechnen
0402  03 02  ADD A,#02      ;Plus 2
0404  39      OUTL P1,A      ;AusPort
0405  83      RET          ;ProzedurEnde
;Anfang
0406  D5      SEL RB1      ;A 8
0407  B8 08  MOV R0,#08
0409  C5      SEL RB0
040A  C5      SEL RB0      ;Amal
040B  94 02  CALL 0402      ;rechnen
040D  D5      SEL RB1
040E  E8 0A  DJNZ R0,040A
0410  C5      SEL RB0
0411  24 F4  JMP 01F4      ;Ende

```

Listing 10.21: Zählschleifen verwenden die Register der zweiten Speicherbank.

Die hier betrachteten Elemente und Strukturen sind relativ einfach. Dagegen ist der kompilierte Programmcode für bedingte Prozeduraufrufe und die Solange-Schleifen in Simpel schwieriger zu analysieren. Erforderlich wäre dazu die Erarbeitung von Prozessor-Flags und ihrer Anwendung auf der

Ebene von Assembler. Für die meisten Kurse wird eher ein Kürzung des hier vorgeschlagenen Umfangs sinnvoll sein. Nähere Informationen zur Programmierung des 8048-Prozessors und zum Aufbau von Simpel finden sich bei Bedarf im Systemhandbuch zur Zelle [1].

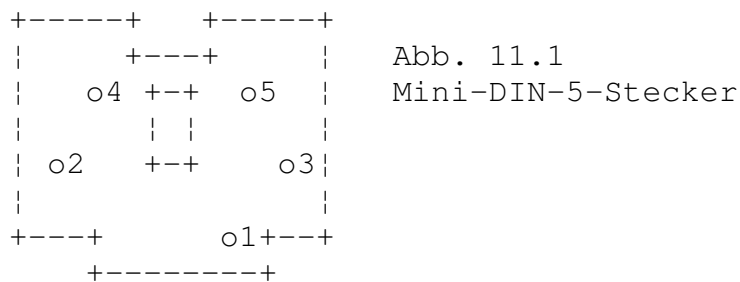
## Anhang A: Anschlüsse der Zelle

Jede der vier Anschlußbuchsen für Meßwandler verfügt über einen Analogeingang (A1...A4), einen Digitaleingang (D1...D4), Masse, Betriebsspannung und einen Ausgang für die Referenzspannung 2,5V.

### Einzelanschlüsse 0 bis 3

Mini-DIN-5 Stecker

Pin 1 Referenzspannung 2,5V  
 Pin 2 Betriebsspannung 4V...6V über EIN-Schalter  
 Pin 3 Masse  
 Pin 4 Analog-Eingang, 0...2,5V (A1...A4)  
 Pin 5 Digital Ein/Ausgang (D1...D4)



Der Sammelanschluß verfügt über alle Anschlüsse der vier Einzelanschlüsse und zusätzlich über den digitalen Steuerport St und Anschlüsse zu den Tasten (")Reset(") und (")Start(").

### Sammelanschluß

Sub-D-15

Pin 1 RESET-Anschluß, aktiv low  
 Pin 2 Start-Tasten-Anschluß, aktiv low  
 Pin 3 Digitaler Steuerport St  
 Pin 4 Masse  
 Pin 5 Analog-Eingang A4  
 Pin 6 Analog-Eingang A3  
 Pin 7 Analog-Eingang A2  
 Pin 8 Analog-Eingang A1  
 Pin 9 Betriebsspannung 4V...6V über EIN-Schalter  
 Pin 10 Referenzspannung 2,5V  
 Pin 11 Masse  
 Pin 12 Digitale Ein/Ausgangsleitung D4  
 Pin 13 Digitale Ein/Ausgangsleitung D3  
 Pin 14 Digitale Ein/Ausgangsleitung D2  
 Pin 15 Digitale Ein/Ausgangsleitung D1

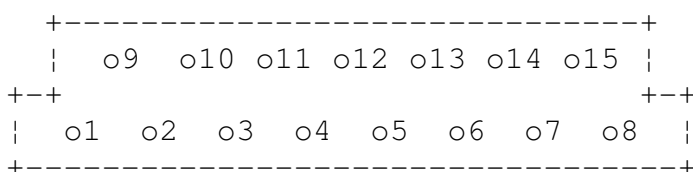


Abb 11.2: Sub-D-15-Stecker

Der Anschluß für die serielle Schnittstelle verfügt über die Sendeleitung TXT, die Empfangsleitung RXD und die Masseleitung:

**Serielle Schnittstelle:**

Mini-DIN-8-Stecker

Pin 3 Sendeleitung TXD (Daten von der Zelle)

Pin 4 Masse

Pin 5 Empfangsleitung RXD (Daten vom PC)

Verbindungskabel sind für PC/XT (25-polig Sub-D), PC/AT (9-polig Sub-D) und für Apple Macintosh (Mini-DIN-8) erhältlich.

## **Anhang B: Funktionsübersichten der Software**

### **B.1 Monitorprogramm**

Das Monitorprogramm ist ein universelles Werkzeug, das alle Grundfunktionen der Zelle unterstützt. Es hat Schnittstellen zur Datenausgabe an andere Programme und zum Einlesen nachladbarer Programme. Das Programm ist menügesteuert und verfügt über ein Hilfesystem.

Das Monitorprogramm dient in erster Linie dazu, einen ersten Überblick über die Funktionen der Zelle zu erhalten und einfache Messungen auszuführen. Eine Auswertung von Meßergebnissen ist anderen Programmen vorbehalten.

