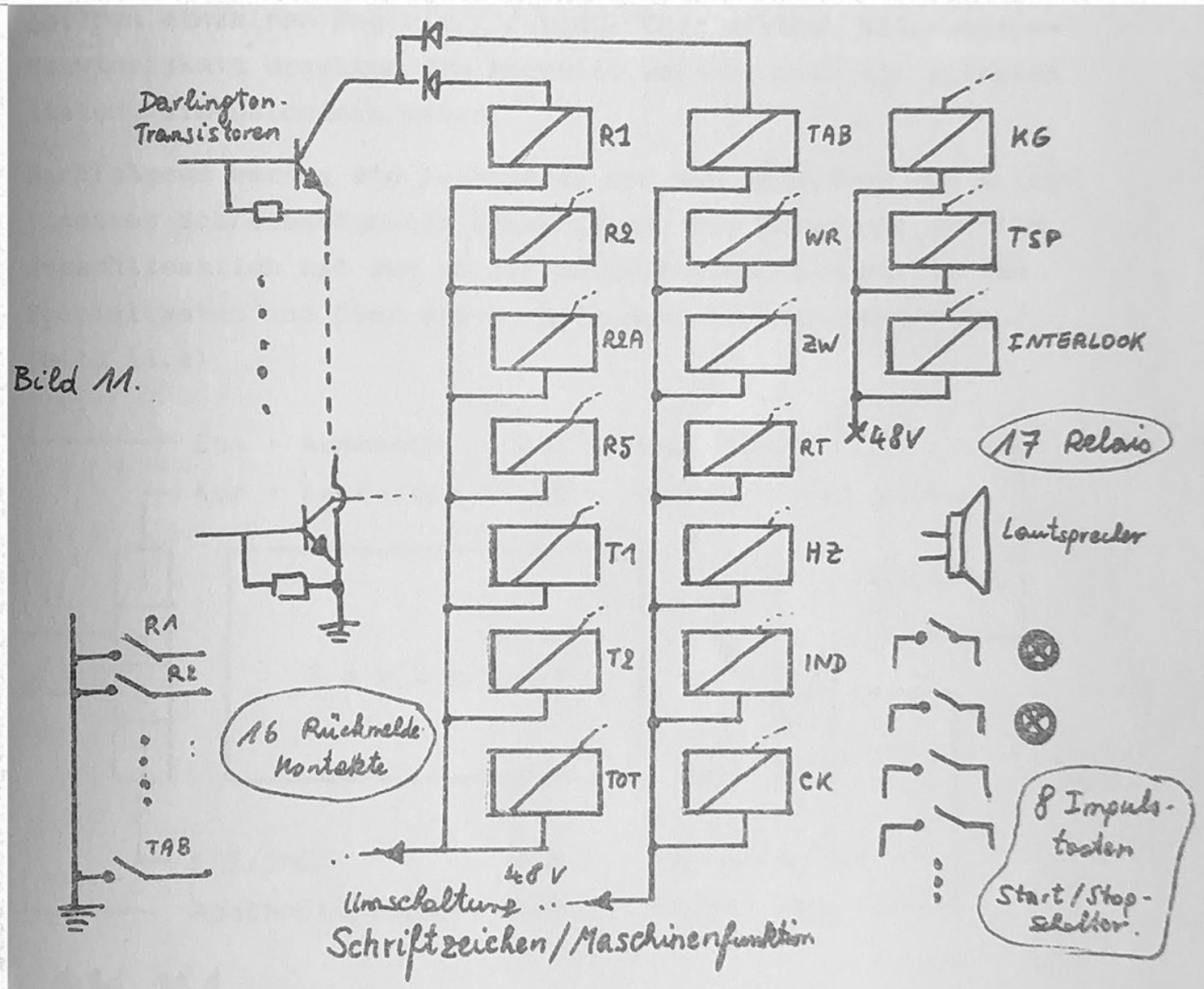


13.1. Textautomat

Zur Realisierung eines leistungsfähigen, preiswerten Textautomaten wurde als Ein / Ausgabeeinheit eine herkömmliche, elektrische Kugelkopfschreibmaschine zum Anschluss an ein Mikroprozessorsystem umgerüstet. Nachstehend wird das Schaltschema angegeben:



Die Auswahl eines Schriftzeichens erfolgt durch 6 Hebel (R1, R2, R2A, R5, T1, T2), welche eine kombinierte Dreh- und Kippbewegung des Kugelkopfes durchführen. Der Anschlag erfolgt mit der Kopplung CK, die Umschaltung von Gross auf Kleinbuchstaben erfolgt mit dem Hebel KG. Die einzelnen Maschinenfunktionen wie Tabulator, Rückschritt, Kopfrücklauf usw. erfolgen über individuelle Hebel und Gestänge, welche alle von dem elektrischen Motor angetrieben werden. Zum Anschluss an einen Rechner

wurde jeder Hebel mit einem Rückmeldekontakt ausgerüstet, zur externen Ansteuerung wurden die entsprechenden Hebel mit Elektromagneten versehen. Da über das zeitliche Verhalten der Rückmeldekontakte keine genauen Angaben vorlagen, wurden Messungen durchgeführt. Nachträglich stellte sich heraus, dass die Toleranzen zwischen einzelnen Maschinen grösser als erwartet waren und erheblich von den aus den Messungen gewonnenen Werten abwichen. Als Abhilfe musste eine zeitraubende Justierung der Kontakte bei den einzelnen Maschinen durchgeführt werden. Eine weitere Schwierigkeit brachten die Magnete, welche nicht für Dauerbelastung dimensioniert waren.

Nachfolgend werden die Funktionen und die Bedienung des intelligenten Schreibautomaten beschrieben. Die Steuerung erfolgt ausschliesslich mit den an der Schreibmaschine angebrachten Spezialtasten und über die normale Schreibmaschinentastatur (Bild 11.1).

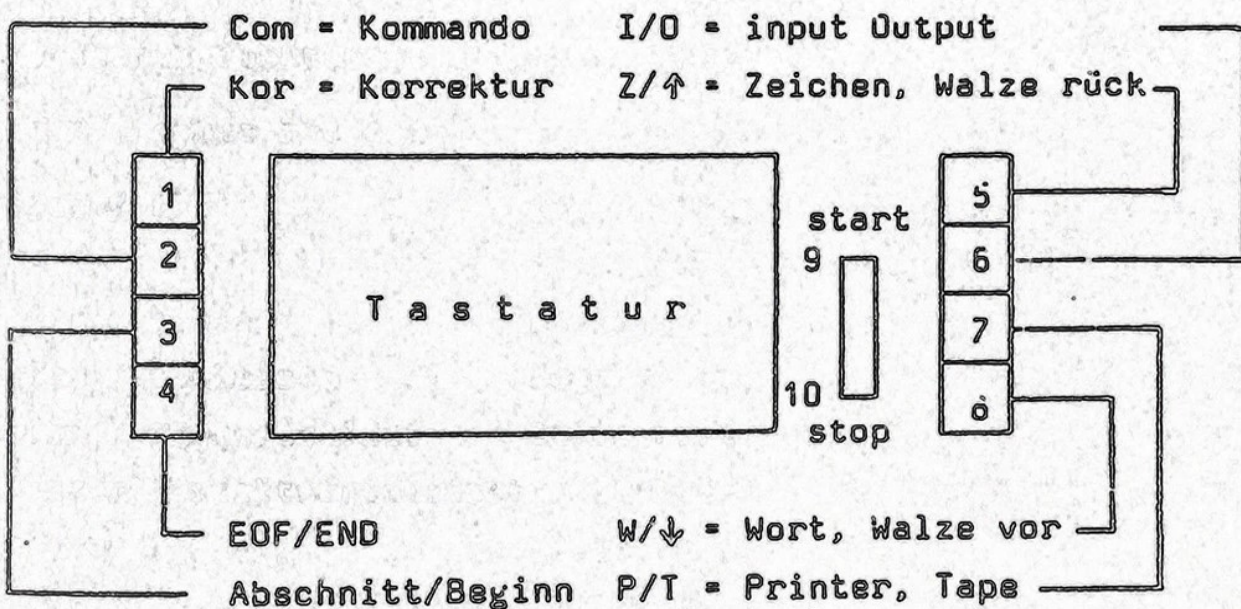


Bild 11.1

- Entwurf und Konzept:

Folgende Randbedingungen waren vor dem eigentlichen Beginn der Entwicklung vorgegeben:

- die obere Kostengrenze für Hard- und Software sowie der Termin für die Fertigstellung waren durch äussere Umstände festgelegt.
- ein Prototyp eines einfachen Schreibautomaten mit dem μ P 4004

- die Bedienung der Maschine sollte nach ergonomischen Kriterien konzipiert werden und die Erfahrungen in der Kommunikation Mensch-Maschine berücksichtigt werden.

- Funktionsbeschreibung:

Die Maschine hat zwei Betriebsarten, Eingabe und Ausgabe, umschaltbar durch die beleuchtete Impulstaste 6 (I/O):

Eingabe - I/O Taste dunkel
Ausgabe - IO/ Taste leuchtet

Die Ein- und Ausgabe kann entweder über die Schreibmaschine oder das Magnetbandgerät erfolgen. Die Umschaltung geht mit der Impulstaste 7 (P/T):

Maschine - P/T Taste dunkel
Band - P/T Taste leuchtet

Die Kommandotaste 2 (Com) dient zur Umschaltung der Funktion aller übrigen Tasten. Sie muss gleichzeitig mit einer anderen Taste gedrückt werden (zeitlich überlappend, d.h. als erste gedrückt und als letzte losgelassen).

Folgende Funktionen stehen zur Verfügung:

Com / n mit n=0 bis 9	Bei der Ausgabe wird der n-te Textblock eingefügt.
Com / t	Bei der Ausgabe wird der nächstfolgende Textblock vom Tonbandgerät eingelesen.
Com / b	Rücksprung auf den Textanfang (Endlosausgabe)
Com / Rücktaste	Zeichen wird gelöscht (Fehlerkorrektur)
Com / Wagenrücklauf	Zeile wird gelöscht (Fehlerkorrektur)
Kor / Korrekturrücktaste	Zeichen wird gelöscht und ein fehlerfreies Original erhalten.
EOF	Blockendung
Com / Begin	Cursor wird auf Anfang des momentanen Blocks gesetzt.
Com / End	Cursor wird auf Ende des momentanen Blocks gesetzt.
Start	Maschine wird gestartet.
Stop	Maschine wird angehalten.
Z	ein Zeichen wird ausgedruckt
W	ein Wort wird ausgedruckt
Com / Start	ein Satz wird ausgedruckt
Com / 2	ein Zeichen wird übersprungen bzw. gelöscht

Com / W	ein Wort wird übersprungen bzw. gelöscht
Com / x	Umschalten in Dialogbetrieb zur Textbearbeitung.
d	delete. Text wird ab Cursorposition gelöscht
s	search. ein Suchwort wird im Textspeicher gesucht. (bis zu 24 Stellen)
f	formattieren. Wagenrückläufe und Trennzeichen werden eliminiert, damit der Text in einem neuen Format wieder ausgedruckt werden kann. (Randsteller)
i	input tape. Einlesen ab Tonband mit Datenkopf
o	output tape. Abspeichern auf Tonband mit Datenkopf zur Identifikation.

Alle Befehle, welche keine direkt sichtbare Aktion auslösen, werden von der Maschine mit einem Piepston (Frequenz hoch) quittiert. Fehlbedienungen werden mit einem Fehlerton (Frequenz tief) quittiert. Im Dialogbetrieb quittiert die Maschine mit Meldungen wie 'OK', 'CAN'T FIND', 'ARE YOU SURE' usw.

