

U. Peters

Technikerschule für Elektronik Meldorf

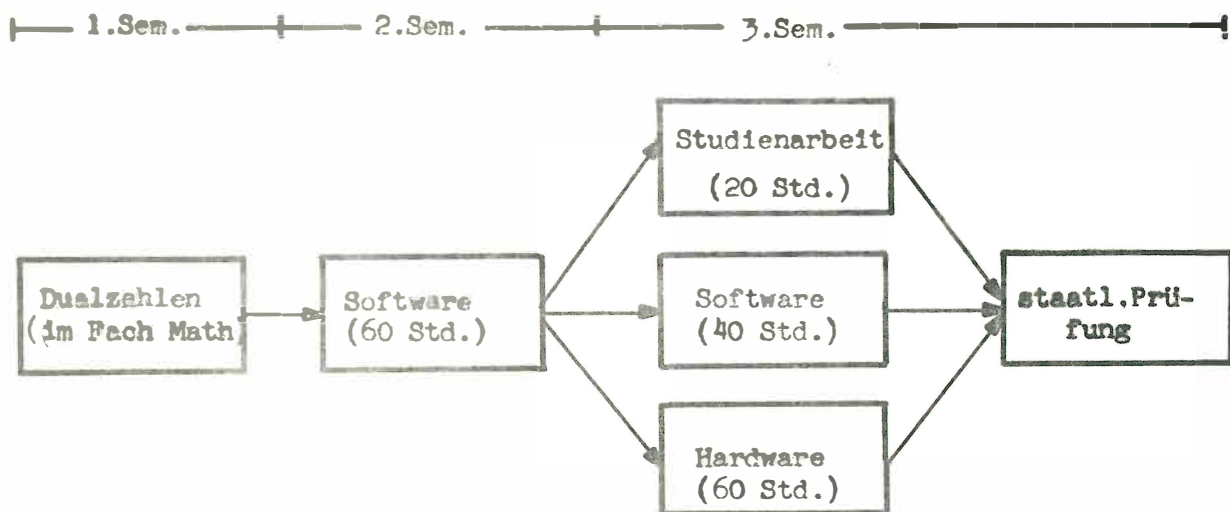
Prozeßrechnerpraktikum an einer Fachschule für Elektronik

Seit knapp zwei Jahren ist in die Ausbildung an der Technikerschule für Elektronik in Meldorf ein Prozeß-rechnerpraktikum integriert. Die Bedingungen für Prozeß-rechnertechnik im Unterricht sind an einer Fachschule in vieler Hinsicht verschieden von denen an einer Fachhochschule. Die Studierenden sind überwiegend Hauptschulabsolventen mit Lehre und mehrjähriger Praxis in einem Elektroberuf. Die Ausbildung dauert nur drei Semester mit insgesamt knapp 2000 Unterrichtsstunden, von denen 180 Stunden auf das Fach Elektronische Datenverarbeitung entfallen.

Unter Beachtung des Rahmenlehrplans für das Fach EDV wird an der Technikerschule Meldorf besonderes Gewicht auf die Prozeßrechneranwendungen gelegt. Dem liegt die Überlegung zugrunde, daß dem zukünftigen Elektronik-Techniker die Datenverarbeitung in erster Linie in Gestalt der Prozeß-automatisierung entgegentritt, während technisch-wissenschaftliche Berechnungen und erst recht die kommerzielle Datenverarbeitung für den Elektroniker in der Industrie im allgemeinen von untergeordneter Bedeutung sind.

Die Technikerschule besitzt für den Unterricht eine Prozeß-rechneranlage vom Typ SIEMENS 300. Die langsam gewachsene Maschinenkonfiguration besteht heute aus einer Zentraleinheit 303 mit 16 KFW Arbeitsspeicher, einem Trommel-speicher, Lochkarten- und Lochstreifengeräten und einer gut ausgebauten Prozeßperipherie.

Bild 1: Ablaufplan für das Fach EDV



Die zeitliche Gliederung des Unterrichtsfachs EDV ist in Bild 1 dargestellt. Es beginnt im 2. Semester mit einem Softwarekurs, der die Grundlagen der Datentechnik sowie eine Einführung in die Maschinenprogrammierung mit Übungen am Bedienungsfeld und eine Einführung in die Assembler-Programmierung enthält. Ferner erwirbt der Studierende in diesem Kurs die für den open shop-Betrieb notwendigen Operator-Kenntnisse. Das Rechnen im dualen und oktalen Zahlensystem ist bereits aus dem Mathematikunterricht des ersten Semesters bekannt. Im 3. Semester wird der Softwarekurs fortgesetzt mit praktischen Übungen zur Programmierung von Prozeßsteuerversuchen in der Assemblersprache. Den Abschluß bildet ein Ausblick auf Compilersprachen und Betriebssysteme.