Der typische Arbeitsgang für eine Messung mit dem Monitorprogramm umfaßt zunächst eine Überprüfung der Hardwarekonfiguration durch direkte Messung im Menü Ein/Aus. Hier kann die Reaktion der Sensoren und die Batteriespannung kontrolliert werden. Die eigentliche Messung wird mit Meßserie/Parameter eingerichtet. Dann kann die Zelle vom Hostrechner getrennt und zum Meßort gebracht werden, wo die Messung mit der Starttaste ausgelöst wird. Die Meßwerte lassen sich dann später mit Meßserie/Einlesen in den PC übertragen. Einen ersten Überblick verschafft das ausdrückbare Meßdiagramm oder die Meßwertetabelle. Für eine weitergehende Auswertung und Darstellung der Ergebnisse müssen die Daten durch Datei/Konvertieren an Auswerteprogramme exportiert werden.

#### **Übersicht der Funktionen:**

##### **Datei**

Hier werden Meßdaten gespeichert und geladen oder für externe Programme konvertiert.

##### **Datei/Laden**

Früher abgespeicherter Meßdaten können mit ihren Kommentartexten eingeladen werden.

##### **Datei/Speichern**

Die gemessenen Daten werden zusammen mit einem eventuell erstellten Kommentartext gespeichert.

##### **Datei/Konvertieren**

Mit diesem Menüpunkt kann eine Ausgabedatei im ASCII-Format erzeugt werden. Zeiten und die Meßwerte der einzelnen Kanäle werden durch TAB-Zeichen getrennt. Diese Datei kann z.B. von Tabellenkalkulationsprogrammen eingelesen werden.

##### **Datei/Ende**

Rückkehr ins Betriebssystem

**Meßserie**

Hier kann eine autonome Meßserie eingerichtet, gestartet, eingelesen und grafisch dargestellt werden.

**Meßserie/Parameter**

Es werden die gewünschten Einstellungen der Zelle gesetzt. Für jede Messung sind die aktiven Kanäle, die Länge eines Meßblocks und die Intervallzeit festzulegen. Der gesamte Zeitbedarf für einen Meßblock beträgt: (Blocklänge) \* (Intervallzeit).

**Meßserie/Start**

Auslösen des ersten Blocks der Serienmessung, ohne daß die Starttaste der Zelle gedrückt wird.

**Meßserie/Einlesen**

Hier werden die gewonnenen Meßdaten der Zelle an den PC übergeben.

**Meßserie/Grafik**

Ausgabe der Meßwerte in Diagrammform. Es werden immer alle aktiven Kanäle zusammen dargestellt.

**Meßserie/Tabelle**

Ausgabe der gemessenen Werte aller aktiven Kanäle und die zugehörigen Zeiten auf dem Bildschirm.

**Meßserie/Kommentar**

Aufruf eines Texteditors, mit dem die Messung kommentiert werden kann.

**Programm**

Es können Programme in die Zelle nachgeladen und gestartet werden. In den meisten Fällen handelt es sich um compilierte Simpel-Programme, deren Quelltexte angesehen werden können.

**Programm/Laden**

Übertragen eines autonomen Programms in die Zelle.

**Programm/Start**

Start des Programms. Nach dem Ausschalten bleibt das Programm aktiv und kann durch Einschalten oder Reset beliebig oft wieder gestartet werden.

### **Programm/Beenden**

Unterbrechung eines aktiven Programms.

### **Programm/Text**

Zeigt den Quelltext des geladenen Programms.

### **Ein/Aus**

Alle aktuellen Eingangszustände werden laufend angezeigt. So können die Zelle, ihre Batteriespannung und eventuell angeschlossene Meßmodule oder Versuche direkt überprüft werden.

### **Ein/Aus/DezAus**

Ausgabe eines Dezimalwertes an die Digitalleitungen D1 bis D4.

### **Ein/Aus/BinAus**

Ausgabe eines Binärwertes an die Digitalleitungen D1 bis D4.

### **Ein/Aus/StAus**

Veränderung des Zustands des Steuerports.

## **B.2 PRISMA**

Das Programm Prisma erlaubt die einfache Messung und Auswertung von bis zu vier unterschiedlichen Meßgrößen mit der Zelle. Dabei ist immer die korrekte Darstellung der Meßwerte mit ihren physikalischen Einheiten gewährleistet. Messungen können im online-Betrieb erfaßt werden, wobei auch eine gleichzeitige grafische Aufbereitung möglich ist. Die Darstellung mehrerer Diagramme auf dem Bildschirm erleichtert Vergleiche und die Untersuchung von gegenseitigen Abhängigkeiten.

Prisma verwendet Definitionsdateien für jeden Sensor zur Interpretation der Meßwerte. Sie enthalten alle Informationen über die Interpretation und Darstellung der Meßwerte. Über spezielle Kalibrierprogramme lassen sich Sensoren softwaremäßig kalibrieren, indem veränderte Definitionsdateien erzeugt werden.

Vor einer Messung werden im Untermenü "Vorbereitung" die gewünschten Meßkanäle, ihnen zugeordnete Meßmodule und die Meßdauer gewählt. Nach dieser Vorbereitung können alle Parameter in die Zelle übertragen werden, um eine autonome Messung durchzuführen. Die gewonnenen Meßdaten werden dann später im Datei-Menü in den PC übertragen und können ausgewertet werden.

Das Untermenü "Messung" erlaubt alternativ zu den autonomen Messungen Echtzeit-Labormessungen mit gleichzeitiger Darstellung. Die dargestellten Meßwerte werden im PC gespeichert und stehen zur späteren Auswertung zur Verfügung.