- Wahl des Mikroprozessors:

Da die Anwendung typisch byte orientiert war, lag es auf der Hand einen 8-bit-Rechner zu verwenden. Eine erste Analyse zeigte, dass a priori keine zeitkritischen Probleme vorhanden waren und dass auch keine anderen Forderungen einen bestimmten Prozessor bevorteilt hätten. Deshalb wurde dem Motorola M6800 der Vorzug gegeben, weil mit diesem Prozessor die meisten Erfahrungen und ein leistungsfähiges Entwicklungssystem vorhanden war.

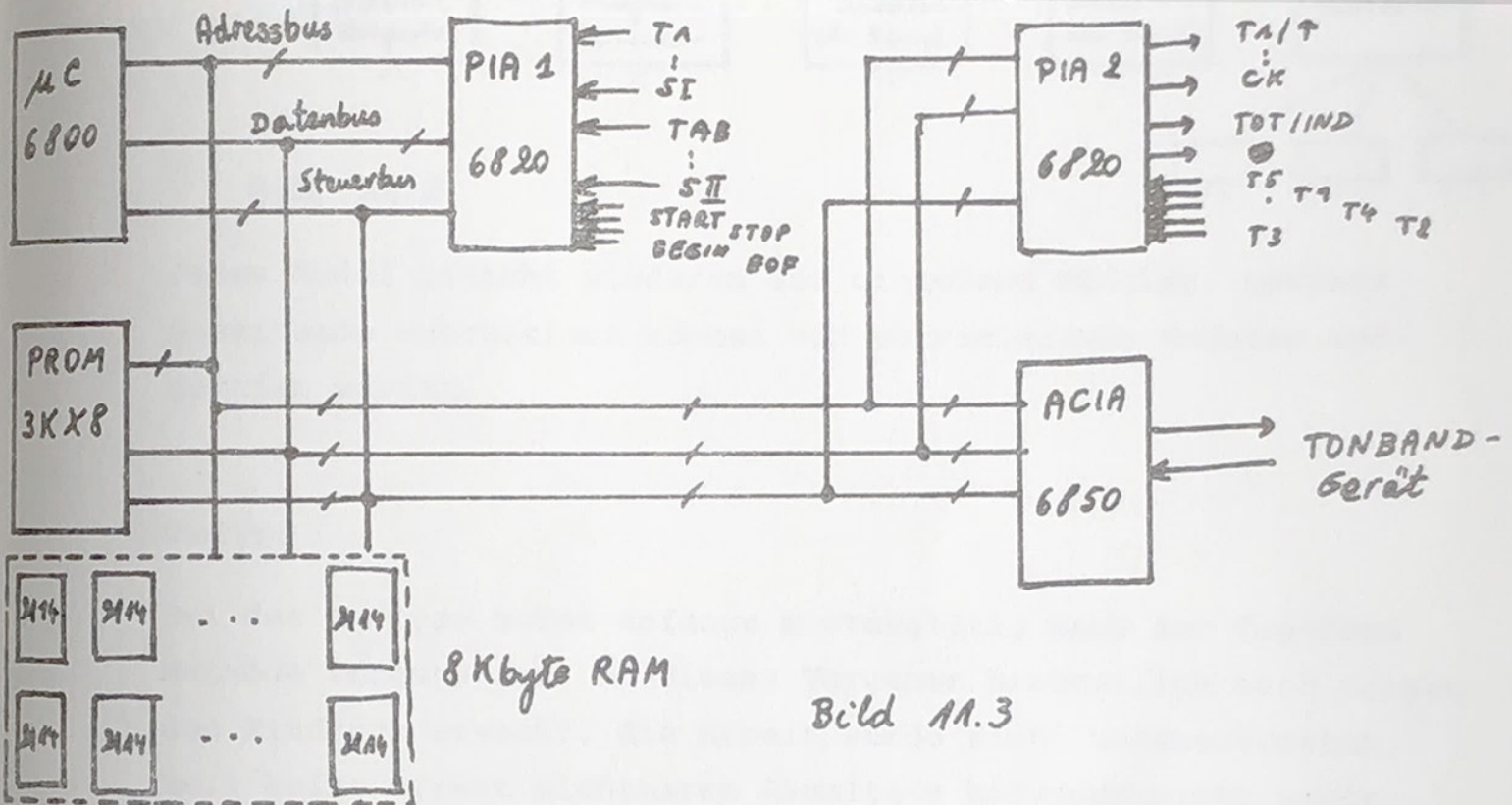
- Aufteilung Hardware/Software:

In der Meinung, die kostengünstigste Lösung sei ein Minimum an Hardware und die Verlagerung aller Funktionen in die Software wurden zum Anschluss der Peripherie nur zwei parallele (PIA's) und ein serielles asynchrones (ACIA) Interface vorgesehen. Als Textspeicher wurden statische RAM's, als Programmspeicher EPROM's benutzt. Rückblickend hätte es sich gelohnt, mehr Funktionen hardwaremässig auszuführen, beispielsweise das Entprellen der Kontakte. (Blockschema Bild 11.3)

- Wahl der Programmiersprache:

Zur Auswahl standen verschiedene 6800 Crossassembler auf einem Minicomputer PDP11 und auf der Grossrechenanlage CDC, sowie die Maschinensprache auf dem 6800 Entwicklungssystem. Da nicht genügend

Erfahrungen mit der Benützung der Crossassemblern bestanden und diese einige Fehler aufwiesen, wurde auf die Vorteile eines Assemblers verzichtet und in Maschinensprache programmiert. Rückblickend war dies keine glückliche Entscheidung, da mehrere Programmierfehler durch die Verwendung eines Assemblers hätten vermieden werden können. Ausserdem wäre die Änderung der festen Adressen einfacher gewesen (Programmspeicher, PIA, ACIA) und die Programmbeschreibung wäre ausführlicher ausgefallen.



- Programmmodule / Schnittstellen

Eine grobe Analyse der notwendigen Software-Funktionen zeigte, dass die Mitteilung gewisser Steuerbefehle wie START, STOP, Umschalten von Ein- auf Ausgabe usw. am zweckmässigsten durch Programmunterbrechungen (Interrupts) durchgeführt werden, da diese Funktionen in fast allen Programmteilen vorhanden sein müssen und da man sich dadurch das kontinuierliche Abfragen der entsprechenden Meldekontakte in den einzelnen Programmmodulen spart. Der Aufbau der Software sieht folgendermassen aus (Bild 11.2.)