Die Hardware-Aspekte der Rechnertechnik werden im 3. Semester in einem eigenen Kurs behandelt. Darüber hinaus wird im 3. Semester von jeweils drei Studierenden gemeinsam eine sogenannte Studienarbeit im Fach EDV angefertigt, die dann in die Programmbibliothek der Schule eingereicht wird.

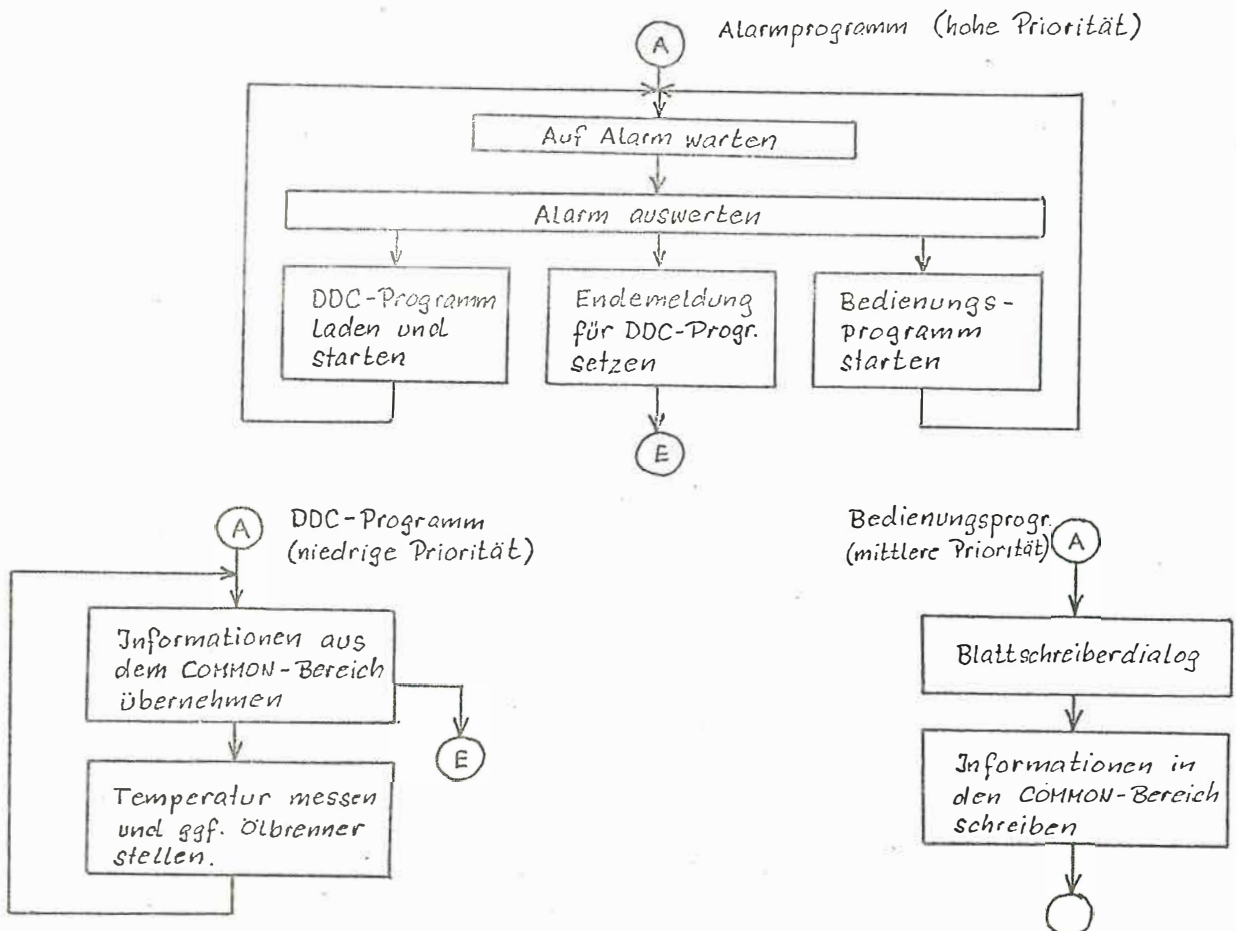
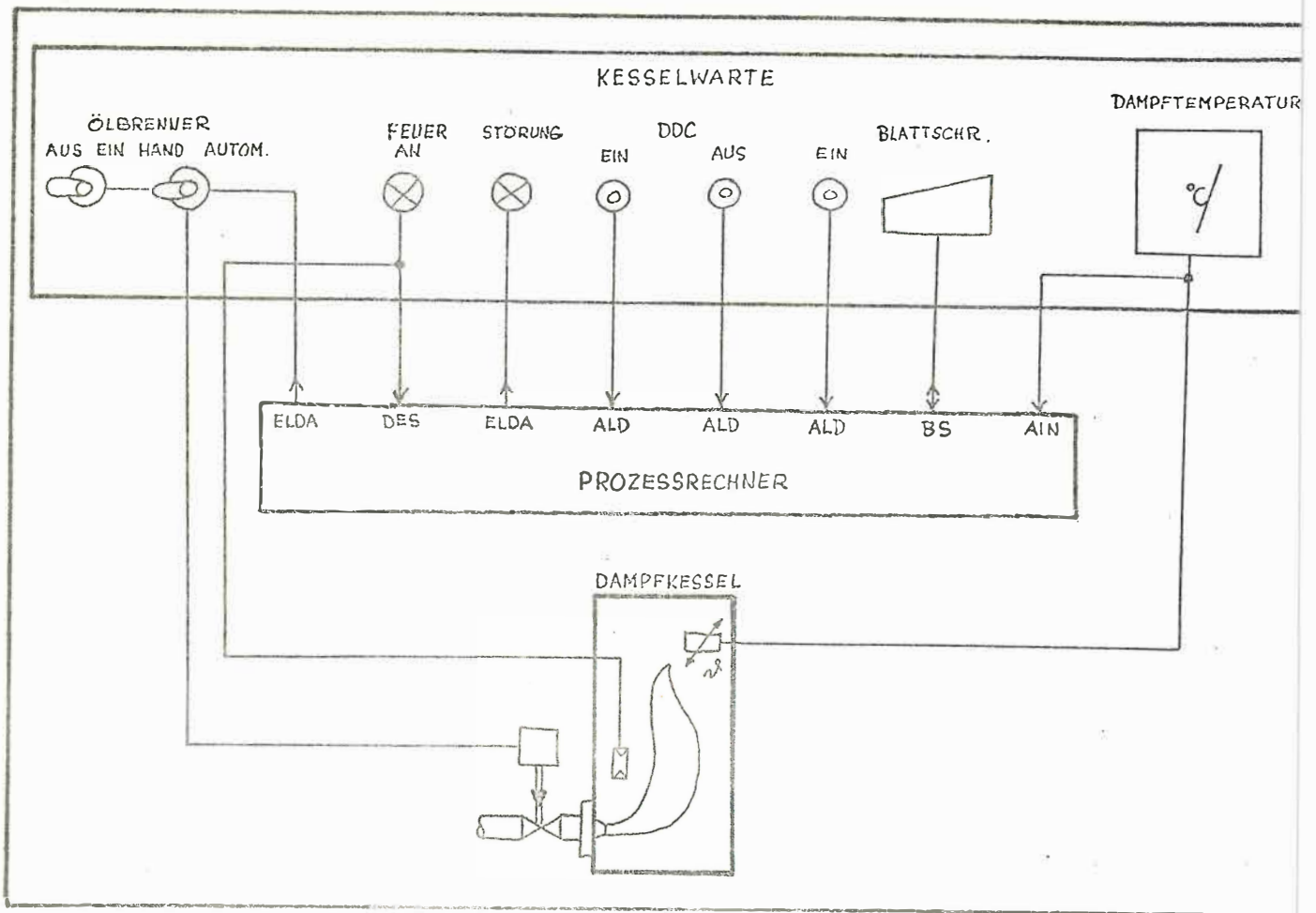
In jedem Halbjahr entstehen auf diese Weise etwa 15 Studienarbeiten. Allerdings wird nicht in jeder Studienarbeit ein neues Assemblerprogramm entwickelt. Ein Teil der Arbeiten befaßt sich mit Hardware-Entwicklungen (Prozeßmodelle) und der Anwendung vorhandener Programme. Beispiele für Studienarbeiten aus den letzten beiden Semestern sind in Tabelle 1 zusammengestellt.