### **Übersicht der Funktionen:**

#### **Datei**

Meßdaten werden aus der Zelle gelesen, gespeichert oder geladen. Daten können mit einer Beschreibung versehen oder für externe Programme konvertiert werden.

#### **Datei/Zelle auslesen**

Nach einer offline-Messung werden die Meßdaten von der Zelle in den Speicher des PC geladen.

#### **Datei/Daten laden**

Daten werden mit ihrem Kommentar von der Diskette geladen.

#### **Datei/speichern**

Meßdaten werden mit einem eventuell vorhandenen Kommentar gespeichert. Zusätzlich erhält die Datei Einträge über die eingestellten Meßmodule und die Diagrammüberschriften.

#### **Datei/konvertieren**

Ein Datensatz wird als Tabelle von Rohdaten exportiert.

#### **Datei/Beschreibung**

Texteditor für Kommentare zur Messung



**Datei/Ende**

Beenden des Programms

**Vorbereitung**

Die Vorbereitung einer Messung besteht in der Auswahl der aktiven Kanäle und ihrer Meßmodule und der Einstellung der Meßzeit. Ein Kanal wird inaktiv, wenn man das Menü der Meßmodule mit <Esc> verläßt.

**Vorbereitung/1. Kanal ... 4. Kanal**

Auswahl von .DEF-Dateien zur Definition der angeschlossenen Meßmodule.

**Vorbereitung/Meßzeit**

Die Einstellung der Meßdauer wird aus einem Auswahlmenü gewählt. Unter dem Menüpunkt "frei" kann eine eigene Einstellung definiert werden.

**Vorbereitung/Meßzeit/frei**

Die Anzahl der Meßpunkte pro Messung (=Blocklänge) und die Intervallzeit können frei gewählt werden.

**Vorbereitung/offline**

Einrichtung der Zelle für eine offline-Messung

**Messung**

Es erfolgen online-Messungen bei gleichzeitiger Darstellung. Die Meßwerte stehen nach der Messung unmittelbar zur Auswertung bereit.

**Messung/Diagramm**

Es werden Meßwerte online erfaßt und in Diagrammen dargestellt. Die Diagrammachsen werden mit ihren Einheiten und Achsenteilungen automatisch nach der Wahl des Meßmoduls und der Meßzeit beschriftet.

**Messung/Multimeter**

Die Meßwerte der aktivierten Kanäle werden digital angezeigt. Die Einheit und die Anzahl der ausgegebenen Dezimalstellen sind durch die Modul-Auswahl festgelegt. Bis zu zwei Meßkanäle werden

in Großziffern angezeigt.

### **Messung/Liste**

Die Meßwerte erscheinen während der online-Messung als Liste auf dem Bildschirm.

### **Auswertung**

Ein Meßdatensatz kann mit verschiedenen Methoden ausgewertet werden. Die Auswertung kann auch an eine Datei oder einen Drucker ausgegeben sowie grafisch dargestellt werden.

### **Auswertung/Darstellung**

Hier werden Einstellungen für die grafische Ausgabe in Auswertung/Diagramm festgelegt.

### **Auswertung/Diagramm**

Die Meßwerte werden grafisch dargestellt.

### **Auswertung/Liste**

Alle Meßwerte werden nacheinander als Liste ausgegeben.

### **Auswertung/Extremwerte**

Es werden Minimum, Maximum und Mittelwert der Messungen ausgewertet und für jeden Kanal angezeigt.

### **Auswertung/Filter**

Die Meßwerte werden einer Tiefpaßfilterung durch gleitende Mittelwertbildung unterzogen.

### **Auswertung/Ausgabe**

Alle Listen und Tabellen der Auswertungen können gleichzeitig auf einem Drucker ausgegeben oder in eine Ausgabedatei geschrieben werden.

### **Auswertung/Ausgabe/Drucker**

Die Druckerausgabe wird eingeschaltet.

### **Auswertung/Ausgabe/Datei**

Die Ausgabedatei PRISMA.AUS wird für die Ausgabe geöffnet und kann nacheinander mehrere

Auswertungen aufnehmen.

### **Auswertung/Ausgabe/Ende**

Die Ausgabedatei wird geschlossen, die Druckerausgabe abgeschaltet.

### **Optionen**

Hier sind verschiedene Hilfsfunktionen des Programms zusammengefaßt.

### **Optionen/Batterie**

Die Batteriespannung der Zelle wird angezeigt.

### **Optionen/Daten löschen**

Die Meßdaten im PC werden gelöscht.

### **Optionen/Erster Block**

Es kann angegeben werden, bei welcher Messung die Auswertung beginnen soll.

### **Optionen/Programmstart**

Programme lassen sich in die Zelle laden und starten.

### **Optionen/Quelltext**

Der Quelltext eines geladenen Programms kann angesehen werden.

### **Optionen/Hilfe**

Die allgemeine Hilfsfunktion wird aufgerufen. Die Hilfsfunktion für jedes Untermenü erhält man mit <?>.