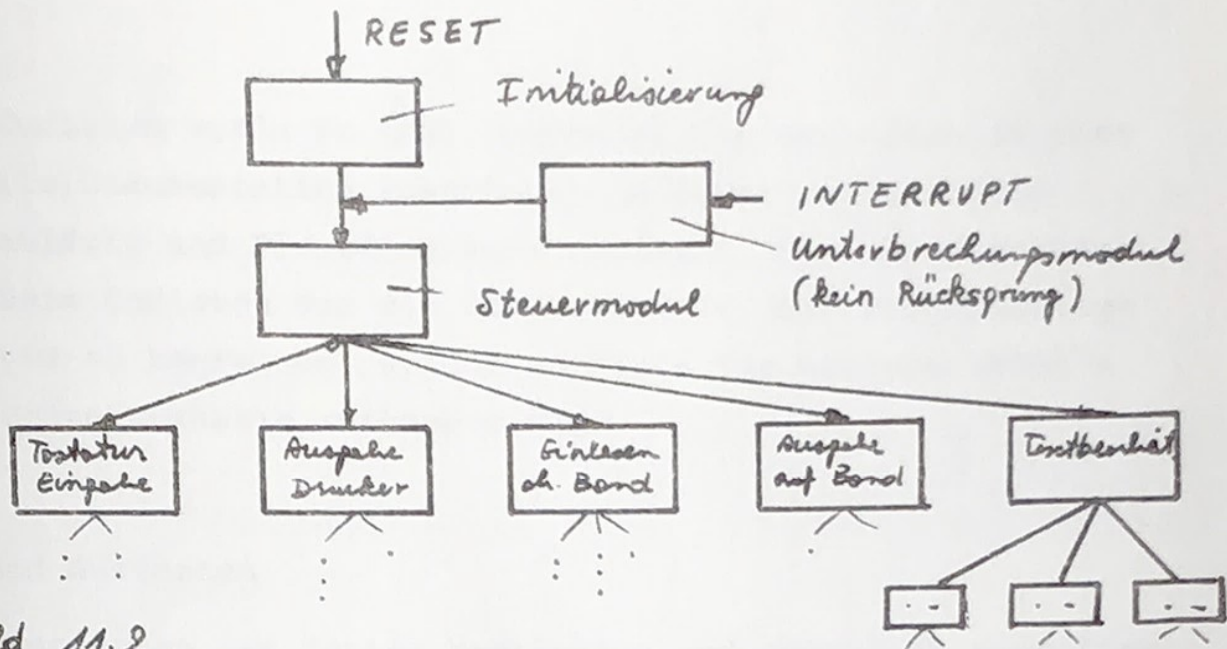


Bild 11.2

Jedes Modul besteht wiederum aus einzelnen Modulen, gewisse gemeinsame Subroutinen können von verschiedenen Modulen aufgerufen werden.

- Analyse

Bei der Analyse wurde anfangs mustergültig nach der Top-Down Methode vorgegangen. Da dieses Vorgehen bekanntlich nach aussen den Eindruck erweckt, die Arbeit würde nicht voranschreiten, weil keine direkt sichtbaren Resultate hervorgebracht werden, wurde bald die Bottom-up Technik aufgegriffen und eine Ausgaberroutine programmiert. Der Vorteil war, dass bald darauf ein sichtbarer Erfolg in Form einer ratternden Maschine vorlag, welche eine vorgegebene Textzeile beliebig oft ausdrückte, der Nachteil war, dass die Ausgaberroutine nachträglich umgeändert und erweitert werden musste, weil sich im Laufe der weiteren Programmentwicklung herausstellte, dass die Schnittstelle zu dem übergeordneten Modul nicht vollständig festgelegt war. (vor allem Probleme mit der Tabulatorfunktion (Nockenschalter) und dem Randschalter). Rückblickend muss man sagen, dass man mehr Zeit für die Entwurfsphase hätte aufwenden müssen und beispielsweise alle Variablen sowie die Belegung der individuellen und der gemeinsamen Speicherplätze in der Entwurfsphase hätte festlegen müssen. Ausserdem war es falsch, von der Top-Down Methode abzuweichen.

- Codieren

Mit dem Codieren wurde zu früh begonnen, was vor allem in eine mangelhafte Dokumentation ausartete, da keine vollständige Programmabläufe und Flussdiagramme vorlagen. Eine Hauptschwierigkeit beim Codieren war die Notwendigkeit, die Programmlänge auf 3K-byte zu begrenzen, weil kein Platz für weitere EPROM's auf der Prozessorkarte vorhanden war.

- Prüfen und Austesten

Bei dem Austesten der fertig bestückten und angeblich geprüften Platinen (Ein-Karten-Computer 6800) einer deutschen Firma zeigte sich, dass fast keine auf Anhieb fehlerfrei funktionierte (meistens infolge von Kurzschlüssen zwischen benachbarten Leiterbahnen durch Aetzfehler). Auch in der Software hatten sich beim Uebergang von der Laborausführung auf die Prototypversionen Fehler eingeschlichen (falsche Adressen usw.). Durch die Verzögerung in der Entwicklung kamen die Tests und Prüfungen im praktischen Einsatz zu kurz. Es traten vor allem Probleme mit der Interruptsteuerung auf, da durch statische Elektrizität falsche Unterbrechungen ausgelöst wurden. Auch die Bedienung warf noch einige Probleme auf und die Justierung der Rückmeldekontakte innerhalb der vorgesehenen Zeitfenster war schwieriger als erwartet. Zur Fehlersuche wurden sowohl Logikzustands- wie auch Logikzeitanalysatoren eingesetzt.