Tabelle 1: Einige abgeschlossene Studienarbeiten

PENDEL	Anpassung der Meßwertgeber für Drehzahl und Drehmoment an der Pendelmaschine an das P1K und Erstellen eines einfachen Protokollierprogramms.
DREHSTROM	Erstellen eines Programmes zur Ausgabe von drei um 120° gegeneinander versetzten Sinusfunktionen über drei Analogausgänge. Wahl der Periodendauer zwischen 4 und 10 Sekunden. Bau von zugehörigen Drehstrommotor-Modellen.
LOGIKTEST	Erstellen eines Programms zum Testen von Verknüpfungsschaltungen. (SIMATIC-N-Lehrbaukasten)
WAERME	Ausführen einer Wärmebedarfsberechnung für ein Wohnhaus mit Hilfe des Formelübersetzers COMP (Programm von der Fachhochschule Paderborn).
MA DEZPUNKT	Erstellen eines Makrobefehls zum Einfügen eines Dezimalpunktes in eine alphanumerische Zeichenfolge.
MA PLOTTEN	Hardwareanschluß eines XY-Plotters und Erstellen eines Makrobefehls zum Ausgeben einzelner Punkte mit absoluten Koordinaten.

Bild 2: Modellversuch "DAMPFKESSELSTEUERUNG"

mit REAL TIME-Programmsystem



Die begrenzte Stundenzahl für den EDV-Unterricht zwingt dazu, den Stoff auf ein überschaubares und eindeutig abgegrenztes Basiswissen zu beschränken. Mit diesem Basiswissen soll es aber möglich sein, ablauffähige Prozeßsteuerprogramme einschließlich kleiner REAL TIME-Programmsysteme zu erstellen und praktisch zu erproben.

Bild 2 zeigt als typisches Beispiel für einen Prozeßsteuerversuch ein Dampfkesselmodell.

Die Abgrenzung des Basiswissens wird durch folgende Beschränkungen im Sprachumfang erreicht:

- Abgekürzte Maschinenbefehlsliste
- Einfache Organisationsaufrufe
- Einfache Unterprogrammtechnik
- Absolut adressierter COMMON-Bereich.

Abgekürzte Maschinenbefehlsliste

Die Liste der im Unterricht benutzten Maschinenbefehle enthält kein Gleitpunktbefehle. Alle arithmetischen Operationen, wie z.B. die Verarbeitung analoger Meßwerte, werden ausschließlich im Festpunktformat programmiert. Die Beherrschung der dabei auftretenden Stellenwert- und Überlaufprobleme stellt einen Teil der Lernziele dar. Ungeklärt ist noch die Rolle der adreßarithmetischen Befehle. Gegenwärtig gehören sie mit zur Maschinenbefehlsliste. Es ist bisher nicht untersucht worden, wie der Unterrichtserfolg sich ändert, wenn die adreßarithmetischen Befehls ebenfalls "unterschlagen" werden und dafür die dann wesentlich unübersichtlichere Programmierung von Adreßrechnungen in Kauf genommen wird.

Tabelle 2: Beispiele aus der Datei MABI

MA DRUCKEN = N, ADR	Text auf Blattschreiber N drucken ADR: Anfangsadresse des Textes
MA BEDIENEN = N	Auf Bedienung am Blattschreiber N warten (max. 8 Zeichen) 1. bis 4. Zeichen im LA 5. bis 8. Zeichen im RA
MA WECKEN = I, N	Programm I regelmäßig alle N Sekunden wecken
MA SCHLAFEN = I	Programm I nicht mehr wecken
MA STELLEN = N	Stellwert von RA nach Ausgang N übertragen (Digitalausgänge und Analogausgänge)
MA ABFRAGEN = N	Digitalwert am Eingang N abfragen und nach RA übertragen
MA MESSEN = N	Analogwert am Eingang N messen und als Dualzahl nach RA übertragen
MA ALARM	Auf Signal am Alarmeingang warten. Alarmbit im RA setzen.
MA SUMMEN	Summer am Bedienungsfeld einschalten

Einfache Organisationsaufrufe

Die häufig benutzten Organisationsaufrufe stehen in Form von speziellen Makrobefehlen zur Verfügung, die besonders einfach anzuwenden sind. Gegenüber der üblichen Makroschreibweise für Aufrufe an das ORG haben diese Makrobefehle folgende Besonderheiten:

- Wenn ein Warteaufruf erforderlich ist, wird dieser mit dem Externaufruf in einem Makrobefehl vereinigt.
Der MA EXWA wird im Unterricht für sich zunächst gar nicht eingeführt.
- Bei der wortweisen Ein- und Ausgabe wird einheitlich der rechte Akkumulator als Nahtstelle zwischen Programm und Außenwelt angesehen. Dadurch vereinfacht sich die Parameterversorgung, wenn der Makrobefehl als Unterprogramm aufgerufen wird.
- Die Makronamen sind keine mnemotechnischen Abkürzungen (wie z.B. ABEI oder BSAU oder P1WA), sondern leicht verständliche Schlagworte wie MESSEN oder DRUCKEN oder ALARM.
- Der Bedienungsaufruf BEWA ist durch einen Eingabeaufruf BEDIENEN ersetzt, der eine Bedienung mit maximal 8 Zeichen erlaubt. Bedienungsprogramme mit eigener Programmnummer werden beispielsweise von einem Alarmprogramm gestartet (siehe Bild 2).