Prisma unterstützt alle Sensoren der Zelle durch eigene Definitionsdateien und Kalibrierprogramme. Neue Sensoren werden mit passenden Dateien geliefert, so daß sie sich ohne weiteres mit Prisma verwenden lassen. Die folgenden Messungen werden derzeit unterstützt:

### **Messungen ohne Meßmodule:**

- SPANNUNG.DEF      direkte Spannungsmessung 0...2,55V
- U\_BAT.DEF          Batteriespannung der Zelle 0...7V (nur Kanal 4)

- FREQUENZ.DEF      Frequenzmessung 0...2,5kHz  
(FREQUENZ.BIN muß gestartet werden)

**Messung mit dem Meßmodul Lärm:**

- LAERM.DEF      Lärmmessung 30...100dB(A)
- KAL\_LRM.EXE      Kalibrierprogramm

**Messungen mit dem Meßmodul Temperatur\_1:**

- TEMP1.DEF      Temperaturmessung -20°C...+107°C
- TEMP1\_30.DEF      eingengter Meßbereich bis 30°C
- KAL\_TMP1.EXE      Kalibrierprogramm

**Messungen mit dem Meßmodul Temperatur\_2 (NTC-Sensor):**

- TEMP2.DEF      Eintauchmessung -20°C...+60°C
- KAL\_TMP2.EXE      Kalibrierprogramm

**Messungen mit dem Meßmodul Belichtungsmesser:**

- LICHT\_E.DEF      Lichtmessung in Lux
- LICHT\_I.DEF      Lichtmessung in W/m<sup>2</sup>
- KAL\_LUX.EXE      Kalibrierprogramm (Lux)
- KAL\_WATT.EXE      Kalibrierprogramm (W/m<sup>2</sup>)

**Messungen mit dem Meßmodul Gas:**

- GAS.DEF      Widerstandsmessung des Sensors in k $\Omega$

## **B.3 ZELLÄRM**

Das Programm Zellärm ist ein Meß- und Auswerteprogramm speziell für die Messung von Geräuschpegeln. Es enthält Standard-Auswerteverfahren für die Beurteilung von Lärmbelastungen. Messungen können unter Echtzeitbedingungen am PC aufgenommen oder ortsunabhängig durchgeführt werden.

### **Übersicht der Funktionen:**

#### **Datei**

Meßdaten werden aus der Zelle gelesen, gespeichert, geladen, an externe Programme exportiert oder kommentiert.

#### **Datei/Zelle auslesen**

Nach einer Offline-Messung werden hier die Meßdaten von der Zelle in den Speicher des PC geladen.

#### **Datei/Daten laden**

Daten werden mit ihrem Kommentar von der Diskette geladen.

#### **Datei/speichern**

Meßdaten werden mit einem eventuell vorhandenen Kommentar gespeichert.

#### **Datei/konvertieren**

Ein Datensatz wird als Tabelle mit Zeiten und Pegeln im ASCII-Format gespeichert.

#### **Datei/Beschreibung**

Eine Messung kann kommentiert werden.

#### **Datei/Ende**

Beenden des Programms

### **Vorbereitung**

Die Vorbereitung einer Messung besteht in der Auswahl der Meßzeit und der Anzahl der Messungen. Es stehen neun Standardeinstellungen zur Verfügung. Drei weitere Einstellungen erlauben die automatische Steuerung der Messung nach unterschiedlichen Kriterien, die sich nach eigenen Wünschen gestalten lassen.

### **Messung**

Online-Messung von Schallpegeln mit gleichzeitiger Darstellung.

**Messung/Diagramm**

Die Meßwerte werden während der Messung in einem Diagramm dargestellt.

**Messung/Großanzeige**

Die Messwerte werden in Großziffern und gleichzeitig mit einem Bargraphen angezeigt.

**Messung/Liste**

Die Meßwerte erscheinen während der online-Messung als Liste auf dem Bildschirm.

**Auswertung**

Ein Meßdatensatz kann mit verschiedenen Methoden ausgewertet werden. Die Auswertung kann auch an eine Datei oder einen Drucker ausgegeben sowie grafisch dargestellt werden.

**Auswertung/Diagramm**

Der Verlauf der Pegel wird grafisch dargestellt.

**Auswertung/Verteilung**

Die Pegel einer Messung werden nach Häufigkeiten ausgewertet.

**Auswertung/L01/Leq/L95**

Jede Messung wird je nach ihrer Dauer vom Programm in bis zu 24 Abschnitte zwischen 10s und 1h aufgeteilt. Für jeden Abschnitt werden die charakteristischen Pegel L01, L95 und der energieäquivalente Mittelwert Leq berechnet. Das Ergebnis kann grafisch dargestellt und ausgedruckt werden.

**Auswertung/Liste**

Gemessene Pegel werden nacheinander als Liste ausgegeben.

**Auswertung/Ausgabe**

Alle Listen und Tabellen der Auswertungen können gleichzeitig auf einem Drucker ausgegeben oder in eine Ausgabedatei geschrieben werden.

**Auswertung/Ausgabe/Drucker**

Die Druckerausgabe wird eingeschaltet.

### **Auswertung/Ausgabe/Datei**

Die Ausgabedatei ZELLAERM.AUS wird für die Ausgabe geöffnet.

### **Auswertung/Ausgabe/Ende**

Die Ausgabedatei wird geschlossen, die Druckerausgabe abgeschaltet.

### **Optionen**

Hier sind verschiedene Hilfsfunktionen des Programms zusammengefaßt.

### **Optionen/Batterie**

Die Batteriespannung der Zelle wird angezeigt.

### **Optionen/Daten löschen**

Die Daten in der Zelle werden gelöscht.

### **Optionen/Clippen**

Diese Funktion erlaubt ein Abschneiden extremer Meßwerte oberhalb eines einstellbaren Pegels.

### **Optionen/Hilfe**

Die allgemeine Hilfefunktion wird aufgerufen. Diese Funktion entspricht dem <?> vom Hauptmenü aus.

## **B.4 MEDA**

Bei dem Programm Meda handelt es sich um einen Daten-Plotter mit integriertem Editor. Die Daten können entweder über die Tastatur eingegeben, von einer Datei gelesen, durch eine mathematische Gleichung erzeugt oder mit einem Interface erfaßt werden. Meßwerte und Kurvenverläufe lassen sich gleichzeitig darstellen, wobei sich die Diagrammachsen laufend selbst anpassen.

Neben anderen Interfaces kann die Meßwerterfassung auch über die Zelle im online-Modus erfolgen. Das Programm ist Maus-gesteuert und benutzt eines GEM-ähnliche Bedieneroberfläche.

### **Übersicht über die Funktionen:**

#### **Datei**

Laden, Speichern und Löschen von Dateien

#### **Datei/Laden**

Laden einer ASCII-Datei von Meßwerten

#### **Datei/Sichern**

Speichern der aktuellen Daten in einer ASCII-Datei

#### **Datei/Sichern als**

Speichern unter einem neuen Dateinamen

#### **Datei/Löschen**

Löschen einer Datei

#### **Edit**

Bearbeiten von Daten im Editorfenster. Jede Funktion wirkt auf die gerade angewählte Spalte. Neben einer X-Spalte gibt es sechs Y-Spalten A-F.

#### **Edit/Ausschneiden**

Zwischenspeichern und leeren der aktuellen Spalte

#### **Edit/Kopieren**

Zwischenspeichern der Spalte

#### **Edit/Einkleben**



Einsetzen der zwischengespeicherten Spalte

### **Edit/Erzeugen**

Erzeugen von Datenreihen. Für die X-Achse lassen sich aufsteigende Werte einsetzen. Für jede Y-Spalte können Werte durch eine Gleichung der Form  $y=f(x)$  erzeugt werden.

### **Edit/Export**

Speichern der aktuellen Spalte auf Diskette

### **Edit/Import**

Einlesen einer Datenspalte von der Diskette

### **Graph**

Beeinflussen des Plotterfensters, das die Spalte A gegen die X-Spalte aufträgt.

### **Graph/Diagramm**

Untermenü für folgende Einstellungen:

- Texte für Überschrift, X-Achse und Y-Achse
- Wahl der Darstellung der Meßpunkte
- Wahl einer Verbindungskurve (direkt, Bezier oder Spline)
- Wahl einer miteingezeichneten, eingegebenen Funktion oder vom Programm berechneten Regressionskurve (lin, pot, exp, log)

### **Graph/X-Achse, Y-Achse**

Wahl der Skala und der Zahlenwerte der Achse, Wahl einer logarithmischen Skala, Wahl von Fehlerbalken

### **Graph/Sort X**

Sortieren der Zahlenwerte der X-Achse

### **Rechnen**

Umrechnen und Verknüpfen der X- und Y-Werte des Plotterfensters

### **Rechnen/X/Y-Achse**

Verknüpfung der X- und Y-Daten

### **Rechnen/X-Achse, Y-Achse**

Umrechnen der Achse, z.B. zur Einheitenumrechnung

### **Rechnen/Regressionen**

Erzeugen der linearen, potentiellen, exponentiellen und logarithmischen Regressionsgleichung

### **Rechnen/Funktion**

Eingabe einer Funktion, die mit den Meßwerten geplottet werden soll

### **Rechnen/Mittelwert**

Anzeige der Mittelwerte

### **Rechnen/Rechner**

Taschenrechner für mathematische Gleichungen

### **Ausgabe**

Ausgabe des Diagramms auf dem ganzen Bildschirm, um z.B. das Diagramm auszudrucken

### **Messen**

Meßwerterfassung über ein Interface

### **Messen/Parameter**

Wahl des Interfaces und des Zeitintervalls (Größenordnung ab ca. 1s). Für die Zelle kann Einkanalmessung oder Vierkanalmessung eingestellt werden.

### **Messen/Start**

Starten einer Messung

## **B.5 Das Programmpaket UNIMESS**

Das Programmpaket Unimess des Arbeitskreises Computer im Chemieunterricht unterstützt sehr viele unterschiedliche Interfaces, darunter auch die Zelle in ihrem Direktmodus. Das Paket besteht im wesentlichen aus den Messprogrammen Unimess 1 bis Unimess 4 zur Meßdatenerfassung an einem bis vier Kanälen und aus dem Auswertprogramm UNI-AUS. Zusätzlich sind Simulationsprogramme und zahlreiche Hilfsprogramme enthalten. Hier soll nur ein kurzer Überblick über die Funktionen von UNI-MESS 1 und UNI-AUS gegeben werden.

### **Übersicht der Funktionen von UNIMESS 1**

#### **Messen**

Einrichtung und Ausführung von Messungen

#### **Messen/Meßgröße, Bereich**

Wahl des Interfaces, der Meßgröße und des Meßbereichs

#### **Messen/Voreinstellung für Meßreihe**

Auswahl, ob während der Messung die Großanzeige eingeschaltet oder eine Punkt- oder Linien-Grafik aufgebaut werden soll, Wahl der X-Achse, Einstellung der Meßzeit usw.