Die so vereinfachten Organisationsaufrufe (Tabelle 2) stehen zusammen mit einer Reihe anderer nützlicher Makrobefehle in der Datei MABI auf dem Trommelspeicher.

Einfache Unterprogramm-Technik

Für die Gliederung eines Quellprogramms bietet PROSA 300 eine für den Anfänger verwirrende Fülle sprachlicher Möglichkeiten. Wiederum muß diese Vielfalt für Unter-richtszwecke stark eingeschränkt werden. Man muß stets im Auge behalten, daß die Fertigkeit im Lösen von tech-nischen Problemen das eigentliche Lernziel ist, nicht aber die perfekte Beherrschung einer formenreichen Programmier-sprache. Nach einigen Experimenten haben wir uns zu folgenden Einschränkungen von PROSA 300 entschlossen:

- keine MC-Anweisung
- keine UB-Anweisung
- keine Dollar-Adressen
- keine Makroerklärungen.

Daraus folgt, daß alle unselbständigen Bibliotheksprogramme die Form einer Makroerklärung haben müssen und in der Datei MABI stehen. Eine Datei UPBI gibt es nicht. Der Anwender braucht also nur die Technik der Makroaufrufe zu kennen, was für den Gebrauch der Organisationsaufrufe ohnehin er-forderlich ist. Wenn ein langer Makrobefehl in einem An-wenderprogramm mehrfach gebraucht wird, kann er zur Speicher-platzersparnis natürlich vom Anwender in die Form eines Unterprogramms gekleidet werden, z.B.

```
DEZDUAL    NOP 0
           MA  DEZDUAL
           SPR (DEZDUAL)
```

Absolut adressierter COMMON-Bereich

Für den Aufbau von Programmsystemen aus getrennt übersetzten Einzelprogrammen wird ebenfalls nicht das eleganteste, sondern vielmehr das am leichtesten durchschaubare V er-fahren benutzt.

Es besteht darin, daß für gemeinsam benutzte Daten ein fester Arbeitsspeicherbereich von 400 Zeilen reserviert ist, der sogenannte COMMON-Bereich.

Er liegt im Arbeitsspeicher im Anschluß an das ORG zwischen den Adressen 2600 und 2999. Am Anfang jedes Anwenderprogramms, das den COMMON-Bereich benutzt, steht die PROSA-Anweisung

COMMON GL 2600

Die Zellen im COMMON-Bereich dürfen nicht explizit, sondern nur relativ zur symbolischen Adresse COMMON adressiert werden, damit die absolute Lage des COMMON-Bereichs im Arbeitsspeicher bei Bedarf durch Neuübersetzung leicht geändert werden kann. Prinzipiell kann der COMMON-Bereich nicht nur gemeinsam benutzte Daten, sondern auch gemeinsam benutzte Unterprogramme aufnehmen. Allerdings wird von dieser Möglichkeit im Anfängerunterricht kein Gebrauch gemacht.

Im Streit um die beste Programmiersprache für Unterrichtszwecke glauben wir damit eine günstige Lösungen gefunden zu haben:

Gegenüber einer Compilersprache wird der Vorteil der Flexibilität und der relativen Nähe zur Hardware beibehalten. Andererseits wird PROSA 300 durch die drastische Beschränkung des Sprachumfangs und durch zahlreiche speziell zugeschnittene Makrobefehle zu einer echt problemorientierten Programmiersprache. Auch der Einwand der Spezialisierung auf einen Hersteller verfängt nicht.

Zwar ist ein Assemblerprogramm nicht kompatibel zu Rechneranlagen anderen Fabrikats. Wohl aber ist ein Assemblerprogrammierer kompatibel in dem Sinne, daß er sich in kürzester Zeit in jede andere Assemblersprache einarbeiten kann.

Abschließend möchten wir uns noch einmal bei den Herren des Rechenzentrums an der Fachhochschule Paderborn, besonders bei Herrn Diplommathematiker Becker bedanken, ohne deren außerordentlich tatkräftige Unterstützung der schnelle Aufbau des Prozeßrechnerbetriebs in Meldorf nicht möglich gewesen wäre.