#### **Messen/ Meßwert als Großanzeige**

Direkte Messung ohne Meßwertspeicherung

#### **Messung/Meßreihe aufnehmen**

Durchführung einer Meßreihe

#### **Messung/Halbautomatische Eichung**

Kalibrierung von Sensoren durch Messung und Angabe von Sollwerten

#### **Datei**

Menü zum Speichern und Laden von Meßdaten und Grafiken

#### **Datei/Daten laden**

Laden von früher aufgenommenen Meßreihen einer gewählten Meßgröße

#### **Datei/Daten speichern**

Speichern einer Meßreihe unter einem frei wählbaren Dateinamen

**Datei/Daten löschen**

Löschen ausgewählter Dateien

**Datei/Graphik laden / speichern / löschen**

Laden, Speichern und Löschen von Diagrammdateien Das Dateiformat kann mit einem Hilfsprogramm z.B. in das PCX-Format umgewandelt werden.

**Datei/Inhaltsverzeichnis**

Zeigt das aktuelle Unterverzeichnis

**Datei/Laufwerk**

Wahl eines Diskettenlaufwerks zur Datenspeicherung

**Datei/Zugriffspfad**

Auswahl eines vorhandenen Unterverzeichnisses zur Datenspeicherung

**Datei/Meßoptionen laden / speichern**

Speichern und Laden aller einstellbaren Versuchsparameter unter einem frei wählbaren Dateinamen

**Datei/Programm beenden**

Rückkehr zum Inhaltsverzeichnis des Programmpaketes oder ins Betriebssystem

**Tabelle**

Menü zur Bearbeitung von Meßreihen in Tabellenform

**Tabelle/Anzeigen, Korrigieren**

Editieren der gemessenen Werte

**Tabelle/Tabelle drucken**

Ausdrucken einer Meßreihe

### **Tabelle/Werte in Großanzeige**

Ausgabe der Tabellenwerte in Großziffern

### **Tabelle/Bezeichnungen**

Ändern von Texten und Beschriftungen

### **Graphik**

Menü zur grafischen Ausgabe und Bearbeitung von Meßdaten

### **Graphik/Anzeigen**

Ansicht eines bereits erstellten Diagramms

### **Graphik/Wechseln der Seite**

Wechsel zwischen zwei Bildschirmseiten für zwei Diagramme

### **Graphik/Hin und Her**

Schnelles Umschalten zwischen beiden Diagrammen

### **Graphik/Nur Koordinatensystem**

Zeichnen eines leeren Koordinatensystems, das als Grundlage für eine manuelle Verarbeitung von Meßwerten dienen kann

### **Graphik/Koordinatensystem und Punkte**

Zeichnen eines Koordinatensystems mit eingetragenen Meßpunkten

### **Graphik/ Punkte zeichnen**

Zeichen von Meßpunkten in ein Diagramm, wobei mehrere Messungen übereinander dargestellt werden können

### **Graphik/Punkte verbinden**

Verbindung der Meßpunkte durch gerade Linien

### **Graphik/Differenzen**

Berechnung und Darstellung von Differenzen zwischen zwei y-Werten

### **Graphik/Spezialkoordinatensystem**

Freie Wahl der Achsen und der Beschriftungen, so daß eine Spreizung von Achsen möglich ist

### **Graphik/Spezialkoordinatensystem und Punkte**

Zeichnen des frei definierten Koordinatensystems und der Meßpunkte

### **Graphik/Einblenden, Korrigieren**

Korrektur von Meßwerten im Diagramm unter Verwendung der Pfeiltasten oder der Maus

### **Graphik/Beschriften**

Eintrag von Beschriftungen an beliebiger Stelle des Diagramms

### **Graphik/PrintScreen**

Ausdrucken eines Diagramms über einen bei der Installation gewählten Drucker (9, 24 Nadeln oder Laser/Tintenstrahl)

### **Graphik/Graphikoptionen**

Wahl des Graphiksymbols für Meßpunkte und der Farben

### **Hilfe**

Ein interaktives Hilfesystem

## B.6 Übersicht der Funktionen von UNIAUS

### Meßgröße

Angabe der Meßgröße, zu der eine Meßdatei geladen werden soll

### Meßgröße/Per Hand eingeben

Eintrag von Meßreihen, die ausgewertet werden sollen

### Datei

Dateimenü wie beim Programm UNI-MESS 1

### Graphik

Darstellung und Auswertung von Meßreihen wie in UNI-MESS 1

Zusätzlich gibt es den folgenden Menüpunkt:

#### **Graphik/Balken zeichnen**

Darstellung der Meßwerte als senkrechte Balken

### Auswertung

Menü für unterschiedliche Auswerteverfahren:

#### **Auswertung/Ausgleichsgerade**

Berechnung einer Ausgleichsgeraden mit Anzeige von Steigung, y-Abschnitt, x-Abschnitt und Korrelation

#### **Auswertung/Eingeraden-Methode**

Ausgleichsgerade durch einen Teilbereich der Meßwerte

#### **Auswertung/Zweigeraden-Methode**

Zwei Ausgleichsgeraden durch gewählte Abschnitte zur Bestimmung des Schnittpunkts (z.B. Schnittpunkte zweier Geraden unterschiedlicher Leitfähigkeit)

#### **Auswertung/Dreigeraden-Methode**

Drei Ausgleichsgeraden (z.B. für die Ermittlung der Wendepunkte bei Säure-Basen-Titrationen)

#### **Auswertung/Spline-Funktionen**

Zeichnen einer Ausgleichskurve der Meßwerte oder ihrer ersten oder zweiten Ableitung

#### **Auswertung/Wendepunkte**

Ermittlung der Wendepunkte durch Berechnung der Differenzenquotienten

#### **Auswertung/Halbäquivalenzpunkt**

Ermittlung der pK-Werte für Säuren durch den Halbäquivalenzpunkt einer Titration

#### **Auswertung/Integration**

Berechnung der unter einem markierten Kurvenbereich liegenden Fläche durch Integration

### **Auswertung/Automatik für Kinetik**

Auswertung von Experimenten zur Ermittlung der Reaktionsordnung durch Berechnung einer Regressionskurve und der Halbwertszeit

## **Rechnen**

Menü für Rechenoperationen, die sich jeweils auf den gesamten Datensatz beziehen

### **Rechnen/Umrechnen der y-Werte, der x-Werte**

- Addition eines konstanten Wertes
- Multiplikation mit einem konstanten Wert
- Logarithmieren (natürlicher Logarithmus)
- Bildung der Kehrwerte
- Bildung der Kehrwerte der Quadrate
- Umrechnung mit selbst einzugebender Funktion
- Uniformieren der Werte  $y=a+b*i$   $x=a+b*i$

### **Rechnen/Spezielle Umrechnungen**

- Vertauschen von x- und y-Werten
- Basislinienkorrektur
- Umrechnen mit Referenzspektrum
- Steigung als y-Werte / y-Werte als x-Werte
- Steigung als y-Werte / y-Quadrat als x-Werte
- Datensatz vor der letzten Umrechnung wiederherstellen
- An Daten vor der letzten Umrechnung anhängen

### **Rechnen/Gehaltsberechnung**

Berechnung der gesuchten Konzentration aus den Meßwerten einer Titration

### **Rechnen/Wissenschaftlicher Taschenrechner**

Alle üblichen Funktionen eines einfachen wissenschaftlichen Taschenrechners

### **Rechnen/Meßwerte ordnen**

Ordnung nach steigenden x-Werten

### **Rechnen/Meßwerte glätten**

Glätten von schwankenden Meßwerten



## **B.7 Der SIMPEL-Compiler**

Das Programm SIMPEL.EXE stellt eine integrierte Entwicklungsumgebung mit Editor, Compiler, Übertragungsprogramm und Terminalprogramm für die Zelle und ihre Programmiersprache Simpel dar. Zusätzlich wurde mit der Version 2.0 ein RAM-Editor zur direkten Bearbeitung von Speicherinhalten der Zelle angefügt. Programme können in Maschinencode eingegeben, gestartet, gespeichert und disassembliert werden.

### **Übersicht der Funktionen**

#### **Datei**

Laden und Speichern von Simpel-Quelltexten

#### **Datei/Laden**

Ein Dateimenü erlaubt die Auswahl eines Quelltextes \*.SIM.

#### **Datei/Neu**

Vorbereitung eines neuen Programms

#### **Datei/Speichern**

Speichern eines Simpel-Quelltextes

#### **Datei/Hilfe**

Es werden Hilfetexte zum Sprachumfang von Simpel, zu den Fehlermeldungen des Compilers und zur Bedienung des Editors gezeigt.

#### **Datei/Ende**

Rückkehr ins Betriebssystem

#### **Editor**

Der Editor dient zum Eingeben oder Verändern von Quelltexten.

#### **Compiler**

Übersetzung des Quelltextes in lauffähigen Programmcode

#### **Compiler/Compiler**

Der Compiler übersetzt den Quelltext unter Verwendung der Übersetzungstabelle SIMPEL.TAB. Der

Übersetzungsvorgang kann am Bildschirm verfolgt werden.

### **Compiler/Speichern**

Das compilierte Programm wird mit der Kennung .BIN abgespeichert.

### **Run**

Compilierte Programme werden in die Zelle übertragen und gestartet.

### **Terminal**

Direkter Datenaustausch mit der Zelle. Einzelne Bytes können gesendet und empfangen werden. Programme, die die RS232 der Zelle verwenden, um Daten zu empfangen und zu senden, können direkt getestet werden.

### **Hexedit**

Direkte Bearbeitung von Speicherinhalten der Zelle

### **Hexedit/Programm**

Hexdump des Speicherinhalts der Zelle. Der Speicherinhalt kann editiert werden, um Maschinenprogramme einzugeben oder zu verändern.

### **Hexedit/Register**

Hexdump der Registerinhalte der Zelle. Alle Register können editiert werden.

### **Hexedit/Starten**

Ein Programm im Speicher der Zelle wird gestartet.

### **Hexedit/Laden**

Ein Programm \*.BIN wird in die Zelle geladen.

### **Hexedit/Speichern**

Der Programmspeicherinhalt der Zelle wird mit der Kennung .BIN abgespeichert.

### **Hexedit/Disassem**

Ein Disassembler für compilierte Simpel-Programme und eingegebene Maschinenprogramme. Es wird ein 8048-Assemblerlisting des Programms erzeugt.

## Anhang C: 8048-Maschinenbefehle

### Register-Operationen

| Register    | R0 | R1 | R2 | R3 | R4 | R5 | R6 | R7 |
|-------------|----|----|----|----|----|----|----|----|
| ADD A,R     | 68 | 69 | 6A | 6B | 6C | 6D | 6E | 6F |
| ADDC A,R    | 78 | 79 | 7A | 7B | 7C | 7D | 7E | 7F |
| ANL A,R     | 58 | 59 | 5A | 5B | 5C | 5D | 5E | 5F |
| ORL A,R     | 48 | 49 | 4A | 4B | 4C | 4D | 4E | 4F |
| XRL A,R     | D8 | D9 | DA | DB | DC | DD | DE | DF |
| INC R       | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
| DEC R       | C8 | C9 | CA | CB | CC | CD | CE | CF |
| DJNZ R,adr  | E8 | E9 | EA | EB | EC | ED | EE | EF |
| MOV A,R     | F8 | F9 | FA | FB | FC | FD | FE | FF |
| MOV R,A     | A8 | A9 | AA | AB | AC | AD | AE | AF |
| MOV R,#data | B8 | B9 | BA | BB | BC | BD | BE | BF |
| XCH A,R     | 28 | 29 | 2A | 2B | 2C | 2D | 2E | 2F |

### Register relativ

| Register     | R0 | R1 |
|--------------|----|----|
| ADD A, @R    | 60 | 61 |
| ADDC A, @R   | 70 | 71 |
| ANL A, @R    | 50 | 51 |
| ORL A, @R    | 40 | 41 |
| XRL A, @R    | D0 | D1 |
| INC @R       | 10 | 11 |
| MOV A, @R    | F0 | F1 |
| MOV @R,A     | A0 | A1 |
| MOV @R,#data | B0 | B1 |
| XCH A, @R,#  | 20 | 21 |
| XCHD A, @R   | 30 | 31 |
| MOVX A, @R   | 80 | 81 |
| MOVX @R,A    | 90 | 91 |

### Sprungbefehle

| Bit 5-7 | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  |
|---------|----|----|----|----|----|----|----|----|
| JMP     | 04 | 24 | 44 | 64 | 84 | A4 | C4 | E4 |
| JBb     | 12 | 32 | 52 | 72 | 92 | B2 | D2 | F2 |
| CALL    | 14 | 34 | 54 | 74 | 94 | B4 | D4 | F4 |

### Ein/Ausgabe

| Port          | 1      | 2  |
|---------------|--------|----|
| IN A,P        | 09     | 0A |
| OUTL p,A      | 39     | 3A |
| ANL p,#data   | 99     | 9A |
| ORL P,#data   | 89     | 8A |
| MOVD A,p      | 0C..0F |    |
| MOVD p,A      | 3C..3F |    |
| ANLD p,A      | 9C..9F |    |
| ORLD p,A      | 8C..8F |    |
| INS a,Bus     |        | 08 |
| OUTL Bus,A    |        | 02 |
| ANL Bus,#data |        | 98 |
| ORL Bus,#data |        | 88 |

### Bedingte Sprungbefehle

|      |    |          |    |      |    |
|------|----|----------|----|------|----|
| JC   | F6 | JMPP a A | B3 | JT0  | 36 |
| JNC  | E6 | JZ       | C6 | JNT0 | 26 |
| JT1  | 56 | JNZ      | 96 | JTF  | 16 |
| JNT1 | 46 | JF0 B6   |    | JNI  | 86 |
| JF1  | 76 | RET      | 83 | RETR | 93 |

### Arithmetische Befehle

|              |    |             |    |       |    |       |    |
|--------------|----|-------------|----|-------|----|-------|----|
| ADD A,#data  | 03 | ANL A,#data | 53 | INC A | 17 | RLA   | E7 |
| ADDC A,#data | 13 | ORL A,#data | 43 | DEC A | 07 | RLC A | F7 |
| DA A         | 57 | XRL A,#data | D3 | CLR A | 27 | RRA   | 77 |
| SWAP A       | 47 |             |    | CPL A | 37 | RRC A | 67 |

### Timer-Befehle

|           |    |
|-----------|----|
| MOV A,T   | 42 |
| MOV T,A   | 62 |
| STRT T    | 55 |
| STRT CNT  | 45 |
| STOP TCNT | 65 |
| EN TCNTI  | 25 |
| DIS TCNTI | 35 |

### Flags

|        |    |
|--------|----|
| CLR C  | 97 |
| CPL C  | A7 |
| CLR F0 | 85 |
| CPL F0 | 95 |
| CLR F1 | A5 |
| CPL F1 | B5 |

### Interne Steuerung

|          |    |
|----------|----|
| ENI      | 05 |
| DISI     | 15 |
| SEL RB0  | C5 |
| SEL RB1  | D5 |
| SEL MB0  | E5 |
| SEL MB1  | F5 |
| ENTO CLK | 75 |

### Sonstige

|             |    |
|-------------|----|
| MOV A,#data | 23 |
| MOV A,PSW   | C7 |
| MOV PSW,A   | D7 |
| MOVP A,@A   | A3 |
| MOVP3 A,@A  | E3 |
| NOP         | 00 |



## Literaturverzeichnis

- 1] LSW, Mobile Meßwerterfassung mit der Zelle - Systemhandbuch - Soest 1993
- 2] LSW, Themenhaft Lärm, Soester Verlagskontor, Soest 1989
- 3] LSW, Interfacesysteme, Teil 1, 2. Auflage, Soest 1992
- 4] H.Weidner, Mikrocontroller angewandt - MCS-48-Familie - Elektor-Verlag 1992
- 5] B.Kainka, Messen, Steuern und Regeln über die RS232-Schnittstelle, Franzis' 5. Auflage 1993
- 6] Die Zelle, Simpel-Programmsammlung, Modul Bus GmbH 1992
- 7] B.Kainka, PC-Schnittstellen angewandt, Elektor-Verlag 1993